Formal Grammars in Linguistics and Psycholinguistics

Volume 1: An Introduction to the Theory of Formal Languages and Automata

Volume 2: Applications in Linguistic Theory

Volume 3: Psycholinguistic Applications

Willem J.M. Levelt

Max Planck Institute for Psycholinguistics,
Nijmegen

John Benjamins Publishing Company

Amsterdam / Philadelphia

Library of Congress Cataloging-in-Publication Data

Levelt, W. J. M. (Willem J. M.), 1938-
[Formele grammatica's in linguïstiek en taalpsychologie. English]
Formal grammars in linguistics and psycholinguistics / Willem J.M. Levelt ; [translation: Andrew Barnas].
p. cm.
Translation of Formele grammatica's in linguistiek en taalpsychologie, 1974.
Includes bibliographical references and index.
1. Mathematical linguistics. 2. Formal languages. 3. Psycholinguistics. I. Barnas, Andrew. II. Title.
P138.L4713 2008
410.1'51--dc22 2008046396
ISBN 978 90 272 3251 9 (Hb; alk. paper)

John Benjamins Publishing Co. · P.O. Box 36224 · 1020 ME Amsterdam · The Netherlands
John Benjamins North America · P.O. Box 27519 · Philadelphia PA 19118-0519 · USA

Contents

III: Psycholinguistics Applications

Postscript

Preface to the 2008 edition

Almost four decades have past since I first conceived of writing *Formal Grammars*. At that time it was still possible to rather comprehensively review for linguists and psycholinguists the relevant literature on the theory of formal languages and automata, on their applications in linguistic theory and in the psychology of language. That is no longer feasible. In all three areas developments have been substantial, if not breath taking. The latter epithet certainly applies to advances in the computational theory of formal languages and automata. There were not only advances in the mathematical foundations, but also in innumerable wide-ranging applications. Any linguist reading an introduction to this field some 30 or 40 years ago felt at home right away. At the center was the Chomsky hierarchy of grammars and the related automata, largely invented for the treatment of generative issues in natural language syntax. That was also the domain of just about all applications. Nowadays, an interested linguist or psycholinguist opening any text or handbook on formal languages can no longer see the wood for the trees. Not only are linguistic applications in the small minority, but it is also by no means evident which formal, mathematical tools are really required for natural language applications. An historical perspective can be helpful here. There are paths through the wood that have been beaten since decades; they can still provide useful orientation. Origins of these paths can be traced in all three volumes of the present re-edition of *Formal Grammars*.

There is, first, the formal theory itself. An introduction to the Chomsky hierarchy of languages with their formal grammars and automata is still the right starting point as an introduction for (psycho-)linguists. It is, for instance, impossible to understand modern work on tree grammars and automata without knowledge of these foundations. Also, the formal treatment of probabilistic grammars, now essential in the handling of large corpora, had already attained its basic formulation.

There is, second, the domain of linguistic applications. A core issue in Chomsky's original work, extensively treated in volume II, was the generative power of grammars. Regular grammars are definitely insufficient for natural language description. Context-free grammars fare a lot better, but are still insufficient for handling various kinds of long-distance dependences. But then, as was proven in 1973, transformational grammars of the *Aspects*-type turned out to have unrestricted generative power. That story is told in volume II. Ever since, the search has been

for a restricted type of formal grammar that nicely matches the generative capacity of natural languages. There is optimism now that this has been achieved in a class called 'mildly context-sensitive grammars'. These new developments can hardly be appreciated without a good understanding of the original work.

There are, third, the psycholinguistic applications. The core issue in language acquisition is no different now from what it was three or four decades ago: how can full mastery of a language be achieved from limited linguistic input? The basic theorems on linguistic inference or 'learnability' of languages, in particular Gold's theorems and the first statistical ones, were available in the early seventies. They have set the framework for all later discussion, in particular on the empirical question of how much negative evidence ('this sentence is not well-formed') the child receives. Again, modern work cannot be fully appreciated without knowledge of these basics. There was also the beginning of modeling the language user. Speaker models, then and now, must in some way incorporate a grammar, because a speaker's every utterance is a grammatical test; it is sometimes failed, for sure, but very often passed. How is that well-formedness achieved? That the speaker's operations do not simply match grammatical rules was known at the time. The still ongoing quest for an empirically founded and formally correct solution is better understood against the background of the many early modeling attempts, often brilliant, but failed.

In a postscript to this book, I have sketched what has become, after all these years, of formal grammars in linguistics and psycholinguistics, or at least some of the core developments. That chapter may, in fact, provide further motivation for the reader to make a trip back to some of the historical sources, among them real gems.

Writing that chapter would have been impossible without an intensive session of tutoring by my friend Aravind Joshi, one of the world leaders in this field. That was, in fact, a *déjà vu* experience for me, because I wrote *Formal Grammars* with helpful Aravind next doors, when we were both members of Princeton's Institute for Advanced Study.

I also thank Gerard Kempen for his helpful comments on this chapter, but also for many years of inspiring collaboration.

I am grateful to John Benjamins Publishers, in particular to Anke de Looper, for their initiative to bring out a new edition of this book, plus an independent textbook edition of volume I, *An introduction to the theory of formal languages.*

This re-edition is a faithful page-by-page copy of the three-volume original. The three volumes, now in one cover, kept their independent prefaces, numbering and indices. I only corrected a few obvious errors and typos.

Willem J. M. Levelt
Nijmegen
August 2008

FORMAL GRAMMARS IN LINGUISTICS AND PSYCHOLINGUISTICS

VOLUME I

An Introduction to the Theory of Formal Languages and Automata

by

W. J. M. LEVELT

1974

MOUTON

THE HAGUE · PARIS

PREFACE

In the latter half of the 1950's, Noam Chomsky began to develop mathematical models for the description of natural languages. Two disciplines originated in his work and have grown to maturity. The first of these is the theory of formal grammars, a branch of mathematics which has proven to be of great interest to information and computer sciences. The second is generative, or more specifically, transformational linguistics. Although these disciplines are independent and develop each according to its own aims and criteria, they remain closely interwoven. Without access to the theory of formal languages, for example, the contemporary study of the foundations of linguistics would be unthinkable.

The collaboration of Chomsky and the psycholinguist, George Miller, around 1960 led to a considerable impact of transformational linguistics on the psychology of language. During a period of near feverish experimental activity, psycholinguists studied the various ways in which the new linguistic notions might be used in the development of models for language user and language acquisition. A good number of the original conceptions were naïve and could not withstand critical test, but in spite of this, transformational linguistics has greatly influenced modern psycholinguistics.

The theory of formal languages, transformational linguistics, psycholinguistics, and their mutual relationships are the theme of this work. Volume I is an introduction to the theory of formal languages and automata; grammars are treated only as formal systems, and no application of the theory, linguistic or other, is made. Volume II in turn deals with applications of those mathe-

matical models to linguistic theory. Volume III treats applications of grammatical systems to models of language user and language learner, as well as the formal questions which have arisen as a result of such applications. The material is cumulative: Volume II supposes a general understanding of Volume I, and Volume III refers to the subjects dealt with in Volumes I and II. Volumes II and III have their own preface, so we can now turn to some introductory remarks with respect to the present volume.

Volume I, independent of the two following volumes, should be seen as an introduction to the theory of formal languages and automata. A number of similar introductions are available at the moment, but I have nevertheless undertaken the present work for three reasons. First, most available texts, because they suppose an acquaintance with sophisticated mathematical theories and methods, are beyond the reach of many students of linguistics and psychology. More often than not, Chomsky's and Miller's contributions to the *Handbook of Mathematical Psychology* prove too difficult for early graduate teaching. The present introduction is kept at a rather elementary level; a general knowledge of college mathematics will be sufficient to follow the text, although familiarity with the elements of set theory and statistics will certainly be an advantage.

Second, existing introductions treat a number of subjects which have little obvious relation to linguistics or psychology. The linguist or the psychologist is obliged to make his own selection from among a series of topics which he does not yet understand, and he might search in vain for a treatment of topics which are especially relevant to his field. Probabilistic grammars and grammatical inference, for example, are not treated in any of the existing introductions. Special attention has been paid to these topics in the present volume, but matters not directly relevant to linguistics or psychology have not been completely excluded, as a balanced presentation of the theory sets its own demands.

The third reason for writing this introduction is to supply readers of the two following volumes with a concise survey of the main notions of formal language theory used there. The subject

index of this volume can be used to find definitions of technical terms: definitions are indicated by italicized page numbers.

Without the help and cooperation of many, these three volumes could not have been realized. A first version was written during a sabbatical year at The Institute for Advanced Study in Princeton, New Jersey. I am deeply grateful to Professor Duncan Luce and to The Institute for the invitation which made my stay possible. Much in this work is due to the help and insights of Professor George Miller, former director of the Harvard Center for Cognitive Studies, where the new psychology of language originated under his guidance. Thanks to him I was granted a Research Fellowship at the Center in 1965, and by happy coincidence, he too was at the Institute for Advanced Study when I was composing the text. His attentive advice was most useful, especially in the writing of the third volume. Likewise, regular discussions with Dr. Philip Johnson-Laird helped to clarify many of the psychological issues. Conversations with Professor Aravind Joshi on the subject matter of the first two volumes were also enormously stimulating and enjoyable; I profited almost daily from his erudition in the fields of both formal systems theory and mathematical linguistics.

Finally, I wish to express my gratitude to all those who have contributed by critically reading the text in the original Dutch version: Professor L. Verbeek, Dr. H. Brandt Corstius, Mr. R. Brons, Dr. G. Kempen, Dr. A. van der Ven, Mr. E. Schils, Mr. L. Noordman, Dr. A. De Wachter-Schaerlaekens, and Professor A. Kraak. Their remarks not only prevented the printing of many disturbing errors, but also led to many enriching additions to the text.

March 1973

W. J. M. Levelt
Nijmegen

TABLE OF CONTENTS

1

GRAMMARS AS FORMAL SYSTEMS

1.1. GRAMMARS, AUTOMATA, AND INFERENCE

The theory of formal languages originated in the study of natural languages. The description of a natural language is traditionally called a GRAMMAR; it should indicate how the sentences of a language are composed of elements, how elements form larger units, and how these units are related within the context of the sentence. The theory of formal languages proceeds from the need to provide a formal mathematical basis for such descriptions.

Chomsky, the founder of the theory, envisaged more than a simple refinement of traditional linguistic description. He was primarily concerned with a more thorough examination of the basis of linguistic theory. This involves such questions as "what are the goals of linguistic theory?", "what conditions must a grammar fulfill in order to be adequate in view of these goals?", and "what is the general form of a linguistic theory?" Without a formal basis, these and similar questions cannot be handled with sufficient precision. Volume II of this book will deal with these issues; it will be shown that a formal language can serve as a mathematical model for a natural language, while a formal grammar can act as a model for a linguistic theory.

From a mathematical point of view, grammars are FORMAL SYSTEMS, like Turing machines, computer programs, propositional logic, theories of inference, neural nets, and so forth. Formal systems characteristically transform a certain INPUT into a particular OUTPUT by means of completely explicit, mechanically applicable rules. Input and output are strings of symbols taken

from a particular alphabet or VOCABULARY. For a formal grammar the input is an abstract START SYMBOL; the output is a string of "words" which constitutes a "sentence" of the formal "language". Therefore a grammar may be considered as a GENERATIVE system; this feature is often emphasized by the use of the term GENERATIVE GRAMMAR. The quotation marks around "word", "sentence", and "language" indicate that these terms are not used in their full linguistic sense, but rather are concepts which must be strictly defined within the formal system. In linguistic applications of formal language theory, as in Volume II of this book, care must be taken to establish the relationships between the formal and linguistic notions. In the present volume, however, we will no longer use the quotation marks, and will omit the adjective "formal" for both language and grammar where the context allows.

A second type of formal system can use the sentences of a language as input; its output is generally an abstract stop symbol. Systems of this type are called AUTOMATA, and may be considered as ACCEPTING SYSTEMS. The theory of automata is older than that of formal language, and historically it was rather surprising that the two theories showed such close parallels that they often appeared to be mere notational variants. One can very well use an automaton rather than a formal grammar as a model for a theory of natural language, but although this has in fact been done, the generative grammar remains the preferred model. The interchangeability of grammars and automata indicates that the distinction between generative and accepting is less fundamental than it may at first appear. It is primarily a conceptual distinction; there are indeed automata with no "preferential direction" such as Turing machines, and grammars which are accepting rather than generative systems such as categorical grammars. However, from the point of view of presentation and application, the dichotomy has its merits. In psycholinguistics in particular it has a natural interpretation with reference to SPEAKER-HEARER models. Volume III of this book will offer several examples of such applications.

The third and last type of formal system which will be discussed

in this volume takes a sample of the sentences of a language as input; its output is a grammar which is in some way adequate for the language. Such systems are called GRAMMATICAL INFERENCE PROCEDURES. They can serve as models not only for linguistic discovery procedures (how can one find a grammar for a given corpus of sentences?) but also for theories of language acquisition.

The mathematical growth of formal language theory has resulted in an enormous extension of its range of applications. Beyond its obvious applications in the analysis of computer languages, the theory is used for the formal description of visual patterns (see Volume III, paragraph 3.6.7. for such picture grammars), for subdivisions of logic, and for several other fields which deal with the formal representation of knowledge.

Conversely, the integration of formal language theory into the theory of formal systems has made various mathematical tools, such as recursive function theory, available to the study of formal languages.

The reader, however, need not be acquainted with such areas of mathematics in order to understand the present work which is meant to be an introduction. Our discussion will be limited to the relationship between formal language theory on the one hand and the theories of automata and inference on the other. Each of these has rather direct linguistic and psycholinguistic applications, and it is precisely the possibility of application which has served as the principal, though not only, criterion for selecting properties of the theories for discussion. This does not alter the fact that it is better to treat the structure of grammar, of automata, and of inference from an abstract than from an applied point of view. Such is the method which we shall follow here, beginning with a formal definition of the concept "grammar".

1.2. THE DEFINITION OF "GRAMMAR"

For the formal definition of "grammar" we must introduce four concepts: terminal vocabulary, nonterminal vocabulary, production rule, and start symbol.

The TERMINAL VOCABULARY V_T is the set of terminal elements with which the sentences of a language may be constructed. Elements of V_T will be denoted by lower case letters from the beginning of the Latin alphabet. We write $a \in V_T$ or a *in* V_T when a belongs to the terminal vocabulary.

The NONTERMINAL VOCABULARY V_N consists of elements which are only used in the derivation of a sentence; they never occur as such in the sentences of the language. Elements of V_N, in upper case Latin letters, are called VARIABLES or CATEGORY SYMBOLS.

V_N and V_T are disjoint: their intersection, $V_N \cap V_T$, is empty. Together V_N and V_T form the vocabulary V of the grammar, thus $V = V_N \cup V_T$. A string of elements in V, regardless of whether they are variables, terminal elements, or both, will be denoted by a lower case letter of the Greek alphabet. A string may have 0, 1, or more elements; the string of 0 elements is called the NULL-STRING, and is represented by λ. A string consisting exclusively of terminal elements may be denoted by a lower case letter from the end of the Latin alphabet.

The symbol V_T^* is used to denote the set of all finite strings of elements from the terminal vocabulary. For example, if V_T consists of two elements, a and b, i.e. $V_T = \{a, b\}$, V_T^* consists of λ, a, b, aa, ab, bb, ba, aaa, aab, aba, bba, ... If we wish explicitly to exclude the null-string λ, we write V_T^+, the set of all strings of positive length. Thus, $V_T^+ = V_T^* - \lambda$. Obviously, therefore, if V_T is not empty, then V_T^* and V_T^+ contain an infinite number of elements (strings). Analogously one can define V^* as the set of all possible strings of vocabulary elements, and V^+ as the set of all possible strings of vocabulary elements except the null-string. The length of a string α is denoted by $|\alpha|$; thus $|a| = 1$, $|aab| = 3$, and $|\lambda| = 0$.

The PRODUCTION RULES or productions of a grammar are ordered pairs of strings. They take the form $\alpha \to \beta$, where $\alpha \in V^+$ and $\beta \in V^*$. This means that string of elements α of positive length can be replaced by, or rewritten as, string of elements β, possibly λ. Such rules apply in any context, i.e. if α is part of a longer string $\gamma\alpha\delta$, then $\gamma\alpha\delta$ may be rewritten as $\gamma\beta\delta$ by the same rule. When a

string is rewritten as another string by a single application of a production rule, we use the symbol $\Rightarrow$; thus $\gamma\alpha\delta \Rightarrow \gamma\beta\delta$. The latter string DERIVES DIRECTLY from the former. If there are productions such that $\alpha_1 \Rightarrow \alpha_2$, $\alpha_2 \Rightarrow \alpha_3$, ... $\alpha_{n-1} \Rightarrow \alpha_n$, we may write $\alpha_1 \overset{*}{\Rightarrow} \alpha_n$, read "$\alpha_1$ derives α_n". The set of productions of a grammar is denoted by P; the set may also be described as a CARTESIAN PRODUCT. The set of all possible rules consists of all ordered pairs of strings which can be constructed in this manner; it may be denoted by $V^+ \times V^*$, the cartesian product of V^+ and V^*. The productions of a grammar are a subset of this product: some strings of V^+ may be replaced by some strings in V^*. Thus $P \subset V^+ \times V^*$.

The START SYMBOL of a grammar is denoted by S (originally for "sentence"); it is a particular element of V_N.

We can at this point define a grammar as follows.

A GRAMMAR $G = (V_N, V_T, P, S)$ is a system consisting of a nonterminal vocabulary V_N, a terminal vocabulary V_T, a set of productions P, and a start symbol S, with the following properties:

(1) V_N, V_T and P are finite, nonempty sets.
(2) $V_N \cap V_T = 0$.
(3) $P \subset V^+ \times V^*$.
(4) $S \in V_N$.

A SENTENCE generated by G is every element s of V_T^* for which $S \overset{*}{\Rightarrow} s$, i.e. it is a terminal string derivable from S by the productions of P.

The LANGUAGE $L(G)$ generated by G is the set of sentences generated by G.

Two grammars G_1 and G_2 are (WEAKLY) EQUIVALENT if $L(G_1) = L(G_2)$, i.e. if they generate the same set of sentences. Another form of equivalence, STRONG EQUIVALENCE, will be discussed in Volume II, paragraph 2.1.

1.3. EXAMPLES

EXAMPLE 1.1. Let $G = (V_N, V_T, P, S)$, where $V_N = \{S\}$, i.e. S is the only nonterminal symbol, $V_T = \{a, b\}$, $P = \{S \to aS, S \to b\}$. Which language is generated by G? Repeated application of the first production gives $S \Rightarrow aS \Rightarrow aaS \Rightarrow aaaS$, etc. None of these strings is a sentence, for all include the nonterminal symbol S. The only way to eliminate S is by use of the second production $S \to b$. This will produce sentences such as b, ab, aab, $aaab$, etc. A sentence generated by G is thus a string of a's followed by a single b. A simple notation for language $L(G)$ is $\{a^*b\}$, where a^* is any string of a's of length $\geqslant 0$.

EXAMPLE 1.2. Let $G = (V_N, V_T, P, S)$, where $V_N = \{S\}$, $V_T = \{a, b\}$, $P = \{S \to aSa, S \to bSb, S \to aa, S \to bb\}$. The first two rules may be applied and repeated in any order. This will produce such derivations as $S \Rightarrow aSa \Rightarrow abSba \Rightarrow abbSbba \Rightarrow abbaSabba$. The only way to derive sentences from such strings is by use of the third or fourth production; these replace S with aa or bb. In all cases the result is a string of a's and b's, followed by the same string in reverse order. G is said to generate language $\{ww^R\}$, where w^R represents the reflection of w, and $|w| \geqslant 1$. $L(G)$ is called a MIRROR IMAGE language.

EXAMPLE 1.3. Let $G = (V_N, V_T, P, S)$, where $V_N = \{S,E,F\}$, $V_T = \{a, b, c, d\}$, $P = \{S \to ESF, S \to EF, E \to ab, F \to cd\}$. By applying the first production of P $n-1$ times, we obtain the string $E^{n-1}SF^{n-1}$ (the exponent indicates the number of successive occurrences of the element). By then using the second production once, one obtains E^nF^n. When, by application of the third and fourth productions respectively, all the E's are replaced by ab and all the F's by cd, the resulting string consists of n ab-pairs followed by n cd-pairs. Language $L(G)$ consists of all sentences of the form $(ab)^n(cd)^n$, where $n \geqslant 1$.

In this example a alternates with b, and c with d in the sentences of $L(G)$. It is possible to modify the grammar in such a way that

the terminal elements will be neatly grouped in the sentences of L: first all a's, then all b's, etc. This will be the case in the following example.

EXAMPLE 1.4. Language $\{a^n b^n c^n d^n\}$, where $n \geqslant 1$, is generated by grammar $G = (V_N, V_T, P, S)$, in which $V_N = \{S, E, F, B, C\}$, $V_T = \{a, b, c, d\}$, and P consists of the following productions:

1. $S \rightarrow ESF$	4. $F \rightarrow Cd$	7. $BC \rightarrow bc$
2. $S \rightarrow EF$	5. $Ba \rightarrow aB$	8. $Bb \rightarrow bb$
3. $E \rightarrow aB$	6. $dC \rightarrow Cd$	9. $cC \rightarrow cc$

The first four productions are essentially the same as those of Example 1.3. They produce strings of the form $(aB)^n(Cd)^n$, where $n \geqslant 1$. The other five productions serve in the further grouping of the elements. By means of production 5 one can replace a string $aBaBaB\ldots$ of arbitrary length by a string of a's followed by a string of B's. Production 6 acts similarly with respect to $CdCdCd\ldots$ sequences. We must now see to it that further rewriting in terminal symbols is possible only when these arrangements have in fact been performed; this is the purpose of rules 7 through 9. Rule 7 serves to replace the pair BC in the center of the string with terminal elements, but it can be applied only if B and C are found in the right place in the center of the string. By means of production 8 the variables B are replaced by the terminal symbol b, on condition that each B is located directly to the left of a b. The process can be completed only when all the B's are already in the correct positions. Finally production 9 acts similarly in the right hand half of the string. The result is a string of the desired form, $a^n b^n c^n d^n$; sentences of other forms cannot be generated by this grammar.

EXAMPLE 1.5. It is possible to write a still more compact grammar for language $\{a^n b^n c^n d^n\}$, namely $G = (V_N, V_T, P, S)$, in which $V_N = \{S, E, F\}$, $V_T = \{a, b, c, d\}$, and P consists of the following productions:

1. $S \rightarrow ESF$
2. $S \rightarrow abcd$
3. $Ea \rightarrow aE$
4. $dF \rightarrow Fd$
5. $Eb \rightarrow abb$
6. $cF \rightarrow ccd$

The reader himself may now experiment with the operation of this grammar.

2

THE HIERARCHY OF GRAMMARS

2.1. CLASSES OF GRAMMARS

The definition of grammar given in the preceding chapter is absolutely general in the following intuitive sense: if a mechanical procedure can be contrived, according to which the sentences of language L can be enumerated in some order, then language L can be generated by a grammar in the defined form. We call this statement intuitive because the concept "mechanical procedure" has not yet been defined. One definition of it will be given in paragraph 7.4., but for the present one can roughly conceive of it as follows. Let us assume that we dispose of a general purpose computer with an unlimited memory. Let us further assume that a program can be written for this computer according to which each sentence of L, and only sentences of L, will appear in the output after a finite number of operations. (The program might, for example, produce the sentences in order of length: first λ if it is in the language, then the sentences of length 1, followed by the sentences of length 2, etc.) We could then say that a procedure exists for the enumeration of the sentences of L, and that L is RECURSIVELY ENUMERABLE. Every recursively enumerable language can be generated by a grammar corresponding to the definition (we shall return to this matter in paragraph 7.4.).

The class of recursively enumerable languages is large, but it is of little interest from a linguistic point of view. One would expect that natural languages have characteristic properties which would rather limit the range of possible syntactic structures in certain

respects. The class of recursively enumerable languages is therefore an unattractive model for natural languages because it is defined by procedures which may be completely arbitrary. Models of empirical interest will result only from the definition of more limited classes of grammars. It is better to reject too strong a model with good reason than to maintain a weak model and never discover the characteristic structure of a language. The class of recursively enumerable languages is the weakest conceivable model.

Chomsky (1959 a, b) devised a schema for the classification of grammars which is now in general use. It is based on three increasingly restrictive conditions on the production rules.

FIRST LIMITING CONDITION: For every production $\alpha \to \beta$ in P, $|\alpha| \leqslant |\beta|$. Thus the grammar contains no productions whose application would result in a decrease of string length.

SECOND LIMITING CONDITION: For every production $\alpha \to \beta$ in P, (1) α consists of only one variable, i.e. $\alpha \in V_N$, and (2) $\beta \neq \lambda$. The productions are of the form $A \to \beta$, where $\beta \in V^+$.

THIRD LIMITING CONDITION: For every production $\alpha \to \beta$ in P, (1) $\alpha \in V_N$, and (2) β has the form a or aB, where $a \in V_T$ and $B \in V_N$. The rules are thus either of the form $A \to a$ or of the form $A \to aB$.

With these limiting conditions, grammars may be classified in the following way.

TYPE-0 GRAMMARS are grammars which are not restricted by any of the limiting conditions. Their definition is simply that of "grammar"; they are also called UNRESTRICTED REWRITING SYSTEMS. Productions are of the form $\alpha \to \beta$.

TYPE-1 GRAMMARS are grammars restricted by the first limiting condition. Productions have the form $\alpha \to \beta$, where $|\alpha| \leqslant |\beta|$. Type-1 grammars are also called CONTEXT-SENSITIVE GRAMMARS for reasons to be mentioned in paragraph 2.4. They obviously constitute a subclass of type-0 grammars. In fact they are a strict subset of the set of type-0 grammars, for there are type-0 grammars

which are not of type-1, namely, those grammars with at least one production where $|\alpha| > |\beta|$. The grammars given in Examples 1.1. through 1.5. satisfy this first condition and are therefore context-sensitive.

TYPE-2 GRAMMARS are grammars restricted by the second limiting condition. Productions have the form $A \to \beta$ where $\beta \neq \lambda$. Grammars of this type are called CONTEXT-FREE GRAMMARS. The second condition implies the first: from $|\beta| \geqslant 1$ and $|A| = 1$ it follows that $|A| \leqslant |\beta|$. Context-free grammars are therefore context-sensitive, but the inverse is not true; the class of context-free grammars is a strict subset of the class of context-sensitive grammars. The grammars given in Examples 1.1., 1.2., and 1.3. are context-free.

TYPE-3 GRAMMARS are grammars restricted by the third limiting condition. Productions have the form $A \to a$ or $A \to aB$. These are REGULAR GRAMMARS (in linguistic literature they are often called FINITE STATE GRAMMARS). In its turn the third limiting condition implies the second. Therefore the class of regular grammars is a subclass of the class of context-free grammars; in fact it is a strict subset. The grammar given in Example 1.1. is a regular grammar.

Language types may be defined according to the various classes of grammars. A type-3 grammar generates a regular language (or finite state language), a type-2 grammar generates a context-free language, a type-1 grammar generates a context-sensitive language, and a type-0 grammar generates a (recursively enumerable) language.

It does not follow, however, from the relations of inclusion which exist among the various types of grammars that corresponding languages are bound by the same relations of inclusion. We cannot exclude the possibility a priori that for every context-free grammar there might exist an equivalent regular grammar. In that case all context-free languages might be generated by regular grammars, and consequently regular languages would not form a strict subset of context-free grammars. However in the following it will become apparent that the language types do show the same relations of strict inclusion as the grammar types: there

are type-0 languages which are not context-sensitive, context-sensitive languages which are not context-free, and context-free languages which are not regular. Figure 2.1. illustrates this hierarchical relation, called the Chomsky Hierarchy.

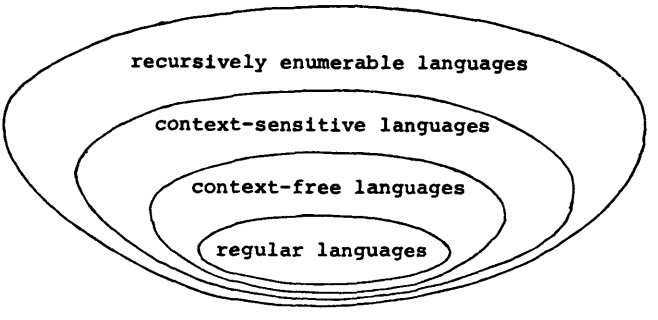


Fig. 2.1. The Chomsky Hierarchy of Languages.

It is obvious that the null-string can be present only in type-0 languages. Sometimes, however, it is convenient to add it to other languages as well. In the following we shall suppose in all cases, except in Chapter 3, that λ has been added to the language, unless otherwise stated.

In the remaining part of this chapter we shall deal with a few properties of each of the grammars.

2.2. REGULAR GRAMMARS

Most properties of regular grammars (*RG*'s) can best be treated on the basis of the theory of automata (cf. chapter 4). Our discussion here will be limited to five theorems which will be needed in the remainder of the present chapter; four of them can easily be explained without reference to automata theory.

We must first introduce a means of visual representation of grammatical derivations, called DERIVATION TREES, TREE DIAGRAMS, or PHRASE MARKERS (*P*-markers). The procedure is a general one which may be used not only for regular grammars, but also for

context-free grammars and some context-sensitive grammars. An example will illustrate the procedure.

EXAMPLE 2.1. Let $G = (V_N, V_T, P, S)$, where $V_N = \{S,B\}$, $V_T = \{a, b\}$, and $P = \{S \to aB,\ B \to bS,\ B \to b\}$. G is thus a regular grammar. The sentences in $L(G)$ consist of alternating a's and b's, beginning with a and ending with b. Thus $L(G) = \{(ab)^+\}$ (by convention $\lambda \in L(G)$).

Let us examine the derivation of the sentence *ababab*; it can be generated only in the following way: $S \Rightarrow aB \Rightarrow abS \Rightarrow abaB \Rightarrow ababS \Rightarrow ababaB \Rightarrow ababab$. Figure 2.2.a. gives the tree diagram for this derivation, clearly illustrating each step. Beginning at S (at the top of the diagram), the tree divides into two branches, one leading to a, the other to B; this is the first step in the derivation. From B two further branches lead to b and to S respectively, showing the second step. The remaining steps in the derivation may be discovered by inspection.

Formally speaking, a (derivation) tree is a system of nodes and branches (or edges). Branches are directed connections between nodes, i.e. branches enter and leave the nodes. A tree has only one node which no branch enters; it is called the root or origin of the tree. Exactly one branch enters each of the remaining nodes. Moreover, a path may be found from each node to the root of the tree. Finally, each node bears a label.

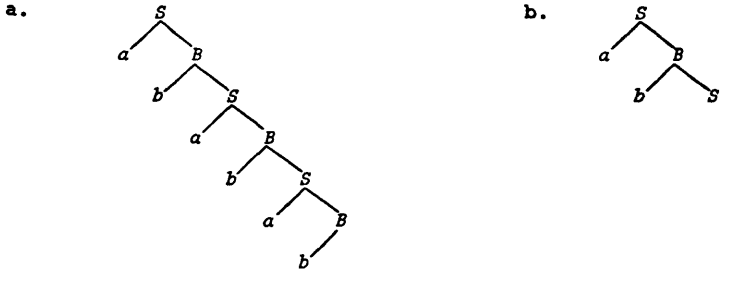


Fig. 2.2. a. Derivation Tree for the Sentence *ababab* (Example 2.1.).
b. Incomplete Derivation Tree.

A derivation in a context-free grammar can be represented by a tree diagram, all the nodes of which are labeled with elements of V. The root is the start symbol S, nodes from which branches leave are elements of V_N, and nodes from which no branches leave are elements of V_T. Each of these features can easily be verified in Figure 2.2.a.

Sometimes it is considered unnecessary to show the entire derivation, and only the first few steps are given in an incomplete tree, as in Figure 2.2.b. In such a case it is possible that nodes from which no branches leave may be labeled as elements of V_N.

We can now return to the subject of regular grammars. It is evident that each string in a regular grammar derivation contains at most one variable, and that this variable is the last element of the string. Consequently, tree diagrams for such derivations branch to the right, i.e. at each step it is the rightmost node which further divides into two branches.

The definition given for regular grammars is in some sense economical. It is possible that the class of languages generated by regular grammars be generated also by grammars with a more complicated rule structure. While this fact is not interesting in itself, it should caution us against concluding on the class to which a language might belong solely on the basis of the type of grammar by which it is generated. An example will serve to illustrate this.

EXAMPLE 2.2. Let $G = (V_N, V_T, P, S)$, with $V_N = \{S\}$, $V_T = \{a\}$, and $P = \{S \to aSa,\ S \to aa,\ S \to a\}$. This is obviously a context-free grammar; the productions are not of the form of those of regular grammars. But $L(G)$ is a regular language, for there is also a regular grammar by which it can be generated. $L(G)$ consists of all possible strings of a's; it can likewise be generated by grammar G' with $P' = \{S \to aS,\ S \to a\}$. G' is thus a regular grammar equivalent to G, and consequently $L(G)$ is a regular language.

A grammar is called RIGHT-LINEAR if all its productions are of the form $A \to xB$ or $A \to x$ (notice that x represents a string of terminal elements).

THEOREM 2.1. The class of right-linear grammars generates precisely the class of regular languages.

PROOF. All regular grammars are right-linear, and therefore all regular languages can be generated by right-linear grammars. The inverse, that each right-linear grammar has an equivalent regular grammar, must also be shown to be true. Let $G = (V_N, V_T, P, S)$ be a right-linear grammar. We must show that there is a regular grammar G' such that $L(G') = L(G)$. Take $G' = (V'_N, V'_T, P', S)$ with the following composition. For every production $A \to x$ in P, where $x = a_1a_2 \ldots a_n$, P' contains the following set of productions: $A \to a_1A_1$, $A_1 \to a_2A_2$, ..., $A_{n-2} \to a_{n-1}A_{n-1}$ and $A_{n-1} \to a_n$. These productions are clearly of the prescribed regular form, and A generates x. If we see to it that the variables A_1, A_2, ..., A_{n-1} do not occur in any other production of P', G' will generate *only* x. Likewise for each production of the type $A \to xB$ in P, where $x = b_1b_2 \ldots b_m$, let P' contain a set of productions $A \to b_1B_1$, $B_1 \to b_2B_2$, ..., $B_{m-1} \to b_mB$, also taking care that the new variables B_1, B_2, ..., B_{m-1} appear only in these productions. Further, let the nonterminal vocabulary V'_N contain V_N plus all the new variables introduced in the above way, and $V'_T = V_T$. It follows from the construction that $L(G') = L(G)$.

THEOREM 2.2. A context-free grammar, with productions such that all derivations are either of the form xB or of the form x, generates a regular language. The same holds if all derivations are of the form Bx or x.

PROOF (summarized). If all the derivations of a context-free grammar must be of the form xB or x, then all the productions must have the form $A \to xB$ or $A \to x$. It follows from Theorem 2.1. that such grammars only generate regular languages. A similar argument holds for grammars, all the derivations of which have the form Bx or x, but it must be shown that grammars with productions exclusively of the form $A \to Ba$ or $A \to a$ generate only regular languages.

THEOREM 2.3. All finite languages are regular.

PROOF. Let L be the finite set $\{s_1, s_2, \ldots, s_n\}$, where $s_i = a_{i1}a_{i2} \ldots a_{ik_i}$. One can generate s_i by a finite set of regular productions, namely $S \to a_{i1}A_{i1}$, $A_{i1} \to a_{i2}A_{i2}$, $\ldots$, $A_{ik_i-1} \to a_{ik_i}$, following the construction used in the proof of Theorem 2.1. The combination of all sets of productions for all s_i gives a finite regular grammar which generates L.

THEOREM 2.4. The union of two regular languages is regular.

PROOF. Let L_1 and L_2 be regular languages. We must show that L_3, where $L_3 = L_1 \cup L_2$ (i.e. L_3 consists of all the sentences of L_1 and all the sentences of L_2), is also regular. Let $G_1 = (V_N^1, V_T^1, P^1, S^1)$ be a regular grammar which generates L_1, and $G_2 = (V_N^2, V_T^2, P^2, S^2)$ be a regular grammar which generates L_2, taking care that $V_N^1 \cap V_N^2 = \emptyset$ (this is always possible). We compose grammar $G_3 = (V_N^3, V_T^3, P^3, S)$ as follows. (1) $V_N^3 = V_N^1 \cup V_N^2 \cup S$, i.e. V_N^2 contains the variables of G_1 and G_2 plus a new variable S, which will also serve as the start symbol of G_3. (2) $V_T^3 = V_T^1 \cup V_T^2$. (3) P^3 contains all productions P^1 and P^2 as well as all possible productions $S \to \alpha$ such that either $S^1 \to \alpha$ is a production in P^1, or $S^2 \to \alpha$ is a production in P^2. Thus $S \Rightarrow \alpha$ in G_3 in precisely the cases where $S^1 \Rightarrow \alpha$ in G_1 and $S^2 \Rightarrow \alpha$ in G_2. Therefore $L_3 = L_1 \cup L_2$. Because all the productions of G_3 are of the required regular form, L_3 is regular.

L_3 may be called the PRODUCT of L_1 and L_2 if L_3 consists of all strings xy with x in L_1 and y in L_2.

THEOREM 2.5. The product of two regular languages is regular. (This theorem will be proven in paragraph 4.4. in connection with the discussion of finite automata.)

2.3. CONTEXT-FREE GRAMMARS

The definition of context-free grammars (*CFG*) is less economical than that of regular grammars. Any production of the form

$A \to \beta$, where $|\beta| \neq 0$, is allowed; β can therefore be any string of terminal and nonterminal elements. However, one can greatly simplify the form of productions without diminishing the generative capacity of the grammars. Such simplified forms of grammars are called NORMAL-FORMS. The most important normal-forms of context-free grammars are the CHOMSKY NORMAL-FORM and the GREIBACH NORMAL-FORM. We shall discuss each of these, and will likewise prove that every context-free grammar is equivalent to a grammar of the Chomsky normal-form.

2.3.1. *The Chomsky Normal-Form*

A grammar is said to be of the Chomsky normal-form if all productions have the form $A \to BC$ or $A \to a$.

THEOREM 2.6. Any context-free language can be generated by a grammar of the Chomsky normal-form.

PROOF. By definition a context-free language can be generated by a grammar with productions of the form $A \to \beta$. We can distinguish three possibilities for such productions: (1) $\beta \in V_T$ (2) $\beta \in V_N$, (3) all other cases. In order to construct a grammar G' in Chomsky normal-form and equivalent to context-free grammar G, we must see if production forms (1), (2), and (3) can be replaced by the appropriate normal production forms. (1) Productions $A \to \beta$, where $\beta = a$, are of the required form and call for no further discussion. (2) If $A \to B$ is a production of G, there are two possibilities: (a) G contains no productions of the form $B \to x$, i.e. B cannot be further rewritten; in this case we can simply ignore the production $A \to B$ in the construction of G'. (b) B can be further rewritten in G, for instance by the productions $B \to \beta_1$, $B \to \beta_2$, ..., $B \to \beta_n$. Without diminishing the generative capacity of the grammar we can now replace these productions, as well as $A \to B$ with the set of productions $A \to \beta_1$, $A \to \beta_2$, ..., $A \to \beta_n$. In spite of rewriting, one or more of these new productions may retain the same form, for instance $A \to C$. In that case we can repeat the procedure and replace $A \to C$ by the productions $A \to \gamma_i$

for every γ_i for which $C \rightarrow \gamma_i$. This can in its turn lead to the same problem, but, as G contains a finite number of variables, the process will reach an end, except if the replacement chain contains a loop (for example $A \rightarrow B$, $B \rightarrow C$, $C \rightarrow A$). But in that case, the variables in the loop are interchangeable, and one of them, A for instance, can replace the others in all the productions of the grammar. The result is that all the newly constructed productions are of form (1) or (3). Those of form (1) are of the Chomsky normal-form. Both the new productions of form (3) and the original form (3) productions from G can be treated as follows. (3) In the remaining productions $A \rightarrow \beta$, β consists of terminal and/or nonterminal elements. We replace all the terminal elements with new variables. Assume that the i^{th} element of β is a terminal element b_i; we replace it with a new variable B_i, and add the production $B_i \rightarrow b_i$, which is of the required normal form. By repeating the operation for all terminal elements in β, we replace the production $A \rightarrow \beta$ by a production $A \rightarrow B_1B_2 \ldots B_n$ and a terminal production of the form mentioned above. Finally we must replace nonterminal productions with productions of the form $A \rightarrow BC$. Here we again apply the construction used in the proof of theorem 2.1., replacing production $A \rightarrow B_1B_2 \ldots B_n$ with a set of productions $A \rightarrow B_1D_1$, $D_1 \rightarrow B_2D_2$, $\ldots$ $D_{n-2} \rightarrow B_{n-1}B_n$, which are all of the required form. It follows from the construction that grammar G' thus obtained is equivalent to G and in the Chomsky normal-form.

EXAMPLE 2.3. Let $G = (V_N, V_T, P, S)$, where $V_N = \{S,A,B\}$, $V_T = \{a,b\}$, and P contains the following productions:

1. $S \rightarrow aSB$
2. $S \rightarrow A$
3. $A \rightarrow ab$
4. $B \rightarrow b$

G generates all strings of the form a^nb^n ($n \geqslant 1$ when λ is excluded). Sentence a^3b^3, for example, has the following derivation: $S \Rightarrow aSB \Rightarrow aaSBB \Rightarrow aaSBb \Rightarrow aaSbb \Rightarrow aaabbb$. We shall now construct a grammar G' in the Chomsky normal-form and equivalent to G.

The only production in the required form is production 4; all others must be replaced. Beginning with production 1, we replace $S \rightarrow aSB$ with two productions $S \rightarrow CSB$ and $C \rightarrow a$, as in (2) in the above proof. $S \rightarrow CSB$ can in turn be replaced by $S \rightarrow CD$ and $D \rightarrow SB$, as in (1).

In production 2 we first replace A with the strings as which it can be directly rewritten. In the present case, the only such string is ab (cf. production 3), and production 2 is thus replaced by $A \rightarrow ab$. The normal-form can be obtained by the replacement of a and b with new variables and the addition of two terminal productions. As we already dispose of terminal productions $C \rightarrow a$ (from production 1) and $B \rightarrow b$ (production 4), it is sufficient to replace production 2 with $S \rightarrow CB$. Production 3 is at the same time replaced by productions of the required form. Thus G' contains the following productions:

1. $S \rightarrow CB$
2. $D \rightarrow SB$
3. $S \rightarrow CD$
4. $C \rightarrow a$
5. $B \rightarrow b$

The derivation of sentence a^3b^3 in G' is therefore $S \Rightarrow CD \Rightarrow aD \Rightarrow aSB \Rightarrow aCDb \Rightarrow aaDb \Rightarrow aaSBb \Rightarrow aaSbb \Rightarrow aaabbb$.

Although grammars G and G' are equivalent, the derivations differ. This can easily be observed from the derivation trees for sentence a^3b^3 given in Figure 2.3.a. (derivation in G) and Figure 2.3.b. (derivation in G').

2.3.2. *The Greibach Normal-Form*

A grammar is in the Greibach normal-form if all the productions are of the form $A \rightarrow a\beta$, where β is a string of 0 or more variables ($\beta \in V_N^*$).

THEOREM 2.7. Any context-free language can be generated by a grammar in the Greibach normal-form.

For the proof of this theorem we refer the reader to Greibach (1965). Our discussion here will be limited to the following example.

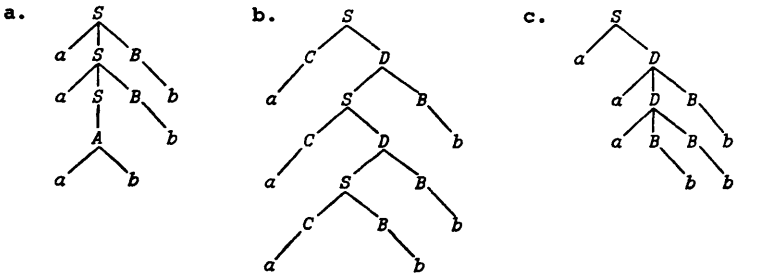


Fig. 2.3. Derivation Trees for a^3b^3.
a. Derivation Tree in G.
b. Derivation Tree in G' (Chomsky normal-form).
c. Derivation Tree in G'' (Greibach normal-form).

EXAMPLE 2.4. Let us once again consider grammar G of Example 2.3. In order to find a grammar G'' in Greibach normal-form which is equivalent to it, we may use grammar G' in Chomsky normal-form as starting point. The variables of G' are S, B, C, and D. We number these in an arbitrary order, indicating the number by subscript: thus, S_1, B_2, C_3, D_4. We shall at this point change the productions in such a way that the direct rewriting of a variable has as its first element either a terminal element or a variable with a higher number. Production 1 ($S_1 \rightarrow C_3B_4$) and production 3 ($S_1 \rightarrow C_2D_4$) already have this form. Production 2 ($D_4 \rightarrow S_1B_2$) can be adapted by first replacing S_1 with the strings as which it can be directly rewritten, namely C_3B_2 and C_3D_4, giving $D_4 \rightarrow C_3B_2B_2$ and $D_4 \rightarrow C_3D_4B_2$. It remains the case that the subscripts decrease (from 4 to 3), but the required form can be obtained by replacing C_3 in both productions with the only string as which it can be rewritten, a (see production 4). This gives the productions $D_4 \rightarrow aB_2B_2$ and $D_4 \rightarrow aD_4B_2$. Productions 4 ($C \rightarrow a$) and 5 ($B \rightarrow b$) are already of the required form. Recapitulating, at this point we have the following productions: $S_1 \rightarrow C_3B_2$, $S_1 \rightarrow C_3D_4$, $D_4 \rightarrow aD_4B_2$, $D_4 \rightarrow aB_2B_2$, $C_3 \rightarrow a$, $B_2 \rightarrow b$.[1]

[1] This example is relatively simple, as the case where the two subscripts are equal does not occur. In that case a special procedure is applied, and it is this which is the heart of Greibach's proof. We refer the reader to her original article, or to Hopcroft and Ullman (1969).

The first two productions are not yet of the Greibach normal-form; we thus replace the variable C_3 in these two productions with the only string as which it can be rewritten, a, thus also eliminating the need for the production $C_3 \rightarrow a$. In this way we arrive at the following productions for grammar G'' in Greibach normal-form (the subscripts are no longer necessary):

1. $S \rightarrow aB$
2. $S \rightarrow aD$
3. $D \rightarrow aBB$
4. $D \rightarrow aDB$
5. $B \rightarrow b$

Grammar G'' will thus generate sentence a^3b^3 as follows: $S \Rightarrow aD \Rightarrow aaDB \Rightarrow aaaBBB \Rightarrow aaaBBb \Rightarrow aaaBbb \Rightarrow aaabbb$. The tree diagram for this derivation is given in Figure 2.3.c.

2.3.3. *Self-embedding*

The economical production forms for context-free languages, especially the Chomsky normal-form ($A \rightarrow a$, $A \rightarrow BC$), show the minute difference in type of production which distinguishes context-free and regular languages (the regular form is $A \rightarrow a$ or $A \rightarrow bC$). What is the characteristic difference between these two classes of languages? One important property characterizing all nonregular context-free languages and absent in regular languages is that of SELF-EMBEDDING.

A context-free grammar $G = (V_N, V_T, P, S)$ is called self-embedding if there is a variable B in V_N, and elements α and γ in V^+ such that $B \overset{*}{\Rightarrow} \alpha B \gamma$.

Thus there is a variable B which, by application of the productions, can be rewritten as a string in which B itself occurs, but neither at the beginning nor at the end. The definition implies that a regular grammar is not self-embedding, since nonterminal symbols occur in regular derivations only at the end of a string.

A language is self-embedding if all grammars generating it are self-embedding.

It is therefore not sufficient that one of its grammars be self-embedding, as some self-embedding grammars merely generate

regular languages. This is the case with the grammar of Example 2.2. Its productions are $S \to aSa$ $S \to aa$, $S \to a$, generating the language $\{a^n | n \geqslant 1\}$. The language is regular, but the grammar is self-embedding because $S \Rightarrow aSa$. The same example showed that G', with productions $S \to aS$ and $S \to a$, generates the same language. Grammar G' is not self-embedding, and generates $L(G)$, and consequently, by definition, $L(G)$ is not self-embedding.

THEOREM 2.8. All nonregular context-free languages are self-embedding, and all self-embedding languages are nonregular.

PROOF. The second member of this theorem follows directly from the definitions. A self-embedding language is generated exclusively by self-embedding grammars; a self-embedding grammar is, as we have seen, nonregular. Therefore a self-embedding language is nonregular.

The first member of the theorem can be otherwise formulated. It must be shown that all grammars of a nonregular context-free language are self-embedding. This can be done by proving that if a language L is generated by a non-self-embedding grammar, it is necessarily a regular language. To do this, however, we shall have to refer to a lemma which in turn will be easy to prove after the discussion of finite automata in Chapter 4.

Lemma. Let L_1 and L_2 be regular languages, and a be a terminal element of L_1. Let L_3 be a language consisting of all sentences in L_1 in which the element a does not occur, as well as all strings which can be obtained by replacing the element a in the remaining sentences of L_1 with a sentence of L_2 (if L_2 is infinite, this can be done in an infinite number of ways). L_3 is then a regular language.

We shall now prove that a language generated by a grammar which is not self-embedding is a regular language. Let language L be generated by a grammar G which is not self-embedding and which contains the variables $A_1, A_2, \ldots, A_n$.

Let us assume that grammar G is connected: a grammar is CONNECTED if for each pair of variables A_i, A_j ($i,j = 1, 2, \ldots, n$, where n is the number of variables in the grammar), there are strings α_1 and α_2 in V^* such that $A_i \overset{*}{\Rightarrow} \alpha_1 A_j \alpha_2$. Let A_i, A_j be an

arbitrary pair of variables in G. Since G is connected, we have $A_i \overset{*}{\Rightarrow} \varphi_1 A_j \varphi_2$ for some pair φ_1, φ_2. Let us further assume that $|\varphi_1| > 0$. Let A_k, A_l also be an arbitrary pair of variables in G, with $A_k \overset{*}{\Rightarrow} \psi_1 A_l \psi_2$, and assume that $|\psi_2| > 0$. Let us examine the consequences of the two conditions $|\varphi_1| > 0$ and $|\psi_2| > 0$. It follows from the fact that G is connected that strings ω_1 and ω_2 exist such that $A_j \overset{*}{\Rightarrow} \omega_1 A_k \omega_2$ and that one can therefore make the following derivation in G: $A_i \overset{*}{\Rightarrow} \varphi_1 A_j \varphi_2 \overset{*}{\Rightarrow} \varphi_1 \omega_1 A_k \omega_2 \varphi_2 \overset{*}{\Rightarrow} \varphi_1 \omega_1 \psi_1 A_l \psi_2 \omega_2 \varphi_2$. But it follows from the same fact that $A_l \overset{*}{\Rightarrow} \xi_1 A_i \xi_2$. Therefore we have the following derivation in G: $A_i \overset{*}{\Rightarrow} \varphi_1 \omega_1 \psi_1 \xi_1 A_i \xi_2 \psi_2 \omega_2 \varphi_2$. It follows from the two additional conditions that A_i is self-embedding in G. But G is not self-embedding. At least one of the additional conditions must not be valid for a grammar to be connected, i.e. if a connected grammar has a pair of variables A_i, A_j, for which $A_i \overset{*}{\Rightarrow} \alpha_1 A_j \alpha_2$ with $|\alpha_1| > 0$, then there is no pair of variables for which $|\alpha_2| > 0$, including the pair A_i, A_j. Therefore all the derivations in G are either all of the forms xA and x, or all of the forms Ax and x. It follows from Theorem 2.2. that G is regular. Theorem 2.8. is thus valid for connected grammars. We must show that the theorem also holds for grammars which are not connected.

A nonconnected grammar has at least one pair of variables A_i, A_j, for which it is not the case that $A_i \overset{*}{\Rightarrow} \alpha_1 A_j \alpha_2$ for some pair α_1, α_2. We shall prove the theorem for such cases by Mathematical induction, in two steps: (i) we must first show that the theorem is valid for grammars with only one variable, S; (ii) then we assume that it holds for all grammars with less than n variables (the induction-hypothesis) and prove that in that case the theorem also holds for grammars with n variables. It follows from (i) and (ii) that the theorem holds for all grammars with one or more variables.
(i) G has only one variable, S. The only possible pair of variables is thus S,S, and consequently there is no pair α_1 and α_2 such that $S \overset{*}{\Rightarrow} \alpha_1 S \alpha_2$. Since all productions are of the form $S \to x$, language $L(G)$ is finite; on the basis of Theorem 2.3. it is regular. The theorem is thus valid for nonconnected grammars with one variable.
(ii) Let us assume that the theorem is valid for all grammars with

less than n variables (the induction-hypothesis). Take grammar G with n variables $A_1, A_2, \ldots, A_n$, where $S = A_1$. Because S is the start symbol, it is true for all variables which may occur in the derivation of a sentence (we suppose without loss of generality that G contains no "dummy" variables from which no derivation is possible) that $S \overset{*}{\Rightarrow} \varphi_1 A_j \varphi_2$ $(j > 1)$ and for strings φ_1 and φ_2 in V^*. Because G is not connected, there must be a variable A_i such that it is not true that $A_i \overset{*}{\Rightarrow} \alpha_1 S \alpha_2$ for a pair α_1, α_2. Otherwise we would have $A_i \overset{*}{\Rightarrow} \alpha_1 \varphi_1 A_j \varphi_2 \alpha_2$, but we know that there is at least one pair A_i, A_j for which this is not the case.

Let us first examine the case where $i > 1$, that is, where $A_i \neq S$. We can construct a grammar G' with $n - 1$ variables by removing all productions of the form $A_i \to \psi$ from G, and by replacing A_i in all productions with a new terminal element a. From the induction-hypothesis it follows that $L(G')$ is regular. Next let us examine the set K of terminal strings x for which $A_i \overset{*}{\Rightarrow} x$ in G, $K = \{x | A_i \overset{*}{\Rightarrow} x\}$. This set can be generated by a grammar G'' which includes all the productions of G except those containing S ($A_i \overset{*}{\Rightarrow} \alpha_1 S \alpha_2$ is impossible), and with A_i as start symbol. Because G'' has fewer than n variables, K is regular (by the induction-hypothesis). $L(G)$, however, is precisely the language which results from the replacement of the element a in the strings of $L(G')$ with strings x from K. It follows from the lemma that $L(G)$ is regular.

Let us now consider the case where $A_i = S$. Take the productions in G of the form $S \to \alpha$; an arbitrary α_i can be rewritten as a string of terminal and/or nonterminal elements $\xi_1, \xi_2, \ldots, \xi_m$. For each ξ_j in α_i we can define a set of strings L_j for which $\xi_j \overset{*}{\Rightarrow} x$ on the basis of the productions in G. Thus $L_j = \{x | \xi_j \overset{*}{\Rightarrow} x\}$. From the induction-hypothesis it follows that L_j is regular for all j's. Let K_i be the set of strings y for which $\alpha_i \overset{*}{\Rightarrow} y$, i.e. $K_i = \{y | \alpha_i \overset{*}{\Rightarrow} y\}$. From the composition of α_i it follows that each y consists of a sequence of x's respectively taken from $L_1, L_2, \ldots, L_m$, all of which are regular. From Theorem 2.5. it then follows that K_i is regular. $L(G)$ is the union of all K_i's. As a consequence of Theorem 2.4., therefore, $L(G)$ is itself regular. This completes the proof of Theorem 2.8.

2.3.4. *Ambiguity*

The generation of a sentence by a context-free grammar can be represented by a tree diagram. This however does not mean that a given tree diagram corresponds to only one way in which a sentence can be derived.

EXAMPLE 2.5. Let G be a context-free grammar with the following productions:

1. $S \to AB$
2. $S \to CD$
3. $S \to bc$
4. $A \to a$
5. $B \to Sd$
6. $C \to aS$
7. $D \to d$

The sentence *abcd* can be derived from this grammar as follows: $S \Rightarrow AB \Rightarrow aB \Rightarrow aSd \Rightarrow abcd$. The corresponding derivation tree is shown in Figure 2.4. There are, however, other derivations of *abcd* which correspond to the same tree, for example, the derivation $S \Rightarrow AB \Rightarrow ASd \Rightarrow Abcd \Rightarrow abcd$, where the productions are applied in a different order. This cannot be detected in the tree diagram, which fact corresponds to our intuition that the two derivations determine the same syntactic structure. Therefore we cannot consider this to be a case of real ambiguity.

In order to define ambiguity in terms of derivations, we must introduce the concept of LEFTMOST DERIVATION. We can speak of a leftmost derivation of x if at each step in the derivation $S \overset{*}{\Rightarrow} x$ it is the variable farthest to the left of the string which is rewritten. A leftmost derivation of the sentence *abcd* can begin with $S \Rightarrow AB$. At this stage the leftmost variable is A; thus the following step will be $AB \Rightarrow aB$. The leftmost variable is now B, and the next

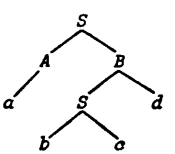


Fig. 2.4. Derivation Tree for the Sentence *abcd* (Example 2.5.).

step is $aB \Rightarrow aSd$, and the final step, $aSd \Rightarrow abcd$. The first derivation given in this example was in fact a leftmost derivation. It is clear that every tree diagram corresponds to no more than one leftmost derivation, and every leftmost derivation with only one tree diagram.

A grammar G is AMBIGUOUS if there is a sentence in $L(G)$ for which there are two or more leftmost derivations.

The grammar given in Example 2.5. is ambiguous, for sentence *abcd* has another leftmost derivation: $S \Rightarrow CD \Rightarrow aSD \Rightarrow abcD \Rightarrow abcd$. The tree diagram for this derivation is shown in Figure 2.5.

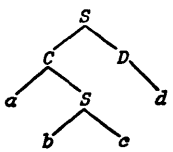


Fig. 2.5. Alternative Derivation Tree for the Sentence *abcd* (Example 2.5.).

A language L is (inherently) ambiguous if all grammars which generate it are ambiguous.

Although grammar G of Example 2.5. is ambiguous, $L(G)$ is not. Language $L(G)$ consists of sentences a^*bcd^*, which can be generated by grammar G' with productions $S \rightarrow aSd$ and $S \rightarrow bc$; G' is not ambiguous. Languages exist, however, which are inherently ambiguous. An example is the union of $\{a^ib^jc^j\}$ and $\{a^ib^jc^j\}$, briefly noted $L = \{a^ib^jc^k | i = j \text{ or } j = k, \text{ where } i, j, k > 1\}$. Any grammar for L will generate sentences with $i = j$ by a different process than sentences with $j = k$. But then sentences with $i = j = k$ can be generated by both processes.

2.3.5. *Linear Grammars*

A production is called LINEAR if it is of the form $A \rightarrow xBy$, i.e. if the string derived contains only one variable. A RIGHT-LINEAR production has the form $A \rightarrow xB$; a LEFT-LINEAR production has the form $A \rightarrow Bx$.

A grammar is linear if each of its productions is either linear or of the form $A \rightarrow x$; a grammar is right-linear if each of its productions is either right-linear or of the form $A \rightarrow x$; a grammar is left-linear if each of its productions is either left-linear or of the form $A \rightarrow x$.

It follows from Theorem 2.1. that a right-linear grammar generates a regular language. Left-linear grammars also generate only regular languages.

An example of a linear grammar is G' mentioned in the preceding paragraph, with productions $S \rightarrow aSd$ and $S \rightarrow bc$. The language generated by it, $\{c^n bcd^n\}$, is not regular; it is therefore self-embedding. Although the class of linear grammars has a greater generative capacity than the class of regular grammars, it does not coincide with the class of context-free languages.

THEOREM 2.9. There are context-free languages for which no linear grammar exists.

For proof of this theorem[2] we refer the reader to Chomsky and Schützenberger (1963). An example of a context-free language for which no linear grammar can be found is language L with sentences $a^{m_1}b^{m_1}a^{m_2}b^{m_2} \ldots a^{m_k}b^{m_k}$, where $m_i > 0$ and $k > 0$, thus strings of alternating sequences of a's and b's, where each sequence of b's is as long as the sequence of a's which precedes it. A grammar for this language has the productions $S \rightarrow SS$, $S \rightarrow aSb$, $S \rightarrow ab$. The first of these productions is not linear. All other grammars for this language likewise have at least one nonlinear production.

2.4. CONTEXT-SENSITIVE GRAMMARS

2.4.1. *Context-sensitive Productions*

The definition of context-sensitive grammars (grammars in which all productions are of the form $\alpha \rightarrow \beta$ where $|\alpha| \leqslant |\beta|$) does not indicate in what way such grammars are "sensitive to context".

[2] Though with an incorrect example grammar, as pointed out to me by Geoffrey Pullum.

The original definition given by Chomsky (1959a) was in fact different from the present one. He defined context-sensitive grammars (*CSG*) as grammars the productions of which have the form $\alpha_1 A \alpha_2 \rightarrow \alpha_1 \beta \alpha_2$, where α_1 and α_2 are elements of V^*, and β is an element of V^+. Thus A can be replaced by β only if A appears in the context $\alpha_1 - \alpha_2$. This type of context-sensitive production can also be written as $A \rightarrow \beta / \alpha_1 - \alpha_2$. In spite of the change of definition, the following theorem remains valid.

THEOREM 2.10. The class of languages generated by grammars exclusively containing context-sensitive productions is the class of type-1 languages.

PROOF. Let G_1 be a type-1 grammar, and G_c be a grammar exclusively containing context-sensitive productions. Every G_c is evidently also a G_1, because for all productions $\alpha \rightarrow \beta$ in G_c it is true that $|\alpha| \leqslant |\beta|$. However it must likewise be shown that for every G_1 there is an equivalent G_c.

Let $G_1 = (V_N, V_T, P, S)$ be a type-1 grammar. There is a grammar $G' = (V'_N, V'_T, P', S')$ equivalent to it, where all the productions $\alpha \rightarrow \beta$ in P' have the following "normal-form": either both α and β are strings exclusively containing variables, or α and β are of the forms A and a respectively (i.e. the productions are of the type $A \rightarrow a$). This will become evident from the following. Let V'_N consist of all the elements in V_N as well as an additional variable X_a for each element a in V_T, thus $V'_N = V_N \cup \{X_a | a \in V_T\}$. To compose P' we must change the productions of P in such a way that every terminal element a in them is replaced by X_a, then add productions $X_a \rightarrow a$ for every a in V_T. Thus all productions in P' are of the "normal-form" (note that this normal-form can also be used for all type-0 grammars), and $L(G') = L(G_1)$.

We must now find a grammar G'' which contains only context-sensitive productions, and is equivalent to G'. Let $\alpha \rightarrow \beta$ be a production in P', with $\alpha = A_1 A_2 \ldots A_m$, and $\beta = B_1 B_2 \ldots B_n$, where $n \geqslant m$. We replace this production with the following set of equivalent context-sensitive productions in P'':

$$\begin{array}{lll} A_1 \to A_1' \,/\, - A_2A_3 \ldots A_m & & A_1' \to B_1 \\ A_2 \to A_2' \,/\, A_1' - A_3 \ldots A_m & \text{and} & A_2' \to B_2 \\ \quad\vdots & & \quad\vdots \\ A_m \to A_m' \,/\, A_1' \ldots A_{m+1}' - & & A_m' \to B_mB_{m+1} \ldots B_n \end{array}$$

The first group of context-sensitive productions (A_1 though A_m) replaces $\alpha = A_1A_2 \ldots A_m$ to a string of new variables $A_1'A_2' \ldots A_m'$. This can in turn be replaced by $B_1B_2 \ldots B_n$ by way of the second group of context-sensitive productions (A_1' through A_m') if $n \geqslant m$. When all the productions of P' have been replaced in this way by sets of context-sensitive productions, and V_N'' includes V_N' and the newly introduced variables, then G'' is equivalent to G' and consequently also to G'. G'', however, is a G_c.

EXAMPLE 2.6. The production $CD \to DC$ is of type-1 form. Application of the procedure mentioned above yields the following set of context-sensitive productions equivalent to $CD \to DC$:

1. $C \to C'/-D$
2. $D \to D'/C'-$
3. $C' \to D$
4. $D' \to C$

An advantage of a type-1 grammar in context-sensitive form (that is, containing productions exclusively in context-sensitive form) is that the derivation of a sentence in it can be represented by means of a tree diagram. Context-sensitive productions, in effect, replace only one variable in the string at each step; each step, therefore, corresponds to the branches leaving only one node. This will be illustrated by the following example.

EXAMPLE 2.7. Let us examine the derivation of sentence *aabbccdd* in grammar G of Example 1.5. G contains the following productions:

1. $S \to ESF$
2. $S \to abcd$
3. $Ea \to aE$
4. $dF \to Fd$
5. $Eb \to abb$
6. $cF \to ccd$

As a first step we replace grammar G with grammar G', containing the following "normal form" productions, obtained by application of the procedure explained in the proof of Theorem 2.10.:

1. $S \to ESF$
2. $S \to X_aX_bX_cX_d$
3. $EX_a \to X_aE$
4. $X_a \to a$
5. $X_dF \to FX_d$
6. $X_b \to b$
7. $EX_b \to X_aX_bX_b$
8. $X_cF \to X_cX_cX_d$
9. $X_c \to c$
10. $X_d \to d$

The productions are now replaced by context-sensitive productions where necessary by application of the procedure given in Example 2.6. This yields the following productions; productions 3-6 and 8-11 were obtained by means of this procedure:

1. $S \to ESF$
2. $S \to X_aX_bX_cX_d$
3. $E \to E' \mid -X_a$
4. $X_a \to X'_a \mid E'-$
5. $E' \to X_a$
6. $X'_a \to E$
7. $X_a \to a$
8. $F \to F' \mid X_d-$
9. $X_d \to X'_d \mid -F'$
10. $F' \to X_d$
11. $X'_d \to F$
12. $X_b \to b$
13. $E \to X_aX_b \mid -X_b$
14. $F \to X_cX_d \mid X_c-$
15. $X_c \to c$
16. $X_d \to d$

These productions can be used to derive the sentence *aabbccdd* in the following way (the numbers over the arrows refer to the productions applied):

$$S \overset{1}{\Rightarrow} ESF \overset{2}{\Rightarrow} EX_aX_bX_cX_dF \overset{3}{\Rightarrow} E'X_aX_bX_cX_dF$$
$$\overset{4}{\Rightarrow} E'X'_aX_bX_cX_dF \overset{5}{\Rightarrow} X_aX'_aX_bX_cX_dF \overset{6}{\Rightarrow} X_aEX_bX_cX_dF$$
$$\overset{8}{\Rightarrow} X_aEX_bX_cX_dF' \overset{9}{\Rightarrow} X_aEX_bX_cX'_dF' \overset{10}{\Rightarrow} X_aEX_bX_cX'_dX_d$$
$$\overset{11}{\Rightarrow} X_aEX_bX_cFX_d \overset{13}{\Rightarrow} X_aX_aX_bX_bX_cFX_d \overset{14}{\Rightarrow} X_aX_aX_bX_bX_cX_cX_dX_d$$
$$\overset{7,12,15,16}{\Longrightarrow} aabbccdd.$$

All sixteen productions have been used in this derivation. Figure 2.6., gives the corresponding tree diagram.

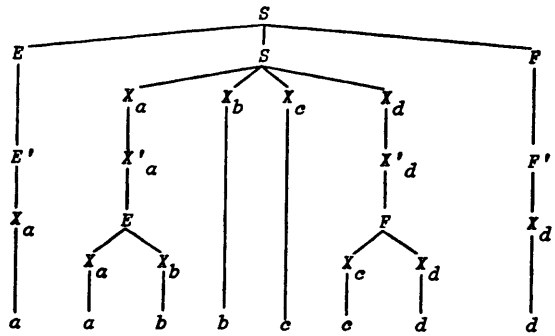


Fig. 2.6. Derivation Tree for the Sentence *aabbccdd* (Example 2.7.).

Nevertheless, tree diagrams for derivations in context-sensitive grammars are less exhaustive in illustrating the precise steps of derivation than tree diagrams for derivations in context-free grammars. More specifically, the diagrams do not show the contextual restrictions operative at the various steps of rewriting in a context-sensitive grammar, and it is possible that two derivations, based on different sets of productions, will be represented by the same tree diagram. For a context-sensitive derivation, as opposed to a context-free derivation, the "ambiguity of x" does not correspond to "more than one possible tree diagram for x".

2.4.2. *The Kuroda Normal-Form*

In the preceding paragraph two restricted forms of context-sensitive productions were discussed; they may be called normal-forms. The first of them contains two types of production, $\alpha \rightarrow \beta$ with α and β in V_N^+ and $|\alpha| \leqslant |\beta|$, and $A \rightarrow a$. The second is the context-sensitive form $A \rightarrow \beta / \alpha_1 - \alpha_2$, with α_1 and α_2 in V^* and β in V^+. We shall now introduce a third normal-form, developed by Kuroda, which is relevant not only to the discussion of the relationship between context-sensitive grammars and automata (chapter 6), but also to the proof of certain essential properties of transformational grammars (Volume II, chapter 5).

THEOREM 2.11. Every context-sensitive grammar is equivalent to

a context-sensitive grammar with productions exclusively in the following forms:

(i) $S \to SB$, (ii) $CD \to EF$, (iii) $G \to H$, (iv) $A \to a$, where the variables A, B, C, D, E, F, and H are different from the start symbol S (G may be identical to S).

PROOF. It is striking that no string in these production forms has more than two elements. We shall first show that if G is context-sensitive, there exists a grammar G' equivalent to it, in which for each production $\alpha \to \beta$, $|\alpha| \leqslant 2$, and $|\beta| \leqslant 2$. In the second place we will prove that there is a grammar G_n in the Kuroda normal-form which is equivalent to G'.

Let $G = (V_N, V_T, P, S)$ be a context-sensitive grammar. We already know that there is an equivalent grammar G'' of the first normal-form, i.e. with production types $A \to a$ and $\alpha \to \beta$, where α and β are strings of variables such that $|\beta| \geqslant |\alpha| > 0$. Suppose that the maximum length of any string of a production of G'' is n. We must construct a grammar $G''' = (V'''_N, V_T, P''', S)$ equivalent to G'' (and thus also to G), for which the maximum string length for any production is not greater than $n - 1$. To do so, we let P''' include all the productions of P'' where the string length is no greater than 2; the remaining productions have string lengths of 3 or more. (If $n = 1$ or $n = 2$, G'' already conforms to the limitation on string length and this step may be omitted.) Let $\alpha \to \beta$ be such a production; we write it then as

$$A\alpha' \to BCD\beta' \text{ (where } |\alpha'| \geqslant 0 \text{ and } |\beta'| \geqslant 0).$$

If $\alpha' = \lambda$, we create two new variables A_1 and A_2, and add the following productions to P''':

$$A \to A_1A_2$$
$$A_1 \to BC$$
$$A_2 \to D\beta'$$

If $|\alpha'| > 0$, α' can be replaced by $E\alpha''$. In that case we add the following productions to P''':

$AE \rightarrow A'E'$
$A' \rightarrow B$
$E'\alpha'' \rightarrow CD\beta'$

It is clear that in both cases no string length is greater than $n - 1$. If we follow this procedure for all the productions of P'' and add the resulting productions to P''', in virtue of the construction G''' will be equivalent to G'', and consequently also to G. By induction on n it follows that there is a grammar $G' = (V_N', V_T, P', S)$ in which the length of the strings in productions is limited to 2, and which is equivalent to G.

At this point we must show that there is a grammar G_n which is equivalent to G' and G, and which contains only productions of types (i) through (iv). Take grammar $G_n = (V_N^n, V_T, P^n, S')$, where $V_N^n = \{V_N' \cup S' \cup Q\}$. Thus we have added two new variables, one of which, S', is a new start symbol. The productions in P^n are the following:

1. $S' \rightarrow S'Q$
2. $S' \rightarrow S$
3. $QA \rightarrow AQ$ } for all variables A in G'
4. $AQ \rightarrow QA$ }
5. $A \rightarrow B$ for all productions $A \rightarrow B$ in G'
6. $A \rightarrow b$ for all productions $A \rightarrow b$ in G'
7. $AB \rightarrow CD$ for all productions $AB \rightarrow CD$ in G'
8. $AQ \rightarrow BC$ for all productions $A \rightarrow BC$ in G'

It is clear that the productions of G_n are subject to the same restriction of string length as the productions of G'; all strings in productions are of a length no greater than 2. Productions 1 through 8, moreover, are all of types (i) through (iv). (Note that the start symbol is S', while S is an ordinary variable.)

Finally, we must prove that G_n is equivalent to G'; to do so it will be necessary to show that if $x \in L(G_n)$, it is also true that $x \in L(G')$, as well as the inverse. (1) If $x \in L(G_n)$, then $S' \stackrel{*}{\Rightarrow} x$. When every S' in the derivation is replaced by S and all Q's are omitted, every step of the derivation is in G'. This may be seen

when the same operation is performed on the eight productions of G_n. The first and second productions become $S \to S$ (which adds nothing essential); the third and fourth productions become $A \to A$ (which is equally uninteresting); the fifth, sixth, and seventh productions remain unchanged, and the eighth production becomes $A \to BC$. Thus if $S' \overset{*}{\Rightarrow} x$, each step in the derivation of x can be simulated by the application of the productions of G', and therefore it is true that $x \in L(G')$.

(2) Let $x \in L(G')$; then $S \overset{*}{\Rightarrow} x$. It is true of every production $\alpha \to \beta$ in G' that it is either contained in G_n or has been replaced by a production of type 8, $AQ \to BC$. Therefore, in order to generate x in G_n we must see to it that there is exactly one Q available for each step of derivation in which a production of the type $A \to BC$ is involved. The Q must be placed directly to the right of the variable A to be rewritten. This can easily be done in G_n: we first count the number of steps in the derivation $S \overset{*}{\Rightarrow} x$ in which the situation occurs, for instance n times. We then begin the derivation of x in G_n by applying the first production n times; this may be written as $S' \overset{*}{\Rightarrow} S'Q^n$. Next we replace S' with S by means of the second production, thus $S'Q^n \Rightarrow SQ^n$. The rest of the derivation can proceed in the same way as the derivation $S \overset{*}{\Rightarrow} x$, except where the eighth type of production is involved. In this latter case we must move one Q to the position directly to the right of the variable to be rewritten; this is done by application of productions of the third and fourth types. The Q is then eliminated upon application of a production of the eighth type. In this way G_n can generate x.

It follows from (1) and (2) that $L(G_n) = L(G')$. Since G' is equivalent to G, G_n in the Kuroda normal-form is also equivalent to G. This concludes the proof of Theorem 2.11.

We would note in conclusion that Kuroda called his normal-form a "linear bounded grammar", analogous to the equivalent automaton of the same name (cf. chapter 6).

3

PROBABILISTIC GRAMMARS

3.1. DEFINITIONS AND CONCEPTS

Until now we have limited the concept of grammar to a system of rules according to which the sentences of a language may be generated. On the basis of such a concept one can distinguish differences in the sentences of a language only in their derivation, also called their STRUCTURAL DESCRIPTION. However one might also consider the differences in frequency with which sentence types occur in a language. One reason for doing so, as we shall see in chapter 8, is to facilitate the choice between two or more grammars which generate the same language. One might determine the efficiency of a grammar on the basis of the frequencies with which particular derivations or sentence types occur in a language. But the concept "efficiency" has not been clearly defined, and the usefulness of a probabilistic interpretation of it will have to be considered in each concrete situation. We shall return to this subject in chapter 8.

We shall limit our discussion in the present chapter to an extension of the concept "grammar" which will enable us to describe the probability of occurrence of sentences in a language. Therefore, we shall first define the concept of a probabilistic grammar.

A PROBABILISTIC GRAMMAR G is a system (V_N, V_T, P, S) in which:

(1) V_N (the nonterminal vocabulary), V_T (the terminal vocabulary), and P (the productions) are finite, nonempty sets.
(2) $V_N \cap V_T = \emptyset$.

(3) Let $V_N \cup V_T = V$; P is composed of ordered groups of three elements $(\alpha_i, \beta_j, p_{ij})$, ordinarily written $\alpha_i \overset{p_{ij}}{\rightarrow} \beta_j$, where $\alpha_i \in V^+$, $\beta_j \in V^*$, and p_{ij} is a real number indicating the probability that a given string α_i will be rewritten as β_j. The number p_{ij} is called the PRODUCTION PROBABILITY of $\alpha_i \rightarrow \beta_j$.

(4) $S \in V_N$.

This definition differs from the original definition of grammar only in that a probability is assigned to every production.

A probabilistic grammar is NORMALIZED if for every production $\alpha_i \overset{p_{ij}}{\rightarrow} \beta_j$, it is true that $\sum_j p_{ij} = 1$ for every α_i in the productions. This means that if α_i occurs in a derivation, the total chance that α_i will be rewritten by means of some production is equal to 1. A production whose probability is equal to 0 cannot be used; it can simply be excluded from P. The reason for allowing the possibility that $p = 0$ is only of practical interest in some calculations. In the following, however, we shall suppose that every $p_{ij} > 0$ unless otherwise mentioned.

We use the notation $\alpha \overset{p}{\Rightarrow} \beta$ for a derivation $\alpha \overset{p_1}{\Rightarrow} \xi_1 \overset{p_2}{\Rightarrow} \xi_2 \ldots \overset{p_n}{\Rightarrow} \beta$, where each step is the result of the application of one production, and where $p = f(p_1, p_2, \ldots, p_n)$. The analogy with standard notation is obvious, but to avoid crowding symbols above the arrow, we shall omit the asterisk, except where doing so might lead to confusion, and write $\alpha \overset{p}{\Rightarrow} \beta$.

Function f is determined by the interdependence, or lack of it, between the various steps of the derivation. A probabilistic grammar is called UNRESTRICTED if the steps of a derivation in it are mutually independent; in this case $p = p_1 \cdot p_2 \cdot \ldots \cdot p_n$. As no considerable literature exists on the subject of restricted probabilistic grammars, we shall limit our discussion to unrestricted probabilistic grammars. In applications of the theory, however, it will be necessary to estimate the validity of the presupposition that the productions are mutually independent.

A SENTENCE generated by a probabilistic grammar is a finite string s of terminal elements, where $S \overset{p}{\Rightarrow} s$ and $p > 0$.

A probabilistic grammar G is AMBIGUOUS if at least one sentence can be derived in it in more than one way. A sentence is k-times ambiguous if there are k derivations $S \overset{p_1}{\Rightarrow} s$, $S \overset{p_2}{\Rightarrow} s$, ..., $S \overset{p_k}{\Rightarrow} s$.

A PROBABILISTIC LANGUAGE L, generated by a probabilistic grammar G, is the set of pairs $(s, p(s))$, where: (1) s is a sentence generated by G, and (2) $p(s) = \sum_{i=1}^{k} p_i(s)$ where k is the number of different ways in which s can be derived from S. We call $p(s)$ the PROBABILITY of s in L. A probabilistic language can also be defined, without reference to a grammar, as a subset of V_T^* for which a probability distribution has been defined (V_T is any finite vocabulary).

Two probabilistic grammars G_1 and G_2 are EQUIVALENT if they generate the same probabilistic language L, i.e. the same set of pairs $(s, p(s))$. Notice that equivalence here requires also that the probabilities of the sentences be the same.

A probabilistic language $L = \{(s, p(s))\}$ is NORMALIZED if $\sum_{s \in L} p(s) = 1$. This means that the language has a total probability of 1. We shall see later that a normalized probabilistic grammar need not generate a normalized probabilistic language.

3.2. CLASSIFICATION

Probabilistic grammars may be classified as follows in a way completely analogous to that used in Chapter 2.

Type-0 probabilistic grammars are all probabilistic grammars which satisfy the definition given above. Type-1 or CONTEXT-SENSITIVE probabilistic grammars are those probabilistic grammars in which, for all productions $\alpha_i \overset{p_{ij}}{\rightarrow} \beta_j$, it is true that $|\alpha_i| \leqslant |\beta_j|$. Type-2 or CONTEXT-FREE probabilistic grammars are those probabilistic grammars in which, for all productions $\alpha_i \overset{p_{ij}}{\rightarrow} \beta_j$, it is true that $\alpha_i = A_i \in V_N$. Type-3 or REGULAR probabilistic grammars are type-2 probabilistic grammars whose productions are exclusively of the forms $A \overset{p}{\rightarrow} aB$ and $A \overset{p}{\rightarrow} a$.

It is obvious that this classification is completely independent

of the probabilistic aspect of the grammars. This is also true of the classification of probabilistic LANGUAGES generated by probabilistic grammars. Thus we have type-0 probabilistic languages, type-1 or context-sensitive probabilistic languages, type-2 or context-free probabilistic languages, and type-3 or regular probabilistic languages.

In the present chapter only regular and context-free probabilistic grammars will be treated, as no results on the other two types are yet available.

3.3. REGULAR PROBABILISTIC GRAMMARS

Three theorems will be treated in this paragraph. The first of them is of direct practical interest. The second, on the other hand, appears to be somewhat alarming from a practical point of view, but the third, which has not as yet been proven, suggests that things might not be as problematic as they seem.

THEOREM 3.1. Every normalized regular probabilistic grammar generates a normalized regular probabilistic language.

In such a case, the probabilistic grammar is said to be CONSISTENT, and the theorem is therefore called a CONSISTENCY-THEOREM.

The theorem is of practical interest in determining the frequencies of sentences in a language. To do so one would wish to be certain that the sum of the corresponding probabilisties is equal to 1. The theorem states that this is guaranteed if the regular grammar in question is normalized.

The proof of this theorem supposes some acquaintance with matrix algebra. For readers who prefer to omit it we shall first present an example which holds the essence of the proof without requiring knowledge of matrix algebra. The general proof will be given later.

EXAMPLE 3.1. Let G be a regular probabilistic grammar with the following productions:

1. $S \xrightarrow{\frac{1}{2}} a$
2. $S \xrightarrow{\frac{1}{2}} aB$
3. $B \xrightarrow{\frac{1}{3}} bA$
4. $B \xrightarrow{\frac{2}{3}} b$
5. $A \xrightarrow{1} a$

G is normalized because for every variable the total chance of being rewritten is equal to 1. Only three sentences can be generated by G: *a*, *ab*, *aba*. The derivations with their respective probabilities are as follows:

$S \xRightarrow{\frac{1}{2}} a$ ..... $p(a) = \frac{1}{2}$

$S \xRightarrow{\frac{1}{2}} aB \xRightarrow{\frac{2}{3}} ab$ ..... $p(ab) = \frac{1}{2} \cdot \frac{2}{3} = \frac{1}{3}$

$S \xRightarrow{\frac{1}{2}} aB \xRightarrow{\frac{1}{3}} abA \xRightarrow{1} aba$ ..... $p(aba) = \frac{1}{2} \cdot \frac{1}{3} \cdot 1 = \frac{1}{6}$

$L(G)$ is evidently normalized, because $\sum_{s \in L(G)} p(s) = \frac{1}{2} + \frac{1}{3} + \frac{1}{6} = 1$. On the basis of this example we shall now show that there is a simple method for determining the chance that a regular probabilistic grammar will generate sentences up to a certain length. To do so we present the probabilities of the productions in G in matrix form [1] as follows:

	S	A	B	V_T
S	0	0	$\frac{1}{2}$	$\frac{1}{2}$
A	0	0	0	1
B	0	$\frac{1}{3}$	0	$\frac{2}{3}$
V_T	0	0	0	1

$= C$

Let us examine the first row (row-element S). It shows the chances for the respective column-elements to appear in direct or "one-

[1] A matrix is a rectangular grid with one or more rows and one or more columns. Each row is denoted by a ROW-ELEMENT x_i, and each column by a COLUMN-ELEMENT y_i. At the intersection of row i and column j is the MATRIX-ELEMENT a_{ij}.

step" derivations from S. There are only two productions for rewriting S, $S \overset{1}{\rightarrow} aB$ and $S \overset{1}{\rightarrow} a$. The matrix-element under B in row S has the value $\frac{1}{2}$ because of the first of these productions, and the matrix-element under V_T in the same row has the value $\frac{1}{2}$ because of the second production. Column V_T thus serves for all productions in which a variable is rewritten as a terminal element, regardless of which terminal element it is. Row A shows how the variable A can be rewritten in one step, and with what probability, thus A can be rewritten only as a terminal element, with probability 1. Row B shows to which elements the variable B can be rewritten, and with what probability, thus it can be rewritten as A with probability $\frac{1}{3}$ and as a terminal element with probability $\frac{2}{3}$. The fourth row, row V_T, is added to the matrix for further calculations; it is composed of zeros, except the rightmost element which has the value 1.

This matrix, which we shall call matrix C, has a pleasant property which may be explained as follows. We know that by definition sentences are derived from S. If we wish to know the chance for a sentence with length 1, we look at row S under V_T, and find the value $\frac{1}{2}$. What then is the chance for a sentence of length 1 or 2? Such sentences are derived by going from S to V_T by two steps at most. The variables S, A, or B may be present in the first derived string. Consequently there are four possibilities of arriving at a sentence with a length of 2:

(1) From S a string is derived in which S is present, then S is replaced by a terminal element. One can immediately see in the matrix that these two steps have respective probabilities of 0 and $\frac{1}{2}$. The total chance of such a derivation is thus $0 \cdot \frac{1}{2} = 0$.

(2) From S the variable A is first derived, then a terminal element is derived from A. The chance for this is $0 \cdot 1 = 0$.

(3) From S a string is derived with the variable B, then a terminal element is derived from B. The chance for this is $\frac{1}{2} \cdot \frac{2}{3} = \frac{1}{3}$.

(4) A terminal element is directly derived from S. The chance for this is $\frac{1}{2}$. The total chance for a sentence with length 1 or 2 is the

sum of these four probabilities, $0 + 0 + \frac{1}{3} + \frac{1}{2} = \frac{5}{6}$. This is precisely the chance for the sentence a $(\frac{1}{2})$ plus the chance for the sentence ab $(\frac{1}{3})$, the only two sentences of the grammar in this category.

This operation can also be carried out systematically by means of MATRIX-MULTIPLICATION. The four steps which we have just performed correspond to the multiplication in pairs of the elements in row S with the elements in column V_T, followed by the addition of the four products: $(0 \cdot \frac{1}{2}) + (0 \cdot 1) + (\frac{1}{2} \cdot \frac{2}{3}) + (\frac{1}{2} \cdot 1) = \frac{5}{6}$. We say then that the row-vector S is multiplied by the column-vector V_T. Let us make a new matrix C^2, and put the result $\frac{5}{6}$ at the intersection of row S and column V_T. The remaining matrix-elements of C^2 are obtained in a similar way, that is the multiplication of a given row-vector in C with a given column-vector in C yields the matrix-element in C^2 for the intersection of the row and column in question. For example, the matrix-element in C^2 for the intersection of row S and column A is $\frac{1}{6}$. This is obtained by multiplying the row-vector S in C by the column-vector A: $(0 \cdot 0) + (0 \cdot 0) + (\frac{1}{2} \cdot \frac{1}{3}) + (\frac{1}{2} \cdot 0) = \frac{1}{6}$. The value $\frac{1}{6}$ means that there is one chance out of six of deriving a string with A from S in no more than two steps. Matrix C^2 is called the square of matrix C.

	S	A	B	V_T
S	0	$\frac{1}{6}$	0	$\frac{5}{6}$
A	0	0	0	1
B	0	0	0	1
V_T	0	0	0	1

$= C^2$

By multiplying C by C^2 (multiplying the row-vectors in C by the column-vectors in C^2) we obtain matrix C^3:

	S	A	B	V_T
S	0	0	0	1
A	0	0	0	1
B	0	0	0	1
V_T	0	0	0	1

$= C^3$

In row S under V_T we find the value 1. This means that the chance of obtaining a sentence the length of which is three or smaller is equal to 1. The grammar, as we have observed, generates no longer sentences.

In this example we see that the critical matrix-element in row S under V_T increases with the power of the matrix from $\frac{1}{2}$ to $\frac{5}{6}$ to 1. The proof of Theorem 3.1 consists of showing that this is a generally valid theorem for matrices such as matrix C. By increasing the power of the matrix, i.e. the sentence length, the critical element approaches the value 1. The sum of the chances for all sentences, i.e. for the sentences of all lengths, is thus equal to 1, and $L(G)$ is normalized.

PROOF. Let G be a normalized regular probabilistic grammar. We suppose that G has no redundant variables, i.e. for each $A \in V_N$ there is at least one production $A \xrightarrow{p} a$, $a \in V_T$, for which $p > 0$. This supposition implies no loss of generality (cf. Huang and Fu 1971). Let us define a matrix $C = [c_{ij}]$, $i, j = 1, 2, \ldots, n+1$, as follows:

$$c_{ij} = \sum_{a \in V_T} p(A_i \to aA_j) \qquad \text{for } i, j \leqslant n, \text{ and where } p \text{ is the production probability of } A_i \to aA_j.$$

$$c_{ij} = \sum_{a \in V_T} p(A_i \to a) \qquad \text{for } i \leqslant n, j = n+1$$

$$c_{ij} = 0 \qquad \text{for } i = n+1, j \leqslant n$$

$$c_{n+1,n+1} = 1$$

C is a stochastic matrix [1] because for each row the sum of the elements is equal to 1, and G is normalized. The right hand column-vector in matrix C^k shows the probability that a string of k or fewer elements will be derived from the variable A_i. If $A_1 = S$, then $c^k_{1,n+1}$ is the probability that the grammar generates a sentence of k or fewer elements. We are interested in the value of $c^k_{1,n+1}$ when $k \to \infty$, i.e. the sum of the probabilities of all sentences

[1] A STOCHASTIC MATRIX is a square matrix, the matrix-elements of which are not negative, and the sums of the rows of which are equal to 1 (cf. Feller 1968).

generated by the grammar. We have supposed that it is true of every variable A that $\sum_{a \in V_T} p(A \to a) > 0$, that is, that there are no redundant variables. C may therefore be written as $C = \left[\begin{array}{c|c} A & B \\ \hline 0 & 1 \end{array}\right]$, where all the elements of column-vector B have a value > 0. Then $C^2 = \left[\begin{array}{c|c} A^2 & AB+B \\ \hline 0 & 1 \end{array}\right]$, and in general, $C^k = \left[\begin{array}{c|c} A^k & (A^{k-1}+A^{k-2}\ldots A^\circ)B \\ \hline 0 & 1 \end{array}\right]$ $= \left[\begin{array}{c|c} A^k & D \\ \hline 0 & 1 \end{array}\right]$. But for each of the row-vectors in A, the sum of the row-elements is smaller than 1, and consequently $\lim_{k \to \infty} A^k = 0$. But C^n is a stochastic matrix because C is a stochastic matrix (this theorem is treated in Feller 1968), and thus for every row in C^k the sum of the row elements is also equal to 1. The limit of each of the row-vectors in C^k is thus $[0\ 0\ \ldots\ 0\ 1]$, and thus $\lim_{k \to \infty} c_{1,n+1} = 1$ as we set out to prove.

A normalized regular grammar generates a normalized regular language. But let us examine the situation from the other side. Let L be a regular language for which a probability distribution has been defined. There is thus a value $p(s)$ for every s in L. Let us suppose that L is normalized, i.e. that $\sum_{s \in L} p(s) = 1$. Is there a regular probabilistic grammar which generates precisely the pairs $(s, p(s))$? This is known as the PROBLEM OF REPRESENTATION. We have the following theorem.

THEOREM 3.2. There is a regular language L, and a probability distribution for the sentences in L with the property $\sum_{s \in L} p(s) = 1$, for which no regular probabilistic grammar exists.

There are thus normalized regular probabilistic languages for which no normalized regular probabilistic grammar exists. The practical implication seems to be that not every sample (corpus) of sentences of a regular language can be described by a regular probabilistic grammar. However, the proof of this theorem, for which reference is made to Ellis (1969), is based on an argument

which is completely without practical implications. It is shown, in effect, that one can assign a normalized probability distribution to a regular language such that for some sentences s, $p(s)$ cannot be the product of any production probabilities whatsoever. The argument is based on the consideration that there are real numbers which are not rational. It supposes that some sentences of L have nonrational probabilities, and shows that in certain circumstances it is impossible to represent those probabilities as the product of production probabilities.

In every empirical situation, however, we have to do with samples of the sentences of a language L, and can therefore write the estimates of $p(s)$ as fractions. On the basis of this consideration, Suppes (1970) suggests the following general representation theorem for probabilistic languages; the theorem has not yet been proven.

THEOREM 3.3. If L is a type-i language, and a normalized probability distribution $p(s)$ has been defined for the sentences of L, then there is a type-i normalized probabilistic grammar which generates a probability distribution $p'(s)$ for the sentences of L, and for every finite sample S of L the null-hypothesis that S is drawn from $(L, p'(s))$ cannot be rejected.

In other words, we can find a probabilistic grammar for every sample (corpus) of sentences, according to which the original probability distribution can be approached so closely that it is impossible to decide (on the basis of a statistical test) if we are dealing with $L(p')$ or with $L(p)$.

3.4. CONTEXT-FREE PROBABILISTIC GRAMMARS

Two normal-forms for context-free grammars were introduced in chapter 2, and it was shown that every context-free grammar is equivalent to a grammar in the Chomsky normal-form and to a grammar in the Greibach normal-form. In the present paragraph

we shall show that these equivalences are also valid for context-free probabilistic grammars. Afterwards we shall discuss the consistency-problem for context-free probabilistic grammars.

3.4.1. *Normal-Forms*

Normal-forms pose an additional problem for context-free probabilistic grammars, for not only must the normal-form grammar be equivalent to the original one with respect to the sentences generated, but it must also be equivalent to the original grammar with respect to the probability of the sentences generated. This can be done only by giving the production probabilities in the normal-form grammar a certain relation to those of the original grammar. It is not certain in advance that this can always be done. For the Chomsky normal-form we shall state and derive the relations. The Greibach normal-form will only be mentioned.

THEOREM 3.4. (Chomsky normal-form). Every normalized context-free probabilistic grammar G is equivalent to a normalized context-free grammar, the productions of which are exclusively of the forms $A \xrightarrow{p} BC$ and $A \xrightarrow{p} a$.

PROOF. The proof is carried out in three steps. We first construct a grammar G' equivalent to G, and in which no productions of the form $A \xrightarrow{p} B$ occur. Next we compose a grammar G'' equivalent to G', and in which the productions are exclusively of the forms $A \xrightarrow{p} a$ and $A \xrightarrow{p} B_1B_2 \ldots B_n$ $(n \geqslant 2)$. Finally we compose G_n in the normal-form, equivalent to G'', and consequently also to G.

(i) Let there be such productions in G of the form $A \xrightarrow{p} B$ that derivations of the form $A \overset{p_1}{\Rightarrow} B_1 \overset{p_2}{\Rightarrow} B_2 \ldots \overset{p_n}{\Rightarrow} B_n \overset{p_{n+1}}{\Longrightarrow} \alpha$, where $\alpha \notin V_N$. We can replace every derivation of this kind by adding a production to P' in the form $A \xrightarrow{p} \alpha$, where

(1) $p = p_1 \cdot p_2 \cdot \ldots \cdot p_{n+1}$

This is only possible where there are no "loops" in such a deriva-

tion chain. For these cases we do the following. We speak of a loop when productions of the following form occur in P [1]:

$$A \xrightarrow{p_o} B$$

$$A \xrightarrow{p_i} \alpha_i \qquad i = 1,\ldots,n$$

$$B \xrightarrow{q_o} A$$

$$B \xrightarrow{q_j} \beta_j \qquad j = 1,\ldots,m$$

These productions can be replaced by the following productions in P':

$$A \xrightarrow{r_j} \beta_j \qquad j = 1,\ldots,m$$

$$B \xrightarrow{s_i} \alpha_i \qquad i = 1,\ldots,n$$

$$A \xrightarrow{t_i} \alpha_i \qquad i = 1,\ldots,n$$

$$B \xrightarrow{u_j} \beta_j \qquad j = 1,\ldots,m$$

where,

$$(2)\ r_j = \frac{p_o q_j}{1 - p_o q_j},\ t_i = \frac{p_i}{1 - p_o q_o},\ s_i = \frac{q_o p_i}{1 - p_o q_o},\ u_j = \frac{q_j}{1 - p_o q_o}$$

To show this, let us examine in detail the productions $A \xrightarrow{r_j} \beta_j$ in G'; the derivation for the other three types follows the same pattern. β_j can be derived in G in an infinite number of ways when there is a loop of the form $A \xrightarrow{p_o} B$ and $B \xrightarrow{q_o} A$, thus:

$$A \overset{p_o}{\Rightarrow} B \overset{q_i}{\Rightarrow} \beta_j$$

$$A \overset{p_o}{\Rightarrow} B \overset{q_o}{\Rightarrow} A \overset{p_o}{\Rightarrow} B \overset{q_j}{\Rightarrow} \beta_j$$

$$A \overset{p_o}{\Rightarrow} B \overset{q_o}{\Rightarrow} A \overset{p_o}{\Rightarrow} B \overset{q_o}{\Rightarrow} A \overset{p_o}{\Rightarrow} B \overset{q_j}{\Rightarrow} \beta_j, \text{ etc.}$$

The total probability that β_j be derived from A is thus

[1] Notation: In the following probabilities p always corresponds to productions where A occurs to the left of the arrow, and q corresponds to productions where B occurs to the left of the arrow.

$$p_o q_j + p_o(q_o p_o) q_j + p_o(q_o p_o)^2 q_j + \ldots =$$

$$p_o q_j \sum_{n=0}^{\infty} (q_o p_o)^n = \frac{p_o q_j}{1 - p_o q_o}.$$

By the same procedure we can deal with t_i, s_i, and u_j.

By eliminating all loops in this way, we obtain grammar G', equivalent to G, and in which there are no productions of the form $A \xrightarrow{p} B$.

(ii) Grammar G'' will contain all the productions of G' except those of the form $A \xrightarrow{p} \beta$, where β consists of terminal elements and possibly also variables ($|\beta| \geqslant 2$). All these productions are rewritten as productions which contain only variables; there will also be a set of terminal productions. If b_i is a terminal element in the string β, we introduce a new variable B_i in G'', and a new terminal production $B_i \xrightarrow{1} b_i$. In this way all the productions of the form $A \xrightarrow{p} \beta$ are replaced by productions of the form $A \xrightarrow{p} B_1 B_2 \ldots B_n$. It is clear that with this set of productions $A \overset{p}{\Rightarrow} \beta_i$ in G'', and in general that G'' is equivalent to G'.

(iii) At this point all productions in G'' which are not of the form $A \xrightarrow{p} a$ or $A \xrightarrow{p} BC$ must be reduced to the form $A \xrightarrow{p} BC$. The only productions in question here are those of the form $A \xrightarrow{p} B_1 B_2 \ldots B_n$ ($n > 2$). We replace each of these productions by a set of new productions as follows:

$$A \xrightarrow{p} B_1 D_1$$

$$D_1 \xrightarrow{1} B_2 D_2$$

$$\vdots$$

$$D_{n-2} \xrightarrow{1} B_{n-1} B_n$$

where D_i is a new variable (i = 1, ... $n - 2$).

When G_n contains these new productions and these new variables as well as the productions of G'' of the form $A \to \beta$ with $|\beta| \leqslant 2$, then G_n is obviously equivalent to G'' and therefore also to G, and moreover G_n is of the Chomsky normal-form.

This proof also shows what the relations must be between the production probabilities of the grammar in the Chomsky normal-form and those of the original grammar. They are found in the proof under (1) and (2).

EXAMPLE 3.2. Let $G = (V_N, V_T, P, S)$ be a context-free probabilistic grammar where $V_N = \{S, A, B\}$, $V_T = \{a,b\}$, and P consists of the following productions:

1. $S \xrightarrow{0.8} aS$
2. $S \xrightarrow{0.2} ABb$
3. $A \xrightarrow{0.5} B \quad (p_o = 0.5)$
4. $A \xrightarrow{0.4} a \quad (p_1 = 0.4)$
5. $A \xrightarrow{0.1} aA \quad (p_2 = 0.1)$
6. $B \xrightarrow{0.4} A \quad (q_o = 0.4)$
7. $B \xrightarrow{0.2} Bb \quad (q_1 = 0.2)$
8. $B \xrightarrow{0.4} b \quad (q_2 = 0.4)$

Grammar G is clearly normalized. To find an equivalent grammar in Chomsky normal-form, we must first construct a grammar G', equivalent to G, and in which the loop $A \xrightarrow{0.5} B$, $B \xrightarrow{0.4} A$ no longer occurs. To do so, we replace productions 3 to 8 with the following eight productions (cf. Proof (i)):

$A \xrightarrow{r_1} Bb$ $\qquad$ $A \xrightarrow{t_1} aA$

$A \xrightarrow{r_2} b$ $\qquad$ $A \xrightarrow{t_2} a$

$B \xrightarrow{s_1} aA$ $\qquad$ $B \xrightarrow{u_1} Bb$

$B \xrightarrow{s_2} a$ $\qquad$ $B \xrightarrow{u_2} b$

In order to calculate the values of r, s, t, and u, we use the following formulas:

$$r_1 = \frac{p_o q_1}{1 - p_o q_o} = \frac{0.5 \times 0.2}{1 - 0.5 \times 0.4} = \frac{0.1}{0.8} = 0.125$$

$$r_2 = \frac{p_o q_2}{1 - p_o q_o} = \frac{0.5 \times 0.4}{0.8} = 0.25$$

$$s_1 = \frac{q_o p_2}{1 - p_o q_o} = \frac{0.4 \times 0.1}{0.8} = 0.05$$

$$s_2 = \frac{q_o p_1}{1 - p_o q_o} = \frac{0.4 \times 0.4}{0.8} = 0.2$$

$$t_1 = \frac{p_2}{1 - p_o q_o} = \frac{0.1}{0.8} = 0.125$$

$$t_2 = \frac{p_1}{1 - p_o q_o} = \frac{0.4}{0.8} = 0.5$$

$$u_1 = \frac{q_1}{1 - p_o q_o} = \frac{0.2}{0.8} = 0.25$$

$$u_2 = \frac{q_2}{1 - p_o q_o} = \frac{0.4}{0.8} = 0.5$$

If we add the first and second productions of G to G', grammar G' is equivalent to G.

Grammar G'' is obtained by replacing the productions in G' with productions exclusively of the forms $A \xrightarrow{p} a$ and $A \xrightarrow{p} \beta$, where every β is made up only of variables. This yields the following productions in G'':

$S \xrightarrow{0.8} A_1 S$	$A \xrightarrow{0.125} BB_2$	$A_2 \xrightarrow{1} a$	$A \xrightarrow{0.5} a$
$A_1 \xrightarrow{1} a$	$B_2 \xrightarrow{1} b$	$B \xrightarrow{0.2} a$	$B \xrightarrow{0.25} BB_3$
$S \xrightarrow{0.2} ABB_1$	$A \xrightarrow{0.25} b$	$A \xrightarrow{0.125} A_3 A$	$B_3 \xrightarrow{1} b$
$B_1 \xrightarrow{1} b$	$B \xrightarrow{0.05} A_2 A$	$A_3 \xrightarrow{1} a$	$B \xrightarrow{0.5} b$

Finally, grammar G_n in Chomsky normal-form can be obtained by replacing the production $S \xrightarrow{0.2} ABB_1$ with $S \xrightarrow{0.2} AC$ and $C \xrightarrow{1} BB_1$.

The grammar in Chomsky normal-form will then contain the seventeen following productions:

1. $S \xrightarrow{0.8} A_1 S$	6. $A \xrightarrow{0.25} b$	10. $B \xrightarrow{0.25} BB_3$	14. $B_1 \xrightarrow{1} b$
2. $S \xrightarrow{0.2} AC$	7. $A_1 \xrightarrow{1} a$	11. $B \xrightarrow{0.05} A_2 A$	15. $B_2 \xrightarrow{1} b$
3. $A \xrightarrow{0.125} BB_2$	8. $A_2 \xrightarrow{1} a$	12. $B \xrightarrow{0.5} b$	16. $B_3 \xrightarrow{1} b$
4. $A \xrightarrow{0.125} A_3 A$	9. $A_3 \xrightarrow{1} a$	13. $B \xrightarrow{0.2} a$	17. $C \xrightarrow{1} BB_1$
5. $A \xrightarrow{0.5} a$			

This grammar is clearly normalized. But one cannot immediately see that a sentence generated by G has the same probability as a sentence generated by G_n. This is because every sentence generated by G has an infinity of possible leftmost derivations as a result of the loop. This emphasizes the advantage of a grammar in the Chomsky normal-form, since such a grammar has only a finite number of leftmost derivations for each sentence.

THEOREM 3.5. (Greibach normal-form) Every normalized context-free probabilistic grammar G is equivalent to a normalized context-free probabilistic grammar G', in which all productions are of the form $A \xrightarrow{p} a\alpha$, where $\alpha \in V_N^*$.

For proof of this theorem, as well as for the derivation of the production probabilities, we refer the reader to Huang and Fu (1971).

3.4.2. *Consistency Conditions for Context-free Probabilistic Grammars*

The theorems on the normal-forms tell us something of equivalence for normalized probabilistic grammars. But it is of interest to recall the definition: two normalized grammars may well generate the same probabilistic language, but that need not mean that the language is also normalized. The following theorem shows that one may not take it for granted that a normalized context-free grammar generates a normalized language. Context-free probabilistic grammars are not necessarily consistent.

THEOREM 3.6. (Inconsistency theorem) There are normalized context-free probabilistic grammars which do not generate normalized probabilistic languages.

PROOF. For proof of this theorem it is sufficient to show an example of such a grammar. Let $G = (\{S\}, \{a\}, P, S)$ be a grammar with the following productions in P:

1. $S \xrightarrow{\frac{3}{4}} SS$ 2. $S \xrightarrow{\frac{1}{4}} a$.

This grammar is normalized (and moreover in Chomsky normal-form); it generates the language $L = \{a^n\}$, where $n \geqslant 1$. The respective derivations of sentences a and aa are as follows:

$$S \overset{\frac{1}{3}}{\Rightarrow} a \qquad p(a) = 1/3$$

$$S \overset{\frac{2}{3}}{\Rightarrow} SS \overset{\frac{1}{3}}{\Rightarrow} aS \overset{\frac{1}{3}}{\Rightarrow} aa \qquad p(a^2) = 2/27$$

For the sentence aaa, there are two leftmost derivations possible:

$$S \overset{\frac{2}{3}}{\Rightarrow} SS \overset{\frac{2}{3}}{\Rightarrow} SSS \overset{\frac{1}{3}}{\Rightarrow} aSS \overset{\frac{1}{3}}{\Rightarrow} aaS \overset{\frac{1}{3}}{\Rightarrow} aaa$$

$$S \overset{\frac{2}{3}}{\Rightarrow} SS \overset{\frac{1}{3}}{\Rightarrow} aS \overset{\frac{2}{3}}{\Rightarrow} aSS \overset{\frac{1}{3}}{\Rightarrow} aaS \overset{\frac{1}{3}}{\Rightarrow} aaa$$

The reader will notice here that these derivations correspond to two different tree diagrams; G is therefore ambiguous. For $p(a^3)$ we find $(\frac{2}{3}\cdot\frac{2}{3}\cdot\frac{1}{3}\cdot\frac{1}{3}\cdot\frac{1}{3}) + (\frac{2}{3}\cdot\frac{1}{3}\cdot\frac{2}{3}\cdot\frac{1}{3}\cdot\frac{1}{3}) = 2\cdot(\frac{2}{3})^2\cdot(\frac{1}{3})^3 = \frac{8}{243}$. In general we can state that $p(a^n) = (n-1)\,(\frac{2}{3})^{n-1}\,(\frac{1}{3})^n$, where $n > 1$. After some calculation it appears that $\sum_{n=1}^{\infty} p(a^n) = \frac{1}{2}$, instead of the 1 required for normalization. G is therefore inconsistent.

It is possible, however, to pose conditions under which a normalized context-free probabilistic grammar will be consistent. For the following discussion of such conditions, some acquaintance with matrix algebra will again be required. We would advise readers who wish to omit the remainder of this paragraph that in any case every nonambiguous normalized context-free probabilistic grammar is consistent.

The conditions of consistency for a context-free grammar can best be discussed on the basis of the $n \times n$ matrix $A = [a_{ij}]$. Before defining the elements a_{ij}, we must first indicate what they are to represent. The value a_{ij} must be the total chance that the variable A_i generates at least one A_j in a derivation. Take the following productions for A_i and the corresponding probabilities:

$$\begin{array}{lll} A_i \to \alpha_1 & & p(A_i \to \alpha_1) \\ A_i \to \alpha_2 & \text{with probabilities} & p(A_i \to \alpha_2) \\ \quad\vdots & & \quad\vdots \\ A_i \to \alpha_k & & p(A_i \to \alpha_k) \end{array}$$

and suppose that in the h^{th} production $A_i \to \alpha_h$, the element A_j appears in the derivation m_{ijh} times. The production will thus be as follows:

$A_i \to \beta_1 A_j \beta_2 A_j \ldots \beta_{m_{ijh}} A_j \beta_{m_{ijh}+1}$,
where $|\beta_l| \geqslant 0$ for $l = 1, \ldots, m_{ijh}+1$.

We define a_{ijh} as follows: $a_{ijh} = m_{ijh} \cdot p(A \to \alpha_h)$. The definition of a_{ij} is then: $a_{ij} = \sum_{h=1}^{k} a_{ijh}$, with $i, j = 1, 2, \ldots, N$, where N is the number of variables in V_N.

In order to construct a consistent context-free probabilistic grammar, we must see to it that $\lim_{n \to \infty} A^n = 0$. This means that finally every variable, and consequently also $A_1 = S$, is rewritten as a terminal element. From this point of view, matrix A here fulfills precisely the same function as matrix C in the proof of Theorem 3.1. It is established (cf. Booth 1969, for example) that the limit is equal to the null-matrix 0, when the eigenvalue of A, with the highest absolute value $\lambda_{\max}$, is smaller than 1. If $\lambda_{\max} > 1$, the grammar is inconsistent; $\lambda_{\max} = 1$ produces various special problems which we will leave out of our discussion.

Let us again consider grammar G of Theorem 3.6., with productions $S \to SS$ and $S \to a$. Let $p(S \to SS) = p$, and $p(S \to a) = 1 - p$. Under what conditions will G be consistent? In this case matrix A has one cell: $A = [2p]$, because S occurs twice to the right of the arrow in the production $S \to SS$ with probability p. The only eigenvalue of A is then $2p$, and the grammar is consequently consistent when $2p < 1$ or $p < \frac{1}{2}$. It is inconsistent if $p > \frac{1}{2}$ (as was the case in the original example where $p = \frac{2}{3}$). In this case the grammar is also consistent when $p = \frac{1}{2}$.

4

FINITE AUTOMATA

In the present chapter we shall regard that which generative systems give as output, as the input of accepting systems. By definition, grammars are finite systems of rules by which potentially infinite sets of sentences can be generated. In this and the following chapters we shall show that for every language-type a mechanism can be constructed which is able to accept precisely the sentences of a language. In other words, given a language L of type-i, an automaton can be devised which can decide, after a finite number of operations, for the sentences of L and for no other string, that a sentence belongs to L. In generating a sentence, a grammar ascribes a structural description to it in passing; in a similar way, when an equivalent automaton accepts a sentence, an equivalent structural description unfolds.

It would, however, be incorrect to conclude from this symmetry that a mechanism finite in size can accept anything which is generated by a finite grammar. Such a mechanism can indeed be of finite description, but in most cases it will have to contain an infinite number of parts. In fact, only one of the language types which we have treated — the class of regular languages — is recognizable through finite means.

In this chapter we shall present a survey of the theory of finite automata, and we shall show (1) that there is a finite recognition-automaton for every regular language, and (2) that for every set of strings which is accepted by a given finite automaton, a regular grammar can be found which generates precisely the same strings. Some special types of finite automata, such as nondeterministic and

k-limited automata, will also be briefly discussed. In the final paragraph we shall mention some of the properties of probabilistic finite automata.

4.1. DEFINITIONS AND CONCEPTS

A FINITE AUTOMATON, FA, is a system (S, I, δ, s_0, F) in which

(1) S is a finite nonempty set of STATES. At any given moment the automaton must be in one of these states. Individual states are generally denoted by the letters s or t, with subscripts when needed.

(2) I is a finite nonempty (INPUT) VOCABULARY. Its elements ("words") are represented by letters from the beginning of the Latin alphabet. I^* is the set of strings, finite in length, composed of the elements of I, including the null-string λ. Elements of I^* may be represented by letters from the end of the Latin alphabet.

(3) δ is a (STATE) TRANSITION FUNCTION which indicates how the automaton changes states under the influence of an input word. The notation is as follows: $\delta(s, a) = t$ means that the automaton in state s changes to state t at the insertion of word a, where s and t are elements of S. The transformation function is defined for every possible pair of state and input-element: for every $s \in S$ and every $a \in I$, $\delta(s, a)$ is either a state in S, or φ, where φ means that the automaton blocks and no further step is possible. The transition function is also said to MAP the cartesian product $S \times I$ into $S \cup \varphi$. Because $S \times I$ is finite, the transition function consists of a finite set of rules called TRANSITION RULES.

(4) s_0 is a particular element of S, called the INITIAL STATE. It is the state of the automaton when the input process begins.

(5) F is a nonempty set of FINAL STATES in S.

A finite automaton $FA = (S, I, \delta, s_0, F)$ is said to ACCEPT a string $x \in I^*$, if FA, first operating in the initial state s_0, passes through a sequence of states, the last of which is a final state in F, under the influence of the successive elements of x.

Ordinarily the δ-notation is not limited to the input of individual

elements of I, but is also used for the input of strings from I^*. If $x = a_1a_2 \ldots a_n$, and FA contains the following transition rules: $\delta(s_1, a_1) = s_2$, $\delta(s_2, a_2) = s_3$, ..., $\delta(s_n, a_n) = s_{n+1}$, where $s_1 = s$ and $s_{n+1} = t$, we may write $\delta(s, x) = t$. Thus $\delta(s, xa) = \delta(\delta(s, x), a)$. By convention $\delta(s, \lambda) = s$. Expanded in this way, the transition function maps $S \times I^*$ in $S \cup \varphi$. We may also say that the automaton ACCEPTS $x \in I^*$ if $\delta(s_0, x) \in F$.

The LANGUAGE T accepted by the finite automaton FA is $\{x|\delta(s_0, x) \in F\}$, the set of strings accepted by the automaton. Such strings are also called SENTENCES.

Two finite automata are EQUIVALENT if they accept the same language.

Finite automata can be pictured as in Figure 4.1. They consist of a CONTROL-UNIT and a READING HEAD along which an INPUT TAPE runs from right to left. A string of input symbols appears on the tape (in the figure $x = a_1a_2 \ldots a_n$). The control-unit can be in only one of a finite number of states at a time. When the reading

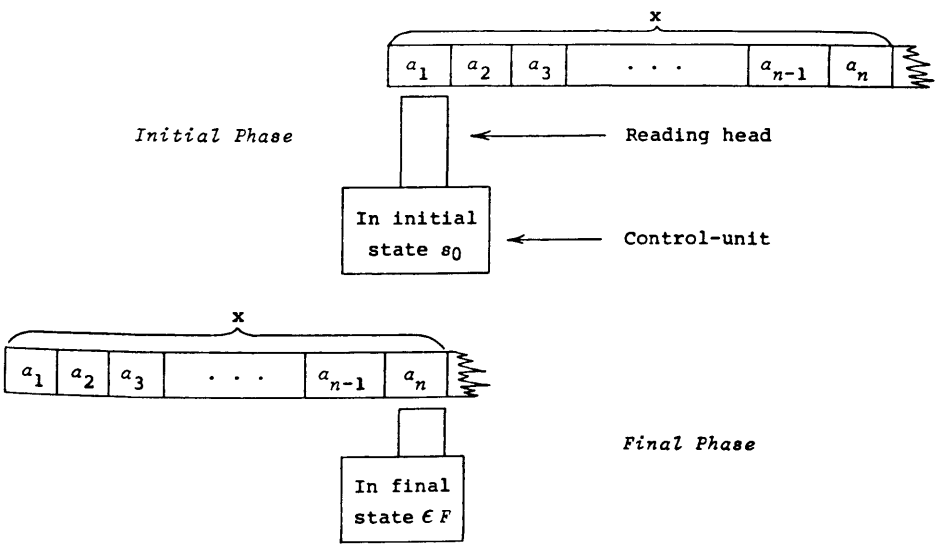


Fig. 4.1. The Accepting of a String $x = a_1a_2 \ldots a_n$ by a Finite Automaton.

head begins to read the first symbol, the control-unit is in the initial state s_o. When the first element (a_1 in the figure) is read, the state of the control-unit can change (according to the transition rule concerned). The tape then moves one space to the left. The next input symbol (a_2 in the figure) is read in the new state, and a second change of state may take place, according to the respective transition rule. The tape again moves one space to the left. This process continues until the control-unit arrives at a final state in F. The string of symbols read up to that point is then said to have been accepted by the automaton. Figure 4.1. shows the initial and final phases.

It is also possible visually to represent what occurs in the control-unit during reading; this is done by means of a TRANSITION-DIAGRAM. We shall illustrate this with a few examples.

EXAMPLE 4.1. Let $FA = (S, I, \delta, s_o, F)$ be a finite automaton with $S = \{s_o, s_1\}$, $I = \{a, b\}$, $F = \{s_1\}$, and where δ contains the following transition rules:

$$\delta(s_o, a) = s_1 \qquad \delta(s_o, b) = \varphi$$
$$\delta(s_1, b) = s_o \qquad \delta(s_1, a) = \varphi$$

The transition-diagram for this automaton is given in Figure 4.2.

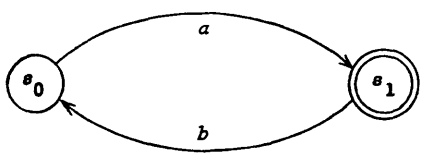


Fig. 4.2. Transition-Diagram for Finite Automaton FA (Example 4.1.).
initial state is s_o
final state (circled twice) is s_1

Such a diagram should be read in the following terms. Every state is shown by means of a circle in which the name of the state is given. For every nonblocking transition rule $\delta(s, a) = t$, there is an arrow in the diagram going from the circle labeled s to the circle

labeled t; the input symbol a is written near the arrow. In Figure 4.2. it is clear that the automaton in question has two states, that it passes from state s_0 to state s_1 when a is read, and that it returns from state s_1 to state s_0 when b is read. String a is obviously accepted by this automaton, because beginning in the initial state s_0, it passes to the (only) final state s_1 when a is read. Another way of coming to the final state s_1 is by reading the string *aba*: the automaton passes successively from s_0 to s_1, then back to s_0, and again to s_1; because s_0 is an initial state and s_1 is a final state, the string *aba*, by definition, is accepted. This automaton accepts all strings *a*, *aba*, *ababa*, ... The language is $T = \{a(ba)^*\}$.

EXAMPLE 4.2. Let $FA = (S, I, \delta, s_0, F)$ be a finite automaton with $S = \{s_0, s_1, s_2\}$, $I = \{a, b, c, d, e, f\}$, $F = \{s_0\}$, and with the following transition rules in δ:

$\delta(s_0, a) = s_1$
$\delta(s_1, b) = s_1$
$\delta(s_1, c) = s_2$
$\delta(s_1, d) = s_2$
$\delta(s_2, e) = s_0$
$\delta(s_2, f) = s_0$
$\delta(-, -) = \varphi$ for all other pairs

The transition-diagram for this automaton is given in Figure 4.3.

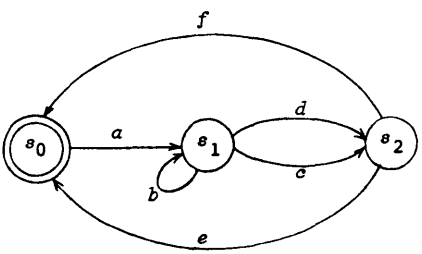


Fig. 4.3. Transition-Diagram for Finite Automaton *FA* (Example 4.2.).

Here s_0 is both an initial and a final state. One can easily see from the diagram that the automaton will accept all strings which bring it from the initial state s_0 back to the final state s_0; these are such strings as *adf*, *ace*, *ade*, *abdf*, *abbce*, etc. Each of these strings is

composed of first an a, then a string of 0 or more b's, then either a d or a c $(d \vee c)$, and finally either an e or an f $(e \vee f)$, thus strings of the form $ab^*(c \vee d)(e \vee f)$. As in the preceding example, however, after returning to the final state s_0, one can make still another turn in the automaton, returning once again to s_0, and continue doing so. The language accepted by this automaton is $T = \{(ab^*(c \vee d)(e \vee f))^*\}$. The machine also accepts λ, because by definition $\delta(s_0, \lambda) = s_0$, bringing the automaton from the initial to the final state.

Beside the fact that initial and final states are identical, this automaton has the peculiarity of allowing LOOPS, by which a state s_1 can be transformed into itself again. Moreover, there are two pairs of EQUIVALENT INPUT SYMBOLS, d and c, and e and f, which under all circumstances have the same effect on the operation of the automaton.

Instead of a transition-diagram, one can also use a TRANSITION-TABLE to show the structure of an automaton. A transition-table is a matrix in which the row-elements represent the states of an automaton, and the column-elements represent the possible input symbols. Every matrix-element shows a state (or φ) which is reached from a given state (row-element) and a given input symbol (column-element). An example of such a matrix is the following transition-table for finite automaton *FA* of Example 4.2.

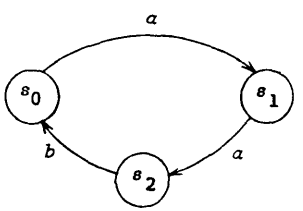
	input elements					
	a	b	c	d	e	f
s_0	s_1	φ	φ	φ	φ	φ
s_1	φ	s_1	s_2	s_2	φ	φ
s_2	φ	φ	φ	φ	s_0	s_0

Ordinarily the φ is omitted in such a matrix. A transition-table contains precisely the same information as a transition-diagram.

Some finite automata are K-LIMITED. A k-limited automaton is a finite automaton the state of which is determined at every moment by the last k (or fewer) accepted input symbols. The automaton of Example 4.2. is 1-limited. As is clear from the

transition-diagram (Figure 4.3.), the automaton, after having accepted *a*, can be only in state s_1; after accepting *b*, only in state s_1; after accepting *c*, only in state s_2; after accepting *d*, only in state s_2; after accepting *e*, only in state s_0; and after accepting *f*, only in state s_0. Likewise in each column of the transition-table, only one state is mentioned.

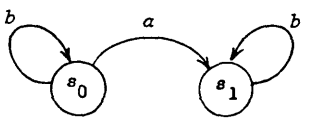
A 2-limited automaton is shown in Figure 4.4., both in diagrammatic and in tabular form. It is clear that immediately after accepting an *a*, the machine can be in one of two states, either s_1 or s_2. The automaton is therefore not 1-limited, but 2-limited, for after accepting *aa*, it is in state s_2; after accepting *ab* it is in s_0, and after *ba*, in s_1. It can never accept *bb*.



	a	b
s_0	s_1	
s_1	s_2	
s_2		s_0

Fig. 4.4. Transition-Diagram and Transition-Table for a 2-limited Automaton.

Figure 4.5. shows that not all finite automata are *k*-limited; it represents an automaton which is *k*-limited for no finite *k*. Even when this automaton has accepted an arbitrarily long string of *b*'s, we do not know if it is in state s_0 or in state s_1.



	a	b
s_0	s_1	s_0
s_1		s_1

Fig. 4.5. Transition-Diagram and Transition-Table for an Automaton which is *k*-limited for no Finite *k*.

If s_0 is the initial state and s_1 the final state, then the language which the automaton accepts is $T = \{b^*ab^*\}$. The *k*-limited auto-

maton is of some interest in dealing with Markov processes (cf. Volume II, 6.1., and Volume III, 3.2.).

4.2. NONDETERMINISTIC FINITE AUTOMATA

The finite automaton defined in the preceding paragraph has the property that for every state and input symbol, the state which follows (or φ) is unambiguously determined. Such an automaton is therefore called a DETERMINISTIC automaton. But, for two reasons, it remains necessary to define the nondeterministic variant of finite automata here. The first reason is that such a definition will allow us more easily to establish the relationship between finite automata and regular grammars. The second reason is that the probabilistic automaton (cf. paragraph 4.4.) is in turn a generalization of the nondeterministic finite automaton.

A NONDETERMINISTIC FINITE AUTOMATON *NFA* is a system (S, I, δ, s_0, F) which is in every way equal to a deterministic finite automaton, except for the transition rules δ. The transition rules of a nondeterministic finite automaton have the following form: $\delta(s, a) = \{t_1, t_2, \ldots, t_k\} = D$, where $0 \leqslant k < \infty$; $s, t_i \in S$, and $D \subset S$. In other words, for every pair of state and input symbols, there is a finite set of states at which the automaton can arrive. δ is said to be a mapping of $S \times I$ into the subset of S (where φ is the empty subset). A deterministic finite automaton is actually a particular case of nondeterministic finite automata: it covers those cases where for all transition rules $k = 1$ or $k = 0$.

When can one say that $x \in I^*$ is accepted by a nondeterministic finite automaton? Suppose that $x = a_1 a_2 \ldots a_n$, and that the finite automaton *FA* contains the following transition rules: $\delta(s_0, a_1) = D_1$, $s_1 \in D_1$; $\delta(s_1, a_2) = D_2$, $s_2 \in D_2$; $\ldots$; $\delta(s_{n-1}, a_n) = D_n$, $s_n \in D_n$ and $s_n \in F$, then x is said to be accepted by the automaton. Thus, if there is some succession of states allowed by the transition rules, according to which x brings the automaton from s_0 to a final state, the nondeterministic finite automaton is said to accept x.

The operation of a nondeterministic finite automaton is also easy to represent by way of a transition diagram, as becomes apparent in the following example.

EXAMPLE 4.3. Let $NFA = (S, I, \delta, s_0, F)$ be a nondeterministic finite automaton where $s = \{s_0, s_1, s_2\}$, $I = \{a, b\}$, $F = \{s_2\}$, and δ contains the following transition rules:

$\delta(s_0, a) = \{s_0, s_1\}$
$\delta(s_1, a) = \{s_2\}$
$\delta(s_1, b) = \{s_1, s_2\}$
$\delta(-, -) = \varphi$ for all other pairs.

Figure 4.6. shows the transition-diagram for this automaton. Among the strings which can bring the automaton from the initial state s_0 to the final state s_2 are the following: *aa*, *ab*, *aaa*, *aab*, *aba*, *abb*, and so forth. In general, the language accepted by this automaton is $T = \{a^*ab^*(a \vee b)\}$.

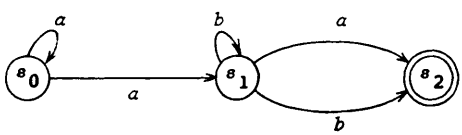


Fig. 4.6. Transition-Diagram for the Nondeterministic Finite Automaton *NFA* (Example 4.3.). The final state s_2 is circled twice.

The following important theorem is valid for nondeterministic finite automata.

THEOREM 4.1. For every nondeterministic finite automaton there exists an equivalent deterministic finite automaton.

The proof of this theorem, for which we refer the reader to Rabin and Scott (1959), will be briefly discussed later. We shall first illustrate it by returning to Example 4.3. We can construct a finite automaton *FA* equivalent to the nondeterministic finite automaton

NFA of that example in the following way. *NFA* had three states, i.e. $S = \{s_0, s_1, s_2\}$; the corresponding *FA* will have seven states, namely, $[s_0]$, $[s_1]$, $[s_2]$, $[s_0, s_1]$, $[s_0, s_2]$, $[s_1, s_2]$, and $[s_0, s_1, s_2]$. These states are thus called after all possible nonempty subsets of S. We maintain the input vocabulary, and in order to establish the new set of transition rules we proceed as follows. Let us begin with $\delta'([s_0], a)$. In *NFA* $\delta(s_0, a) = \{s_0, s_1\}$; in *FA* let $\delta'([s_0], a) = [s_0, s_1]$. Notice that this latter is one state and not two. Further let $\delta'([s_1], a) = [s_2]$ because $\delta(s_1, a) = \{s_2\}$, and $\delta'([s_2], a) = \varphi$ because $\delta(s_2, a) = \varphi$. For $\delta'([s_0, s_1], a)$ we proceed as follows. In *NFA* $\delta(s_0, a) = \{s_0, s_1\}$ and $\delta(s_1, a) = \{s_2\}$. The union of $\delta(s_0, a)$ and $\delta(s_1, a)$ is thus $\{s_0, s_1, s_2\}$, and in *FA* we let $\delta'([s_0, s_1], a) = [s_0, s_1, s_2]$. Again the latter is a single state. Similarly we construct $\delta'([s_0, s_2], a) = [s_0, s_1]$, etc. This procedure leads to the establishment of the following list of transition rules:

$\delta'([s_0], a) = [s_0, s_1]$ $\quad$ $\delta'([s_0, s_2], a) = [s_0, s_1]$
$\delta'([s_1], a) = [s_2]$ $\quad$ $\delta'([s_1, s_2], a) = [s_2]$
$\delta'([s_1], b) = [s_1, s_2]$ $\quad$ $\delta'([s_1, s_2], b) = [s_1, s_2]$
$\delta'([s_0, s_1], a) = [s_0, s_1, s_2]$ $\quad$ $\delta'([s_0, s_1, s_2], a) = [s_0, s_1, s_2]$
$\delta'([s_0, s_1], b) = [s_1, s_2]$ $\quad$ $\delta'([s_0, s_1, s_2], b) = [s_1, s_2]$
For all other $\delta'(-, -)$, $\delta'(-, -) = \varphi$.

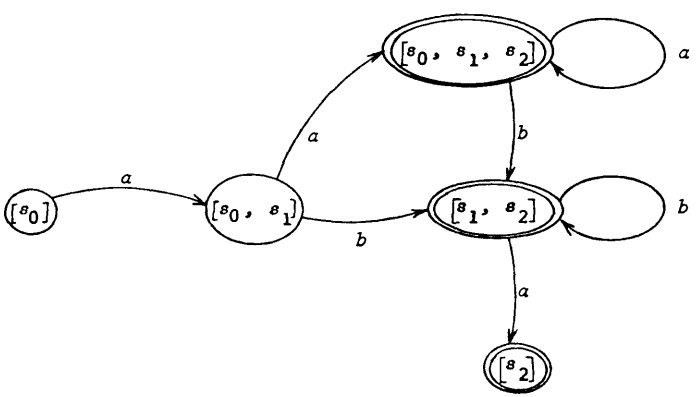


Fig. 4.7. Deterministic Finite Automaton Equivalent to the Nondeterministic Finite Automaton in Figure 4.6.

The set of final states F' in FA is defined as consisting of those states in which the label of a final state of NFA occurs. The only final state in NFA is s_2, and therefore $F' = \{[s_2], [s_0, s_2], [s_1, s_2], [s_0, s_1, s_2]\}$. Finally we take $[s_0]$ as the initial state in FA, and we affirm that FA is equivalent to NFA.

The transition-diagram for FA is given in Figure 4.7. The final states in the diagram are circled twice. The reader should notice that states $[s_1]$ and $[s_0, s_2]$ do not appear in the figure; this is because neither of them serves as the output of any transition rule. They are superfluous and consequently omitted. The diagram shows that FA accepts precisely the language $\{a^*ab^*(a \vee b)\}$.

PROOF OF THEOREM 4.1. (résumé). The proof follows the construction which we have just described. The states of FA correspond to the nonempty subsets of S in NFA. The transition rules are constructed as we have shown, and the set of final states F' in FA consists of those states which have one or more elements of F in their labels. By induction on the length of the string of input symbols it can be shown that FA is equivalent to NFA.

Because, inversely, deterministic finite automata are special cases of nondeterministic finite automata, we can conclude that the class of finite automata is equivalent to the class of nondeterministic finite automata; they accept the same class of languages.

4.3. FINITE AUTOMATA AND REGULAR GRAMMARS

In this paragraph we shall give proof of the equivalence of finite automata and regular grammars. The languages accepted by finite automata are exactly the same as those generated by regular grammars, and vice versa.

THEOREM 4.2. For every finite automaton FA there exists a regular grammar G such that $T(FA) = L(G)$.

PROOF. Let $FA = (S, I, \delta, s_0, F)$ be a finite automaton. We must construct a regular grammar $G = (V_N, V_T, P, S)$ such that

(i) $V_N = S$
(ii) $V_T = I$
(iii) $S = s_0$
(iv) $A \to aB$ is in P as $\delta(A, a) = B$
$A \to a$ is in P as $\delta(A, a) = C$, where $C \in F$
(notice that B and C are used here as labels for states)

We shall now show that G is equivalent to FA. For this, two conditions must be fulfilled: (1) If $x \in T(FA)$, then $x \in L(G)$, and (2) if $x \in L(G)$, then $x \in T(FA)$.

(1) $x \in T(FA)$. If this is so, then by definition $\delta(s_0, x)$ in F. We write x as $a_1a_2 \ldots a_n$. We presuppose that $\lambda \notin T(FA)$, and that therefore $n > 0$. In that case $\delta(s_0, x) = \delta(\delta(s_0, a_1a_2 \ldots a_{n-1}), a_n)$ (cf. paragraph 4.1. (5)), and continuing in the same way $\delta(s_0, x) = \delta(\delta(\ldots (s_0, a_1), a_2), \ldots), a_n)$. Because $\delta(s_0, x)$ in F, there is a sequence of states $s_0, s_1, \ldots, s_n$ ($s_i \in S$; s_i and s_j are not necessarily different) such that $\delta(s_0, a_1) = s_1$, $\delta(s_1, a_2) = \delta(\delta(s_0, a_1), a_2) = s_2$, $\ldots$, $\delta(s_{n-1}, a_n) = s_n$, where $s_n \in F$. But then there are also productions $S = s_0 \to a_1s_1$, $s_1 \to a_2s_2$, $\ldots$, $s_{n-1} \to a_n$ in P, on the basis of the construction of G. It is then clear that $S \overset{*}{\Rightarrow} a_1a_2 \ldots a_n = x$.

(2) $x \in L(G)$. By definition $S \overset{*}{\Rightarrow} x$. Let x be written as $a_1a_2 \ldots a_n$. Then there are productions $S = s_0 \to a_1s_1$, $s_1 \to a_2s_2$, $\ldots$, $s_{n-2} \to a_{n-1}s_{n-1}$ and $s_{n-1} \to a_n$ in P for certain s_i in V_N. But that means that FA contains the following transition rules: $\delta(s_0, a_1) = s_1$, $\delta(s_1, a_2) = s_2$, $\ldots$, $\delta(s_{n-2}, a_{n-1}) = s_{n-1}$, $\delta(s_{n-1}, a_n) = s_n$ with s_n in F (this follows from the definition of G). It is evident that with these transition rules FA accepts the string $a_1a_2 \ldots a_n = x$.

It follows from (1) and (2) that FA and G are equivalent for sentences of length > 0. If FA also accepts λ, the theorem holds only if we maintain the convention of paragraph 2.1., i.e. that by definition G also generates λ.

EXAMPLE 4.4. Let us construct a grammar equivalent to the finite automaton FA in Example 4.1. We recall that $FA = (S, I, \delta, s_0, F)$, where $S = \{s_0, s_1\}$, $I = \{a, b\}$, $F = \{s_1\}$, and with the following

transition rules: $\delta(s_0, a) = s_1$ and $\delta(s_1, b) = s_0$ (for all other pairs $\delta(-, -) = \varphi$).

The construction as shown in the proof is as follows: $G = (V_N, V_T, P, S)$, with $V_N = \{s_0 = S, s_1\}$, $V_T = \{a, b\}$, and $P = \{s_0 \to as_1, s_0 \to a, s_1 \to bs_0\}$. Notice that on the basis of (iv), the transition rule $\delta(s_0, a) = s_1$ leads to two productions in G: $s_0 \to as_1$ and $s_0 \to a$.

THEOREM 4.3. For every regular grammar G there exists a finite automaton FA such that $T(FA) = L(G)$.

PROOF. We shall prove that a nondeterministic finite automaton NFA can be found so that $T(NFA) = L(G)$. The theorem is then valid because for every nondeterministic finite automaton NFA there exists an equivalent finite automaton FA (Theorem 4.1.).

Let $G = (V_N, V_T, P, S)$ be a regular grammar. We construct $NFA = (S, I, \delta, s_0, F)$ as follows:

(i) $S = V_N \cup X$
(ii) $I = V_T$
(iii) $\delta(A, a)$ contains X (*inter alia*) if $A \to a$ in P
$\delta(A, a)$ contains every B for which $A \to aB$ in P
$\delta(X, a) = \varphi$ for every a in V_T
(iv) $s_0 = S$
(v) $F = \{X\}$, if $\lambda \notin L(G)$; $F = \{X, S\}$, if $\lambda \in L(G)$

Once again the proof of equivalence takes place in two steps. First it must be shown that if $x \in L(G)$, where $x = a_1a_2 \ldots a_n$, then $x \in T(NFA)$. Afterward the inverse must be shown.

(1) $x \in L(G)$. If $x \in L(G)$ and $|x| > 0$, then there is a derivation $S \Rightarrow a_1A_1 \Rightarrow \ldots \Rightarrow a_1a_2 \ldots a_{n-1}A_{n-1} \Rightarrow a_1a_2 \ldots a_n$ for some sequence $A_1, \ldots, A_{n-1}$ of variables in V_N. P thus contains the productions $S \to a_1A_1$, $A_1 \to a_2A_2$, $\ldots$ $A_{n-1} \to a_n$. It appears, then, from the construction of NFA that $A_1 \in \delta(S, a_1)$, $A_2 \in \delta(A_1, a_2)$, $\ldots$ $X \in \delta(A_{n-1}, a_n)$. But if the transition rules are valid, $x = a_1a_2 \ldots a_n$ is in $T(NFA)$. If $\lambda \in L(G)$, then $S \in F$ (see (v)), and because $\delta(S, \lambda)$ contains S by definition, $\lambda \in T(NFA)$.

(2) $x \in T(NFA)$. If $|x| > 0$ and x is accepted by NFA, then there are states $S, A_1, \ldots, A_{n-1}, X$, where $A_1 \in \delta(S, a_1)$, $A_2 \in \delta(A_1, a_2)$, $\ldots$, $X \in \delta(A_{n-1}, a_n)$. But from the construction of NFA it appears that P must also have productions $S \to a_1A_1$, $\ldots$, $A_{n-1} \to a_n$. It follows from this that $S \overset{*}{\Rightarrow} a_1a_2 \ldots a_n = x$. If $\lambda \in T(NFA)$, then $S \in F$. But $S \in F$ only if $\lambda \in L(G)$ (see (v)).

The equivalence of G and NFA follows from arguments (1) and (2). It follows from Theorem 4.1. that there must also exist an FA equivalent to G.

EXAMPLE 4.5. Let us construct a nondeterministic finite automaton NFA which accepts the language generated by regular grammar G in Example 2.1. We recall that $G = (V_N, V_T, P, S)$ where $V_N = \{S, B\}$, $V_T = \{a, b\}$, and $P = \{S \to aB, B \to bS, B \to b\}$, and that $L(G) = \{(ab)^*\}$. We shall now construct $NFA = (S, I, \delta, s_0, F)$ according to the procedure given in the proof. Thus $S = \{S, B, X\}$, $I = \{a, b\}$, δ contains the following transition rules: $\delta(S, a) = \{B\}$, $\delta(B, b) = \{X, S\}$, $\delta(-, -) = \varphi$ for all other pairs; finally, $F = \{X, S\}$. The transition-diagram for automaton NFA is given in Figure 4.8.

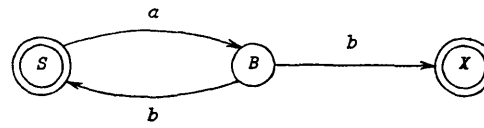


Fig. 4.8. Transition-Diagram for Nondeterministic Finite Automaton NFA which accepts language $\{(ab)^*\}$.

Together Theorems 4.2. and 4.3. show the equivalence of finite automata and regular grammars. We can employ this equivalence in order to prove certain theorems concerning regular grammars by means of theorems concerning finite automata, and vice-versa. Theorem 2.5. is a good example of this.

THEOREM 2.5. The product of two regular languages is regular.

PROOF. Let L_1 and L_2 be regular languages; let L_3 consist of the

strings xy where $x \in L_1$ and $y \in L_2$. There is a regular grammar for L_1, and therefore we know, on the basis of the equivalency theorem, that there is also a finite automaton which accepts L_1. We shall call this finite automaton $FA_1 = (S, I_1, \delta_1, s_o, F_1)$. Likewise there is a finite automaton $FA_2 = (T, I_2, \delta_2, t_o, F_2)$ which precisely accepts L_2. F_1 and F_2 can always be chosen such that they have no states in common. We must now construct a nondeterministic finite automaton $NFA = (U, I_3, \delta_3, u_o, F_3)$, which, in a way, connects FA_1 and FA_2 "in series". We define NFA as follows:

(i) $U = S \cup T$

(ii) $I_3 = I_1 \cup I_2$

(iii) $\delta_3(u, a) = \{\delta_1(s, a)\}$ for every s in $S - F_1$. In this way NFA can begin with a given input as if it were FA_1.

$\delta_3(u, a) = \{\delta_1(s, a), \delta_2(t_o, a)\}$ for every s in F_1. If NFA arrives at a final state of FA_1, it can freely (nondeterministically) either continue to another state of FA_1 (if this is also possible for FA_1) or pass on to FA_2. This latter is possible only when NFA has already reached a final state of F_1 (the transition rule is applicable only if s is in F_1) and when a can be the first symbol of a sentence of L_2 (notice that the initial state of FA_2 is t_o).

$\delta_3(u, a) = \{\delta_2(t, a)\}$ for every t in T. This guarantees that once NFA has "transferred" to FA_2 it will continue to operate as FA_2.

(iv) $u_o = s_o$

(v) $F_3 = F_2$ if $\lambda \notin L_2$. This guarantees that NFA accepts the input when the end of a sentence of L_2 is reached.

$F_3 = F_1 \cup F_2$ if $\lambda \in L_2$. If FA_2 accepts the null-string, it accepts all sentences $x\lambda = x$, i.e. the sentences of L_1. The automaton must be able to accept in each of the final states of F_1.

The construction of NFA guarantees that it will accept precisely the sentences $xy \in L_3$. But, on the basis of Theorem 4.1., there is also a deterministic finite automaton FA which does the same.

It follows from Theorem 4.2. that there is a regular grammar for L_3, and that L_3 is consequently regular.

The reader may now himself prove the lemma which was used at the proof of Theorem 2.8., with the help of finite automata.[1]

4.4. PROBABILISTIC FINITE AUTOMATA

We shall mention probabilistic automata only in the present paragraph. It is only on the subject of probabilistic finite automata that literature of any considerable size is available.

The probabilistic finite automaton (*PFA*) is a generalization of the nondeterministic finite automaton; a probability is assigned to every possible transition. Before presenting a formal definition of probabilistic finite automata, we shall discuss the manner, step by step, in which the generalization is made.

If it is true for a nondeterministic finite automaton *NFA* that $\delta(s, a) = \{s_1, s_2, \ldots, s_n\}$, we can define $p_i(s, a)$ for a probabilistic finite automaton *PFA* as the chance that the automaton will pass from state s to state s_i, given the input symbol a. We shall suppose that every probabilistic finite automaton is normalized, i.e. $\sum_{i=1}^{n} p_i(s, a) = 1$. In other words, the total chance for a state transition under the influence of a given input is 1. We shall return to the merits of this convention at the end of this paragraph. There is no reason why the chance for transition to a particular state could not be zero. In general we shall suppose that $1 \geqslant p_i(s, a) \geqslant 0$. Because transitions which cannot take place in a nondeterministic finite automaton can in a probabilistic finite automaton be considered as transitions where $p = 0$, we may give a more general definition of the transition function δ in a probabilistic finite automaton. If such an automaton *PFA* has n states, then $\delta(s, a)$ can

[1] To do so one should construct a nondeterministic finite automaton *NFA* which normally operates as FA_1 (which accepts L_1) except with transitions $\delta(s, a)$ where a is the critical terminal element. In such cases FA_2 (which accepts L_2) should be made to "take over" until a state in F_2 is reached. This should then act as $\delta(s, a)$, in order for *NFA* to be able to go on functioning as FA_1.

unambiguously be regarded as a row (vector) $(p_1, p_2, \ldots, p_n)$, where $p_i = p_i(s, a)$. For impossible transitions $p_i = 0$; for all other transitions p_i is the transition probability. Thus for every pair (s, a) where $s \in S$ and $a \in I$, δ is a vector of n numbers. If, for an element a, we wish to represent all the vectors, we can show them in matrix form as follows:

	s_1	s_2	...	s_j	...	s_n
$\delta(s_1, a)$	P_{11}	P_{12}	...	P_{1j}	...	P_{1n}
$\delta(s_2, a)$	P_{21}	P_{22}	...	P_{2j}	...	P_{2n}
	⋮	⋮		⋮		⋮
$\delta(s_i, a)$	P_{i1}	P_{i2}	...	P_{ij}	...	P_{in}
	⋮	⋮		⋮		⋮
$\delta(s_n, a)$	P_{n1}	P_{n2}	...	P_{nj}	...	P_{nn}

For the sake of brevity we shall call this entire matrix $M(a)$, the TRANSITION-MATRIX for element a. Matrix-element p_{ij} in $M(a)$ means that if the automaton is in state s_i and reads the input symbol a, there is a chance of p_{ij} that a transition to state s_j will take place. Normalization guarantees that the sum of the elements in a row in this matrix is equal to 1. The matrix is square $(n \times n)$, and is thus a stochastic matrix.

To include all the transition rules in *PFA* we would have to compose similar matrices for each of the input elements. If $I = \{a_1, a_2, \ldots, a_m\}$, we define M as the set of transition-matrices for the elements of I. Thus $M = \{M(a_1), M(a_2), \ldots, M(a_m)\}$.

Finally, we wish to open the possibility that the initial state of *PFA* is also random. For each of the n states we must define an INITIAL PROBABILITY $p(s)$, which represents the chance that at the first input the automaton is in state s. Since we wish *PFA* with certainty to be initially in one of the n states, we let $\sum_{i=1}^{n} p(s_i) = 1$. One can now no longer speak of an initial state, but rather of an INITIAL DISTRIBUTION; this simply means the string of initial probabilities $(p(s_1), p(s_2), \ldots, p(s_n))$. This vector is denoted by s_0.

At this point we can define a probabilistic finite automaton.

A PROBABILISTIC FINITE AUTOMATON is a system $PFA = (S, I,$

M, s_o, F), in which S is a finite set of states, I is a finite input vocabulary, M is the set of transition-matrices, s_o is the initial distribution and $F \subset S$ is the set of final states.

EXAMPLE 4.6. Take the probabilistic finite automaton $PFA = (\{s_1, s_2\}, \{a, b\}, \{M(a), M(b)\}, (1, 0), \{s_2\})$ with $M(a) = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$ and $M(b) = \begin{bmatrix} \frac{2}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{2}{3} \end{bmatrix}$. *PFA* has two states and the probability of starting in s_1 is 1 (because $s_o = (1,0)$). From transition-matrix $M(a)$ we learn that when the automaton is in state s_1 and reads the input symbol a, it has a chance of 1 to change to state s_2; if in state s_2 input of a leads with probability 1 to transition to s_2, i.e. *PFA* remains in s_2. Transition-matrix $M(b)$ shows what happens when the input is the symbol b. Once again all this is better shown by a transition-diagram. In a transition-diagram for a probabilistic finite automaton, the various arrows are labelled not only with the respective input elements, but also with the corresponding transition probabilities. Figure 4.9. gives the diagram for the automaton in this example. Arrows for transitions the probabilities of which are equal to 0 have been omitted.

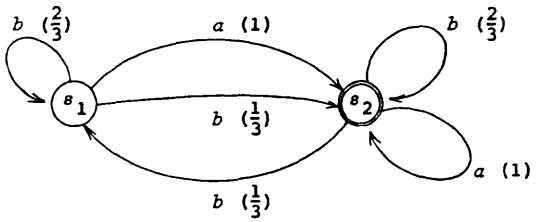


Fig. 4.9. Transition-Diagram for a Probabilistic Finite Automaton (Example 4.6.).

The diagram shows that starting in state s_1 the automaton has a chance of 1 to pass to final state s_2 when the input symbol a is read; this chance becomes $\frac{1}{3}$ when the input symbol is b. What will be the chance for the transition if the input is the string ab?

The element a brings the automaton, with a probability of 1, to state s_2; the element b will maintain the automaton in state s_2 with a probability of $\frac{2}{3}$. If the transitions are independent of each other (which is our presupposition here), the string ab brings the automaton to state s_2 with a probability of $1 \cdot \frac{2}{3} = \frac{2}{3}$. What then will be the chance that the string ab will bring the automaton back to state s_1? Obviously this will be $1 \cdot \frac{1}{3} = \frac{1}{3}$. Likewise the string ab will take the automaton from state s_2 back to state s_2 with the probability $1 \cdot \frac{2}{3} = \frac{2}{3}$, and from state s_1 back to state s_1 with probability $1 \cdot \frac{1}{3} = \frac{1}{3}$. In this way we have in fact found a transition-matrix for the string ab:

$$M(ab) = \begin{bmatrix} \frac{1}{3} & \frac{2}{3} \\ \frac{1}{3} & \frac{2}{3} \end{bmatrix}.$$

It is also quite easy to see that $M(ab)$ is the matrix product of $M(a)$ and $M(b)$:

$$M(ab) = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \frac{2}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{2}{3} \end{bmatrix} = \begin{bmatrix} \frac{1}{3} & \frac{2}{3} \\ \frac{1}{3} & \frac{2}{3} \end{bmatrix}.$$

In general we can define the TRANSITION-MATRIX $M(x)$ FOR A STRING $x = a_1 a_2 \ldots a_n$ as the product $M(x) = M(a_1) \cdot M(a_2) \cdot \ldots \cdot M(a_n)$. In such a matrix one can read, for all pairs s_i, s_j, the probability that the entry of an input x will cause the probabilistic finite automaton to change from state s_i to state s_j.

For the interested reader we can likewise easily indicate, in matrix notation, the chance that a final state be reached at all with a given string, given vector s_o, the string of initial probabilities. For this purpose, we define a FINAL VECTOR s_f as a string of n numbers, analogous to s_o, corresponding to the n states in S and in the same order. For every state, the corresponding number is 1 if the state is a final state, and 0 when this is not the case. Thus $s_f = (q_1, q_2, \ldots, q_n)$ where $q_i = 1$ if $s_i \in F$, and $q_i = 0$ if $s_i \notin F$. The final vector in Example 4.6. is thus (0, 1), for only s_2 is a final state. The chance that x will bring the automaton to a final state is given in matrix

notation as $s_o M(x)\, s'_f$.[1] Thus the chance that the string ab will bring the automaton of Example 4.6. to a final state is equal to

$$(1, 0) \cdot \begin{bmatrix} \frac{1}{3} & \frac{2}{3} \\ \frac{1}{3} & \frac{2}{3} \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} = (\tfrac{1}{3} \cdot \tfrac{2}{3}) \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \tfrac{2}{3}.$$

With these means at our disposition, we are able to define the language which is accepted by a probabilistic finite automaton. We should like to define that language as the set of strings by which the automaton reaches a final state with a certain minimum probability. What that minimum probability precisely is remains quite arbitrary. We can call it the CUT-POINT PROBABILITY, η.

The η-STOCHASTIC LANGUAGE $T(PFA, \eta)$ is the set of strings which bring the probabilistic finite automaton PFA to a final state with a probability $> \eta$. Formally stated, $T(PFA, \eta) = \{x | s_o M(x) s'_f > \eta\}$.

If $\eta = 0$, the situation is simple; every sentence by which a final state can be reached belongs to T. But stricter conditions can be posed. The opposite extreme is $\eta = 1$. However, the chance is never greater than 1 that a sentence will bring the automaton to a final state, and thus $T(PFA, 1)$ is empty for every PFA.

THEOREM 4.4. A regular language is η-stochastic for $0 \leqslant \eta < 1$.

PROOF. Let L be a regular language, and FA, a finite automaton, where $T(FA) = L$. We begin to construct probabilistic finite automaton PFA by borrowing I and F from FA. The set of states S' in PFA will be $S \cup s_\varphi$, where s_φ is a "dummy" state. A transition-matrix is composed for every $a \in I$ in PFA as follows: $p_{ij} = 1$ if $\delta(s_i, a) = s_j$; $p_{ij} = 0$ if $\delta(s_i, a) \neq s_j$, for every pair s_i, s_j in S. We let $p_{i\varphi} = 1$ if $\delta(s_i, a) = \varphi$, and $p_{i\varphi} = 0$ in all other cases, for $s_i \in S$. Finally, we let $P_{\varphi\varphi} = 1$, and $p_{\varphi i} = 0$ for every $s_i \in S$. In this way every matrix $M(a)$ is stochastic, and for every sentence x in $T(FA)$

[1] s'_f is the TRANSPOSITION of the row-vector, i.e. the row-vector is set up vertically like a column, with the leftmost element at the top. Notice that the definition of a transition-matrix for x supposes the stochastic independence of the transitions.

there is a probability of 1 that x will be accepted by *PFA*, while a final state will be reached with no other string. Because for every sentence s in L, the probability $p(s) = 1$ in $T(PFA)$, it is true for every $0 \leqslant \eta < 1$ that $T(PFA, \eta) = L$.

The inverse of Theorem 4.4. does not hold, but the following theorem is valid.

THEOREM 4.5. Every 0-stochastic language is regular.

PROOF. Let $PFA = (S, I, M, s_0, F)$ be the probabilistic finite automaton which accepts the 0-stochastic language T. We must first construct a nondeterministic finite automaton $NFA(i)$ for a state s_i with initial probability in PFA: $p(s_i) > 0$. We make $NFA(i)$ such that it accepts every sentence which brings PFA from state s_i to a final state, with probability > 0. For this purpose we let the initial state of $NFA(i)$ be s_i, F be the set of final states in $NFA(i)$, and s_l in $\delta(s_j, a_k)$ if the element p_{jl} is greater than 0 in the transition-matrix $M(a_k)$. The language T_i accepted by $NFA(i)$ is regular (Theorems 4.1. and 4.2.).

If we construct a $NFA(i)$ for every s_i in S for which $p(s_i) > 0$, it follows that every sentence which is accepted by *PFA*, with probability greater than 0, will also be accepted by at least one of the *NFA*, and that every sentence accepted by one of the *NFA* will also be accepted by *PFA* with probability greater than 0. We conclude that the union of all the languages T_i is also regular (Theorem 2.5.).

We close this paragraph with a remark on normalization as used with probabilistic finite automata. The basis for normalization $\sum_{i=1}^{n} p_i(s, a) = 1$ is the input symbol: each input symbol leads to a transition with a probability of 1. The consequence of this normalization is that it is not generally valid that the sentence probabilities in a stochastic language add up to 1. In the degenerate case, for example, where the matrix contains only 1's and 0's, every sentence of the language has a probability of 1, while the language can indeed contain more than one sentence. There is therefore no

simple relationship between probabilistic finite automata and regular probabilistic grammars which are normalized on the basis of a nonterminal element. As we have seen, in that case a normalized probabilistic language is generated. Probabilistic finite automata can, of course, also be normalized on another basis, namely the state. In that case the total chance for transition from a given state, taken over all inputs, is equal to 1, thus $\sum_i \sum_j p_i(s, a_j) = 1$. It then becomes possible to show equivalences to probabilistic grammars.

5

PUSH-DOWN AUTOMATA

In the preceding chapter we showed that regular languages can be accepted by finite automata. For languages of a higher order we shall have to refer to systems which are, in some way, infinite in size. To clarify the notion, let us consider a digital computer.

A digital computer is a finite automaton because it has a finite number of parts — for instance, n (including storage) — each of which can be in a finite number of states — let us say k at most. The machine will therefore have no more than k^n states, a finite number. Consequently a computer can accept, in principle, only regular languages; it cannot accept context-free or higher order languages.

One may wonder if there is any practical interest in studying automata which can accept higher order languages, since, in principle, they can never be built. However, the theoretical infiniteness of such automata is of little consequence in practice. The value of n for a sizable computer can easily reach 10^6, and if k is equal to 2, k^n is an astronomically high number. For practical purposes, then, a computer is of unlimited size. It can, within limits which in practice are never reached, accept higher order languages. Most computer languages, such as ALGOL, are in fact context-free or higher order languages.

In this chapter we will discuss one simple infinite automaton, the PUSH-DOWN AUTOMATON. This automaton is infinite because its store, the PUSH-DOWN STORE, is of unlimited capacity. In all other respects it is a finite automaton. We shall show that push-down automata are equivalent to context-free grammars.

5.1. DEFINITIONS AND CONCEPTS

A push-down automaton is a finite automaton to which an unlimited push-down store has been added. A push-down store is somewhat comparable to a narrow knapsack. Imagine that a hiker has placed his matches at the very bottom of his knapsack, then put in his jacket and other articles of clothing, and finally a can of soup, a can opener, and cooking utensils. When the hiker becomes hungry and reaches a brook, he may wish to eat the soup. He removes the cooking utensils, can opener, and the can of soup; this poses no problems, as the last articles placed in the sack are the first to come out. Also, he can add water from the brook. But if he wishes to light a fire to warm the soup, he must first remove the clothing and jacket before he is able to reach the matches: the first things placed in the sack are the last to come out.

We can make an analogy between the hiker and a push-down automaton: the knapsack can be compared to the push-down store (with the matches as the start element), the water and firewood to inputs, and warmth and satisfaction for hunger to state transitions.

The formal definition of a push-down automaton is as follows. A PUSH-DOWN AUTOMATON PDA is a system $(S, I, \Gamma, \delta, s_0, \gamma_0)$ where:

(1) S is a finite nonempty set of STATES, with $s_0 \in S$ as INITIAL STATE.

(2) I is a finite (INPUT) VOCABULARY.

(3) Γ is a finite PUSH-DOWN VOCABULARY, with $\gamma_0 \in \Gamma$ as push-down START SYMBOL, the only element in the store when input begins. Other push-down symbols are $\gamma_1, \gamma_2, \ldots$. The set of finite strings of push-down symbols is Γ^*. Elements of Γ^* are represented by lower case letters from the end of the Greek alphabet, such as χ, ψ, ω. The topmost symbol which at a given moment is found in the push-down store is called the TOP SYMBOL.

(4) δ is the set of TRANSITION RULES. Each rule indicates what will occur when, at a given state, with a given top symbol, a given input symbol (possibly also λ) is introduced, i.e. it shows what the following state will be and by what the top symbol will be replaced. The top symbol may be replaced by (a) an element of Γ;

(b) itself — a special case of (a), the content of the store remains unchanged; (c) an element of Γ^*, thus, a STRING of symbols replaces the top symbol; or (d) the null-string λ — a special case of (c), this amounts to simply removing the top symbol. The notation for these cases is as follows:

(a) $\delta(s_i, a, \gamma_k) = (s_j, \gamma_1)$, where s_i and s_j are states in S, a is an input symbol in I, and γ_k and γ_1 are push-down symbols in Γ.

(b) $\delta(s_i, a, \gamma_k) = (s_j, \gamma_k)$

(c) $\delta(s_i, a, \gamma_k) = (s_j, \chi)$, where χ is a string in Γ^*. If $\chi = \psi\gamma_k$ for some ψ in Γ^*, and thus $\delta(s_i, a, \gamma_k) = (s_j, \psi\gamma_k)$, then ψ is added to the store. Notice that the last added element is noted at the left.

(d) $\delta(s_i, a, \gamma_k) = (s_j, \lambda)$. Because λ is the null-string, this simply means that the top symbol γ_k is removed.

It can also occur that $\delta(s_i, a, \gamma_k) = \varphi$; the automaton is then said to BLOCK.

The function δ maps the cartesian product $S \times (I \cup \lambda) \times \Gamma$ into $S \times \Gamma^* \cup \varphi$.

A CONFIGURATION in a push-down automaton is a combination of state and store content. A transition rule in δ can bring the automaton from one configuration to another. If there is a rule $\delta(s_i, a, \gamma_k) = (s_j, \chi)$, then the introduction of the input element a can change the configuration from $(s_i, \gamma_k\omega)$ to $(s_j, \chi\omega)$. The notation for this is:

$$a\colon (s_i, \gamma_k\omega) \vdash (s_j, \chi\omega).$$

This change is called a TRANSITION in the automaton. Unless otherwise stated, we shall suppose that $\delta(s, \lambda, \gamma) = (s, \gamma)$ for every s in S and for every γ in Γ; in other words, the input of λ changes neither state nor store content. Thus:

$$\lambda\colon (s, \omega) \vdash (s, \omega) \text{ for every } s \in S \text{ and every } \omega \in \Gamma^*.$$

In specially mentioned cases where it is permitted that $\delta(s, \lambda, \gamma) \neq (s, \gamma)$ (i.e. where the automaton can make a real change of state without input), we must allow that $\delta(s, a, \gamma) = \varphi$ for every a in I,

for otherwise the automaton could make various different transitions when the input a is introduced. The INITIAL CONFIGURATION of a push-down automaton is by definition (s_o, y_o).

We write $x = a_1 a_2 \ldots a_n$: $(s, \omega) \vdash^* (s', \omega')$, if δ allows transitions a_i: $(s_i, \omega_i) \vdash (s_{i+1}, \omega_{i+1})$, where $i = 1, 2, \ldots, n$, such that $s_1 = s$, $\omega_1 = \omega$, $s_{n+1} = s'$, and $\omega_{n+1} = \omega'$. String x makes the automaton change from configuration (s, ω) to configuration (s', ω').

A string x is ACCEPTED by a *PDA* if at the end of the processing of x the push-down store is empty. Formally, string x is accepted by *PDA* if x: $(s_o, \gamma_o) \vdash^* (s, \lambda)$. Note that this definition is not based on the attainment of a final state, as was the case with finite automata. There exists a description of push-down automata which does refer to the attainment of a final state; it is completely equivalent to the description used here, and we shall not bring it into the discussion.

The LANGUAGE $T(PDA)$ accepted by a push-down automaton is the set of strings which are accepted by that automaton, $T(PDA) = \{x | x: (s_o, \gamma_o) \vdash^* (s, \lambda)\}$.

Figure 5.1 shows how a push-down automaton accepts a string.

EXAMPLE 5.1. In order to demonstrate the operation of the push-down automaton, we take a *PDA* which only uses its store, and never changes states. The automaton accepts strings of a's, b's, and c's, with as many a's as b's, and one c at the end of the string: e.g. *c*, *abc*, *aabbc*, *baabc*, etc.

$PDA = (S, I, \Gamma, \delta, s_o, \gamma_o)$, with $S = \{s_o\}$, $I = \{a, b, c\}$, $\Gamma = \{\gamma_o, \gamma_a, \gamma_b\}$, and where δ consists of the following transition rules:

1. $\delta(s_o, a, \gamma_o) = (s_o, \gamma_a\gamma_o)$
2. $\delta(s_o, a, \gamma_a) = (s_o, \gamma_a\gamma_a)$
3. $\delta(s_o, a, \gamma_b) = (s_o, \lambda)$
4. $\delta(s_o, b, \gamma_o) = (s_o, \gamma_b\gamma_o)$
5. $\delta(s_o, b, \gamma_b) = (s_o, \gamma_b\gamma_b)$
6. $\delta(s_o, b, \gamma_a) = (s_o, \lambda)$
7. $\delta(s_o, c, \gamma_o) = (s_o, \lambda)$

For all other (s, c, γ), $\delta(s, c, \gamma) = \varphi$.

By convention $\delta(s, \lambda, \gamma) = (s, \gamma)$ for all s, γ.

We shall now show how the automaton accepts the string

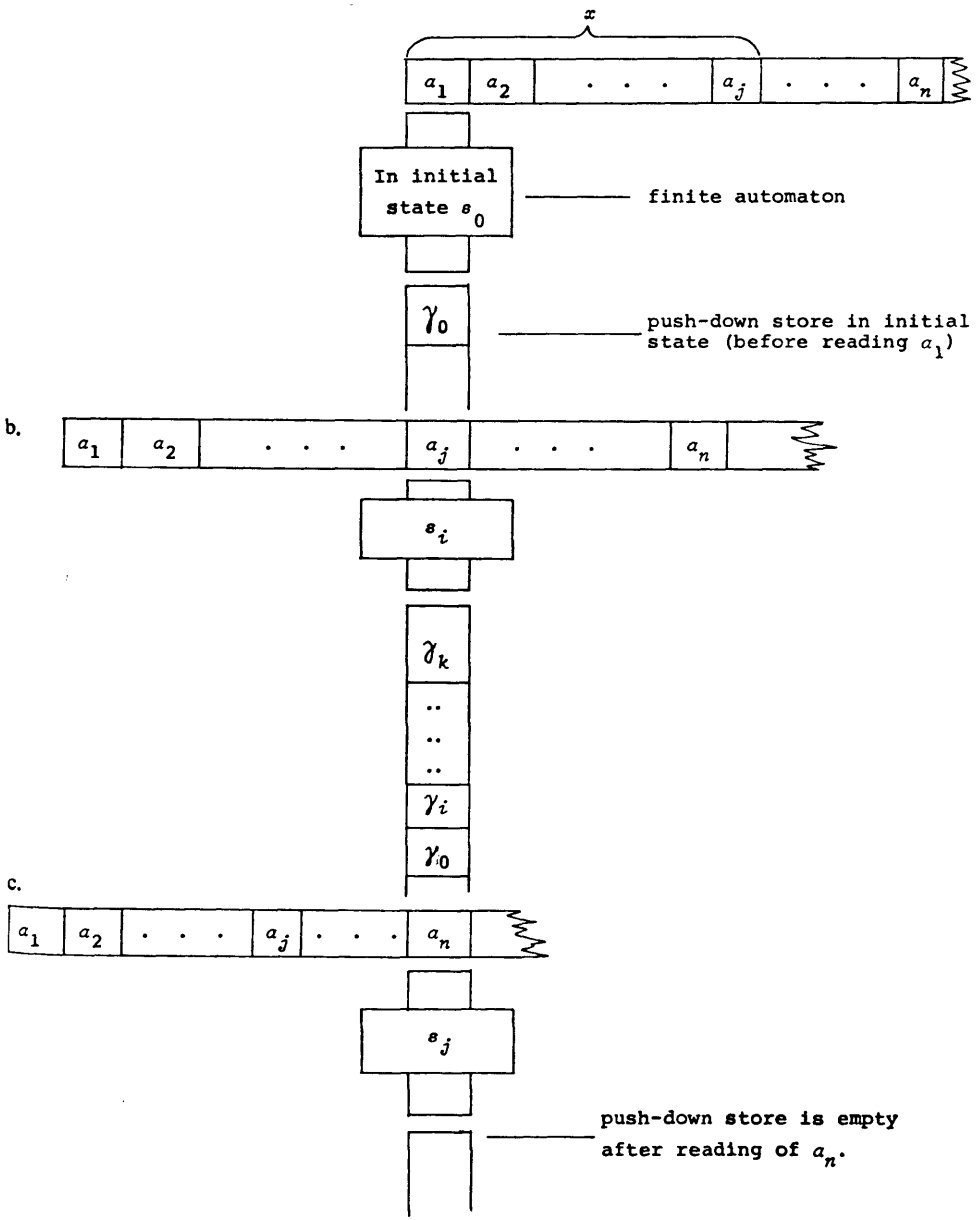


Fig. 5.1. A Push-Down Automaton in Operation
a. Situation at start
b. Automaton while processing string x
c. Automaton after accepting string x

aabbbbaac. The following list gives the successive transitions and the rules applied.

a: $(s_o, \gamma_o) \vdash (s_o, \gamma_a\gamma_o)$	(rule 1)
a: $(s_o, \gamma_a\gamma_o) \vdash (s_o, \gamma_a\gamma_a\gamma_o)$	(rule 2)
b: $(s_o, \gamma_a\gamma_a\gamma_o) \vdash (s_o, \gamma_a\gamma_o)$	(rule 6)
b: $(s_o, \gamma_a\gamma_o) \vdash (s_o, \gamma_o)$	(rule 6)
b: $(s_o, \gamma_o) \vdash (s_o, \gamma_b\gamma_o)$	(rule 4)
b: $(s_o, \gamma_b\gamma_o) \vdash (s_o, \gamma_b\gamma_b\gamma_o)$	(rule 5)
a: $(s_o, \gamma_b\gamma_b\gamma_o) \vdash (s_o, \gamma_b\gamma_o)$	(rule 3)
a: $(s_o, \gamma_b\gamma_o) \vdash (s_o, \gamma_o)$	(rule 3)
c: $(s_o, \gamma_o) \vdash (s_o, \lambda)$	(rule 7)

Thus $aabbbbaac$: $(s_o, \gamma_o) \vdash^* (s_o, \lambda)$.

EXAMPLE 5.2 Let $PDA = (S, I, \Gamma, \delta, s_o, \gamma_o)$ be a push-down automaton where $S = \{s_o, s_1\}$, $I = \{a, b, c\}$, $\Gamma = \{\gamma_o, \gamma_a, \gamma_b\}$, with the following transition rules:

1. $\delta(s_o, a, \gamma_o) = (s_o, \gamma_a\gamma_o)$	7. $\delta(s_o, c, \gamma_o) = (s_o, \lambda)$
2. $\delta(s_o, a, \gamma_a) = (s_o, \gamma_a\gamma_a)$	8. $\delta(s_o, c, \gamma_a) = (s_1, \gamma_a)$
3. $\delta(s_o, a, \gamma_b) = (s_o, \gamma_a\gamma_b)$	9. $\delta(s_o, c, \gamma_b) = (s_1, \gamma_b)$
4. $\delta(s_o, b, \gamma_o) = (s_o, \gamma_b\gamma_o)$	10. $\delta(s_1, a, \gamma_a) = (s_1, \lambda)$
5. $\delta(s_o, b, \gamma_b) = (s_o, \gamma_b\gamma_b)$	11. $\delta(s_1, b, \gamma_b) = (s_1, \lambda)$
6. $\delta(s_o, b, \gamma_a) = (s_o, \gamma_b\gamma_a)$	12. $\delta(s_1, \lambda, \gamma_o) = (s_1, \lambda)$

$\delta(s, \lambda, \gamma) = (s, \gamma)$ for every other s, γ and in all other cases $\delta(s, -, \gamma) = \varphi$.

This push-down automaton accepts all symmetric sentences, where c may occur only in the middle of the sentence. If w is a string of a's and b's, and w^R is the "mirror image" of w, then the language accepted by PDA is $\{wcw^R\}$. In essence, the PDA places a γ_a into the store for every incoming a, and a γ_b for every incoming b until a c is introduced. From that point the state changes from s_o to s_1, and the process is reversed: for every incoming a it removes the top symbol if it is γ_a, and for every incoming b it removes the top symbol if it is γ_b. This continues until γ_o is the top symbol, and by rule 12 the automaton removes γ_o without further input.

The sequence of transitions for string *aabbcbbaa* is as follows:

$(s_0, \gamma_0) \vdash (s_0, \gamma_a\gamma_0) \vdash (s_0, \gamma_b\gamma_a\gamma_0) \vdash (s_0, \gamma_b\gamma_b\gamma_a\gamma_0) \vdash (s_1, \gamma_b\gamma_b\gamma_a\gamma_0) \vdash (s_1, \gamma_b\gamma_a\gamma_0) \vdash (s_1, \gamma_a\gamma_0) \vdash (s_1, \gamma_0) \vdash (s_1, \lambda)$.

It is obvious that push-down automata can do more than finite automata. The languages which are accepted by the automata in the last two examples are both context-free languages, and there is no finite automaton which can accept them. But push-down automata cannot accept all context-free languages; the languages which they accept are called DETERMINISTIC LANGUAGES. A class of grammars is known which generates precisely these deterministic languages, namely the class of *LR(k)*-GRAMMARS. These are equivalent to push-down automata. We shall not discuss *LR(k)*-grammars here. The interested reader may consult Knuth (1965).

However, there is equivalence between context-free languages and nondeterministic push-down automata.

5.2. NONDETERMINISTIC PUSH-DOWN AUTOMATA AND CONTEXT-FREE LANGUAGES

A nondeterministic push-down automaton *NPDA* differs from a *PDA* only in that each of its transition rules is of the following form:

$$\delta(s, a, \gamma) = \{(s_1, \gamma_1), (s_2, \gamma_2), \ldots, (s_n, \gamma_n)\}.$$

This means that in each configuration the automaton is not limited to a single possible transition, but can make a "choice" among the elements of a set of transitions.[1] The construction of a nondeterministic push-down automaton is completely analogous to that of a nondeterministic finite automaton, and the same is true of the definition of accepting. A *NPDA* ACCEPTS a string x, if, when x is

[1] At this point we drop the condition that if $\delta(s, \lambda, \gamma) \neq \varphi$, then $\delta(s, a, \gamma) = \varphi$ for every a in I. This condition was necessary in order to exclude the possibility of a nondeterministic transition when an input a is introduced into the automaton.

introduced as input, there is at least one possible sequence of transitions for which x: $(s_o, r_o) \vdash^* (s, \lambda)$.

EXAMPLE 5.3. Let us construct a simple *NPDA* which will accept the language $\{a^n b^n \mid n \geqslant 1\}$. Let $NPDA = (\{s_o\}, \{a, b\}, \{\gamma_o, \gamma_a, \gamma_b\}, \delta, s_o, \gamma_o)$, with the following transition rules in δ:

1. $\delta(s_o, \lambda, \gamma_o) = \{(s_o, \gamma_a\gamma_b), (s_o, \gamma_a\gamma_o\gamma_b)\}$
2. $\delta(s_o, a, \gamma_a) = \{(s_o, \lambda)\}$
3. $\delta(s_o, b, \gamma_b) = \{(s_o, \lambda)\}$

By convention, $\delta(s, \lambda, \gamma) = (s, \gamma)$ for every s and γ, and $\delta(s, -, \gamma) = \varphi$ for all other δ.

Only rule 1 is nondeterministic. To show how *NPDA* operates, we give the successive transitions in the accepting of the string *aaabbb*:

λ: $(s_o, \gamma_o) \vdash (s_o, \gamma_a\gamma_o\gamma_b)$	(rule 1)
a: $(s_o, \gamma_a\gamma_o\gamma_b) \vdash (s_o, \gamma_o\gamma_b)$	(rule 2)
λ: $(s_o, \gamma_o\gamma_b) \vdash (s_o, \gamma_a\gamma_o\gamma_b\gamma_b)$	(rule 1)
a: $(s_o, \gamma_a\gamma_o\gamma_b\gamma_b) \vdash (s_o, \gamma_o\gamma_b\gamma_b)$	(rule 2)
λ: $(s_o, \gamma_o\gamma_b\gamma_b) \vdash (s_o, \gamma_a\gamma_b\gamma_b\gamma_b)$	(rule 1)
a: $(s_o, \gamma_a\gamma_b\gamma_b\gamma_b) \vdash (s_o, \gamma_b\gamma_b\gamma_b)$	(rule 2)
b: $(s_o, \gamma_b\gamma_b\gamma_b) \vdash (s_o, \gamma_b\gamma_b)$	(rule 3)
b: $(s_o, \gamma_b\gamma_b) \vdash (s_o, \gamma_b)$	(rule 3)
b: $(s_o, \gamma_b) \vdash (s_o, \lambda)$	(rule 3)

Thus $aaabbb = \lambda a\lambda a\lambda abbb$: $(s_o, \gamma_o) \vdash^* (s_o, \lambda)$.

This example also shows how a push-down automaton can make spontaneous transitions (when the input is λ), and how the initial symbol γ_o can be removed from the store before the store is empty.

Theorems 5.1 and 5.2 together show the equivalence of nondeterministic push-down automata and context-free grammars.

THEOREM 5.1. For every context-free language L, there is a nondeterministic push-down automaton which accepts L and only L.

PROOF. In fact we shall prove a somewhat stronger theorem,

namely, that there is a nondeterministic push-down automaton with only one state which can accept the context-free language L.

Let L be a context-free language, and $G = (V_N, V_T, P, S)$, a grammar in Greibach normal-form which generates language L (according to Theorem 2.7., such a grammar exists). The productions in G are thus exclusively of the form $A \to a\alpha$, where α is a string of 0 or more variables. We construct a nondeterministic push-down automaton $NPDA = (S, I, \Gamma, \delta, s_o, \gamma_o)$ as follows: $S = \{s_o\}$, $I = V_T$ (with elements a_i), $\Gamma = V_N \cup V_T = V$ (with elements a_i in V_T and elements A_i, S in V_N), $\gamma_o = S$. The input vocabulary of $NPDA$ is the terminal vocabulary of G; the push-down symbols of $NPDA$ are the elements of V in G, and the push-down start symbol of $NPDA$ is the start symbol S of G. Let $NPDA$ have the following transition rules:

1. $\delta(s_o, \lambda, A)$ contains $(s_o, a\alpha)$ for every production $A \to a\alpha$ in P (where α can have length 0).
2. $\delta(s_o, a, a) = \{(s_o, \lambda)\}$ for every a in V_T.

The push-down automaton will in general be nondeterministic, for if A can be rewritten in more than one way in G (e.g. $A \to \alpha$ and $A \to \beta$), then $\delta(s_o, \lambda, A)$ likewise has more than one possible transition ((s_o, α) and (s_o, β) in the present example).

We must show that $T(NPDA) = L(G)$. We shall first show that if $x \in L(G)$, then $x \in L(NPDA)$; afterwards we shall show the inverse.

(1) If $x = a_1 a_2 \ldots a_n$ in $L(G)$, then $S \overset{*}{\Rightarrow} x$ with the following leftmost derivation: $S \Rightarrow a_1\alpha_1 \Rightarrow a_1a_2\alpha_2 \Rightarrow \ldots \Rightarrow a_1a_2 \ldots a_{n-1}A_{n-1} \Rightarrow a_1a_2 \ldots a_n$. This derivation is performed by rewriting the leftmost variable of α_i at each step. If we wish explicitly to show this variable in the derivation, we can write $S \Rightarrow a_1A_1\beta_1 \Rightarrow a_1a_2A_2\beta_2 \Rightarrow \ldots \Rightarrow a_1a_2 \ldots a_{n-1}A_{n-1} \Rightarrow a_1a_2 \ldots a_n$, where β_i represents the string of remaining variables. The following shows how $NPDA$ precisely simulates this leftmost derivation for $x = a_1a_2 \ldots a_n$:

λ: $(s_o, S) \vdash (s_o, a_1A_1\beta_1)$ (rule 1)
a_1: $(s_o, a_1A_1\beta_1) \vdash (s_o, A_1\beta_1)$ (rule 2)

$$\begin{aligned}
\lambda&: (s_o, A_1\beta_1) \vdash (s_o, a_2A_2\beta_2) && \text{(rule 1)}\\
a_2&: (s_o, a_2A_2\beta_2) \vdash (s_o, A_2\beta_2) && \text{(rule 2)}\\
&\vdots\\
a_{n-1}&: (s_o, a_{n-1}A_{n-1}) \vdash (s_o, A_{n-1}) && \text{(rule 2)}\\
\lambda&: (s_o, A_{n-1}) \vdash (s_o, a_n) && \text{(rule 1)}\\
a_n&: (s_o, a_n) \vdash (s_o, \lambda) && \text{(rule 2)}
\end{aligned}$$

Thus $x \in T(NPDA)$.

(2) If $x = b_1b_2 \ldots b_m$ is accepted by *NPDA*, then $b_i \in I$. The transitions in *NPDA* in accepting x take place when the input b is introduced, or "spontaneously" when the input is λ. We can therefore write $x = a_1a_2 \ldots a_n$, where $a_i = \lambda$, or $a_i = b_j$, while maintaining the order and in such a way that exactly one transition of *NPDA* goes together with each a_i in the acceptance of x. Thus we have the following steps for accepting x:

$$\begin{aligned}
a_1&: (s_o, S) \vdash (s_o, \omega_1)\\
a_2&: (s_o, \omega_1) \vdash (s_o, \omega_2)\\
&\vdots\\
a_n&: (s_o, \omega_{n-1}) \vdash (s_o, \lambda)
\end{aligned}$$

With regard to rule 2, it follows directly that $\omega_{n-1} = a_n$, and tritely $\omega_{n-1} \overset{*}{\Rightarrow} a_n$ in grammar G. We shall now take as an inductive hypothesis that $\omega_i \overset{*}{\Rightarrow} a_{i+1} \ldots a_n$ in G, and show that $\omega_{i-1} \Rightarrow a_i \ldots a_n$. It then follows by induction (going back to $n-1$, for which the theorem is valid) that $\omega_o = S \overset{*}{\Rightarrow} a_1 \ldots a_n$.

We thus suppose that $\omega_i \overset{*}{\Rightarrow} a_{i+1} \ldots a_n$. We know that a_i: $(s_o, \omega_{i-1}) \vdash (s_o, \omega_i)$. There are two possibilities: $a_i \in V_T$ or $a_i = \lambda$. Let us first suppose that $a_i \in V_T$. In that case the transition a_i: $(s_o, \omega_{i-1}) \vdash (s_o, \omega_i)$ can only have taken place by means of rule 2, and consequently $\omega_{i-1} = a_i\omega_i$. But because $\omega_i \overset{*}{\Rightarrow} a_{i+1} \ldots a_n$ (induction hypothesis), it is true that $\omega_{i-1} = a_i\omega_i \overset{*}{\Rightarrow} a_ia_{i+1} \ldots a_n$, that which we had to prove.

Now let us suppose that $a_i = \lambda$. In this case the transition $a_i = \lambda$: $(s_o, \omega_{i-1}) \vdash (s_o, \omega_i)$ can only have taken place by means of rule 1, and consequently $\omega_{i-1} = A\omega'_{i-1}$ and $\omega_i = a\alpha\omega'_{i-1}$. Because $A \rightarrow a\alpha$ is by definition a production in G, it is true that $A\omega'_{i-1} \Rightarrow a\alpha\omega'_{i-1}$, or otherwise formulated $\omega_{i-1} \Rightarrow \omega_i$. According

to the induction hypothesis, however, $\omega_i = a_{i+1} \ldots a_n$, and consequently we have the following derivation: $\omega_{i-1} \Rightarrow a_{i+1} \ldots a_n = \lambda a_{i+1} \ldots a_n = a_i a_{i+1} \ldots a_n$, which is what we had to prove. We conclude, then, that $\omega_o = S \overset{*}{\Rightarrow} x$.

To illustrate Theorem 5.1., we offer the following example.

EXAMPLE 5.4. Take context-free language $L = \{a^n c b^n\}$, $n \geqslant 0$. A simple grammar for L is $G = (\{S, B\}, \{a, b, c\}, \{S \rightarrow aSB, B \rightarrow b, S \rightarrow c\}, S)$, which is in Greibach normal-form. According to the procedure given in the proof of Theorem 5.1., we construct the following push-down automaton which accepts language L: $NPDA = (S, I, \Gamma, \delta, s_o, \gamma_o)$, with $S = \{s_o\}$, $I = V_T = \{a, b, c\}$, $\Gamma = V = \{a, b, c, S, B\}$, $\gamma_o = S$, and with the following transition rules in δ:

1. $\delta(s_o, \lambda, S) = \{(s_o, aSB), (s_o, c)\}$
2. $\delta(s_o, \lambda, B) = \{(s_o, b)\}$
3. $\delta(s_o, a, a) = \{(s_o, \lambda)\}$
4. $\delta(s_o, b, b) = \{(s_o, \lambda)\}$
5. $\delta(s_o, c, c) = \{(s_o, \lambda)\}$

The following list shows the various steps by which *NPDA* accepts the sentence *aacbb*:

λ: $(s_o, S) \vdash (s_o, aSB)$	(rule 1)
a: $(s_o, aSB) \vdash (s_o, SB)$	(rule 3)
λ: $(s_o, SB) \vdash (s_o, aSBB)$	(rule 1)
a: $(s_o, aSBB) \vdash (s_o, SBB)$	(rule 3)
λ: $(s_o, SBB) \vdash (s_o, cBB)$	(rule 1)
c: $(s_o, cBB) \vdash (s_o, BB)$	(rule 5)
λ: $(s_o, BB) \vdash (s_o, bB)$	(rule 2)
b: $(s_o, bB) \vdash (s_o, B)$	(rule 4)
λ: $(s_o, B) \vdash (s_o, b)$	(rule 2)
b: $(s_o, b) \vdash (s_o, \lambda)$	(rule 4)

To complete the proof of equivalence between nondeterministic push-down automata and context-free grammars, we must prove the following theorem.

THEOREM 5.2. For every language T which is accepted by a nondeterministic push-down automaton, there is a context-free grammar G which generates precisely T.

PROOF. Let T be the language accepted by $NPDA = (S, I, \Gamma, \delta, s_o, \gamma_o)$. We must construct a context-free grammar $G = (V_N, V_T, P, S)$ as follows:

(i) V_N consists of compound elements $[s_i, \gamma, s_j]$, where s_i and s_j are elements of S, and γ is an element of Γ. V_N also contains S, which is not compound.

(ii) $V_T = I$.

(iii) P contains the following productions:

1. $S \to [s_o, \gamma_o, s]$ for every s in S.
2. $\{[s, \gamma, s_{n+1}] \to a[s_1, \gamma_1, s_2]\,[s_2, \gamma_2, s_3] \ldots [s_n, \gamma_n, s_{n+1}]$ for any numbering of states in $S\}$ for every transition rule in δ of the form: $\delta(s, a, \gamma)$ contains $(s_1, \gamma_1\gamma_2 \ldots \gamma_n)$.

The second rule gives productions in G for every transition rule in $NPDA$. These productions are in Greibach normal-form: to the right of the arrow there is a terminal element followed by 0 or more variables. The case of 0 variables occurs when $\gamma_1\gamma_2 \ldots \gamma_n = \lambda$, thus in transition rules in which $\delta(s, a, \gamma)$ includes (s_1, λ); this gives the following productions in G: $[s, \gamma, s_i] \to a$ for all s_i in S.

Although the first production is not Greibach normal-form, every leftmost derivation of G is as follows: $S \Rightarrow \alpha_o \Rightarrow a_1\alpha_1 \Rightarrow a_1a_2\alpha_2 \Rightarrow \ldots \Rightarrow a_1a_2 \ldots a_n$, where every α is a string of variables. Each of these variables is composed of three elements. If we examine the components γ in these variables, we find that they stand for every α_i precisely in the order they take on in the push-down store when $a_1a_2 \ldots a_i$ is introduced into the automaton. Thus the grammar simulates the push-down automaton. Before continuing the proof of the theorem, we present an example in which this simulation is clearly to be seen.

EXAMPLE 5.5. Let $NPDA = (S, I, \Gamma, \delta, s_o, \gamma_o)$ be a nondeterministic push-down automaton with $S = \{s_o, s_1\}$, $I = \{a, b\}$,

$\Gamma = \{\gamma_o, \gamma_1\}$, and the transition rules given in Table 5.1. We must construct a grammar $G = (V_N, V_T, P, S)$ according to the above procedure: V_N consists of S and all triples $[s_i, a \vee b, s_j]$. For convenience we use a separate upper case letter to denote each of these compound variables:

$$A = [s_o, \gamma_o, s_o],\ B = [s_o, \gamma_o, s_1],\ C = [s_o, \gamma_1, s_o],\ D = [s_o, \gamma_1, s_1],$$
$$E = [s_1, \gamma_o, s_o],\ F = [s_1, \gamma_o, s_1],\ G = [s_1, \gamma_1, s_o],\ H = [s_1, \gamma_1, s_1].$$

Further $V_T = \{a, b\}$; the productions are given in Table 5.1. in both complete and abbreviated notation, grouped according to the corresponding transition rules. The abbreviated notation clearly shows that only the numbered productions lead to terminal strings.

TABLE 5.1. Transition Rules of *NPDA* and Corresponding Productions of Equivalent Grammar G (Example 5.5.).

Transition Rules *NPDA*	Productions G	Abbreviated Notation
	1. $S \to [s_o, \gamma_o, s_o]$	$S \to A$
	2. $S \to [s_o, \gamma_o, s_1]$	$S \to B$
(a) $\delta(s_o, a, \gamma_1) = \{(s_1, \gamma_1)\}$	$[s_o, \gamma_1, s_o] \to a[s_1, \gamma_1, s_o]$	$C \to aG$
	3. $[s_o, \gamma_1, s_1] \to a[s_1, \gamma_1, s_1]$	$D \to aH$
(b) $\delta(s_o, b, \gamma_o) = \{(s_o, \gamma_1\gamma_o)\}$	$[s_o,\gamma_o,s_o] \to b[s_o,\gamma_1,s_o]\,[s_o,\gamma_o,s_o]$	$A \to bCA$
	4. $[s_o,\gamma_o,s_o] \to b[s_o,\gamma_1,s_1]\,[s_1,\gamma_o,s_o]$	$A \to bDE$
	$[s_o,\gamma_o,s_1] \to b[s_o,\gamma_1,s_o]\,[s_o,\gamma_o,s_1]$	$B \to bCB$
	5. $[s_o,\gamma_o,s_1] \to b[s_o,\gamma_1,s_1]\,[s_1,\gamma_o,s_1]$	$B \to bDF$
(c) $\delta(s_o, b, \gamma_1) = \{(s_o, \gamma_1\gamma_1)\}$	$[s_o,\gamma_1,s_o] \to b[s_o,\gamma_1,s_o]\,[s_o,\gamma_1,s_o]$	$C \to bCC$
	$[s_o,\gamma_1,s_o] \to b[s_o,\gamma_1,s_1]\,[s_1,\gamma_1,s_o]$	$C \to bDG$
	$[s_o,\gamma_1,s_1] \to b[s_o,\gamma_1,s_o]\,[s_o,\gamma_1,s_1]$	$D \to bCD$
	6. $[s_o,\gamma_1,s_1] \to b[s_o,\gamma_1,s_1]\,[s_1,\gamma_1,s_1]$	$D \to bDH$
(d) $\delta(s_o, \lambda, \gamma_o) = \{(s_o, \lambda)\}$	7. $[s_o, \gamma_o, s_o] \to \lambda$	$A \to \lambda$
(e) $\delta(s_1, a, \gamma_o) = \{(s_o, \gamma_o)\}$	8. $[s_1, \gamma_o, s_o] \to a[s_o, \gamma_o, s_o]$	$E \to aA$
	9. $[s_1, \gamma_o, s_1] \to a[s_o, \gamma_o, s_1]$	$F \to aB$
(f) $\delta(s_1, b, \gamma_1) = \{(s_1, \lambda)\}$	10. $[s_1, \gamma_1, s_1] \to b$	$H \to b$
	$[s_1, \gamma_1, s_o] \to b$	$G \to b$

In order to show how G simulates *NPDA*, we give first the acceptance of the sentence *bbabba* by *NPDA*, and then the generation of the same sentence by G. Acceptance by *NPDA*:

b: $(s_o, \gamma_o) \vdash (s_o, \gamma_1\gamma_o)$	(rule b)
b: $(s_o, \gamma_1\gamma_o) \vdash (s_o, \gamma_1\gamma_1\gamma_o)$	(rule c)
a: $(s_o, \gamma_1\gamma_1\gamma_o) \vdash (s_1, \gamma_1\gamma_1\gamma_o)$	(rule a)
b: $(s_1, \gamma_1\gamma_1\gamma_o) \vdash (s_1, \gamma_1\gamma_o)$	(rule f)
b: $(s_1, \gamma_1\gamma_o) \vdash (s_1, \gamma_o)$	(rule f)
a: $(s_1, \gamma_o) \vdash (s_o, \gamma_o)$	(rule e)
λ: $(s_o, \gamma_o) \vdash (s_o, \lambda)$	(rule d)

Derivation by G:

$S \Rightarrow A$	(production 1)
$A \Rightarrow bDE$	(production 4)
$bDE \Rightarrow bbDHE$	(production 6)
$bbDHE \Rightarrow bbaHHE$	(production 3)
$bbaHHE \Rightarrow bbabHE$	(production 10)
$bbabHE \Rightarrow bbabbE$	(production 10)
$bbabbE \Rightarrow bbabbaA$	(production 8)
$bbabbaA \Rightarrow bbabba$	(production 7)

It should be noticed that the last step in this derivation is an abbreviation although this is theoretically not permitted with a context-free grammar. The abbreviation is a result of production 7 in Table 5.1, but this production is actually only a formalization of the convention introduced in paragraph 2.1., that λ can be added to a context-free language.

We can now continue with the proof of Theorem 5.2. We must show that $T(NPDA) = L(G)$. The proof follows two steps: first we must show that if $x \in T$, then x is also generated by G; then we must show the inverse of this statement.

(1) If $x = a_1a_2 \ldots a_m$ is in $T(NPDA)$, then $S \overset{*}{\Rightarrow} x$. To prove this we must show by induction that for every n the following is true: if x: $(s_i, \gamma) \overset{*}{\vdash} (s, \lambda)$ in n transitions, then $[s_i, \gamma, s_j] \overset{*}{\Rightarrow} x$ by the productions of G. We first prove the theorem for $n = 1$, then show that it is also valid for $n - 1$ or fewer steps, and consequently

that it holds for n steps; thence follows general validity. From that point it is not difficult to show that if x is accepted by *NPDA*, then it is also generated by G.

If $n = 1$, then either $x = a$ (where $a \in I$), or $x = \lambda$. In both cases $x\colon (s_i, \gamma) \vdash (s_j, \lambda)$, and therefore (s_i, x, γ) must include (s_j, λ), so that G (according to production 2) includes the production $[s_i, \gamma, s_j] \to x$. It follows directly that $[s_i, \gamma, s_j] \Rightarrow x$ is a derivation of G.

Let us now suppose that the theorem holds for fewer than n transition steps. Let us examine $x = a_1 a_2 \ldots a_m$ $(m \geqslant 0)$, for which $x\colon (s_i, \gamma) \vdash^* (s_j, \lambda)$ in precisely n transitions. The first step in this process is as follows: $a\colon (s_i, \gamma) \vdash (s_1, \gamma_1\gamma_2 \ldots \gamma_k)$. The element a here is either λ, or the first element a_1 of x. After the first step, the push-down store thus contains $\gamma_1\gamma_2 \ldots \gamma_k$, and $n - 1$ transitions remain to be made before this string is completely removed from the store. We know that this does finally occur, and that the respective γ_i's are successively removed. This, however, need not proceed directly, and might, on the contrary, follow various detours (γ_i might, for example, be replaced by a whole string of new push-down symbols, which will be removed when later elements of x are introduced into the input). Nevertheless it must remain possible to articulate the string $x = a_1 a_2 \ldots a_m$ in such a way that it can be written as $aw_1w_2 \ldots w_k$ where $a = \lambda$ or $a = a_1$ (dependent on the nature of the first step), and where every w_i leads to the removal of γ_i, when the operation on the step began in the proper state s_i. But if γ_i can be removed from the store with w_i as input, then it also holds that if γ_i should be the only element in the push-down store while the automaton is in state s_i, $w_i\colon (s_i, \gamma_i) \vdash^* (s_{i+1}, \lambda)$, where s_{i+1} is precisely the state beginning with which w_{i+1} would empty the store if only γ_{i+1} were in it. For every w this process of emptying takes fewer than n steps, and there are productions in G such that $[s_i, \gamma_i, s_{i+1}] \overset{*}{\Rightarrow} w_i$ (induction hypothesis). It holds also that the string of variables $[s_1, \gamma_1, s_2]$ $[s_2, \gamma_2, s_3] \ldots [s_k, \gamma_k, s_{k+1}]$ can be rewritten by means of the productions in G as the terminal string $w_1w_2 \ldots w_k$. From $a\colon (s_i, \gamma) \vdash (s_1, \gamma_1\gamma_2 \ldots \gamma_k)$, however, we know that $(s_1, \gamma_1\gamma_2 \ldots \gamma_k)$ is an ele-

ment of $\delta(s_i, a, \gamma)$, and therefore G (according to production 2) includes the production $[s_i, \gamma, s_{k+1}] \rightarrow a[s_1, \gamma_1, s_2]\,[s_2, \gamma_2, s_3] \ldots [s_k, \gamma_k, s_{k+1}]$. It therefore holds that $[s_i, \gamma, s_{k+1}] \overset{*}{\Rightarrow} aw_1w_2 \ldots w_k = x$, from which we see that the theorem also holds for n transitions. By induction, the theorem is valid in general.

It is true of every x which is accepted by *NPDA* that x: $(s_o, \gamma_o) \overset{*}{\vdash} (s, \lambda)$, and consequently, by the theorem as proven, $[s_o, \gamma_o, s] \overset{*}{\Rightarrow} x$ in G. According to production 1, $S \rightarrow [s_o, \gamma_o, s]$ for every s in S; therefore $S \overset{*}{\Rightarrow} x$.

(2) If $S \overset{*}{\Rightarrow} x$, then $x \in T(NPDA)$. We shall first prove that for every $n > 0$, if $[s_i, \gamma, s_j] \overset{*}{\Rightarrow} x$ in G in n transitions, then x: $(s_i, \gamma) \overset{*}{\vdash} (s_j, \lambda)$ in *NPDA*. Let $n = 1$. Then $[s_i, \gamma, s_j] \rightarrow x$ is a production of G, and consequently, given the construction of G, either $x \in V_T$ or $x = \lambda$. Likewise $\delta(s_i, x, \gamma)$ includes (s_j, λ), from which follows that the theorem holds for $n = 1$.

Let the theorem hold for derivations in G with fewer than n steps (induction hypothesis). Let $[s, \gamma, t] \overset{*}{\Rightarrow} x = a_1a_2 \ldots a_m$ be a derivation which demands exactly n steps. This is possible, given the form of production 2, if a leftmost derivation is as follows: $[s, \gamma, t] \Rightarrow a[t_1]\,[t_2] \ldots [t_k] \overset{*}{\Rightarrow} aw_1[t_2]\,[t_3] \ldots [t_k] \overset{*}{\Rightarrow} \ldots \overset{*}{\Rightarrow} aw_1w_2 \ldots w_k = a_1a_2 \ldots a_m = x$. Here $[t_i]$ represents the triad $[s_i, \gamma_i, s_{i+1}]$, and w_i is a string of one or more successive elements a from x. Every w_i can be derived from $[t_i]$ by the productions of G, and in general $[t_i] \overset{*}{\Rightarrow} w_i$ in fewer than n steps. On the basis of the induction hypothesis, however, w_i: $(s_i, \gamma_i) \overset{*}{\vdash} (s_{i+1}, \lambda)$ for every $i = 1, \ldots, k$. But then it is also the case that $w_1w_2 \ldots w_k$: $(s_1, \gamma_1\gamma_2 \ldots \gamma_k) \overset{*}{\vdash} (s_2, \gamma_2 \ldots \gamma_k) \overset{*}{\vdash} \ldots \overset{*}{\vdash} (s_{k+1}, \lambda)$, and consequently also x: $(s, \gamma) \overset{*}{\vdash} (t = s_{k+1}, \lambda)$. By induction, the theorem holds for every $n > 0$.

The derivation $S \overset{*}{\Rightarrow} x$ can be written $S \Rightarrow [s_o, \gamma_o, s] \overset{*}{\Rightarrow} x$. If x is generated by G, then $[s_o, \gamma_o, s] \overset{*}{\Rightarrow} x$, so that, on the basis of Theorem 5.2, x: $(s_o, \gamma_o) \overset{*}{\vdash} (s, \lambda)$, which by definition means that $x \in T(NPDA)$.

It follows from Theorems 5.1. and 5.2. that the class of languages which are accepted by nondeterministic push-down automata is precisely the same as the class of languages generated by context-free grammars.

6

LINEAR BOUNDED AUTOMATA

An automaton has been discovered which accepts precisely the languages of the context-sensitive class. Like the push-down automaton, it is unlimited, but in an interesting way. In effect, it disposes of as much storage capacity as the input string is long: the store is small for a short string, large for a long string. It is as if one had to calculate the sum of two numbers and were given exactly the same amount of space on a blackboard for counting as the two original numbers occupy. One would be allowed to write and to erase as often as desired, but could use no more space than that allowed.

The automaton in question is called LINEAR BOUNDED AUTOMATON, *LBA*. In this chapter we shall show that linear bounded automata are equivalent to context-sensitive grammars. But the proof of this equivalence is considerably more complicated than those in the preceding chapters, and we will not be able to discuss it fully within the scope of this book. Therefore we shall limit ourselves here to a global proof of the theorem that for every context-sensitive grammar there is an equivalent linear bounded automaton. We have chosen this particular theorem for proof because it refers to the Kuroda normal-form which will be used later in dealing with linguistic applications (in Volume II), and because it provides a good illustration of the way linear bounded automata work.

6.1. DEFINITIONS AND CONCEPTS

In several ways linear bounded automata resemble finite automata. In chapter 4 we observed that finite automata begin operating in an initial state and first read the leftmost symbol on the input tape. They then proceed to read the input symbols from left to right, until a final state is reached. Like finite automata, linear bounded automata also have a limited number of states, and they too begin their operation in an initial state by reading the leftmost symbol on the input tape. But linear bounded automata are capable of more than finite automata in two respects. In the first place, they can both read and write: they can write over a symbol which they have read, and replace it with another symbol. In the second place, they can move the input tape not only from left to right, but also from right to left; moreover, at a transition (a change of state and or the replacement of a symbol in the input tape), they can remain at the same position on the tape. In writing they can use "auxiliary symbols" which are not part of the input vocabulary. Because linear bounded automata may write only within the bondaries of the original input string, two boundary symbols (#) are placed on the tape, to the left of the first element and to the right of the last. Linear bounded automata always start in an initial state at the left-hand boundary symbol; they are said to accept the input when they pass over the right-hand boundary symbol in a final state. This latter is possible, of course, only after they have dealt with each element between the boundary symbols. The formal definitions are as follows.

A linear bounded automaton is a system $LBA = (S, I, \Gamma, \delta, s_0, \#, F)$ in which:

(1) S is a finite, nonempty set of STATES, with $s_0 \in S$ as INITIAL STATE, and $F \in S$ as the set of FINAL STATES. (States are, as usual, denoted by the letter s with a subscript, or by r, s, t, ...)

(2) I is a finite INPUT-VOCABULARY (notation as usual).

(3) Γ is a finite set of TAPE SYMBOLS, the vocabulary of symbols which can appear on the tape. I belongs to this set, as do all auxiliary symbols which can be used in writing. (Notation: tape

symbols are in general denoted by γ with a subscript; strings of auxiliary symbols are denoted by lower case letters from the end of the Greek alphabet, χ, ψ, ω. If it is known that a tape symbol belongs to the input vocabulary, the notation for I can be used.) There is also a special tape symbol $\#$, the BOUNDARY SYMBOL.

(4) δ is a finite set of TRANSITION RULES. A transition rule indicates for a pair of state and tape symbols what the following state and tape symbol will be; it also indicates if the band remains at the same place, goes one place to the right, or one place to the left. This is written as follows: we say that (s_m, γ_n, k) is *in* $\delta(s_i, \gamma_j)$ if the automaton, in state s_i and reading γ_j, *can change* to state s_m and write γ_n in the place of γ_j. The letter k shows in which direction the automaton moves on the tape: $k = -1$ indicates that it goes to the left; $k = 1$ indicates that it goes to the right; $k = 0$ indicates that it remains in the same place and reads the symbol it has written in the place of γ_n. By convention, $\delta(s, \gamma)$ always contains $(s, \gamma, 0)$. We say "can change" because linear bounded automata are nondeterministic; a linear bounded automaton has in principal several possible transitions for each configuration. δ maps the cartesian product $S \times \Gamma$ into subsets of $S \times \Gamma \times \{-1, 0, 1\} \cup \varphi$. In every operation the boundary symbols must remain in place; thus, whenever the automaton reads $\#$ it writes $\#$ over it. In formal terms, if (s', γ, k) is in $\delta(s, \#)$, then $\gamma = \#$ for every s', and vice versa if $(s', \#, k)$ is in $\delta(s, \gamma)$, then $\gamma = \#$.

The concept of "configuration" calls for some further clarification. This can best be done with a visual representation of the operation of a linear bounded automaton, as in Figure 6.1. In that figure we see the initial and final situations in the process of accepting the string $x = a_1 a_2 \ldots a_n$, as well as two possible situations during the operation.

A useful way of showing the entire configuration of automaton and tape is to write the state of the automaton to the left of the symbol which is being read. The configuration in Figure 6.1.a. can thus be denoted by $s_0 \# a_1 \ldots a_n \#$ because the automaton is in state s_0 and is reading the left-hand boundary symbol. For the configuration in Figure 6.1.b. we write $\# \gamma_1 \gamma_2 \ldots \gamma_k s_j a_{k+1} \ldots a_n \#$,

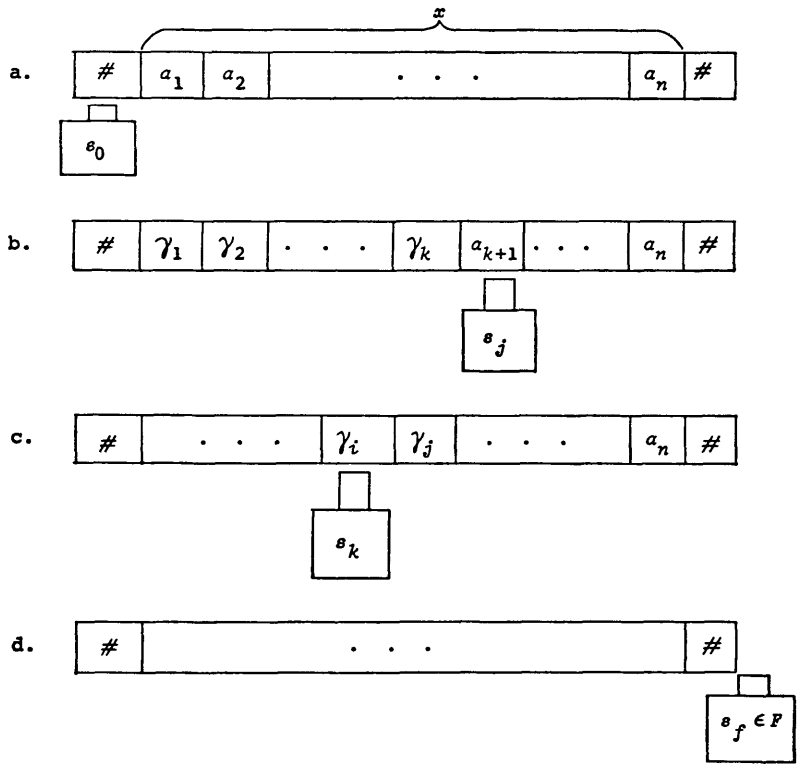


Fig. 6.1. A Linear Bounded Automaton in Operation

a. Situation at start.

b. } Possible situations during operation.
c. }

d. Situation after accepting x.

in which we see that the tape symbol a_{k+1} is being read in state s_j. The configuration in Figure 6.1.c. is written $\# \ldots s_k\gamma_i\gamma_j \ldots a_n\#$; that represented in Figure 6.1.d. is written $\# \ldots \#s_f$. If the automaton passes from configuration C to configuration C' in one step we write $C \vdash C'$, and when the change takes place by an undetermined number of transitions, the notation is $C \vdash^* C'$.

A linear bounded automaton *LBA* ACCEPTS a string x when

$s_o\#x\# \vdash^* \#\omega\#s_f$, where $x \in \Gamma^*$, $\omega \in \Gamma^*$, and $s_f \in F$. The LANGUAGE $T(LBA)$ accepted by LBA is the set of strings which are accepted by LBA: $T(LBA) = \{x | s_o\#x\# \vdash^* \#\omega\#s_f, x \in \Gamma^*, \omega \in \Gamma^*, s_f \in F\}$.

EXAMPLE 6.1. Let $LBA = (S, I, \Gamma, \delta, s_o, \#, F)$ be a linear bounded automaton in which $S = \{s_o, s_1, s_2, s_3, s_4, s_f\}$, $I = \{a, b\}$, $\Gamma = \{a, b, \gamma_a, \gamma_b, \#\}$, $F = \{s_f\}$, and with the following transition rules in δ:

1. $\delta(s_o, \#) = \{(s_1, \#, 1)\}$
2. $\delta(s_1, a) = \{(s_2, \gamma_a, 1)\}$
3. $\delta(s_1, \#) = \{(s_f, \#, 1)\}$
4. $\delta(s_1, \gamma_b) = \{(s_1, \gamma_b, 1)\}$
5. $\delta(s_2, a) = \{(s_2, a, 1)\}$
6. $\delta(s_2, b) = \{(s_2, b, 1)\}$
7. $\delta(s_2, \gamma_b) = \{(s_3, \gamma_b, -1)\}$
8. $\delta(s_2, \#) = \{(s_3, \#, -1)\}$
9. $\delta(s_3, b) = \{(s_4, \gamma_b, -1)\}$
10. $\delta(s_4, a) = \{(s_4, a, -1)\}$
11. $\delta(s_4, b) = \{(s_4, b, -1)\}$
12. $\delta(s_4, \gamma_a) = \{(s_1, \gamma_a, 1)\}$

$\delta(s, \gamma) = \varphi$ for all other cases for which no convention holds.

It is immediately obvious that this automaton is deterministic: there is never more than one possible transition. We shall first show how the automaton accepts the string ab. The input tape carries the string $\#ab\#$, and the first configuration is $s_o\#ab\#$, i.e. LBA is reading the left-hand boundary symbol in the initial state s_o. The successive steps are as follows:

$s_o\#ab\# \vdash \#s_1ab\#$ (rule 1)
$\#s_1ab\# \vdash \#\gamma_a s_2 b\#$ (rule 2)
$\#\gamma_a s_2 b\# \vdash \#\gamma_a b s_2\#$ (rule 6)
$\#\gamma_a b s_2\# \vdash \#\gamma_a s_3 b\#$ (rule 8)
$\#\gamma_a s_3 b\# \vdash \#s_4\gamma_a\gamma_b\#$ (rule 9)
$\#s_4\gamma_a\gamma_b\# \vdash \#\gamma_a s_1\gamma_b\#$ (rule 12)
$\#\gamma_a s_1\gamma_b\# \vdash \#\gamma_a\gamma_b s_1\#$ (rule 4)
$\#\gamma_a\gamma_b s_1\# \vdash \gamma\#_a\gamma_b\#s_f$ (rule 3)

The following shows in short how the automaton accepts the string $aabb$: $s_o\#aabb\# \vdash \#s_1aabb\# \vdash \#\gamma_a s_2 abb\# \vdash^* \#\gamma_a abbs_2\# \vdash \#\gamma_a abs_3 b\# \vdash \#\gamma_a as_4 b\gamma_b\# \vdash^* \#s_4\gamma_a ab\gamma_b\# \vdash \#\gamma_a s_1 ab\gamma_b\# \vdash \#\gamma_a\gamma_a s_2 b\gamma_b\# \vdash \#\gamma_a\gamma_a bs_2\gamma_b\# \vdash \#\gamma_a\gamma_a s_3 b\gamma_b\# \vdash \#\gamma_a s_4\gamma_a\gamma_b\gamma_b\# \vdash \#\gamma_a\gamma_a s_1\gamma_b\gamma_b\# \vdash^* \#\gamma_a\gamma_a\gamma_b\gamma_b\#s_f$.

Thus this automaton shifts back and forth between the boundary symbols until every a has been converted into γ_a, and every b into γ_b. It can reach the final state s_f only if there are as many γ_a's as γ_b's, and when the γ_a's are in the left-hand half of the tape, and the γ_b's in the right hand half. This automaton accepts the language $\{a^n b^n \mid n \geqslant 0\}$.

6.2. LINEAR BOUNDED AUTOMATA AND CONTEXT-SENSITIVE GRAMMARS

The equivalence of linear bounded automata and context-sensitive grammars is established in Theorems 6.1. and 6.2.

THEOREM 6.1. For every context-sensitive language L, there is a linear bounded automaton which accepts L and only L.

PROOF (summarized). Let L be a context-sensitive language. According to Theorem 2.11., there is a grammar G in Kuroda normal-form which generates L. We must construct a linear bounded automaton such that $T(LBA) = L(G)$. Let $G = (V_N, V_T, P, S)$. The automaton $LBA = (S, I, \Gamma, \delta, s_0, \#, F)$ must have the following construction:

(i) $S = \{s_0, s_1, t_0, t_1, \{t_A\}, r_0, r_1\}$, with s_0 as both initial and final state: $F = \{s_0\}$.

(ii) $I = V_T$

(iii) $\Gamma = V_N \cup V_T \cup \#$

(iv) δ contains the following transition rules:

1. $\delta(s_0, \#) = \{(s_1, \#, 1)\}$	
2. $\delta(s_1, a) = \{(s_1, a, 1)\}$	for every a in V_T
3. $\delta(s_1, \#) = \{(t_0, \#, -1)\}$	
4. $\delta(t_0, A)$ contains $(t_0, A, 1)$	for every A in V_N
5. $\delta(t_0, A)$ contains $(t_0, A, -1)$	for every A in V_N
6. $\delta(t_0, a)$ contains $(t_0, a, 1)$	for every a in V_T
7. $\delta(t_0, a)$ contains $(t_0, a, -1)$	for every a in V_T
8. $\delta(t_0, B)$ contains $(t_0, A, 0)$	for all productions $A \to B$ in P

9. $\delta(t_0, a)$ contains $(t_0, A, 0)$ — for all productions $A \to a$ in P
10. $\delta(t_0, C)$ contains $(t_A, A, 1)$ } — for all productions $AB \to CD$ in P
11. $\delta(t_A, D)$ contains $(t_0, B, 0)$ }
12. $\delta(t_0, S)$ contains $(r_0, S, -1)$
13. $\delta(r_0, \#) = \{(r_1, \#, 1)\}$
14. $\delta(r_1, S) = \{(t_1, \#, 1)\}$ } — for all productions $S \to SA$ in P
15. $\delta(t_1, A) = \{(t_0, S, 0)\}$ }
16. $\delta(t_1, \#) = \{(s_0, \#, 1)\}$

In all other cases where no convention holds, $\delta(s, \gamma) = \varphi$.

We shall now show, without complete proof by mathematical induction, that this linear bounded automaton simulates the derivations of G and only those of G. The states s_0 and s_1 function to verify that a string of terminal elements is found between the two boundary symbols $\#$. Rules 1 and 2 show that the automaton starting at the left-hand boundary symbol passes over all terminal elements until the right-hand boundary symbol is reached. Rule 3 indicates that at that point state t_0 is reached. If symbols other than terminal elements are found between the boundary symbols, the machine blocks and the string is not accepted. Rules 4 through 7 see to it that the automaton can move freely to the left or to the right without altering the content of the input; it can simply write the symbol it reads. Rules 8 through 11 see to it that the automaton can transpose elements or pairs of elements only according to the productions in P. Rules 12 through 15 see to the correct inversion of productions $S \to SA$, the only rules in Kuroda normal-form in which S can appear to the right of the arrow. Because these are the only expanding productions in the grammar, it must be possible to derive the input string x in grammar G as $S \Rightarrow SA \Rightarrow SAA \Rightarrow \ldots \Rightarrow SA \ldots A \overset{*}{\Rightarrow} x$. This is simulated in reverse order by the linear bounded automaton by replacing $\#SAB\ldots\#$, where possible, with $\#\#SB\ldots\#$. This can occur because when the automaton in the "work-state" t_0 reads S, it changes to state r_0 (rule 12) and moves one place to the left to see if there is an S next to the boundary symbol $\#$. If that is the

case, the automaton changes to state r_1 and, provided that $S \rightarrow SA$ is a production of P, rules 14 and 15 replace SA with $\#S$, and the work-state t_o is again reached. The automaton then sees if SB can be reduced to S; if it is, $\# \# \# S \ldots \#$ appears on the tape, and the process continues. In this way the string $\# \# \ldots \# S \#$ will appear on the tape only if x can be derived from S. Once the automaton has reached state t_o, rules 12, 13, and 14[1] see to it that it goes on to state t_1 and proceeds to the right in order to read the last boundary symbol. According to rule 16, when the automaton reaches the final state s_o and the tape is pushed out, string x is accepted.

If we wish to have LBA also accept the null-string λ, we must add a new state t_λ, and two new transition rules: $\delta(t_o, \#)$ contains $(t_\lambda, \#, 1)$, and $\delta(t_\lambda, \#)$ contains $(s_o, \#, 1)$. With these, when the input is λ, the final state is reached immediately after completion of the steps required by rules 1, 2, and 3.

EXAMPLE 6.2. Take grammar $G = (V_N, V_T, P, S)$, with $V_N = \{S, A, B\}$, $V_T = \{a, b\}$, and the following productions:

(a) $S \rightarrow SA$
(b) $S \rightarrow B$
(c) $BA \rightarrow AB$
(d) $A \rightarrow a$
(e) $B \rightarrow b$

Because of production (c) it is clear that grammar G is context-sensitive and that it is in Kuroda normal-form. G generates the language $L(G) = \{a^i b a^j \mid i + j \geqslant 0\}$. The sentences are thus strings of a's with one b in them. Production (a) generates the string SA^n; production (b) replaces the single S with B; by production (c) the B can be moved any number of places to the right. Productions (d) and (e) replace the variables with terminal symbols.

[1] Notice that rule 14 exists only if there is indeed a production $S \rightarrow SA$ in P. If this were not the case, the operation would stop. When no such production exists, language $L(G)$ consists exclusively of sentences of length 1, and it obviously remains possible to construct a linear bounded automaton which accepts that language and only that language. Also rule 14 strictly violates the convention that no new boundary symbols may be written. Paragraph 7.1 gives an easy way out.

We can construct a linear bounded automaton *LBA* which accepts $L(G)$, according to the procedure given in the proof of Theorem 6.1. Thus $LBA = (S, I, \Gamma, \delta, s_0, \#, F)$, with $S = \{s_0, s_1, t_0, t_1, t_B, r_0, r_1\}$, $I = \{a, b\}$, $\Gamma = \{S, A, B, a, b, \#\}$, $F = \{s_0\}$, and the following transition rules in δ:

1. $\delta(s_0, \#) = \{(s_1, \#, 1)\}$
2. $\delta(s_1, a) = \{(s_1, a, 1)\}$ because $a \in V_T$
3. $\delta(s_1, b) = \{(s_1, b, 1)\}$ because $b \in V_T$
4. $\delta(s_1, \#) = \{(t_0, \#, -1)\}$
5. $\delta(t_0, S) = \{(t_0, S, 1), (t_0, S, -1), (r_0, S, -1)\}$ because $S \in V_N$
6. $\delta(t_0, A) = \{(t_0, A, 1), (t_0, A, -1), (t_B, B, 1)\}$ because $A \in V_N$, and $BA \to AB$ in P
7. $\delta(t_0, B) = \{(t_0, B, 1), (t_0, B, -1), (t_0, S, 0)\}$ because $B \in V_N$, and $S \to B$ in P
8. $\delta(t_0, a) = \{(t_0, a, 1), (t_0, a, -1), (t_0, A, 0)\}$ because $a \in V_T$, and $A \to a$ in P
9. $\delta(t_0, b) = \{(t_0, b, 1), (t_0, b, -1), (t_0, B, 0)\}$ because $b \in V_T$, and $B \to b$ in P
10. $\delta(t_B, B) = \{(t_0, A, 0)\}$ because $BA \to AB$ in P
11. $\delta(r_0, \#) = \{(r_1, \#, 1)\}$
12. $\delta(r_1, S) = \{(t_1, \#, 1)\}$ because $S \to SA$ in P
13. $\delta(t_1, A) = \{(t_0, S, 0)\}$ because $S \to SA$ in P
14. $\delta(t_1, \#) = \{(s_0, \#, 1)\}$

The following shows the consecutive configurations in *LBA* for the acceptance of the sentence *abaa*; the numbers over the transition symbols $\vdash$ indicate the rule used in the transition.

$$s_0\#abaa\# \overset{1}{\vdash} \#s_1abaa\# \overset{2}{\vdash} \#as_1baa\# \overset{3}{\vdash} \#abs_1aa\# \overset{2}{\vdash} \#abas_1a\# \overset{2}{\vdash} \#abaas_1\# \overset{4}{\vdash} \#abat_0a\# \overset{8}{\vdash} \#abat_0A\# \overset{6}{\vdash} \#abt_0aA\# \overset{8}{\vdash} \#abt_0AA\# \overset{6}{\vdash} \#at_0bAA\# \overset{9}{\vdash} \#at_0BAA\# \overset{7}{\vdash} \#t_0aBAA\# \overset{8}{\vdash} \#t_0ABAA\# \overset{6}{\vdash} \#Bt_BBAA\# \overset{10}{\vdash} \#Bt_0AAA\# \overset{6}{\vdash} \#t_0BAAA\# \overset{7}{\vdash} \#t_0SAAA\# \overset{5}{\vdash} r_0\#SAAA\# \overset{11}{\vdash} \#r_1SAAA\# \overset{12}{\vdash} \#\#t_1AAA\# \overset{13}{\vdash} \#\#t_0SAA\# \overset{5,11,12,13}{\vdash} \#\#\#t_0SA\# \overset{5,11,12,13}{\vdash} \#\#\#\#t_0S\# \overset{5,11,12}{\vdash} \#\#\#\#\#t_1\# \overset{14}{\vdash} \#\#\#\#\#\#s_0.$$

To complete the statement of equivalence between linear bounded automata and context-sensitive grammars, we mention the following theorem.

THEOREM 6.2. For every linear bounded automaton *LBA*, there is a context-sensitive grammar G such that $T(LBA) = L(G)$.

A large number of rules are needed for the construction of such an equivalent context-sensitive grammar. The proof of this theorem is beyond the scope of this book; for it we refer the reader to Landweber (1963) and Kuroda (1964).

7

TURING MACHINES

An obvious question at this point is whether it is possible to design an automaton which could accept type-0 languages. The answer is affirmative; in fact some time before the theory of formal languages came into existence, Turing had described an automaton which later proved capable of accepting type-0 languages. The TURING MACHINE, as the automaton is called, is in principle capable of performing every operation which one might intuitively qualify as a MECHANICAL (EFFECTIVE)PROCEDURE (cf. paragraph 2.1.). In this chapter we will make the notion of "procedure" more explicit in order to facilitate an understanding of a number of important properties of natural languages. However, we shall first show that Turing machines accept type-0 languages and only type-0 languages, and that there exists a type-0 grammar for every language accepted by a Turing machine.

In this chapter, more than in the preceding chapters, theorems will be stated without proof. The theory of Turing machines has recourse to refined fields of mathematics, such as recursive function theory, with which we can suppose no acquaintance on the part of the reader. Moreover Turing machines are less of interest to linguistics and psycholinguistics than automata of more limited capacity. Therefore, we shall state and discuss only a limited number of theorems which are of some importance to linguistics.

7.1. DEFINITIONS AND CONCEPTS

Several different but equivalent terminologies have been used in describing Turing machines. The terminology which we shall use here is closely akin to that of linear bounded automata used in the preceding chapter

Like linear bounded automata, a Turing machine is made up of a finite automaton and a tape. A Turing machine can read and write tape symbols in the same way as a linear bounded automaton, but it is not subject to linear limitation: it can read and write to the left and to the right of the original input. We must suppose that the length of the tape is infinite, and that at the beginning of an operation a limited and continuous portion of the tape carries input symbols, bordered left and right by boundary symbols. To facilitate further formulation, we also suppose that the remainder of the tape is filled with boundary symbols. The machine can read the boundary symbols and replace them with other tape symbols, but cannot itself write boundary symbols. Consequently the tape carries a continuous string of input symbols which cannot be interrupted by boundary symbols. On the other hand, there may be "pseudo-boundary symbols", equivalent in every respect to the ordinary boundary symbols except in that they may also be written; in informal treatment of Turing machines, the distinction between the two types of boundary symbols is often neglected.

The notation will be the same as that used for linear bounded automata.

In formal terms, a Turing machine TM is a system $(S, I, \Gamma, \delta, s_0, \#, F)$, in which:

(1) S is a finite set of STATES, with s_0 as the INITIAL STATE, and $F \subset S$ as the set of FINAL STATES.

(2) I is a finite set of INPUT SYMBOLS.

(3) Γ is a finite set of TAPE SYMBOLS, of which I is a subset. Elements of Γ which are not elements of I are called AUXILIARY SYMBOLS, one of which is the BONUDARY SYMBOL $\#$. In the initial configuration the tape carries a string from I^*, bordered on the left and on the right by strings of boundary symbols of infinite length.

(4) δ is a finite set of TRANSITION RULES which indicate, for every pair of state and input symbol, what the machine must write (the boundary symbol cannot be written by the machine), what the following state will be, and whether the machine will remain at the same place on the tape, or move one step to the left or right. It is also possible for the machine to block. We can therefore say that δ maps $S \times \Gamma$ in $S \times \{\Gamma - \#\} \times \{-1, 0, 1\} \cup \varphi$. The transition rules have the form $\delta(s, \gamma) = (s', \gamma', k)$, where $k = -1$, 0, or 1. They should be interpreted as follows: if the Turing machine is in state s and reads the symbol γ, it passes to state s', writes γ' over the symbol γ, and moves the tape according to the value of k. Turing machines are deterministic; for every combination of state and tape symbol, only one transition is possible. It is possible, of course, to define nondeterministic Turing machines, but these are equivalent to deterministic Turing machines.[1] (We shall use nondeterministic Turing machines in the proof of Theorem 7.1.).

Before defining the language accepted by a Turing machine, we must indicate what is meant here by configuration. As was the case for linear bounded automata, a configuration in a Turing machine includes the content of the tape, the state of the automaton, and the position of the tape content in relation to the automaton. The notation is the same as for configurations in linear bounded automata, but redundant boundary symbols are omitted. Thus, for example, $s\#\gamma_1\gamma_2 \ldots \gamma_n\#$ stands for $\ldots\#\#s\#\gamma_1\gamma_2 \ldots \gamma_n\#\#\#\ldots$, and means that the Turing machine is in state s and is reading the boundary symbol directly to the left of the tape content $\gamma_1\gamma_2 \ldots \gamma_n$. The initial configuration is $s_0\#w\#$, where $w \in I^*$. A final configuration is every configuration in which the Turing machine is in a final state: $\omega s_f \chi$, where ω and χ are elements of Γ^*, and s_f is an element of F. In this case the automaton is said to STOP (stopping should not be confused with blocking). A string x in I^* is accepted by a Turing machine when $s_0\#x\# \vdash^* \omega s_f \chi$. The LANGUAGE accepted by a Turing machine is the set of strings in I^* accepted by the machine. Figure 7.1. illustrates an initial

[1] It is not known whether deterministic and nondeterministic linear bounded automata are also equivalent.

configuration, a configuration during operation, and a final configuration of a Turing machine in the process of accepting the input string $x = a_1 \ldots a_m$.

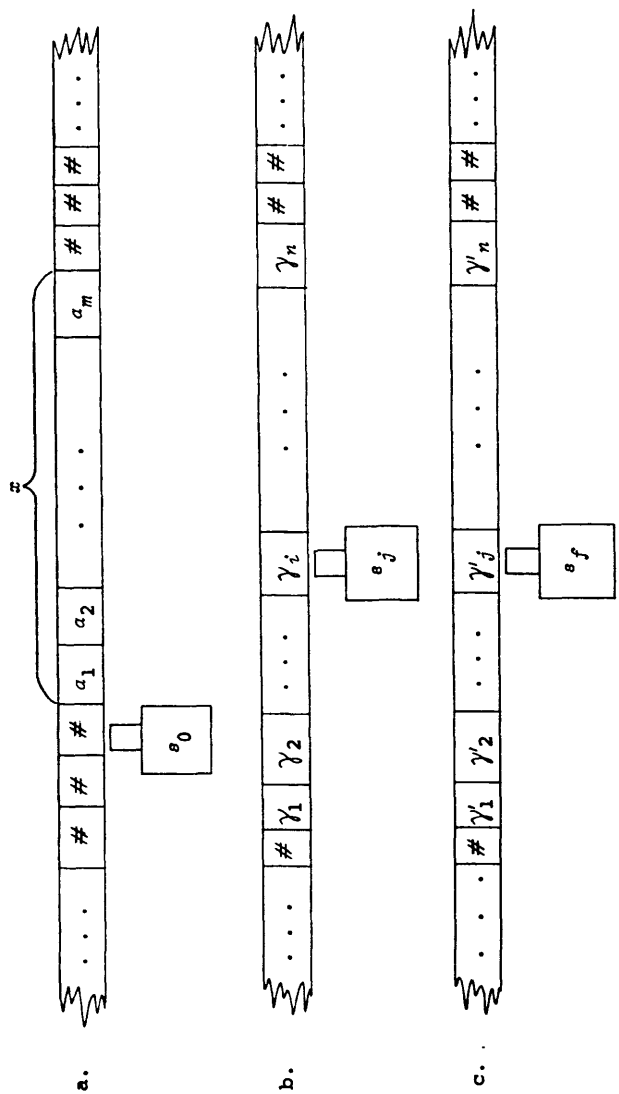


Fig. 7.1. A Turing Machine in Operation a. Initial configuration. b. The machine in operation. c. Final configuration.

7.2. A FEW ELEMENTARY PROCEDURES

In this paragraph we shall give a few examples of operations which can be performed by a Turing machine. The operations given here will later serve as elementary procedures in the comparison of Turing machines and type-0 grammars.

EXAMPLE 7.1. The transfer of information on the tape
In several cases it is necessary to transfer parts of the original input, or of the tape content which develops later, to a different place on the tape. In this way information can be stored while other operations are carried out. A simple example of this may be seen in the following Turing machine:

$TM = (S, I, \Gamma, \delta, s_0, \#, F)$, with $S = \{s_0, s_A, s_B, s_1, s_2, s_3\}$, $I = \{a, b\}$, $\Gamma = \{\#, a, b, c, A, B\}$, $F = \{s_3\}$, and where δ contains te following transition rules:

1.	$\delta(s_0, \#) = (s_0, \#, 1)$	13	$\delta(s_B, A) = (s_B, A, 1)$
2.	$\delta(s_0, a) = (s_A, c, 1)$	14.	$\delta(s_B, B) = (s_B, B, 1)$
3.	$\delta(s_0, b) = (s_B, c, 1)$	15.	$\delta(s_B, \#) = (s_1, B, -1)$
4.	$\delta(s_0, A) = (s_2, a, 1)$	16.	$\delta(s_1, a) = (s_1, a, -1)$
5.	$\delta(s_0, B) = (s_2, b, 1)$	17.	$\delta(s_1, b) = (s_1, b, -1)$
6.	$\delta(s_A, a) = (s_A, a, 1)$	18.	$\delta(s_1, c) = (s_0, c, 1)$
7.	$\delta(s_A, b) = (s_A, b, 1)$	19.	$\delta(s_1, A) = (s_1, A, -1)$
8.	$\delta(s_A, A) = (s_A, A, 1)$	20.	$\delta(s_1, B) = (s_1, B, -1)$
9.	$\delta(s_A, B) = (s_A, B, 1)$	21.	$\delta(s_2, A) = (s_2, a, 1)$
10.	$\delta(s_A, \#) = (s_1, A, -1)$	22.	$\delta(s_2, B) = (s_2, b, 1)$
11.	$\delta(s_B, a) = (s_B, a, 1)$	23.	$\delta(s_2, \#) = (s_3, \#, 0)$
12.	$\delta(s_B, b) = (s_B, b, 1)$	$\delta(-, -) = \varphi$ in all other cases.	

This Turing machine will replace every string x in I^+, where $|x| = n$, with a string $c^n x$; the original string of a's and b's is moved exactly its length to the right and is replaced by a string of c's whose length is equal to that of the string of a's and b's. Let us take for example the transfer of the string aab. The following gives the successive configurations in the machine; the number of the transition rule involved is given over the transition symbol,

except where a sequence of operations is repeated, in which case an asterisk * appears over the transition symbol.

$$s_0\#aab\# \overset{1}{\vdash} \#s_0aab\# \overset{2}{\vdash} \#cs_Aab\# \overset{6}{\vdash} \#cas_Ab\# \overset{7}{\vdash} \#cabs_A\# \overset{10}{\vdash} \#cas_1bA\# \overset{17}{\vdash} \#cs_1abA\# \overset{16}{\vdash} \#s_1cabA\# \overset{18}{\vdash} \#cs_0abA\# \overset{2}{\vdash} \#ccs_AbA\# \overset{*}{\vdash} \#ccbs_1AA\# \overset{19}{\vdash} \#ccs_1bAA\# \overset{17}{\vdash} \#cs_1cbAA\# \overset{18}{\vdash} \#ccs_0bAA\# \overset{3}{\vdash} \#cccs_BAA\# \overset{*}{\vdash} \#cccAAs_B\# \overset{15}{\vdash} \#cccAs_1AB\# \overset{*}{\vdash} \#cccs_0AAB\# \overset{4}{\vdash} \#cccas_2AB\# \overset{21}{\vdash} \#cccaas_2B\# \overset{22}{\vdash} \#cccaabs_2\# \overset{23}{\vdash} \#cccaabs_3\#.$$

EXAMPLE 7.2. The comparison of two strings

At times it is necessary to decide whether two strings of elements are identical. One can easily see that this is possible with a Turing machine. Imagine that we are interested in two strings r_1 and r_2 over a vocabulary V. We place the string r_1cr_2 on the tape, where $c \notin V$. The language $T = \{wcw\}$ is then a context-sensitive language with a vocabulary $V \cup c$. This means that there is a context-sensitive grammar which generates the sentences wcw and only the sentences wcw. There is consequently a linear bounded automaton *LBA* which accepts language T, and since Turing machines are a generalization of the linear bounded automaton, there is a Turing machine which accepts language T. In other words, a Turing machine accepts a string r_1cr_2 on condition that $r_1 = r_2$, and can therefore be considered an automaton which determines the identity of two strings.

7.3. TURING MACHINES AND TYPE-0 LANGUAGES

It is possible to construct a "Universal Turing machine" *UTM*, which can simulate the operation of any given Turing machine. A description of the *TM* (its transition rules, etc.) would be placed on the input tape of the *UTM*, while the input of the *TM* would appear in another place on the input tape of the *UTM*. Thus "programmed", the *UTM* would imitate the operation of the *TM* precisely. It is even possible to construct a *UTM* with only two states, but it would need an extremely large tape vocabulary.

However, it is not our intention to discuss Universal Turing machines here. We have mentioned them only to render the proposition acceptable that various elementary procedures for which Turing machines have been constructed can be combined in a single Turing machine. Such a machine could switch over from one procedure to another, just as a digital computer can switch from one subroutine to another. (The only essential difference between a computer and a Turing machine is that the latter disposes of an unlimited store: all information presented can be stored on a tape of infinite length.) With this background, we can discuss the following theorem.

THEOREM 7.1. For every type-0 language L there is a Turing machine such that $T(TM) = L$.

PROOF (summary). The construction of a TM which accepts language L is roughly as follows. Let L be a type-0 language, and G the type-0 grammar which generates it. Let x be a sentence in L. We put the string x on the input tape as $\#x\#$, and build in a procedure according to which the symbols c and S (neither of which are elements of V_T) are added to the string as follows: $\#xcS\#$. For every production $\alpha \rightarrow \beta$ in G we construct such transition rules for TM that a string α can be rewritten on the tape as β. If α is not of the same length as β, it will be necessary at rewriting to transfer the information directly to the right of α, either to the left or to the right, so that β will fit precisely into place. Therefore we must include a transfer procedure in the Turing machine, similar to that of Example 7.2.

TM can nondeterministically replace S with some β, where $S \rightarrow \beta$ is a production in G. Let $\beta = B_1 B_2 \ldots B_n$ (where B_i is an element of V, but not necessarily of V_N). In that case the tape shows $\#xcB_1B_2 \ldots B_n\#$.

Next we must build a procedure into TM according to which the left-hand members (α_i) of the productions $\alpha_1 \rightarrow \beta_i$ can be rewritten as an identification symbol. The automaton now nondeterministically chooses an α_i and a B_j from the string mentioned

above, and switches over to a comparison procedure which compares α_i element for element with B_jB_{j+1} .... Example 7.2. showed that such a comparison procedure is possible in principle. If string α_i is identical to string B_jB_{j+1} ..., it is replaced by β_i, the right-hand member of the production $\alpha_i \to \beta_i$. By continued replacement of strings between c and $\#$ according to the productions of G, a string of terminal elements is (nondeterministically) composed between c and $\#$. At this point the Turing machine can switch back to the comparison procedure in order to compare this new string with string x. If the two are identical, the machine reaches a final state and stops. It is clear that the terminal strings between c and $\#$ can only be sentences of $L(G)$, and that any sentence in $L(G)$ can appear there. Thus TM accepts the sentences of $L(G)$ and only the sentences of $L(G)$. If there is a nondeterministic Turing machine which accepts $L(G)$ and only $L(G)$, then there is a deterministic Turing machine which does the same.

THEOREM 7.2. For every language T accepted by a TM, there is a type-0 grammar G such that $L(G) = T(TM)$.

PROOF (summary). Let T be the language accepted by Turing machine TM. For every x in T, TM goes from its initial state to a final state in a finite number of operations: $s_o\#x\# \vdash^* \#\omega s_f \chi\#$, with $s_f \in F$ and $\omega, \chi \in \Gamma^*$. We write x as $a_1a_2 \ldots a_n$ $(n > 0)$. The first step in the process of accepting is as follows: $s_o\#a_1a_2\ldots a_n\# \vdash \#s_oa_1a_2\ldots a_n\#$. Another transition arbitrarily chosen is $\#\psi\gamma_1s\gamma_2\sigma\# \vdash \#\psi s'\gamma_1\gamma_2'\sigma\#$ if TM moves to the left (with s, $s' \in S$, γ_1, $\gamma_2, \gamma_2' \in \Gamma$, and $\psi, \sigma \in \Gamma^*$). This can be described as rewriting triads:

(1) $\gamma_1s\gamma_2 \to s'\gamma_1\gamma_2'$.

Nothing else changes in the configuration, and given the construction of TM, the transition is completely determined by the triad $\gamma_1s\gamma_2$. There is a similar pair of triads for the case that the machine moves to the right. The transition has the form $\#\psi s\gamma_1\gamma_2\sigma\# \vdash \#\psi\gamma_1's'\gamma_2\sigma\#$ and can be represented as a rewrite:

(2) $s\gamma_1\gamma_2 \to \gamma_1's'\gamma_2$.

If the machine remains in place, we write:

(3) $s\gamma_2 \rightarrow s'\gamma_2'$.

Because the number of states s and tape symbols γ for each Turing machine is finite, the number of pairs or triads is also finite. A subset of the set of these pairs gives a complete description of the possible operations of the Turing machine. Because Turing machines are deterministic, for every triad or pair to the left of the arrow there is only one possible triad or pair which can follow to the right of the arrow. Therefore, we can conclude that the operation of every Turing machine can be completely described by means of a finite set of deterministic rewrite rules.

Let TM accept x. We have seen that the final configuration has the form $\#\omega s_f \chi \#$. It is not difficult to construct a Turing machine TM' equivalent to TM, which has as final configuration $\# s_f S' \#$. For this purpose we build TM' in such a way that, just before reaching a final configuration, it will follow a procedure to replace all the remaining tape symbols with (pseudo) boundary symbols, except the last which is replaced by the as yet unused tape symbol S'. The initial and final configurations are therefore respectively $s_0 \# x \#$ and $\# s_f S' \#$.

We can now construct a grammar G for which $L(G) = T(TM) = T(TM')$. We collect all the rules of types (1), (2), and (3) in TM'. If $\beta \rightarrow \alpha$ is a rule of TM', we make $\alpha \rightarrow \beta$ a production of G. Given the deterministic character of rules $\beta \rightarrow \alpha$, if $\alpha \rightarrow \beta$ and $\alpha' \rightarrow \beta$, then $\alpha = \alpha'$. Next we add to the productions of G the productions $S \rightarrow s_f S'$ for every s_f in F, and the production $s_0 \# \rightarrow \#$. It is clear that by means of these productions, the derivations $S \Rightarrow s_f S' \overset{*}{\Rightarrow} x$ and only these can be made for every x in T and only if $x \in T$. G is a type-0 grammar, and consequently the theorem is proven.

It follows from Theorems 7.1. and 7.2. that Turing machines are equivalent to type-0 grammars or unrestricted rewrite systems.

7.4. MECHANICAL PROCEDURES, RECURSIVE ENUMERABILITY, AND RECURSIVENESS

Given a type-0 grammar G with a vocabulary V_T, there is a Turing machine TM which will stop in a final state after a finite number of transitions for every string x in V_T^* where $x \in L(G)$. We call this a mechanical procedure. In general we can define a mechanical (effective) procedure as an operation which can be performed by a Turing machine in a finite number of steps. Thus we replace the temporary definition of "procedure" given in paragraph 2.1. with the more precise definition "that which can be performed by means of a Turing machine". In paragraph 2.1. we imagined a procedure as a computer program by which an operation can be performed systematically. It does not at first seem evident that anything that can be performed systematically in a mechanical way (that is, without the use of human intuition), possibly by computer, can also be done on a Turing machine. The Turing machine appears to be far too simple a mechanism. But since the publication of Turing's original article (1936) it has become increasingly evident that the Turing machine can indeed perform anything which we might intuitively qualify as a procedure. For a good survey of the question, see Minsky (1967). It is therefore clearly justified formally to define the concept "procedure", as we have done, in terms of Turing machines. This opens the possibility of establishing with exactitude the problems for which no procedure exists, for such are the problems for which no Turing machine can be constructed. In the remainder of this chapter we shall speak freely of Turing machines whenever it is clear that a mechanical procedure must exist. Whenever we can explicitly indicate the consecutive steps of an operation, we conclude that the operation can be performed on a Turing machine.

The acceptance of a sentence by a Turing machine is by definition a mechanical procedure, but the same is true of the acceptance of sentences by more limited automata. It follows from the hierarchy of languages that for every language which is accepted by a finite automaton, a nondeterministic push-down automaton, or

a linear bounded automaton, there exists a Turing machine which also accepts it. We can therefore treat the acceptance of languages and sentences by automata in general in terms of procedures.

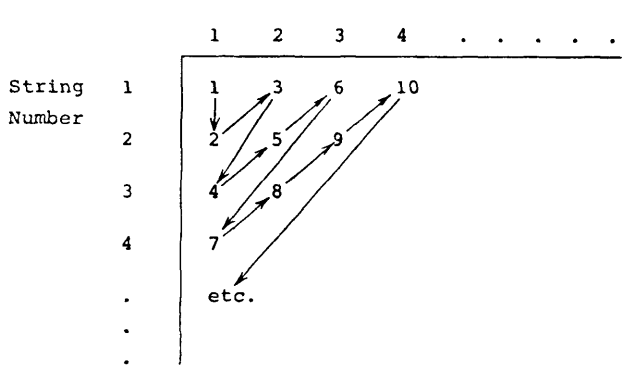
We would point out that the definition of "accepting" has been rather weak for all automata. We know that if $x \in L$, there is a procedure (TM) which will confirm that x is an element of L. But what happens if a string in V_T^* which is not an element of L is introduced as input? The Turing machine cannot reach a final state, but rather becomes blocked or goes on endlessly computing. We shall return to this point, but we shall first show that for every type-0 language L there is a mechanical procedure by which each sentence in L can be enumerated within a finite amount of time. L is then said to be RECURSIVELY ENUMERABLE.

THEOREM 7.3. Every type-0 language is recursively enumerable.

PROOF. It is easy to see that the strings in V_T^* can be enumerated by means of a mechanical procedure. If V_T contains k elements, the strings of V_T^* can be considered as numbers in a system with a base k, plus the null-string. If, for example, there are ten elements in V_T, we can give them the labels 0, 1, 2, ..., 9. strings of V_T^+ are thus numbers of the decimal system: 0, 1, 2, ..., 10, 11, ..., 100, 101, ..., and it is certainly possible to design a Turing machine which will write these sentences in sequence on its tape (the Turing machine must be able to perform the operation $n+1$). Each of these numbers appears on the tape after a finite number of operations, and no number is omitted. The same will hold for k. Furthermore, we know that there is a procedure which can determine whether a string is an element of L (Theorem 7.1.). This procedure can be applied to every newly enumerated string of V_T^*, in order to enumerate the sentences of L. There is a problem, however, for we do not know what will occur if the string in question is not an element of L. It is possible that the machine will go on endlessly computing and will never come to enumerate and test the following strings. This situation can be avoided by interrupting the test procedure at a given moment in the following way. We number

the strings in V_T^*: $\lambda = 1$, $a_1 = 2$, $a_2 = 3$, etc. (this is possible, as we have seen), and we indicate by number how many transitions the *TM* can undergo at a given stage of the test procedure for a given string. The process takes place as shown in Table 7.1. In

TABLE 7.1. Test Procedure for the Enumeration of the Sentences of *L*.



fact we have constructed a new Turing machine, *TM'*, which simulates the test procedure of *TM*. *TM'* first tests string 1 to see if it is an element of *L* by simulating one transition of the procedure of *TM*. If *TM'* finds that the string is an element of *L*, it enumerates the string and proceeds to test string 2. If it is not yet clear whether or not string 1 is an element of *L*, *TM'* still proceeds to test string 2. According to the table, *TM'* may simulate again only one transition of *TM*. String 2 is or is not enumerated according to the results of this test; according to the table, *TM'* then goes back to string 1 and simulates two steps from *TM* to test the string. According to the results of this test, the string is or is not enumerated, and *TM'* then goes on to test string 3 with one step from *TM*. It goes on in the same way to test string 2 with two transitions, string 1 with three transitions, string 4 with one transition, and so forth. In this way the automaton returns to each string and performs one step more than the preceding time to test it. Thus each string in

V_T^* is successively tested for membership in L by way of a finite number of transitions. For each x in L the procedure finally leads to the acceptance and enumeration of x.

We state without proof that the inverse of Theorem 7.3. is also valid: every recursively enumerable language can be generated by a type-0 grammar.

We have seen that the recursive enumerability of a type-0 language follows from the existence of an accepting procedure for the sentences of L, and have remarked that this is a weak theorem. We do not know what the Turing machine will do to a string in V_T^* which does not belong to the language. In order to discuss this question further, we define the COMPLEMENT OF A LANGUAGE L, with vocabulary V_T, as $V_T^* - L$. This is the set of strings over the terminal vocabulary which are not elements of the language. Linguists call this the set of UNGRAMMATICAL SENTENCES. The complement of a language is denoted by CL.

A stronger form of acceptance would be a procedure according to which for every string in V_T^* it would be indicated if the string belongs to L or to CL. One might imagine a "twin Turing machine" which would reach a final state for a string in CL, while the original Turing machine would do the same for a string in L. One might also imagine a Turing machine with two sets of final states, one for accepting, the other for rejecting. For every string x in V_T^*, the Turing machine would reach a final state: the accepting final state when $x \in L$, and the rejecting final state when $x \in CL$. If such a procedure exists for language L, the automaton is said to RECOGNIZE (as opposed to accept) L. A recognition procedure of this sort is usually called an ALGORITHM. An algorithm is thus a procedure according to which for every x in V_T^*, it can be determined whether or not x belongs to L. Because algorithms lead to decisions for every string in V_T^*, the language $L \subset V_T^*$ is called a DECIDABLE (RECURSIVE) SET if an algorithm exists for the recognition of L. It follows from the construction of the twin Turing machines that a language is recursive if both the language and its complement are recursively enumerable.

We know that type-0 languages, and consequently also type-1,

type-2, and type-3 languages are recursively enumerable, but are the complements of these languages also recursively enumerable? That is not the case in general. We state without proof that there are type-0 languages which are not recursive, because they have complements which are not recursively enumerable. This means that the complements are not type-0 languages. However, the complement of a context-sensitive language is recursively enumerable, and consequently context-sensitive, context-free and regular languages are all recursive. There are (recognition) algorithms for all of these languages.

We have seen that the complement of a type-0 language is not necessarily itself of type-0, but what of the other language types? It is not yet known if the complement of a context-sensitive language is context-sensitive; all we know is that it is recursively enumerable, and consequently of type-0. It has been proven that no general procedure exists for determining whether the complement of any context-free language is also context-free. In any case it does not hold in general that the complement of a context-free language is also context-free; the complement of a deterministic context-free language is, however, also deterministic and context-free. It is also known that the complement of a regular language is likewise regular.

8

GRAMMATICAL INFERENCE

8.1. HYPOTHESES, OBSERVATIONS, AND EVALUATION

Is it possible on the basis of samples of a language to decide on an acceptable grammar for that language? In its present form, this question cannot be answered, but the day to day work of the linguist, as well as the fast growing language capacity of the young child, suggest that an affirmative answer might be expected to at least some forms of the question. The answer depends on (1) what is known about the grammar, (2) the composition of the sample of data, and (3) what is understood by "acceptable". The investigation of these matters is known as the study of GRAMMATICAL INFERENCE.

That which is already known or supposed of a grammar is referred to by the term HYPOTHESIS-SPACE. The terminal vocabulary V_T, for instance, is ordinarily given. Certain suppositions can also be made as to the class to which the grammar belongs (regular, context-free, etc.). In the case of a probabilistic grammar, not only can suppositions be made about the type of grammar, but inference can also have the more limited goal of finding the most acceptable production probabilities for a grammar which is given. This latter has rather direct possibilities of application, and we will deal with it in some detail in paragraph 8.2. Paragraph 8.3. will treat a number of general findings relative to nonprobabilistic hypothesis-space, and paragraph 8.4. will discuss the most general kind of hypothesis-space, probabilistic grammars for which both productions and production probabilities must be found.

The term OBSERVATION-SPACE refers to the composition of the data sample; it can take on various forms. If L is the language investigated and x is a given string in V_T^*, we can obtain positive information, $x \in L$, or negative information, $x \notin L$ (i.e. $x \in CL$), about L. In the former case we speak of a POSITIVE INSTANCE, in the latter, of a NEGATIVE INSTANCE. The information available is called an INFORMATION SEQUENCE. If all the instances in the sequence are positive, we have a POSITIVE INFORMATION SEQUENCE; if negative instances also occur, we have a MIXED INFORMATION SEQUENCE. A COMPLETE INFORMATION SEQUENCE is a mixed information sequence in which all positive and negative instances are enumerated; such sequences are generally infinite in length. A COMPLETE POSITIVE INFORMATION SEQUENCE is the enumeration of all positive instances; it is called TEXT PRESENTATION, since the language is presented, sentence for sentence, as a text. Repetitions may occur, provided that the enumeration is complete, i.e. every sentence of the language must occur after a finite number of other sentences. INFORMANT PRESENTATION is the term for a complete mixed information sequence, or a sequence in which every positive and negative instance over V_T^* occurs after a finite number of other instances. One might picture this as a researcher who wishes to find the grammar of a language and reads each string of V_T^* to an informant who in turn tells him for every string whether it belongs to the language or not. A STOCHASTIC TEXT PRESENTATION is an infinite sequence $I = x_1, x_2, \ldots$, where x_i is an element of L, and L is a probabilistic language in which for every x_i, $p(\underset{\sim}{x}_i = x_i) = p(\underset{\sim}{x} = x)$;[1] this means that the chance that string x will be in position i is constant and equal to the probability of the string in the language. The sentences thus appear successively with their respective probabilities in L. Notice that the definition of a stochastic text presentation does not include the property of completeness. At the limit, however, the relative frequency of a sentence in a stochastic text presentation is equal to its probability in L. The chance of occurrence of a sentence x in L can be increased by

[1] $p(\underset{\sim}{x} = x)$ is the probability of x in L. We suppose the variables x_i to be independent, i.e. $p(\underset{\sim}{x}_i = x_i \mid \underset{\sim}{x}_j = x_j) = p(\underset{\sim}{x}_i = x_i)$.

increasing the length of the information sequence. A sample of a stochastic text presentation of size k consists of the first k elements of that text presentation. On the basis of the assumption of independence,[2] the probability of this particular sample is the product of the probabilities of its k elements.

What is an "acceptable" grammar? Suppose that the information consists of an information sequence up to a given point k: x_1, x_2, ..., x_k. Any grammar which corresponds to the elements x_1, ..., x_k is, in a weak sense, acceptable. By "corresponds" we mean that the positive instances in the sequence are generated by the grammar, and the negative instances are not. But the criterion of correspondence will in general allow an infinity of possible grammars. If we concentrate our attention on the positive instances in the text presentation, we find that the one extreme is a grammar which generates only the k elements of the information, whereas the other extreme is a universal (regular) grammar over V_T which generates all the strings of V_T^*. Both these grammars correspond to the information, but the former is "unnecessarily" complex, and the latter would correspond to any sample, and therefore does not "fit". Both complexity and fit must decidedly be included in the standard of evaluation of the acceptability of a grammar. To a large extent, complexity is a matter of taste and of the preferences of the researcher. That the standard is relative is probably the only point on which one could expect all to agree. Grammars may be compared on the basis of various criteria, such as the number of symbols, the number of productions, the number of alternatives for each production, etc. These criteria make up the context of evaluation; on it depends the complexity of a grammar. The use of the mechanism of probabilistic grammars can permit a definition of context (without excluding other definitions, as complexity remains a matter of taste) in terms of the a priori probability of alternative grammars in the hypothesis-space. This will be done in paragraph 8.4; it will at the same time permit an evaluation, by way of the Bayes theorem, of the fit of various probabilistic grammars.

[2] See note 1.

In the following paragraph, however, we shall deal only with the classical statistical evaluation procedure. This method is more efficient in that context, and yields results for large samples which scarcely deviate from those of a Bayes analysis.

8.2. THE CLASSICAL ESTIMATION OF PARAMETERS FOR PROBABILISTIC GRAMMARS

We will be dealing here with the simple case in which, except for the production probabilities, the entire grammar is given. The discussion will be limited to nonambiguous context-free grammars.

On the basis of a sample of language L, we must determine which probabilistic grammar will be the best for L, that is, we must find an optimal estimate for the production probabilities of the grammar.

Let G be a nonambiguous context-free grammar with N productions. The respective production probabilities are labelled p_1, p_2, ..., p_N. To normalize the grammar, we must see to it that for every variable A in V_N, $\sum_i p\ (A \to \alpha_1) = 1$. If there are $l (l > 0)$ productions in which A occurs to the left of the arrow, then for the productions $A \to \alpha_i$ (where $i = 1, 2, \ldots, l$), $l-1$ production probabilities must be found. (If G has only one production, $A \to x$, then $p(A \to x) = 1$.) If V_N has M variables, and the number of independent production probabilities in the grammar is denoted by k, then $k = N - M$. On the basis of the sample, estimates must be found for these k parameters, $q_1, q_2, \ldots, q_k$. When that is done, the production probabilities $p_1, p_2, \ldots, p_N$ will follow directly from the normalization.

Given a sample from language L, we proceed as follows. Let the sample contain n different sentences (or sentence types, since a particular sentence can occur more than once in the sample). The leftmost derivation $S \overset{*}{\Rightarrow} s_i$ must be determined for every sentence s_i (where $i = 1, \ldots, n$). If the productions used in the derivation are independent, then $p(S \overset{*}{\Rightarrow} s_i) = p(s_i)$ can be expressed as the product of the production probabilities p_i of the various

steps in the derivation. For the derivation $S \overset{p_j}{\Rightarrow} \alpha \overset{p_k}{\Rightarrow} \beta \overset{p_l}{\Rightarrow} \gamma \overset{p_j}{\Rightarrow} s_i$, for example, this is $p(s_i) = p_j^2 p_k p_l$. This product for each of the n sentence types is denoted by π_i, and each of its terms can be expressed in parameters $q_1, \ldots, q_k$.

We define the likelihood function $\mathscr{L}$ for the sentences $s_1, \ldots, s_n$ and the parameters $q_1, \ldots, q_k$ as follows:

$$\mathscr{L}(s_1, \ldots, s_n; q_1, \ldots, q_k) = \pi_1^{f_1} \pi_2^{f_2} \ldots \pi_n^{f_n},$$

where f_i is the number of times sentence type i occurs in the sample. Using logarithms, this is:

$$\log \mathscr{L} = f_1 \log \pi_1 + f_2 \log \pi_2 + \ldots + f_n \log \pi_n = \sum_i f_i \log \pi_i.$$

The best estimate of the parameters $q_1, \ldots, q_k$ is that which gives a maximum for $\mathscr{L}$, and thus also for $\log \mathscr{L}$. With these parameters, the chance of drawing precisely this sample is at a maximum. The various parameter estimates $\hat{q}_1, \hat{q}_2, \ldots, \hat{q}_k$, are found by expressing every π_i in parameters, and then determining the k partial derivatives of $\mathscr{L}$ according to $q_1, \ldots, q_k$. This yields a system of k equations $\dfrac{\delta \log \mathscr{L}}{\delta q_i} = 0$, the solutions of which are the desired estimates $\hat{q}_1, \ldots, \hat{q}_k$. At this point the probabilities $p_1, \ldots, p_N$ can be calculated.

EXAMPLE 8.1. Let L be a language over the vocabulary $\{a, b, c\}$. Suppose we have a sample of L consisting of 100 sentences with the following distribution of sentence types: c (22 times), *aca* (42 times), *abcba* (19 times), *abbcbba* (12 times), *abbbcbbba* (4 times), and *abbbbcbbbba* (once). A possible grammar for these sentence types has the following productions:

$$S \overset{q_1}{\rightarrow} aAa \qquad A \overset{q_2}{\rightarrow} bAb$$

$$S \overset{1-q_1}{\longrightarrow} c \qquad A \overset{1-q_2}{\longrightarrow} c$$

Above the arrows we find the production probabilities expressed in parameters, and in such a way that the grammar is normalized. The leftmost derivations of the sentences in the sample are given

below with the probability of the production concerned at each step.

$S \overset{1-q_1}{\Longrightarrow} c$	$p(c) = 1-q_1$
$S \overset{q_1}{\Rightarrow} aAa \overset{1-q_2}{\Longrightarrow} aca$	$p(aca) = q_1(1-q_2)$
$S \overset{q_1}{\Rightarrow} aAa \overset{q_2}{\Rightarrow} abAba \overset{1-q_2}{\Longrightarrow} abcba$	$p(abcba) = q_1q_2(1-q_2)$
etc.	$p(abbcbba) = q_1q_2^2(1-q_2)$
	$p(abbbcbbba) = q_1q_2^3(1-q_2)$
	$p(abbbbcbbbba) = q_1q_2^4(1-q_2)$

The likelihood function then becomes:

$\mathscr{L} = [(1-q_1)]^{22}\,[q_1(1-q_2)]^{42}\,[q_1q_2(1-q_2)]^{19}\,[q_1q_2^2(1-q_2)]^{12} \times [q_1q_2^3(1-q_2)]^4\,[q_1q_2^4(1-q_2)] = q_1^{78}q_2^{59}(1-q_1)^{22}(1-q_2)^{78}$, and the natural logarithm of $\mathscr{L}$ is:

$\ln\mathscr{L} = 78 \ln q_1 + 59 \ln q_2 + 22 \ln (1-q_1) + 78 \ln (1-q_2)$. The most likely values of q_1 and q_2 are found by taking partial derivatives of $\ln\mathscr{L}$ with respect to q_1 and q_2, putting them equal to zero, and solving the equations:

$$\frac{\delta \ln\mathscr{L}}{\delta q_1} = \frac{78}{q_1} - \frac{22}{1-q_1} = 0 \qquad \frac{\delta \ln\mathscr{L}}{\delta q_2} = \frac{59}{q_2} - \frac{78}{1-q_2} = 0$$

thus $\hat{q}_1 = 0.78$ thus $\hat{q}_2 = 0.43$

With these estimates of the parameters, we can calculate the probabilities of the sentence types in the sample. For c we have $1\text{-}q_1 = 0.22$, for aca, $q_1(1\text{-}q_2) = 0.78 \times 0.57 = 0.445$, and so forth. In a sample of 100 sentences we would expect the sentence c 22 times, and the sentence aca, 44.5 times, etc. All the values are given in Table 8.1., together with the observed values. The correspondence between observed and expected values can be measured and evaluated with standard statistical tests such as, for example, the chi-square test for goodness of fit.

TABLE 8.1. Observed and Expected Frequencies of Sentence Types (Example 8.1.).

Sentence Type	Observed	Expected	Sentence Type	Observed	Expected
c	22	22	*abbbcbbba*	4	3.5
aca	42	44.5	*abbbbcbbbba*	1	1.5
abcba	19	19.1	other	0	1.2
abbcbba	12	8.2			

8.3. THE "LEARNABILITY" OF NONPROBABILISTIC LANGUAGES

A number of theorems concerning the "learnability" of nonprobabilistic languages were presented by Gold in a fundamental article (1967). In this paragraph we shall state some of his more important findings without proving them.

Suppose we have a complete (text or informant) information sequence for a language of a given class (finite, regular, etc.). An algorithm must be found with the following characteristics:[1]
(1) each time a new input element x_i is introduced, the algorithm produces a grammar (or a code for a grammar) of the given class which is consistent with the information received up to that point.
(2) after a finite number of elements has been received, the output remains constant: the grammar produced as output is always the same or equivalent, and is a grammar of L.

A language is said to be IDENTIFIABLE IN THE LIMIT OR LEARNABLE if such an algorithm exists for it for every complete information sequence. A class of languages is learnable if every language in it is learnable. The most important conclusions drawn by Gold from his investigation concerning the various classes of languages are given in Table 8.2.; in it, the symbol + denotes "learnable", and the symbol —, "not learnable".

[1] "Algorithm" is used in the same sense here as in the preceding chapter: a Turing machine which stops (produces an output) after every input. Gold also analyzes learnability as a procedure, but we will not discuss his findings here; they are not much different from the results for algorithms.

TABLE 8.2. "Learnability" of Languages of Various Classes according to Text or Informant Presentation

Language Class	Text	Informant
Type-0	—	—
Type-0 (recursive)	—	—
Type-0 (primitive recursive)	—	+
Context-Sensitive	—	+
Context-Free	—	+
Regular	—	+
Finite	+	+

The table calls for some explanation on (a) the broad difference between "learnabilty" on the basis of text presentation and "learnability" on the basis of informant presentation, and (b) the fine differentiation within the class of type-0 languages.

(a) Text presentation involves learnability for finite languages only. The fact that a finite language can be learned through text presentation can easily be understood as follows. Every sentence of the language appears after a finite number of earlier instances (since the presentation is complete). The algorithm can simply be to enumerate all different sentences which have appeared in the presentation up till and including the last instance. This list of sentences can as well be written as a grammar with rules $S \rightarrow x_i$ with one rule for every sentence x_i. After a finite amount of time, all the sentences of the language will have passed in review (as the number of sentences is finite), and from that point the grammar will remain unchanged. The grammar thus produced will certainly be a grammar of the language.

The process, however, will only succeed with finite languages; not even regular languages are learnable, according to Gold's definition of the term, on the basis of text presentation. One might imagine the following algorithm for the learning of regular languages on the basis of text presentation: the first and all following outputs of the algorithm would be a universal grammar U, with productions $S \rightarrow a$ and $S \rightarrow aS$ for every a in V_T. As such a grammar can generate any string in V_T^+, all subsequent outputs would

be the same grammar, which will be consistent with all further information. But this algorithm would not satisfy condition (2) of the definition, because the grammar produced is not a grammar of the language (unless the language is the universal language V_T^+). The grammar would then be "too broad" for the language. The algorithm should be set up in such a way that the grammar is as narrow as possible at first, and is broadened according to the incoming information. As the class of finite languages is contained by the class of regular languages (Theorem 2.3.), it is not impossible that the language here in question be finite. The algorithm must begin here with the narrowest conjecture, namely that the language is finite. If it is more broadly supposed that the language is infinite, while in fact the language is finite, the algorithm would never receive information incompatible with that supposition. We might, of course, imagine an algorithm which decides that a language is finite if it finds k repetitions of the same set of sentences, but this still would not solve the problem. Although such an algorithm would yield a correct grammar for a finite language, it could mistake an infinite for a finite language. Suppose, for example, that from infinite language L a text presentation is prepared as follows: take from L subsets $F_1, F_2, \ldots$ of increasing size. Begin presenting the sentences in F_1 with k or more repetitions. The algorithm will then incorrectly decide that the language is finite. When F_2 is introduced, the algorithm must review its judgment, but if there are also k or more repetitions of the sentences in F_2, it will return to its original decision that the language is finite. But the same process will occur when F_3 is introduced, and so forth. The presentation is complete, for every sentence of the language will be presented after a finite amount of time, but the algorithm would always produce nothing other than grammars for finite languages. Thus an algorithm which functions flawlessly for finite languages cannot learn an infinite language, and an algorithm adapted to infinite languages will, when presented with a finite language, produce grammars which are too broad. Therefore it is impossible to "learn" an infinite language only on the basis of text presentation.

(b) In the preceding chapter it was stated that type-0 languages

are generally not recursive. However there are type-0 languages which are recursive, but not context-sensitive; the set of recursive type-0 languages does not coincide completely with that of context-sensitive languages. The table shows that only "primitive recursive" type-0 languages, a subset of recursive type-0 languages, are learnable according to Gold's definition of the word. Primitive recursive languages cannot be defined without recourse to the theory of recursive functions.[1] Suffice it to note that "most" recursive languages are primitive recursive (also, in the history of mathematics, it has been difficult to find exceptions to this), and that the distinction between recursive and primitive recursive languages is of little importance to the study of natural languages. All recursive grammars (i.e. grammars of decidable languages) which will be mentioned below are in fact primitive recursive.

8.4. INFERENCE BY MEANS OF BAYES' THEOREM

In paragraph 8.2. we found by "classical" means optimal statistical parameters for a given nonambiguous context-free grammar. We renounced the possibility of choosing from among several grammars. In paragraph 8.3. the procedure was inverse, in a sense. We examined the conditions of presentation under which a grammar may be selected from the class of a priori possible grammars, renouncing the probabilistic formulation. The notion of "learnability" had to be defined in terms of equivalent grammars, as the algorithms cannot select an optimal or "most efficient" (cf. 3.1.) grammar from the class of equivalent adequate grammars.

Horning (1969) combined the two approaches, and developed a method of selecting an optimal probabilistic grammar from a

[1] A language is PRIMITIVE RECURSIVE if its characteristic function is primitive recursive. The characteristic function C_L of a language L, where $L \subset V_T^*$, has the value 1 for every string in V_T^* which is an element of L, and the value 0 for every string in V_T^* which is not an element of L.

Definitions of recursive functions may be found in Kleene (1952), Minsky (1967), Nelson (1968), *et alibi.*

given class on the basis of a given information sequence. We shall state some of his most important findings here concerning non-ambiguous context-free grammars.

We have seen that a standard of evaluation must express two aspects: the complexity of the grammar, and the degree to which it fits the information which is available at a given moment (paragraph 8.1.). The complexity of a grammar depends on the context, which includes at least (1) the size of the nonterminal vocabulary, (2) the number of alternative rewrites for a given variable, and (3) the length of those alternatives. (In practical and linguistic situations the context can include far more than this. The three aspects mentioned here, however, are constant themes in the linguistic literature on the subject.) The relative importance to be attributed to each of these aspects of context is a matter of taste, but there is a method by which this can at least be done in an exact manner. The method is by means of a so-called GRAMMAR-GRAMMAR. We will now introduce this notion.

A grammar is a finite string of symbols; a set of grammars (an hypothesis-space) may be regarded as a set of such strings, and thus as a kind of "language". A grammar-grammar is a grammar which generates such a "language". If the grammar-grammar is probabilistic, it will define a probability distribution over the "sentences" of the "language", and thus over the class of grammars which it generates. The complexity of a grammar can then be defined as minus the base two logarithm of its probability, as in information theory. The probabilistic grammar-grammar is thus a precise definition of the context; moreover, the more variables, the more alternatives for each variable, or the longer the alternatives in a generated grammar, the smaller its probability and the greater its complexity. The relative importance of each of the aspects can be varied by varying the production probabilities of the grammar-grammar.

We illustrate this with an example. To avoid confusion, name, variables, and arrow of the grammar-grammar are given in bold face type, while those of grammars are in ordinary type.

EXAMPLE 8.2. Let G be a probabilistic grammar-grammar with the following productions:

1. $\mathbf{S} \xrightarrow{0.5} \mathbf{R}$
2. $\mathbf{S} \xrightarrow{0.5} \mathbf{RR}$
3. $\mathbf{R} \xrightarrow{1} \mathbf{N} \rightarrow \mathbf{P}$
4. $\mathbf{P} \xrightarrow{0.5} \mathbf{A}$
5. $\mathbf{P} \xrightarrow{0.5} \mathbf{P}, \mathbf{A}$
6. $\mathbf{A} \xrightarrow{0.5} \mathbf{T}$
7. $\mathbf{A} \xrightarrow{0.5} \mathbf{TN}$
8. $\mathbf{T} \xrightarrow{0.5} a$
9. $\mathbf{T} \xrightarrow{0.5} b$
10. $\mathbf{N} \xrightarrow{0.5} S$
11. $\mathbf{N} \xrightarrow{0.5} A$

This grammar-grammar generates regular grammars with one or two variables (S, A) and one or two terminal symbols (a, b). We shall show the leftmost derivation of a regular grammar G with the following productions:

$$S \rightarrow b, bS, aA \qquad A \rightarrow a, bA, aS$$

These are in fact six productions: the commas indicate alternative rewrites for a single variable. If we know that G is a context-free grammar, and thus that the first member of every production is a single variable, the grammar can be written without ambiguity as follows:

$$S \rightarrow b, bS, aAA \rightarrow a, bA, aS$$

(In the triad aAA, the reader should imagine a caesura between A and A.) This is precisely the "sentence" which we wish to derive from $\mathbf{G}$; its leftmost derivation is as follows:

$$\begin{aligned}
\mathbf{S} &\overset{0.5}{\Rightarrow} \mathbf{RR} \\
&\overset{1}{\Rightarrow} \mathbf{N} \rightarrow \mathbf{PR} \\
&\overset{0.5}{\Rightarrow} S \rightarrow \mathbf{PR} \\
&\overset{0.5}{\Rightarrow} S \rightarrow \mathbf{P}, \mathbf{AR} \\
&\overset{0.5}{\Rightarrow} S \rightarrow \mathbf{P}, \mathbf{A}, \mathbf{AR} \\
&\overset{0.5}{\Rightarrow} S \rightarrow \mathbf{A}, \mathbf{A}, \mathbf{AR} \\
&\overset{0.5}{\Rightarrow} S \rightarrow \mathbf{T}, \mathbf{A}, \mathbf{AR} \\
&\overset{0.5}{\Rightarrow} S \rightarrow b, \mathbf{A}, \mathbf{AR} \\
&\overset{0.5}{\Rightarrow} S \rightarrow b, \mathbf{TN}, \mathbf{AR} \\
&\overset{0.5}{\Rightarrow} S \rightarrow b, b\mathbf{N}, \mathbf{AR}
\end{aligned}$$

$\overset{0.5}{\Rightarrow} S \to b,\ bS,\ \mathbf{AR}$

$\overset{0.5}{\Rightarrow} S \to b,\ bS,\ \mathbf{TNR}$

$\overset{0.5}{\Rightarrow} S \to b,\ bS,\ a\mathbf{NR}$

$\overset{0.5}{\Rightarrow} S \to b,\ bS,\ aA\mathbf{R}$

$\overset{1}{\Rightarrow} S \to b,\ bS,\ aA\mathbf{N} \to \mathbf{P}$

$\overset{0.5}{\Rightarrow} S \to b,\ bS,\ aAA \to \mathbf{P}$

$\overset{0.5}{\Rightarrow} S \to b,\ bS,\ aAA \to \mathbf{P},\ \mathbf{A}$

$\overset{0.5}{\Rightarrow} S \to b,\ bS,\ aAA \to \mathbf{P},\ \mathbf{A},\ \mathbf{A}$

$\overset{0.5}{\Rightarrow} S \to b,\ bS,\ aAA \to \mathbf{A},\ \mathbf{A},\ \mathbf{A}$

$\overset{0.5}{\Rightarrow} S \to b,\ bS,\ aAA \to \mathbf{T},\ \mathbf{A},\ \mathbf{A}$

$\overset{0.5}{\Rightarrow} S \to b,\ bS,\ aAA \to a,\ \mathbf{A},\ \mathbf{A}$

$\overset{0.5}{\Rightarrow} S \to b,\ bS,\ aAA \to a,\ \mathbf{TN},\ \mathbf{A}$

$\overset{0.5}{\Rightarrow} S \to b,\ bS,\ aAA \to a,\ b\mathbf{N},\ \mathbf{A}$

$\overset{0.5}{\Rightarrow} S \to b,\ bS,\ aAA \to a,\ bA,\ \mathbf{A}$

$\overset{0.5}{\Rightarrow} S \to b,\ bS,\ aAA \to a,\ bA,\ \mathbf{TN}$

$\overset{0.5}{\Rightarrow} S \to b,\ bS,\ aAA \to a,\ bA,\ a\mathbf{N}$

$\overset{0.5}{\Rightarrow} S \to b,\ bS,\ aAA \to a,\ bA,\ aS$

The product of the probabilities of the rewrites is $p(G) = 0.5^{25}$, and the complexity of G in context $\boldsymbol{G}$ is thus $-{}^2\log 0.5^{25} = 25$. The reader can verify for himself that grammar U with productions $S \to a,\ b,\ aS,\ bS$ (this is the universal grammar which generates all strings in V_T^*) has a complexity of 15 in context $\boldsymbol{G}$.

If we consider it particularly important that a grammar should have few variables, we make production 2 less probable; the probability of a grammar with two variables decreases, and the complexity increases. If, on the other hand, we wish the number of alternative rewrites important, we can reduce the probability of production 5, which determines the number of alternatives for rewriting of a variable. Finally, if we wish to increase the importance of rewrite length, we reduce the probability of production 7. Many other variations are possible.[1]

We suppose that a complexity distribution is defined over the grammars in the hypothesis-space by means either of a grammar-

[1] One should, however, remain cautious. A grammar-grammar which generates all grammars of a certain type (e.g. regular grammars) will have a terminal vocabulary of infinite size, since the nonterminal vocabulary of every grammar generated is a subset of the terminal vocabulary of the grammar-grammar. Solutions to this problem have been found by Feldman, et al. (1969) and Horning (1969).

grammar or of some other context. We express the "credibility" of a grammar G_i in the hypothesis-space as a number $p(G_i)$, such that it is an inverse function of complexity (whichever way this is defined), with $0 < p(G_i) \leqslant 1$, and $\sum_i p(G_i) = 1$ for the grammars in the hypothesis-space. These propositions hold automatically in the context of a consistent probabilistic grammar-grammar. The p-values will be treated in all other regards as probabilities. We also suppose that the grammars in the hypothesis-space can be enumerated according to the order of their a priori credibility or "probability" p. (From this point we shall use the word "probability" exclusively.)

The observation-space is assumed to be a stochastic text presentation (cf. paragraph 8.1.).

As the OPTIMAL GRAMMAR we consider the a priori most probable grammar which is stochastically equivalent to the grammar by which the text was derived.

A procedure must be devised (in the sense of a Turing machine) which at receiving each new instance can maximize the chance of conjecturing the optimal grammar, i.e. it must conjecture the grammar with the highest a posteriori probability, given the text and the a priori probabilities of the grammars. In order to investigate the existence of such a procedure we must, therefore, first explicate the relations between a priori and a posteriori probabilities of grammars.

The a priori probability of a grammar G_i in the hypothesis-space is denoted by $p(G_i)$. The probability of an information sequence (a sample) S_j, up to a given moment of the text presentation and given the hypothesis-space, is $p(S_j)$. The conditional probability that S_j will occur when G_i is really the grammar of the language is $p(S_j|G_i)$, and this is equal to the product of the probabilities of the sentences in the sample, given grammar G_i (cf. paragraph 8.1). Therefore, if the sample contains the sentences s_1, s_2, ..., s_k, then $p(S_j|G_i) = p(s_1|G_i) \cdot (p(s_2|G_i) \cdot \ldots \cdot p(s_k|G_i)$, or simply:

$$(1)\quad p(S_j|G_i) = \prod_{j=1}^{k} p(s_j|G_i).$$

On the other hand we indicate the chance that G_i is really the grammar of L, given the sample S_j, as $p(G_i|S_j)$, which, according to an elementary rule of probability theory, is equal to $\frac{p(G_i, S_j)}{p(S_j)}$, where $p(G_i, S_j)$ is the chance that G_i is correct and that the sample S_j occurs. Therefore:

$$(2)\quad p(G_i, S_j) = p(S_j) \cdot p(G_i|S_j).$$

This means that the joint probability of G_i and S_j is the a priori probability of S_j, multiplied by the conditional probability that G_i is the real grammar when S_j occurs. For the sake of symmetry, this can also be written as follows:

$$(3)\quad p(G_i, S_j) = p(G_i) \cdot p(S_j|G_i).$$

On the basis of (1) and (2) we can find the a posteriori probability of G_i:

$$(4)\quad p(G_i|S_j) = \frac{p(G_i) \cdot p(S_j|G_i)}{p(S_j)}$$

(This is a form of the Bayes theorem.)

If we determine the a posteriori probabilities of all grammars in the hypothesis space, given the sample and the a priori probabilities, the denominator in (4), $p(S_j)$, remains constant, and only the two terms of the numerator vary. To find the optimal grammar, we must therefore find the grammar which yields the greatest numerator $p(G_i) \cdot p(S_j|G_i)$. We can write this product as $p'(G_i|S_j)$. If the sample contains k sentences, by substitution of (1) we get:

$$(5)\quad p'(G_i|S_j) = p(G_i) \cdot \prod_{j=1}^{k} p(s_j|G_i).$$

Horning has proven that a procedure does exist by which at every new instance that G in the hypothesis-space can be found for which (5), and thus its posteriori probability, is at a maximum. We shall neither describe the procedure here nor prove the theorem, but only wonder if indeed the optimal grammar can, in the long run, be found in this way. In Gold's terms, the procedure does not

lead, after a finite number of instances, to the reproduction at every new instance of the same grammar or stochastic equivalents which are grammars of the language. It only leads to the somewhat weaker result, that every nonoptimal grammar in the hypothesis-space is rejected after a finite number of instances. In other words, the chance that a nonoptimal grammar be conjectured decreases as the number of instances increases. This can also be regarded as a definition of "learnability", although it is weaker than that given by Gold. Taken in this sense, however, Horning has shown that probabilistic nonambiguous context-free grammars are "learnable" by means of a stochastic text presentation.

Until now we have assumed that the hypothesis-space consists of probabilistic grammars. However, if the hypothesis-space is generated by a probabilistic grammar-grammar this is not the case. Example 8.2. showed that the output of such a grammar-grammar is a grammar and its corresponding probability. Additionally, a way must be found to obtain optimal parameter estimates for production probabilities in the grammars in the hypothesis-space. Horning presents a (Bayes) procedure for this as well, and shows that the conclusions on learnability which we have just mentioned still hold in essence for this complete case.

HISTORICAL AND BIBLIOGRAPHICAL REMARKS

The theory of formal languages, except for the probabilistic part, is largely based on Chomsky's work. The original publication in which the hierarchy of grammars was introduced is Chomsky (1959 a, b.) A later survey is Chomsky (1963) in which the hierarchy of grammars was somewhat refined. Grammars with productions exclusively in the context-sensitive form were given a separate type number, and consequently the numeration differs there from that of the earlier work. We have followed current usage and maintained the original numeration.

The term "regular language" has a history of its own. Originally (Chomsky and Miller 1958; Bar-Hillel, Gaifman, and Shamir 1960) these languages were called "finite state languages" because of the connection with finite or finite state automata. But in mathematics, the theory of recursive functions dealt independently with, among other things, "regular sets", which can be recursively generated by "regular expressions", and Kleene showed the equivalence of these sets and the sets accepted by finite automata. As type-3 grammars are equivalent to finite automata (as in Theorems 4.2. and 4.3. proven by Chomsky and Miller 1958), type-3 languages are regular sets. Consequently type-3 grammars and languages are now generally called "regular grammars" and "regular languages".

Context-free grammars are treated in great detail in Chomsky's original work. The expression "normal-form" originated in Chomsky's notion of a "normal grammar" (Chomsky 1963). He said that normal grammars are the kind of grammars usually dealt with in

linguistic discussions on constituent structure analysis: productions $A \to a$ concern the LEXICON of the language, and productions $A \to BC$ lead to binary divisions into CONSTITUENTS. At present, however, the term "normal-form" is used only to denote standardized forms for the productions of grammars. The Greibach normal-form is presented in Greibach (1965). The self-embedding theorem (Theorem 2.8.) for context-free languages was first formulated by Chomsky (1959a); a complete proof can be found in Salomaa (1969). The notion of ambiguity was first handled by Parikh (1961). For later developments see Ginsburg and Ullman (1966). For linear grammars see Greibach (1963) and (1966) and others. A textbook on context-free grammars is Ginsburg (1966).

The equivalence of type-1 grammars and grammars with productions only in the context-sensitive form was treated by Chomsky (1963). Grammars of the form which we have called the Kuroda normal-form were called "linear bounded grammars" by Kuroda and several other authors, by analogy with the automaton. The normal-form theorem (Theorem 2.11.) was first proven by Kuroda (1964).

The earliest publications on the subject of probabilistic grammars are Grenander (1967), Ellis (1969), and Booth (1969). It was an obvious matter to relate them to the Chomsky hierarchy. The consistency theorem for regular grammars (Theorem 3.1.) was proven by Ellis (1969) as was Theorem 3.2. The hypothesis formulated in Theorem 3.3. may be found in Suppes (1970). The Chomsky and Greibach normal-form theorems were originally proven by Ellis (1969); in the proof given here, we have followed Huang and Fu (1971). The conditions of consistency for probabilistic context-free grammars were investigated by Booth (1969) and Ellis (1969) where the reader may find more details on the subject.

The investigation of finite automata originated in the work of McCulloch and Pitts (1943), in which they gave models for neural networks which could be regarded as FINITE STATE MACHINES. Of the many early publications on this subject, we mention Rabin and Scott (1959), in which the proof of Theorem 4.1. can be found, and Kleene (1956). Later surveys are those by S. Ginsburg

(1962) and by A. Ginzburg (1968). The equivalence of finite automata and regular grammars (Theorems 4.2. and 4.3.) was proven by Chomsky and Miller (1958). Probabilistic finite automata were introduced by Rabin (1963). Much work in this area was done by Salomaa, who gives a good survey in Salomaa (1969).

The notion of the "push-down store" was introduced by Newell, Shaw, and Simon (1959). The first formulation of the relationship between push-down automata and formal languages is that of Oettinger (1961). The relationship between context-free grammars and push-down automata (Theorems 5.1. and 5.2.) was formulated by Chomsky (1963) and Evey (1963) more or less independently. The equivalence of deterministic push-down automata and $LR(k)$-grammars was proven by Knuth (1965).

Deterministic linear bounded automata were introduced by Myhill (1960); Landweber (1963) gave proof of Theorem 6.2. on deterministic linear bounded automata. Kuroda (1964) introduced the nondeterministic linear bounded automaton and proved the equivalence of them and context-free grammars (Theorems 6.1. and 6.2.).

The Turing machine was presented by Turing (1936) as a machine which could perform any computation for which an explicit procedure is known. For an introduction to the subject of mechanical (effective) procedures, see Minsky (1967); in the same work models by Post and Church, similar to the Turing machine, are also discussed. The relationship between Turing machines and type-0 languages formulated in Theorems 7.1. and 7.2. was first mentioned by Chomsky (1959a). We have borrowed the argumentation for Theorem 7.1. from Hopcroft and Ullman (1969). The argumentation for Theorem 7.2. was taken from Chomsky (1963), who in turn refers to Davis (1958), starting from the fact that type-0 languages are recursively enumerable sets. The argumentation for Theorem 7.3. was borrowed from Hopcroft and Ullman (1969). The first surveys of the relationship between formal languages and automata were Chomsky (1963) and Chomsky and Miller (1963) on the one hand, and Bar-Hillel (1964) on the other.

The earliest publication on grammatical inference is Miller and

Chomsky (1957). Solomonoff (1958, 1964 a, b) was the first to develop these ideas. The Feldman group, with among them Horning, has also done important work in this field (Feldman et al. 1969).

The best recent surveys of the subjects treated in this volume are Nelson (1968) where various topics are treated within the theory of formal systems, and Hopcroft and Ullman (1969) to which the present work is indebted and which would serve as excellent further reading. Neither of these books, however, deals with probabilistic grammars or probabilistic automata. For the latter, we refer the reader to Salomaa (1969). There are no standard texts on probabilistic grammars or grammatical inference.

BIBLIOGRAPHY

Bar-Hillel, Y.

1964 *Language and Information. Selected Essays on Theory and Application* (Reading, Mass.: Addison-Wesley).

Bar-Hillel, Y., C. Gaifman, and E. Shamir

1960 "On Categorical and Phrase Structure Grammars", *Bull. Res. Council of Israel* 9F: 1-16. (See also Bar-Hillel 1964.)

Booth, T. L.

1969 "Probability Representation of Formal Languages", *IEEE Tenth Annual Symposium on Switching and Automata Theory* (November).

Chomsky, N.

1959a "On Certain Formal Properties of Grammars", *Information and Control* 2: 137-67.

1959b "A Note on Phrase Structure Grammars", *Information and Control* 2: 393-95.

1962 "Context-Free Grammars and Pushdown Storage" (= *RLE Quart. Prog. Rept. No. 65*) (Cambridge, Mass.: MIT).

1963 "Formal Properties of Grammar", *Handbook of Mathematical Psychology*, R. D. Luce, R. R. Bush, and E. Galanter (eds.) (New York: Wiley).

Chomsky, N., and G. A. Miller

1958 "Finite State Languages", *Information and Control* 1: 91-112.

1963 "Introduction to the Formal Analysis of Natural Languages", *Handbook of Mathematical Psychology*, R. D. Luce, R. R. Bush, and E. Galanter (eds.) (New York: Wiley).

Chomsky, N., and M. P. Schützenberger

1963 "The Algebraic Theory of Context-free Languages", *Computer Programming and Formal Systems*, P. Braffort and D. Hirschberg (eds.) (Amsterdam: North-Holland).

Davis, Martin

1958 *Computability and Unsolvability* (New York: McGraw-Hill, 1958).

Ellis, G. A.

1969 "Probabilistic Languages and Automata" (= *Rept. no. 355. Dept. Comp. Sc.*) (University of Illinois, Urbana, Ill.).

Evey, R. J.

1963 "The Theory and Application of Pushdown Machines", *Mathematical*

Linguistics and Automatic Translation (= *Computation Lab. Rept.* NSF-10) (Cambridge, Mass.: Harvard).

Feldman, J. A., J. Gips, J. J. Horning, and S. Reder
1969 *Grammatical Complexity and Inference* (= *Techn. Rep. No. CS 125*) (Computer Science Dept., Stanford Univ.).

Feller, W.
1968 *An Introduction to Probability Theory and its Applications*, third edition (New York: Wiley).

Ginsburg, S.
1962 *An Introduction to Mathematical Machine Theory* (Reading, Mass.: Edison-Wesley).
1966 *The Mathematical Theory of Context-free Languages* (New York: McGraw-Hill).

Ginsburg, S., and J. Ullman
1966 "Ambiguity in Context-free Languages", *J. Assoc. Comp. Mach.* 13: 62-88.

Ginzburg, A.
1968 *Algebraic Theory of Automata* (New York: Academic Press).

Gold, E. M.
1967 "Language Identification in the Limit", *Information and Control* 10: 441-74.

Greibach, S. A.
1963 "The Undecidability of the Ambiguity Problem for Minimal Linear Grammars", *Information and Control* 6: 117-25.
1965 "A New Normal Form Theorem for Context-free Phrase Structure Grammars", *J. Ass. Comp. Mach.* 12: 42-52.
1966 "The Unsolvability of the Recognition of Linear Context-free Languages", *J. Ass. Comp. Mach.* 13: 582-87.

Grenander, U.
1967 "Syntax-controlled Probabilities", *Rept. Division Appl. Mathem.* (Brown University, Providence, R.I.).

Hopcroft, J. E., and J. D. Ullman
1969 *Formal Languages and Their Relation to Automata* (Reading, Mass.: Addison-Wesley).

Horning, J. J.
1969 "A Study of Grammatical Inference" (= *Technical Report* CS 139 *Stanford Artificial Intelligence Project*) (Stanford: Computer Science Department).

Huang, T., and K. S. Fu
1971 "On Stochastic Context-free Languages", *Information Sciences* 3: 201-24.

Kleene, S. C.
1952 *Introduction to Metamathematics* (Princeton: Van Nostrand).
1956 "Representation of Events in Nerve Nets and Finite Automata", *Automata Studies*, C. E. Shannon and J. McCarthy (eds.) (Princeton: Princeton University Press).

Knuth, D. E.
1965 "On the Translation of Languages from Left to Right", *Information and Control* 8: 607-39.
Kuroda, S. Y.
1964 "Classes of Languages and Linear-bounded Automata", *Information and Control* 7: 201-23.
Landweber, P. S.
1963 "Three Theorems on Phrase Structure Grammars of Type 1", *Information and Control* 6, 131-36.
McCulloch, W. S., and W. Pitts
1943 "A Logical Calculus of the Ideas Immanent in Nervous Activity", *Bull. Math. Biophysics* 5: 115-33.
Miller, G. A., and N. Chomsky
1957 *Pattern Conception* (Paper for Conference on Pattern Detection, University of Michigan).
Minsky, M. L.
1967 *Computation. Finite and Infinite Machines* (Englewood Cliffs: Prentice-Hall).
Myhill, J.
1960 *Linear Bounded Automata* (= *WADD Technical Note* 60-165) (Wright Air Development Division, Wright-Patterson Air Force Base, Ohio).
Nelson, R. J.
1968 *Introduction to Automata* (New York: Wiley).
Newell, A., J. C. Shaw, and H. A. Simon
1959 "Report on a General Problem-solving Program", *Information Processing, Proc. Intern. Conf. on Information Processing, UNESCO* (Paris, June).
Oettinger, A.
1961 "Automatic Syntactic Analysis and the Pushdown Store", *Structure of Language and its Mathematical Aspects*, R. Jakobson (ed.) (Providence: Amer. Math. Soc.).
Parikh, R. J.
1961 "Language Generating Devices", *Quart. Progr. Rep. MIT Res. Lab. Electr.* 60: 199-212.
Rabin, M. O.
1963 "Probabilistic Automata", *Information and Control* 6: 230-54.
Rabin, M. O., and D. Scott
1959 "Finite Automata and Their Decision Problem", *IBM J. Res.* 3: 115-25.
Salomaa, A.
1969 *Theory of Automata* (Oxford: Pergamon Press).
Solomonoff, R. J.
1958 "The Mechanization of Linguistic Learning", *Proc. Second Intern. Congr. Cybernetics* (Namur), 180-93.
1964a "A Formal Theory of Inductive Reference. Part I", *Information and Control* 7: 1-22.

1964b "A Formal Theory of Inductive Reference. Part II", *Information and Control* 7: 224-54.

Suppes, P.

1970 "Probabilistic Grammars for Natural Languages", *Synthese* 22: 95-116.

Turing, A. M.

1936 "On Computable Numbers, With an Application to the Entscheidungsproblem", *Proc. London Math. Soc.* 42: 230-65.

AUTHOR INDEX

SUBJECT INDEX

(italicized numbers refer to definitions)

FORMAL GRAMMARS IN LINGUISTICS AND PSYCHOLINGUISTICS

VOLUME II

Applications in Linguistic Theory

by

W. J. M. LEVELT

1974

MOUTON

THE HAGUE · PARIS

PREFACE

Since the publication of Chomsky's *Syntactic Structures* (1957), linguistic theory has been strongly under the influence of the theory of formal languages, particularly as far as syntax is concerned. Investigations have dealt not only with the extent to which "pure" regular or phrase structure grammars can be used as models for a linguistic theory, but also with "mixed models", i.e., grammars to which a transformational component is added.

The most influential mixed model was that of Chomsky's *Aspects of the Theory of Syntax* (1965), but a number of other transformational grammars have been developed, such as dependency grammars and mixed adjunct grammars, with very different formal structures. Each of these grammars has its own specific advantages and disadvantages to offer. This volume presents a survey of the most important pure and mixed models, their formal structure, their mutual relations, and their linguistic peculiarities.

The formal structure of many transformational grammars has not been worked out in detail. This holds in particular for the syntax of *Aspects*. This fact may be considered as a simple esthetic fault, but on closer examination many deeper problems appear to be connected with it. The formalization of the grammar in *Aspects* has proven that the grammar in its standard form, as well as in later developments of it, cannot handle essential linguistic questions, such as that of the learnability of natural languages and the existence of a universal base grammar. A separate chapter deals with these problems.

Finally, attention will be given to the application of probabilistic grammars in linguistics.

This volume is concerned exclusively with linguistic questions. Psychological matters closely connected with them, such as the distinction between *competence* and *performance*, and the structure of linguistic intuitions, will be treated in Volume III.

Volume II presupposes acquaintance with the essentials of the material on formal grammar theory contained in Volume I. Cross-references to this Volume are made throughout the text. In its turn, the present Volume is preparatory to Volume III.

Nijmegen, June 1973

TABLE OF CONTENTS

1

LINGUISTICS: THEORY AND INTERPRETATION

In this volume we shall discuss the ways in which formal languages are used as models of natural languages, and formal grammars as models of linguistic theories. It was pointed out in Chapter 1 of the first volume that several concepts which have been incorporated into the theory of formal languages have lost something of the meaning they had in linguistics. As in the present volume our attention will be turned to natural languages, it will be necessary to re-examine such essential concepts as "sentence", "language" and "grammar", but in order to do so, we must first make a careful distinction between linguistic theory on the one hand, and its empirical domain on the other.

1.1. THE EMPIRICAL DOMAIN OF A LINGUISTIC THEORY

With respect to linguistic THEORY the problem of definitions mentioned above is trivial, and hardly anything need be changed in the given formal definitions. The formulation of linguistic theory must also aim at EXPLICITNESS and CONSISTENCY. The propositions of a linguistic theory (for example, "sentence x belongs to language L", "string y is a nominalization in language L", etc.) must be *explicit*, that is, it must be possible for anyone verifiably to deduce them from the principles of the theory, without making use of knowledge of the language in question obtained outside the theory. Consequently concepts such as "sentence", "language", etc. may not have the intuitive vagueness in the theory which

they have in ordinary speech. As in formal language theory, a LANGUAGE in a theory of natural languages is a set of sentences, a SENTENCE is a string of elements (to be defined more fully), which satisfies the condition that it be generated by a GRAMMAR, which is a system of production rules, defined over a terminal and a nonterminal VOCABULARY. Such a theory is *consistent* if it does not lead to contradictions, that is, if it is impossible to deduce both a proposition and its negation within the theory. Obviously it is impossible to decide whether a theory is consistent or not if it is not explicit.

On one point, however, due to historical circumstances, a certain terminological ambiguity has come into being. As we have mentioned, a formal grammar is complete in that it is an exhaustive description of the sentences of a language and of their structure. In linguistics, the notion of grammar originally was used primarily for "syntax and morphology", the study of syntactic structure and the structure of words. Seen in this way, a linguistic grammar is not as complete as a formal grammar. A linguistic theory is not complete without phonological and semantic descriptions, concerning respectively the aspects of sound and meaning in the natural language. At first, applications of formal grammars to linguistic theory dealt exclusively with grammatical aspects in the original linguistic sense of the word: semantics was excluded and phonology was considered a more or less independent component and was studied separately. The impression was often given that the formalization thus obtained enjoyed the same degree of completeness in the description of natural languages as formal grammar in the description of formal languages. The notion of "grammar" became synonymous with that of "linguistic theory". It is still often used in this general sense, even now that the essential interest of semantics to linguistic theory is again emphasized in all quarters. If semantics is considered to be a subdivision of grammar, or if it is seen as indistinguishable from syntax, it remains something essential to grammar, and not something aside or apart from it. Some linguists maintain the original terminology, and use the word "grammar" only for syntax-and-morphology. There is no point in rejecting

either terminology as "incorrect"; it is simply a question of scientific tradition and pragmatic considerations. On the basis of such pragmatic considerations, we shall use the word "grammar" only in its more limited sense, for it is quite evident that the clearest and most influential applications of formal language theory to linguistics have been related to syntax and morphology. Phonology has indeed been greatly formalized, but this has seldom been the result of *direct* applications of formal language theory. There have also been applications to semantics, but these have by far been neither as deep nor as extensive as those to syntax. As the subject of this volume is in fact applications to syntax, we shall, unless otherwise mentioned, use the notion of "grammar" as limited to syntax and morphology. Phonology and semantics will only rarely enter the discussion (semantics primarily in Volume III). Therefore we can, without risk of confusion, use the term "grammar" for "grammar in the limited sense", thus maintaining the connection with Volume I as far as possible.

Even within these limitations, however, it still holds that grammatical concepts to be used in linguistic theory should not have the vagueness (and wealth) of connotation which they might have in ordinary usage. Concepts must be defined entirely within the theory, and this holds as well for the concepts just mentioned as for other linguistic concepts such as "verb" and "noun phrase" which have not yet been discussed. They should all be fully defined within the formal description, and the relationships among them are established by the definitions and rules of the grammar. There is never reason for rejecting such concepts separately, but at most for rejecting the grammar as a whole.

But a linguistic theory is also an EMPIRICAL theory: it is designed to explain certain observable phenomena in verbal communication among human beings. As a whole, the observable phenomena with which a theory is concerned is called its EMPIRICAL DOMAIN. The size of the domain is not determined beforehand. Some verbal phenomena which seldom or never occur spontaneously might be elicited by various means; one can, for example, pose directed questions to the native speaker. These observable phe-

nomena correspond with that which is called LANGUAGE in the theory. The theory is an abstract description of the kind of system a natural language is. This description must maintain a direct and understandable relationship with certain aspects of the observable linguistic phenomena. Thus the concept of "sentence" must in some way be related to that which is observable as an "utterance", the concept of "grammatical" (i.e. "generated by the grammar") ought to have something to do with the native speaker's judgment of which utterances are or are not "acceptable" or "good" English, Dutch, etc. The theoretical concept of "paraphrase" is perhaps related to the judgment of a hearer that a speaker means the same thing with two different utterances, and so forth. The network of theoretical concepts must be composed in such a way that the theory reflects the linguistic reality. In order to determine whether or not a theory satisfies this condition, as complete a description as possible must be given of the relations between the theoretical concepts on the one hand, and the empirical domain on the other.

In linguistics there are many cases in which the relations between theory and observable phenomena are simple and acceptable. We already know that various sentences (the relationship with "utterances" will be discussed later) must enjoy the status of "grammatical" in the theory. Any native English speaker will confirm that *the boy walks on the street* is good English, and that *on walks the street boy the* is not. Therefore a theory of the English language should be constructed in such a way that the grammar generates the first string and not the second. Such incontrovertible data are numerous enough to allow the construction of a linguistic theory and to test certain aspects of it, and we might hope that for less clear cases the theory itself might decide (for example, is *if he comes then she will go then he will come* a grammatical sentence?). This means that we can simply notice whether or not the sentence can be generated by the grammar which is composed on the basis of clear cases.

This method has the advantage that a maximum of theoretical construction can be realized with a minimum of troubling about

procedures of data gathering and processing. The history of transformational grammars has shown that this kind of capitalization on immediate intuitive evidence is indeed an extraordinarily fruitful approach. The understanding of the structure of natural languages has probably never grown as rapidly as since the publication of Chomsky's *Syntactic Structures* (1957) in which this very programme was presented.

The reader should notice that this method is based upon a more or less explicit conception of the empirical domain of linguistics. This domain is in fact much less broad than that which we understood by "observable linguistic phenomena". This type of theory is descriptive of only some aspects of verbal phenomena, and the best generic name for those aspects is *linguistic intuitions*. Intuitions are observable in the form of *metalinguistic judgments*, that is, judgments the objects of which are verbal entities. Thus, from the point of view of theory, the objects of judgments on paraphrase relations and grammaticality are SENTENCES, the objects of judgments on cohesion within sentences are PHRASES, and so forth. It may be said that such linguistic theories concern metalinguistic data. It is the reflection of the native speaker on his own speech, which is formalized in the theory. This can also be stated in another way. One of the more noticeable characteristics of a natural language is that it is its own metalanguage; this means that by using a language one can speak about that language itself. The attention of theoretical linguistics (in the sense we have just mentioned) is principally directed toward those verbal activities whose object is the language itself. This restriction is scarcely necessary, and we shall repeatedly return to its attractive and unattractive implications in Volume III, Chapters 1 and 2. Yet until now the application of formal grammars to linguistic theory has in general presupposed such a limitation of the empirical domain.

In the following chapters we shall discuss the adequacy of various formal grammars as models, on the basis of linguistic intuitions as described above. This means that we suppose for the sake of discussion that the relationship between theory and

observation is directly visible and acceptable, or in other words, that every reliable data processing procedure will support the intuitive insight.

1.2. THE INTERPRETATION PROBLEM

This, of course, does not close the discussion. Not only is the restriction of the empirical domain of a linguistic theory to linguistic intuitions far from always clear or attractive, but it is also the case that we do not always dispose of such direct evidence, and even when we do, some very essential questions remain to be answered. We speak of the INTERPRETATION PROBLEM when the relationship between theory and observation is itself the object of investigation. The question then becomes how the theory should be interpreted in terms of observable phenomena. We shall at this point mention three cases in which important linguistic interpretation problems occur.

The first case, in which the problem of interpretation makes itself increasingly felt, is that of the *use of linguistic intuitions* which we have just mentioned. It has slowly but surely become clear that it is not possible, on the basis of incontrovertible, directly evident data, to construct a theory so extensive that all less obvious cases can be decided upon by the grammar itself. It is becoming more and more apparent that decisions on very important areas of theory are dependent on very unreliable observations. In Volume III, Chapter 2 we shall mention an experiment (Levelt, 1972) which shows the frightful unreliability in judgments on grammaticality which occurs in modern transformational linguistic investigations. There is a tendency toward preoccupation with extremely subtle distinctions, not the importance, but rather the direct observability of which can seriously be called into question. Better methods must be developed for testing linguistic intuitions, and this is certainly a realizable possibility (cf. Volume III, Chapter 2, paragraphs 3 and 4). Moreover, a methodological tradition has existed in linguistics for some time, in which more value is given

to some intuitions than to others. This tradition in general is based on implicit conceptions of the problem of interpretation. This can clearly be seen, for example, in the treatment of very long sentences. As we have noticed, there is a tendency to establish a relationship between theoretical grammaticality and the judgment 'sentence x is good English'. But there are sentences which an informant might call 'bad English' or 'not English' on the basis of circumstances which we could not easily formalize in a LINGUISTIC theory. Such is the case for very long sentences. They are unacceptable because our limited memory capacity makes it impossible for us to understand them. It seems undesirable to include an upper limit of sentence length into a grammar; it would be wiser to handle the limitation of length in a psychological theory as a systematic property of human memory, than in a linguistic theory as a systematic property of natural language.

The fact that such an intuition is disregarded by the linguist clearly shows that psychological presuppositions are implicit in the theoretical interpretation of certain linguistic observations. The example, moreover, is by no means incidental. Motivational, socio-psychological and other psychological factors must also be sifted out by the linguist in the interpretation of linguistic intuitions.

This first interpretation problem is thus of a psychological nature. One might wish that every linguist would fully recognize the role of psychological assumptions in the formulation of his theories. Unfortunately this is not the case. On the contrary, many linguists maintain that an adequate psychological theory of verbal behavior is possible only to the extent that linguistic knowledge is available (cf. Chomsky, 1965). The only consequence of this attitude is that at present psychological theory is implicitly working its way into the formulation of linguistic theory, instead of explicitly being taken into account and thus held in control. There is no reason to suppose that the common sense psychology of linguists is in any way better than the common sense linguistics of psychologists.

The second case of the interpretation problem is related to the

first; it occurs when an adequate grammar is given and the question is asked as to whence the linguistic structures described by the grammar proceed. Before developing this matter, we must first clarify the notion of "adequacy". A grammar is called OBSERVATIONALLY ADEQUATE if it generates the sentences of a given language and only the sentences of that language. Because judgment on the observational adequacy of a concrete grammar can be given only on the basis of a concrete and therefore finite corpus of sentences (and at best a finite set of non-sentences), Chomsky calls a grammar observationally adequate when "the grammar presents the observed primary data correctly" (Chomsky, 1964). A grammar is DESCRIPTIVELY ADEQUATE if it gives a correct formalization of the linguistic intuitions. A descriptively adequate grammar is obviously also observationally adequate, because the decision as to whether or not a sentence belongs to a language is also based on an intuitive judgment of the native speaker. A descriptively adequate grammar moreover gives a correct reflection of intuitions about the structure of sentences, the relations between words and word groups, the relations among similar sentences, etc. There are always several possibilities of writing an observationally adequate grammar for a language. A sufficient number of examples of this have been given in volume I to make the point (see, for example, Figure 2.3 and the accompanying text).

In the same way a linguist can probably also dispose of various options for writing a DESCRIPTIVELY adequate grammar. One way of choosing from among several formulations is to compare them with the grammars of other languages. In a GENERAL linguistic theory, the elements common to all natural language, the general systematic properties of natural language, also called UNIVERSALS, will be described. There is thus only a limited degree of freedom in the description of a specific language. Neither for general linguistics nor for the description of individual languages are there generally accepted criteria for the choice of adequate grammars. We refer the reader to Volume I, Chapter 8 in which a number of problems are discussed which can appear in the comparison of grammars, and to this volume, Chapter 5 in which it is shown that certain

very common suppositions on the form of a general linguistic theory cannot be tested empirically.

In spite of these important and unsolved problems concerning observational and descriptive adequacy, still a third form of adequacy can theoretically be imagined. Given a descriptively adequate general theory of linguistics, one can wonder by which psychological, biological and cultural factors this systematic structure of natural languages is determined. A linguistic theory which also answers these questions is called EXPLANATORILY ADEQUATE. This is clearly an extremely hypothetical field. Finally, there is as yet very little in linguistics which might be called adequate even from an observational point of view. Nevertheless, reasoning back from this abstraction, a number of important questions can be posed concerning the problem of interpretation, questions which lend themselves to empirical investigation even without disposing of complete and adequate grammars.

The explanation of the existence of certain grammatical properties must in the first place be brought back to an explanation of the linguistic intuitions with which it is connected. At present practically nothing is known of the nature of linguistic intuitions. We do not know how they come into being, how they are related to conceptions of one's own linguistic behavior in concrete situations, how they change under the influence of situational circumstances, what interaction there may be between them and perceptual aspects of time and place, how they are related to the systematic physical and social structure of the environment. It is also unknown whether or not, and if so to what extent such intuitions are trainable and how they develop in the growth of the child.

In the second place the explanation of a grammar must be brought back to the genetic question of how the language develops in the child. Research should be done on the means with which the child makes the language of his environment his own, to what extent these means are the same as those which the child uses in learning perceptual and conceptual skills in general, and whether the nature of these "cognitive strategies" is also determinant for some structural properties of natural language. Is it possible to

give a general characterization of the relationship between language structure and the learnability of a language? A few mathematical aspects of this question have already been treated in Volume I, Chapter 8, and we shall return to them in Volume III, Chapter 4. It should suffice here to point out that the nature of linguistic intuitions and their development in the child is one of the most fundamental facets of the problem of interpretation. It is quite obvious that both of these aspects are very largely of a psychological nature.

A third case in which the problem of interpretation is of great importance in the formulation of linguistic theory occurs when one has to deal with the *analysis of a given corpus,* without the benefit of access to linguistic intuitions. Not only does the linguist have to cope with this situation in the study of dead languages, but also in applied linguistics he will find it to be less the exception than the rule. Translation and style analysis, for example, are most often performed on the basis of a corpus and without further access to the person who produced the text. From a formal point of view, however, the problem of corpus analysis has been most difficult in the analysis of children's language. One cannot base the development and testing of a grammar for the language of a three-year-old principally on linguistic intuitions. The number of metalinguistic utterances which a small child makes is quite small, and it is possible only on a very limited scale to elicit linguistic judgments from him. The small child is not comparable to the adult as an informant. If the domain of linguistic theory is limited to linguistic intuitions, the study of children's languages becomes a nearly impossible task. The data which can be obtained consists primarily of spontaneous utterances from the child and of observations relative to the circumstances under which they are produced. With ingenious experiments some information may be added to this, but the problem of determining what it is in verbal behavior which corresponds to the theoretical concept of "sentence" still remains, and usually demands a study in itself. It would not be advisable, for example, to consider every recorded utterance as a sentence in the child's language. Different utterances will often be

taken for separate occurrences of a single sentence on the ground of acoustical criteria. If agreement can be reached at all on what these criteria are, it is still possible that not every class of equivalent utterances thus obtained will correspond to sentences in the theory. The grammar can often be simplified by excluding certain utterances or classes of utterances such as, for example, imitations which clearly have not been understood, and utterances of a very infrequent sort. Statistical and other data processing procedures will sometimes need to be adopted for the interpretation of the theory in terms of observable verbal behavior. Should the status of "sentence" be accorded to every utterance which the child understands? Decidedly not, for, as everyone knows, the child can understand much more than he can produce. But where can one draw the line? A sentence not produced by the child can very well have the status of "sentence" in the grammar. This is in fact the interpretation problem *par excellence*. Further interpretation problems occur in applied linguistics. In style analysis one might like to make use of data on distributions of sentence length and word frequencies. In the analysis of children's languages likewise, such data are often useful as parameters of growth or verbal skill. These matters can be considered as linguistic applications of inference theory (cf. Volume I, Chapter 8), where the interpretation problem is of central importance.

1.3. A FEW DESCRIPTIVE DEFINITIONS

In the preceding, careful distinctions were made between theory and observation, and between theoretical and empirical concepts. Theoretical concepts are determined entirely by their formal relations within the theory; in this connection we have already repeated the formal definitions of "sentence", "language" and "grammar", which may be found in greater detail in Volume I, Chapter 1. To these we can add definitions of "morpheme" and of "syntactic category". Unless otherwise stated, MORPHEMES are the terminal elements of the grammar; together they form the

terminal vocabulary. For the sake of simplicity, we shall often refer to the terminal elements as WORDS, except where this might lead to confusion. We would point out that in transformational grammar another term, FORMATIVES, is also used for the terminal elements.[1] The nonterminal vocabulary (cf. Volume I, Chapter 1), by definition, is made up of SYNTACTIC CATEGORIES. The symbolic abbreviations for these are called CATEGORY SYMBOLS.

As a support to the intuitions which will often be called upon in the following chapters, we present a few descriptive definitions of the correspondents in the empirical domain of the concepts "sentence", "morpheme", "word", and "syntactic category". These are not formal definitions, and are meant only to make the ideas a bit clearer. They pretend to no completeness, but refer to each other as do the theoretical concepts.

SENTENCE. This theoretical concept has, as mentioned above, something to do with the empirical concept of "utterance". A hearer might consider two utterances to be identical, in spite of acoustical differences among them. Whether it is John or Mary who says *the weather is nice*, in the intuition of the native speaker, the two acoustical forms are simply occurrences of the same sentence. The intuition "the same statement (question, exclamation, etc.)" thus determines classes of equivalent utterances. Let us call each of those classes a *linguistic construction*. The relationship between this abstract empirical concept "linguistic construction" and the theoretical concept "sentence" remains complicated. On the one hand, there are linguistic constructions which we might prefer to represent theoretically as combinations of sentences; a story, for example, is a linguistic construction which we would ordinarily prefer to analyze as a sequence of sentences. On the other hand, there are linguistic constructions which we would rather consider to be parts of sentences than complete sentences; thus, for

[1] Some authors (Katz and Postal et al., 1964) make a further distinction between morphemes and formatives. Others (Chomsky, 1965) treat only the concept "formative".

example, the answer to the question *where is the hat?* is *on the table. On the table* is a linguistic construction (which might sound very different when spoken by different speakers or at different times), but we do not consider it a sentence. The principal reason for this is that *on the table* is dependent on the question which precedes it. Thus *on the table* cannot follow the question *what is your age?*. The principle used here is that of *distributional independence* (for a definition and discussion of the principle, see Lyons, 1968). Within a linguistic construction to which we should like to accord the status of sentence, various distributional dependences exist. Thus *the lazy nurse stood up* is good English, but *the lazy stone stood up* is not (it could be good English at best in a metaphorical context, but we shall not discuss such cases here). There are limitations of the nouns which can follow the adjective *lazy*. Names of inanimate objects are excluded in this connection (abstraction made of idioms such as "lazy day", etc.). This is a distributional limitation. We let the concept "sentence" correspond to linguistic constructions *within* which distributional limitations hold, but *among* which no distributional limitations hold. Consequently *on the table* is not a sentence, because it is dependent on the earlier question. However this does not yet solve the problem, as we see in the following question and answer situation: *where is your aunt?, she is coming*. The distributional dependence between these sentences is expressed in the intuition that at first sight *he is coming* is an unacceptable sequence to *where is your aunt?*. Yet we would like to represent *she is coming* in the grammar as a sentence. To allow for this, we can make an exception for pronouns in the rule of distributional independence. In other words, we represent these linguistic constructions as sentences in the grammar, adding that the pronoun stands for the noun mentioned in the other sentence. But this too falls short of solving all the problems. Additional criteria can always be given to provide the theoretical concept "sentence" with as careful an empirical basis as possible. Criteria concerning the intonation of the utterance would be an example of this. For the ends of the present volume, however, no further differentiation is needed.

MORPHEME (*formative*). Just as the theoretical concept "sentence" corresponds to the empirical concept "utterance", the theoretical concept "morpheme" corresponds to the empirical concept "morph". Roughly defined, morphs are the smallest meaning-carrying elements of a linguistic construction. Thus *the boys walked* can be segmented as *the-boy-s-walk-ed*; each segment is a morph with a functional or referential meaning. Some morphs can occur "independently" in a linguistic construction. This is the case here for *boy* and *walk*. Others occur only in combination, such as *s* (for the plural) and *ed* (for the past tense) in the present example. The status of *the* is less clear in this connection. Nevertheless we do not wish to limit the terminal elements of a grammar to such observable elements. The linguistic construction *the children ran*, segmented as *the-child-ren-ran*, makes the reason for this quite clear. The meaning of plural is carried by the morph *ren*, and we might thus consider the morph *s* of the preceding example and the morph *ren* as variants of the same grammatical element. The corresponding morpheme in the grammar can be written abstractly as *plural*, or simply as *pl*. The question becomes more abstract, however, when we compare *walk-ed* and *ran*. Our intuition tells us that *walk* is related to *walked* in the same way as *run* is related to *ran*, but in the latter case there is no separate morph which expresses the past tense. Change of tense occurs in the form of a vowel shift in the root. By analogy with *walked*, we can represent *ran* as *run* + *past tense*, or simply as *run* + *past t*, where *past t* represents the past tense morpheme. The consequence of this, however, is that without further additions such words as *ran*, *walked*, *boys*, *children* can no longer be generated by the grammar. The terminal strings will contain such pairs as *run* + *past t*, *boy* + *pl*, etc. Therefore rules must be added to the grammar to change these strings to the correct froms (*ran*, *boys*, etc.). The part of grammar called *morphology* is concerned with such rules. Morphological rules will not be explicitly discussed here, and we shall suppose that a morphology exists for changing the terminal strings of morphemes into the proper word forms, and will represent the terminal elements directly as words, unless this in a given case

might lead to confusion. Morphemes such as *run*, *boy*, *walk* are called *lexical formatives*, and *pl* and *past t* are called *grammatical formatives.*

WORD. This concept will be used only to mean "terminal element", as mentioned above. Theoretically this concept would more properly belong to morphology, which, as we have stated, will be left largely untouched. A rather good definition of the concept "word" is 'a minimal free form'. There are various ways of interchanging morphemes in a sentence and of adding new morphemes without changing the character of the sentence. If, for example, *the boys are walking* is a sentence, then *are the boys walking* is also a sentence, and *the big boys are walking* likewise. In shifts of this kind, some morphs always remain coupled, like *walk* and *ing*, and *boy* and *s*. Such internally connected groups are called words. The smallest free forms in the example are *the*, *boys*, *are*, *walking*, and the form *big* which was added later. This definition is certainly not exhaustive, but should be sufficient to serve as a memory aid for the rest of the book.

SYNTACTIC CATEGORY. This concept is the most difficult to define. Two things may be borne in mind in connection with it. In the first place, one can relate the concept to that which is ordinarily called a "phrase", such as "a noun phrase" (e.g. *the big boy*, *John's carpentry*, *old folks*) or "a verb phrase" (e.g. *goes to school*, *does not give himself away*, *is a bit lazy*, *plays the piano*). In the second place, the concept relates to *classes of morphemes*, such as "number" for the class of morphemes consisting of "singular" and "plural", "tense" for the class of morphemes of time (*past t*, *pres t*, etc.), "verb" for the class of morphemes like *walk*, *run*, *sing*, etc. Formal models of natural languages tend to show considerable divergence in the choice of syntactic categories. Therefore in the following we shall give supplementary definitions when needed.

2

PURE MODELS: PHRASE-STRUCTURE GRAMMARS

In this chapter formal grammars of the pure types 3, 2, and 1 will be examined on their value as models for linguistic grammars. When these grammars are used in linguistics, they are denoted by the generic term PHRASE STRUCTURE GRAMMARS, or CONSTITUENT STRUCTURE GRAMMARS. These designations are related to the fact, discussed in Volume I, Chapter 2, that derivations in such grammars can be represented by means of tree-diagrams. The reader may remember that this held for type-1 grammars only when their production rules were of context-sensitive form (cf. Volume I, paragraph 2.4.1); in the following we shall continue to respect that condition. A tree-diagram clearly shows the phrases of which a sentence is composed. Phrases may also be called CONSTITUENTS, whence the second term for this family of grammars. In linguistics, tree-diagrams for sentences are often called PHRASE MARKERS or simply P-MARKERS.

2.1. GENERATIVE POWER AND STRUCTURAL DESCRIPTION

The order of subjects to be discussed in this chapter will be determined by the methodological principles which have consistently served as the basis of the investigation of formal models in linguistics. Thus the strongest possible model is chosen first to see if that model can be maintained for the description of natural languages. Only if the model can convincingly be rejected can one go a step higher in the hierarchy and repeat the procedure. In this way one

can be sure that the grammar used will never be too broad for the language (cf. Volume I, Section 2.1). Some clarification of what is meant by a "model which can be maintained" or a "tenable model" will be useful. It is only in the more limited sense of "observational adequacy" that we can see precisely what is required, namely that a grammar generate all and only the sentences of a language. One can speak here of the WEAK GENERATIVE POWER of the grammar; this is the language which is generated by the grammar. The weak generative power of a class of grammars (for example, that of the class of regular grammars) is the set of languages generated by the grammars in that class. Thus the weak generative power of the class $\{G_1, G_2, \ldots\}$ is the set $\{L_1, L_2, \ldots\}$, where language L_i is generated by grammar G_i.

It is much less easy to decide whether or not a grammar is descriptively adequate, that is, whether or not it correctly reflects the intuitions of the native speaker. This requirement is often operationalized in the criterion of whether or not the grammar assigns the correct STRUCTURAL DESCRIPTION to the sentences generated. The structural description is the information about the sentences, given by the grammar. This information is contained completely in the GENERATION of the sentence (the LEFTMOST DERIVATION for context-free grammars; cf. Volume I, paragraph 2.3.4). It shows how the sentence is composed of terminal elements, the syntactic categories to which words and phrases in the sentence belong, which production rules were used in the derivation and in what order. On the basis of such structural data other intuitions can also be formalized, such as intuitions concerning the relations among various sentences. Structural descriptions for regular and context-free grammars are identical with the P-marker. Derivations in context-sensitive and type-0 grammars cannot unambiguously be shown in tree-diagrams, and consequently further definition of "structural description" will be necessary. For the present, however, we may decide that the structural description of a sentence will be denoted by the symbol Σ. The set of structural descriptions given or generated by a grammar G is called the ANALYZED LANGUAGE $A(G)$, generated by G. Thus

$\Lambda(G) = \{\Sigma_i \mid \Sigma_i$ generated by $G\}$. $\Lambda(G)$ is also called the STRONG GENERATIVE POWER of G. The strong generative power of a class of grammars $\{G_1, G_2, \ldots\}$ is $\{\Lambda_1, \Lambda_2, \ldots\}$, where $\Lambda_i = \Lambda(G_i)$.

It is possible that an observationally adequate grammar might assign structural descriptions to sentences, while those structural descriptions conflict with various intuitions. It does not seem likely, however, that we might ever be able to expect proof of the intenability of a grammar on grounds of descriptive inadequacy; such a grammar would rather be rejected on the basis of increasing inconvenience in working with it. We shall see later, moreover, that contrary to current opinions, observational inadequacy has never been strictly proven for any class of grammars whatsoever.

Having defined strong generative power in addition to weak generative power, we must give the same extension to the concept of EQUIVALENCE of grammars. In Volume I, paragraph 1.2, we stated that two grammars G_1 and G_2 are weakly equivalent if $L(G_1) = L(G_2)$. We add to this that two grammars G_1 and G_2 are STRONGLY EQUIVALENT if $\Lambda(G_1) = \Lambda(G_2)$. If G_1 and G_2 are strongly equivalent context-free grammars, they assign the same (set of) tree-diagrams to sentences. ("Set" is added between parentheses to cover cases where sentences are ambiguous and may therefore have several different tree-diagrams; cf. Volume I, Figures 2.4 and 2.5). The inverse, however, does not hold in all cases: for context-sensitive grammars the same tree-diagram can be obtained for two different derivations (cf. Volume I, paragraph 2.4.1). The concept of "strong equivalence" is of linguistic interest because of the problem of the descriptive adequacy of grammars. Thus if G_1 and G_2 are strongly equivalent and G_1 is descriptively adequate, then G_2 is also descriptively adequate. Yet the concept presented in its usual form is rather trivial; two strongly equivalent grammars are identical, with the possible exception of a few uninteresting details. They may only differ in unusable production rules, i.e. rules which, if used, do not lead to terminal strings, or in vocabulary elements which cannot be used. Linguistics is decidedly in need of formalization of "equivalence of structural description", but the

strength of that concept should be attuned to the tolerance of our intuitions toward syntactic structures. The only effort in this direction known to us is that of Kuroda (1972).

2.2. REGULAR GRAMMARS FOR NATURAL LANGUAGES

How can a regular grammar be imagined as a model for a linguistic grammar? This can best be illustrated by an example.

EXAMPLE 2.1. Let $G = (V_N, V_T, P, S)$ be a regular grammar, where $V_N = \{S, A, B\}$, $V_T = \{the, bites, dog, cat, scratches, black\}$, and P contains the following productions:

1. $S \rightarrow the\ A$
2. $A \rightarrow black\ A$
3. $A \rightarrow cat\ B$
4. $A \rightarrow dog\ B$
5. $B \rightarrow bites$
6. $B \rightarrow scratches$
7. $B \rightarrow bites\ S$
8. $B \rightarrow scratches\ S$

This grammar can generate such sentences as *the dog bites*, *the black cat bites*, *the black cat scratches*. The derivation of the last sentence, for example, is $S \overset{1}{\Rightarrow} the\ A \overset{2}{\Rightarrow} the\ black\ A \overset{3}{\Rightarrow} the\ black\ cat\ B \overset{6}{\Rightarrow} the\ black\ cat\ scratches$ (the numbers written above the arrows refer to the productions used in the derivation step). The corresponding tree-diagram is given in Figure 2.1.

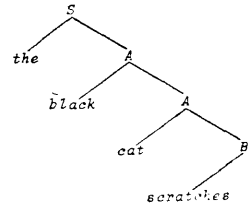


Fig. 2.1. P-marker for the sentence *the black cat scratches* (Example 2.1).

The grammar in fact generates an infinity of sentences. By virtue of production 2 which is recursive, the adjective *black* can be indefintely repeated, as in *the black black black dog bites*. The grammar

can also generate compound sentences thanks to productions 7 and 8 which reintroduce the start symbol S. This produces such sentences as *the dog bites the black cat scratches*, etc.

The equivalence of regular grammars and finite automata shown in Volume I, Chapter 4 suggests that a finite automaton (FA) can be constructed which will be equivalent to this grammar. The following FA is equivalent to G:

$FA = (S, I, \delta, s_0, F)$, with $S' = \{S, A, B\}$, $I = \{$*the, bites, dog, cat, scratches, black*$\}$, $s_0 = S$, $F = \{S\}$, and the following transition rules:

$\delta(S,$ *the*$) = A$
$\delta(A,$ *black*$) = A$
$\delta(A,$ *dog*$) = B$
$\delta(A,$ *cat*$) = B$
$\delta(B,$ *bites*$) = S$
$\delta(B,$ *scratches*$) = S$
$\delta(\text{-},\text{-}) = \varphi$ for all other cases

The transition diagram for this automaton is given in Figure 2.2. The diagram clearly shows which sentences the automaton accepts,

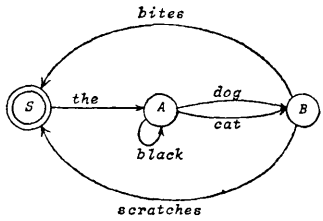


Fig. 2.2. Transition diagram for the finite automaton in Example 2.1.

and consequently the sentences which the grammar generates. Every path shown by the arrows from the initial state S to the final state S corresponds to a grammatical sentence. To complete the presentation, we give the transition table for this automaton in Table 2.1. The attentive reader will have noticed that this example is formally identical with Example 4.1 of Volume I.

It should be evident that many variations are possible here. It might be so arranged that the terminal vocabulary is made up of

TABLE 2.1. Transition Table for the Finite Automaton in Example 2.1.

	Input Symbols						
	the	*black*	*cat*	*dog*	*bites*	*scratches*	
States	*S*	*A*					
	A		*A*	*B*	*B*		
	B					*S*	*S*

morphemes instead of words; it could even be made up of letters or phonemes. The grammar can be rendered more abstract by composing its terminal vocabulary of *classes* of words and morphemes. Thus, regularities may be formulated as, for example, "an article can be followed by a noun or by an adjective", "a noun can be followed by a verb", etc. In this way SENTENCE SCHEMAS are generated, such as "article-adjective-noun-verb". The grammar must then be completed with "lexical rules" showing which words are articles, which are adjectives, and so forth.

One may search in vain in linguistic literature for an explicit proposal to model linguistic theory on regular grammars, in spite of appearances to the contrary.[1] Some confusion exists on this point, since linguists have not seldom used the terms "finite grammar" and "finite state grammar" interchangeably, and readers may be led to think they are referring to a regular grammar when they only mean a finite system of rules. The most explicit use of the model may be found in the application of communication theory to natural languages. The origin of a verbal message is described as a so-called *Markov-source*, which in essence is a probabilistic finite automaton. We shall return to this in Chapter 6, paragraph 1. Suffice it here to point out that this has never been presented as a model of linguistic theory in the strict sense, but only as a model for summing up global statistical properties of a text (oral or written). It has never touched the structure of such messages in detail. Linguistics has indeed gone through a period of 'flirtation' with the

[1] Diligent searching revealed one exception to this, Reich (1969). The large number of essential errors in this article, however, gives rise to some doubt as to the carefulness of the editors of *Language*.

model under the influence of communication theory, but this never went beyond the implicit. Since Chomsky's explicit linguistic formulation and his rejection of the model in 1956, no linguist has seriously proposed it as a model for linguistic grammars.

Chomsky (1956, 1957) rejects the model on grounds of observational inadequacy. The enormous influence which this argument has had on the development of modern linguistics justifies a rather detailed discussion of it. It is also the case that the argumentation as given in *Syntactic Structures* is not completely balanced (the same is true, to a lesser degree, of Chomsky's treatment of the question in 1956). A consequence of this has been that the same sort of evidence is incorrectly used for the rejection of other types of grammars, and, as we shall see, simply erroneous conclusions have been drawn.

Before dealing with the form of the reasoning, we must first consider the fact that every argument is based on the supposition that a natural language contains an infinite number of sentences. Every finite set of sentences can, in effect, be generated by a regular grammar (cf. Theorem 2.3 of Volume I). What is the linguistic justification for this supposition? Three reasons are ordinarily advanced. The first of these has already been mentioned in Chapter 1, paragraph 2 of the present volume, namely, that from a linguistic point of view it is not advisable to set an upper limit to sentence length. The inacceptability of very long sentences can be justified better on the basis of a psychological theory than on the basis of a linguistic theory. A language is infinite if for every sentence another sentence can be found which is longer than the first, and this is clearly an intuitive fact. The second reason, closely related to the first, is the possibility of coordination of sentences. If sentences s_1 and s_2 are declarative (for example, *John is walking* and *it is raining*), then s_1 *and* s_2 also form a declarative sentence (*John is walking and it is raining*). Thus for every pair of declarative sentences, a new declarative sentence can be found which is longer than either. If a language contains one declarative sentence, it contains an infinity of them. The third and principal reason, however, is the following. Imagine a natural language of

finite size. According to Theorem 2.3 of Volume I, a regular grammar can therefore be written for it. But this grammar will have no recursive production rules (i.e., production rules which make it possible to use a given nonterminal element repeatedly for an indefinite number of times in a derivation, like productions 2, 7, and 8 in Example 2.1). Excluding trivial cases, such rules lead to the generation of infinite languages. But if recursive production rules are excluded from the grammar, the number of production rules will be of the same order of magnitude as the number of sentences in the language. Such a grammar would scarcely be helpful in clarifying the structure of the language; a list of all the sentences would be quite as good. The assumption of infinitude is, in other words, a fundamental decision designed for finding a characterization of the language which is as general and as simple as possible.

The argument of inadequacy advanced in *Syntactic Structures* is of the following form: (a) a language with property x cannot be generated by a regular grammar, (b) natural language L has property x, therefore (c) L is not a regular language. The argumentation here is balanced, but the difficulty lies in demonstrating (b). Let us examine this more closely on the basis of the argument.

For property x we shall take self-embedding. From Theorem 2.8 in Volume I we know that self-embedding languages are not regular. This is step (a) in the argument. We must now show for (b) that English is a self-embedding language. This is done by referring to self-embedding subsets (called *subparts* in *Syntactic Structures*) in English. Thus, for example, if s_1 is grammatical, one can add a relative clause to it without loss of grammaticality, as in s_2:

s_1: *the rat ate the malt*
s_2: *the rat the cat killed, ate the malt*

One can now add a relative clause to the relative clause in s_2, as in s_3:

s_3: *the rat the cat the dog chased, killed, ate the malt*

The embedded structure of s_3 becomes obvious when we add parentheses:

(the rat (the cat (the dog chased) killed) ate the malt)

There is no fundamental limit to the number of possible self-embeddings of this kind. The sentences become complicated, but always remain completely unambiguous in meaning. When necessary, one can verify or falsify such a statement, as the following, completely unnatural sentence:

(William II (whom William III (whom William IV (whom William V succeeded) succeeded) succeeded) succeeded William I)

Another example of self-embedding in English is the following sequence:

s_1: *If John says it is raining, he is lying*
s_2: *If John says Joe says it is raining, he is lying*
s_3: *If John says Joe says Mike says it is raining, he is lying*
and so forth.

It would not be difficult to think of other examples. The conclusion is that on the basis of the self-embedding character of English (c) follows, i.e. English is not a regular language.

The self-embedding property (b) of English is however not yet demonstrated, in spite of appearances to the contrary. The only thing which has been proven is that English has self-embedding subsets. But it by no means follows from this that English is a self-embedding language. This can easily be seen in the following. Let language L consist of all strings over the vocabulary $V = \{a, b\}$, so that $L = V^+$. Language L is regular, because it is generated by a regular grammar with production rules $S \to aS$, $S \to bS$, $S \to a$, $S \to b$. Let us now consider the set $X = \{ww^R\}$, the set of symmetrical "mirror-image" sentences, where $w \in V^+$. It is clear that X is a subset of L. Moreover, X cannot be generated by any regular grammar, given its self-embedding property. Nevertheless L is a regular language. The reason why the argument errs is that sen-

tences which are excluded by a grammar for X are nevertheless sentences of L. The omission in the argument for inadequacy is that nothing is said of the grammatical status of relative sentences (or sentences of other types discussed) which are not self-embedding.

Chomsky's original argumentation (1956), to which he refers in *Syntactic Structures*, is considerably more precise. In it he shows that it is necessary for the proof to demonstrate that a certain change in the sentences of a self-embedding subset must always be accompanied by a certain other change, on pain of ungrammaticality. But in the demonstration of that theorem he does not test whether or not this is in fact the case for English. Chomsky chooses the following intuition concerning English: if s_1 and s_2 are English sentences, then *if* s_1 *then* s_2 is also an English sentence. Repeated embedding shows *if* (*if* s_1 *then* s_2) *then* s_2 also to be an English sentence, and in general, *if*^{n} s (*then* s_2)^n, $n \geqslant 1$. Let us suppose that this holds for English (although this is itself an open question); it must then also be shown that *if*^{n} s_1 (*then* s_2)^m is ungrammatical if $n \neq m$.[1] Chomsky, however, does not do this, and, moreover, it does not hold. Grammatical counter-examples are *if John sleeps he snores* and *John drank coffee, then he left*. Similar objections may be made to the other examples in Chomsky (1956) and (1957).

Fewer problems occur when the "proof" is stated as follows (this is due to Dr. H. Brandt Corstius, personal communication). We follow a procedure of indirect demonstration. Assume that English is regular. We now construct the following regular set T. $T = \{$*William (whom William)*^{n} *succeeded*^{m} *succeeded William* $|$ $n, m \geqslant 1\}$.[2] It has been proven by Bar-Hillel (see Hopcroft & Ullman, 1969) that the intersection of two regular sets is a regular set. Therefore, the intersection of English and T should be regular.

[1] It must at least be shown that $n \geqslant m$ for all sentences, or $n \leqslant m$ for all sentences, because not only is $\{a^n b^n\}$ non-regular, but $\{a^n b^m \mid n \leqslant m\}$ and $\{a^n b^m \mid n \geqslant m\}$ are likewise non-regular.

[2] A right linear grammar for T is: $S \rightarrow$ *William* A, $A \rightarrow$ *whom William* B, $B \rightarrow$ *succeeded* C, $B \rightarrow$ *whom William* B, $C \rightarrow$ *succeeded* C, $C \rightarrow$ *succeeded William*. A language with a right linear grammar is regular (cf. Theorem 2.1 in Volume I).

Let us therefore have a closer look at English $\cap$ T. Intuitively, the only grammatical sentences in T are those for which $n = m$, though some people have the intuition that one may delete occurrences of *succeeded* so that the grammatical sentences in T are those for which $n \geqslant m$. In both cases ($n = m$, $n \geqslant m$), however, the intersection is self-embedding; there is no regular grammar which can generate sets like $\{a^n b^n\}$, or $\{a^n b^m \mid n \geqslant m\}$. The intersection is, therefore, not regular. This contradicts the fact that the intersection should be regular, and hence our only assumption must be wrong, namely that English is a regular language.

Although this form of proof avoids the formal difficulties, the "proof" remains as weak as the empirical observation on which it is based. However, it is upon reaching this level of empirical evidence that one can decide in theoretical linguistics to formulate the state of affairs as an axiom: *natural languages are non-regular*. Given the independent character of a theory (see the preceding chapter), this is a more correct method of work than simply acting as though one were dealing with a *theorem* which could be proven, as linguists often do. The latter method is an incorrect mixture of theory and observation.

2.3. CONTEXT-FREE GRAMMARS FOR NATURAL LANGUAGES

Example 2.2 gives a context-free grammar for (part of) a natural language.

EXAMPLE 2.2. Let $G = (V_N, V_T, P, S)$ be a context-free grammar with $V_N = \{S, NP, VP, D, N, V, A\}$, $V_T = \{$*nice, the, and, congratulate, big, boys, children, little, malicious, girls, tease, defend*$\}$, and the following productions in P:

1. $S \rightarrow NP + VP$
2. $NP \rightarrow NP + and + NP$
3. $NP \rightarrow (D) + (A) + NP$
4. $VP \rightarrow V + NP$
5. $NP \rightarrow N$
6. $N \rightarrow \{$*boys, girls, children*$\}$
7. $V \rightarrow \{$*defend, tease, congratulate*$\}$
8. $A \rightarrow \{$*malicious, nice, big, little*$\}$
9. $D \rightarrow$ *the*

Explanation of the notation: The category symbols stand for usual linguistic quantities, *S* for "sentence", *NP* for "noun phrase", *VP* for "verb phrase", *V* for "verb", *N* for "noun", *D* for "determiner", and *A* for "adjective". The sign + only indicates the concatenation of elements. It is used to avoid typographical indistinctness which could come about when elements are printed directly next to each other. Productions with elements surrounded by parentheses are in fact sets of productions; elements placed between parentheses may be used optionally in derivations. Thus production 3 stands for four productions: $NP \rightarrow D + A + NP$, $NP \rightarrow A + NP$, $NP \rightarrow D + NP$, $NP \rightarrow NP$. Braces indicate that only one of the several elements they surround may be used in a rewrite. Thus, in applying production 6 one may replace the *N* with either *boys* or with *girls*, or with *children*. The rule thus stands for three productions.

Sentences which may be generated by *G* are *boys defend girls*, *the little girls congratulate the big children*, *malicious big children tease nice little girls*, and so forth.[1] A leftmost derivation of *malicious boys and girls tease little children* is as follows (the numbers above the arrows indicate the productions applied):

$S \overset{1}{\Rightarrow} NP + VP \overset{2}{\Rightarrow} NP + and + NP + VP \overset{3,8}{\Rightarrow}$ *malicious* $+ NP +$ *and* $+ NP + VP \overset{5,6}{\Rightarrow}$ *malicious* $+$ *boys* $+$ *and* $+ NP + VP \overset{5,6}{\Rightarrow}$ *malicious* $+$ *boys* $+$ *and* $+$ *girls* $+ VP \overset{4}{\Rightarrow}$ *malicious* $+$ *boys* $+$ *and* $+$ *girls* $+ V + NP \overset{7}{\Rightarrow}$ *malicious* $+$ *boys* $+$ *and* $+$ *girls* $+$ *tease* $+ NP \overset{3,9}{\Rightarrow}$ *malicious* $+$ *boys* $+$ *and* $+$ *girls* $+$ *tease* $+$ *the* $+ A + NP \overset{8,3,5}{\Rightarrow}$ *malicious* $+$ *boys* $+$ *and* $+$ *girls* $+$ *tease* $+$ *the* $+$ *little* $+$ *children*. The *P*-marker for this sentence is given in Figure 2.3.

At this point we can proceed to the discussion of a few attractive qualities of context-free grammars for linguistics, problems of weak generative power, and problems of strong generative power. Much of what will be said here will hold also for context-sensitive grammars.

[1] The example has no pretensions; the grammar can also generate "sentences" like *nice the big boys congratulate girls*.

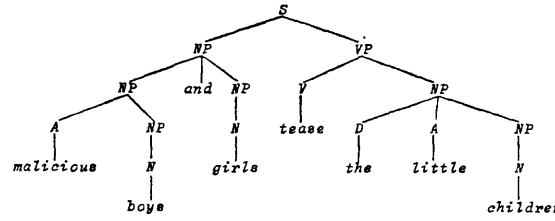


Fig. 2.3. Phrase marker for the sentence *malicious boys and girls tease the little children* (Example 2.2).

2.3.1. *Linguistically Attractive Qualities of Context-free Grammars*

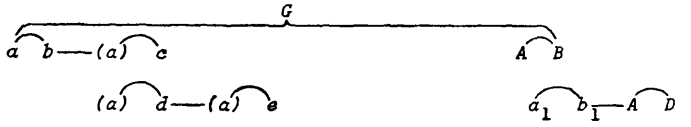
Referring to the discussion in paragraph 2.2, we would first point out that context-free grammars have no difficulties with self-embedding. If a natural language is not regular, it is self-embedding, according to Theorem 2.8 of Volume I. A linguistic rendering of self-embeddingness calls at least for a context-free grammar. *NP* in Example 2.2 is a self-embedding element, as may be seen in Figure 2.3.

But context-free grammars were used as linguistic models in more or less implicit form, long before linguists became aware of the self-embedding property. An important reason for this was the undeniable need of *sentence parsing* in linguistics. Linguists have always been analyzing sentences into phrases. Sentence parts were labelled according to type, and their hierarchical articulation determined the levels of linguistic description. Thus at the lowest level minimal syntactic elements are distinguished which were called morphemes or otherwise. On a somewhat higher level one finds words. An element of higher level is composed of elements of a lower level: a word is composed of morphemes, a phrase is composed of words. A still higher level is found in the traditional distinction between subject and predicate, and so on. Context-free (and context-sensitive) grammars are very well suited to parsing in the form of *levels of labelled syntactic elements*, and we find these ideas in the most diverse linguistic traditions. For a survey of such models in modern English linguistics, we refer the reader to Postal (1964a); the article, although a bit one-sided, shows the "phrase

structure" character of Hockett's linguistics, Lamb's *stratificational syntax*, Pike's *tagmemics*, and a few other theories, including that of the English linguist Halliday. But hierarchical parsing of sentences is a much older tradition, especially in Europe. Take, for example, Jespersen's "analytic syntax", in which parts of sentences are labelled according to function (subject, object, indirect object, etc.), or the important work of Wundt (1900) which is especially interesting for psycholinguistics. We can give an example from the latter work, Wundt's analysis of the following sentence from Goethe's *Wahlverwandtschaften.*

Als er sich aber den Vorwurf sehr zu Herzen zu nehmen schien ($a \frown b$) und immer aufs neue beteuerte (c), dasz er gewisz gern mitteile (d), gern für Freunde tätig sei (e), so empfand sie ($A \frown B$), dasz sie sein zartes Gemüt verletzt habe ($a_1 \frown b_1$), und sie fühlte sich als seine Schuldnerin ($A \frown D$).

Wundt gives the following phrase marker for this:



The *G* stands for *Gesamtvorstellung* or "general image", the psychological equivalent of "sentence". The brace and curves combine lower level elements "apperceptively" into higher level elements. "Apperceptively" means that there is a part-whole relationship between the lower level element and the higher level element. Straight lines indicate that the relationship is "associative", that is, there is no intrinsic relationship of part-to-whole, but only an accidental connection of elements. Notice also that Wundt sometimes puts elements between parentheses. Such elements repeatedly play a grammatical role in the sentence, but are not repeatedly pronounced. We shall return to this phenomenon of deletion, which got a first formalization in Wundt's diagrams.

In this tradition of parsing, the linguistic method of distributional analysis could thrive. Particular attention was paid to finding a

distributional definition of syntactic elements which can play a certain part in sentence structure. This in turn led to distinguishing elementary sentence schemas. The hierarchical relations of inclusion among the labelled syntactic elements in Figure 2.3 give a very satisfying representation of our intuitions concerning the sentence they compose. Finally, we point out that such relations of inclusion make it possible to give justification for some structural ambiguities. The sentence given in Figure 2.3, *malicious boys and girls tease the little children*, is an example of an ambiguous sentence. It is an intuitive datum that *malicious* can refer to *boys and girls* (1), or only to *boys* (2). Even before the formalization of context-free grammars linguists of the "immediate constituent analysis" tradition knew that such ambiguities could be justified by way of inclusion relations. From a formal point of view a sentence is ambiguous, relative to a context-free grammar, when two leftmost derivations of it are possible in that grammar (cf. Volume I, section 2.3.4), and it consequently has two tree-diagrams. In Figure 2.3 we read meaning (1), for *malicious boys* is one noun phrase, and *girls* is another. It is easy to see that the grammar in Example 2.1 also generates the other structure, as given in Figure 2.4. It is quite clear that the correct treatment of structural ambiguities is one of the most important touchstones for the descriptive adequacy of a grammar.

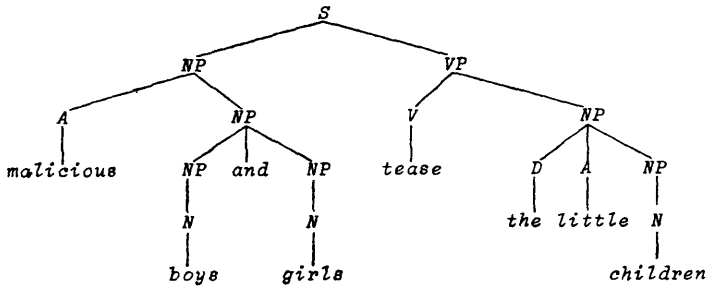


Fig. 2.4. Phrase marker for the sentence *malicious boys and girls tease the little children* (Example 2.2, alternative analysis).

2.3.2. *Weak Generative Power of Context-free Grammars*

Can context-free grammars be observationally adequate, that is, can they generate all and only the sentences of a natural language? Despite contentions of a different tenor in linguistic literature (Postal (1964b)), this question still remains unanswered. Postal "proves" the theorem (his term) that the North American Indian language Mohawk is not context-free by following the argumentation schema of *Syntactic Structures*: (a) a language with the property of "string repetition", as in the language $\{ww\}$ in which w is a string of elements from the terminal vocabulary such that every sentence consists of a string and its repetition, is not context-free. (b) Mohawk has the property of string repetition: there are sentences of the form $s = a_1a_2 \ldots a_nb_1b_2 \ldots b_n$, where a_1 "corresponds" to b_1, a_2 to b_2, and in general a_i corresponds to b_i. Therefore (c) Mohawk is not a context-free language.

This reasoning is as defective as the one, which we criticized, on the proposition that natural languages are not regular. It is erroneous to conclude the non-context-freeness of a language from the existence of non-context-free subsets.

To our knowledge, the literature does not yet contain a correct demonstration of the observational inadequacy of context-free grammars. However, Brandt Corstius (personal communication) recently proposed a proof along the following lines.

The proof is by indirect demonstration. Assume English is context-free. Consider the following regular set: $T = \{$*The academics, accountants, actors, admirals, ... in respectively Belgium, Bulgaria, Burundi, Brasil, ... are respectively calm, candid, canny, careless, ...*$\}$, or abbreviated: $T = \{$*The* a^k, *in respectively* b^m *are respectively* $c^n \mid k, m, n \geqslant 0\}$. It is not difficult to write a right-linear grammar for T. It has been proven by Bar-Hillel (see Hopcroft & Ullman, 1969) that the intersection of a regular language and a context-free language is context-free. Since we assumed English to be context-free it should be the case that $T \cap$ English is context-free. Let us therefore consider which sentences in T are grammatical English sentences. Intuitively,

these are the strings for which $k = m = n \geqslant 1$. However, it is known (see Hopcroft & Ullman, 1969) that there is no possible context-free grammar for the language $\{a^n b^n c^n \mid n \geqslant 1\}$, i.e. the intersection of T and English is not context-free. This contradicts our earlier conclusion, namely that the intersection is in fact context-free. Hence, the only assumption that was made, i.e. that English is context-free must be wrong.

Again this "proof" is as strong as the intuitions about the grammatical subset of T. The *respectively*-construction is rather unnatural. One could probably use Postal's observations for proving non-context-freeness of Mohawk. But Postal is quite cryptic about the grammatical status of strings that do not exactly adhere to string repetition.

Much more convincing, at any rate, are other arguments against the context-free character of natural languages. But they will have to be advanced entirely in terms of strong generative power.

2.3.3. *The Descriptive Inadequacy of Context-free Grammars*

The impossibility for context-free grammars, and for phrase structure grammars in general, to describe a natural language in an intuitively satisfying way has been discussed in great detail in several places (see, for example, Chomsky (1957), and Postal(1964a) and their references). We shall give a short account of a few of the most important arguments here.

(1) A correct representation of the structure of a sentence often, if not always, calls for more than one phrase marker. The identification of structural description and (a single) phrase marker, as is the case for context-free grammars, leaves various intuitive syntactic insights undescribed. A few cases in which there is need of more than one phrase marker are *discontinuities*, *deletions*, and *phenomena of correspondence*.

Discontinuous constituents may be seen in sentences like *John put his coat on*. Intuitively, *put on* belongs together, just as in the nearly synonymous sentence, *John put on his coat*. A context-free grammar gives two different phrase markers, and in the case of the

first sentence *put* and *on* fall into different phrases. The correct word order is thus described, but that is a question of observational adequacy rather than of the intuition that the words belong together. Therefore a *pair* of phrase markers is needed, one of which would group *put* and *on* together (in the same way for both sentences), while the other would give justification for the word order as it is met in fact (different in each sentence). This problem is felt more acutely when one is dealing with languages with freer word orders, such as Latin. An important generalization is lost if for every permutation of words in a sentence a new phrase marker must be made, although the meaning of the sentence does not change essentially because of the permutation.

In the case of deletions we have to do with words or phrases which do function in the sentence, but need not be repeated explicitly. As we have seen, Wundt put such elements between parentheses. This is just another way of showing that more than one phrase marker is involved in the description of the sentence in question, namely, the phrase marker which does contain the elements, and that which does not. The phenomenon of deletion occurs very frequently in coordinative constructions. If we wish adequately to describe the paraphrase relationship between the sentences *John came and auntie came as well* and *John came and auntie as well*, we will have to find some way of making the relationship between *auntie* and *came* explicit, and at the same time we will have to show that *came* does not appear a second time because of the influence of the use of *John came*. Phenomena of coordination will be mentioned separately under (2).

A third general case in which more than one phrase marker seems necessary for the description of a sentence occurs in the representation of correspondence. Compare the sentences *the painters mix the paint* and *the painter mixes the paint*; we see the correspondence here between the number (singular or plural) of the subject and that of the verb. Transgression of the rules of such relations of correspondence leads to ungrammaticality, as may be seen in **the painters mixes the paint* or **the painter mix the paint*.[1]

[1] It is customary to mark non-grammatical sentences with the sign*.

We are obviously dealing here with a very general property of English which should be expressed in the grammar. For this it will be necessary that *painters* be generated as *painter*+*pl*, *painter* as *painter*+*sg*, *mix* as *mix*+*pl*, and *mixes* as *mix*+*sg*. It must also be shown in some way that the morpheme *pl* must be added to *mix* only when the subject *(painter)* appears with *pl*, and the morpheme *sg* must be added to *mix* only when the subject appears with *sg*. In other words, the "underlying" form *mix* is changed to *mix*+*pl* or to *mix*+*sg* under certain conditions elsewhere in the sentence. But this means that *mix* and *mixes* must be described in two ways: on the one hand it must be shown that *mix* is *mix*+*pl* and that *mixes* is *mix*+*sg*, and on the other hand that *pl* or *sg* are not intrinsic to the verb, but rather dependent on a *pl* or *sg* earlier in the sentence.

(2) The description of coordination is a touchstone for every grammar, and therefore also for phrase-structure grammars (for a thorough study of this phenomenon, see Dik, 1968). In example 2.2 we find a context-free description of *NP*-coordination. By production 2 of the grammar, *NP* can be replaced by *NP*+*and*+*NP*. But what will happen when we want to coordinate several *NP*'s? We can apply the production repeatedly, but the hierarchical structure thus obtained would be rather uninforming. The noun phrase *boys and girls and children* will be set out either as *(boys and girls) and children* or as *boys and (girls and children)*. An ambiguity is thus introduced for which there is no intuitive basis: in this and other examples of coordination we prefer to see the elements as ordered really *coordinatively*. We might do this, for example, by making rules like $NP \rightarrow NP+and+NP+and+NP$, but then we would need a new production rule for every new string length. If there is no upper limit to the length of such coordinations, there will be an infinity of such productions. Another solution to the problem is the so-called *rule schema*: $NP \rightarrow NP^n+and+NP$, $n > 0$, by which strings like *boys, girls and children* of indefinite length can be generated. But whatever such rule schemas may be (there is noticeably little known of their mathematical structure relative to formal languages), they are not context-free production

rules. Thus context-free grammars give too much structure in the description of coordination phenomena.

But they also give too little. The phenomenon of deletion which often accompanies coordination is not satisfyingly accounted for by context-free grammars, as we have mentioned under (1). Especially for compound sentences like *Peter plays the guitar daily and John weekly*, a context-free grammar will either generate the deleted element, in which case no account is given for the deletion, or it will not, in which case no account is given for intuitively essential syntactic relations.

(3) Context-free and context-sensitive grammars treat ambiguities correctly only in some cases. Such a case was construed in the grammar of Example 2.2 which was capable of rendering the ambiguity *malicious boys and girls* correctly. There are, however, many cases in which phrase structure grammars fail concerning ambiguities. A few typical examples should make this point clearer. In *Italians like opera as much as Germans*, *Germans* either like or are liked; in *John watched the eating of shrimp*, *shrimp* either eat or are eaten; in *John is the one to help today*, *John* either helps or is helped. In all of these examples it is impossible to represent the ambiguities in an intuitively satisfying way by regrouping the syntactic elements, that is, by assigning alternative phrase markers to the sentences. In such cases a context-free grammar shows too little structure, as we have already seen in the case of deletions.

(4) A context-free grammar will often fall short of an intuitively satisfying representation of the relations *between* sentences. The passive sentence *the house was built by the contractor* is very directly related to the active sentence *the contractor built the house*. It is not clear how a context-free grammar might show that these sentences in important ways are paraphrases of each other. As soon as a similar structure is outlined for both sentences, as would be justified by intuition, account must be given for the fact that the sentences are very different in their elements and word order. To represent such relations, then, it will again be necessary to have a structural description which consists of more than one phrase

marker per sentence. Moreover we cannot write this off as an incidental case, given the generality of the active/passive relationship in English. Other general relations also yield such problems. An English yes/no question which contains an auxiliary verb stands in a simple relationship with the affirmative sentence; compare, for example, *has Peter been joking?* with *Peter has been joking.* In general this concerns a permutation of subject and auxiliary verb. But permutations yield discontinuous constituents, and the related problems for context-free grammars which we have already mentioned. It becomes much more difficult still to imagine a context-free grammar which correctly represents the relationship between the following sentences: *father gave mother roses* and *mother received roses from father.*

These and other kinds of inconveniences have slowly but surely led to the conviction that context-free grammars are descriptively inadequate, whatever their weak generative power may prove to be.

2.4. CONTEXT-SENSITIVE GRAMMARS FOR NATURAL LANGUAGES

For context-sensitive grammars the concept of "structural description" cannot be identified with the "phrase marker", as was the case for context-free grammars. In the first place, it is possible to construct phrase markers only when the grammar exclusively contains context-sensitive production rules (cf. Volume I, paragraph 2.4.1). In the second place, even in the latter case the phrase marker will not represent the derivation unambiguously. The contexts in which the various rewrites took place is especially not expressed. Likewise the sequence of strings obtained in the derivation of the sentence does not show what the contexts were in each step of rewriting. A context-sensitive structural description must, therefore, not only show the sequence of strings, but also the sequence of contexts. Context-sensitive phrase-structure grammars, that is, context-sensitive grammars with context-sensitive production rules, give structural descriptions which can best be defined

as phrase markers, every nonterminal node of which is provided with the context in which it has been generated. This definition of structural description for context-sensitive phrase-structure grammars is used in Example 2.3.

Context-sensitive grammars can resolve some of the problems mentioned in the preceding paragraph, but quite as many new problems appear. Example 2.3 shows how, by the use of a context-sensitive grammar, one can treat the discontinuity which arises when an interrogatory sentence is generated. The example gives a very reduced grammar, developed especially for this problem, and without further pretensions.

EXAMPLE 2.3. Let $G = (V_N, V_T, P, S)$ be a context-sensitive grammar with $V_N = \{S, NP, NP', VP, N, V, V'\}$, $V_T = \{freedom, slavery, is\}$, and the following productions in P:

1. $S \rightarrow NP + VP$
2. $NP \rightarrow N$
3. $VP \rightarrow V + NP$
4. $NP \rightarrow NP' \,/\, -V$
5. $V \rightarrow V' \,/\, NP'-$
6. $NP' \rightarrow V$
7. $V' \rightarrow NP$
8. $N \rightarrow \{freedom, slavery\}$
9. $V \rightarrow is$

This grammar can easily generate the sentence *freedom is slavery*, but it can also generate the interrogatory sentence *is freedom slavery?* This latter is derived as follows: $S \overset{1}{\Rightarrow} NP+VP \overset{3}{\Rightarrow} NP+V+NP \overset{4}{\Rightarrow} NP'+V+NP \overset{5}{\Rightarrow} NP'+V'+NP \overset{6}{\Rightarrow} V+V'+NP \overset{7}{\Rightarrow} V+NP+NP \overset{2,2}{\Rightarrow} V+N+N \overset{8,9}{\Rightarrow} is+freedom+slavery$. Figure 2.5

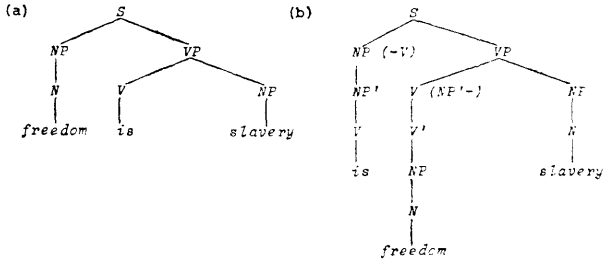


Fig. 2.5. Structural descriptions for the declarative (a) and the interrogative (b) sentences in Example 2.3.

shows the structural descriptions of the two sentences, that is, the phrase markers, to which the rewrite contexts have been added where necessary.

Permutations of elements can be realized with context-sensitive grammars, as may be seen in these illustrations. But it is also clear from the example that this is done in a highly unsatisfying way. The resulting phrase markers are very strange. Thus we see that the interrogative sentence (b) is composed of a *NP* and a *VP*, but that the *NP* is ultimately realized as *is*, and the *VP* as *freedom slavery*, thus in conflict with our dearest intuitions. Context-sensitive grammars supply the need for more than one phrase marker per sentence as badly as do context-free grammars.

Deletions, too, cannot in general be treated by context-sensitive grammars. It is possible, of course, to indicate the context in which a deletion occurs, but this necessarily leads to a type-0 production rule because the string is shortened. Correspondence, on the other hand, can be treated by context-sensitive production rules. There it is simply a matter of adding an element within a given context. Number correspondence, for example, in the sentence *the painter mixes the paint* could be dealt with by a production such as $Num \rightarrow sg/\ NP+sg+V—$, in which *Num* stands for the syntactic category *number*. If we are able to derive the string $NP+sg+V+Num$, application of this production will yield $NP+sg+V+sg$ in which the correspondence of number is realized (a similar argument holds for the plural). This is in fact the method used by Chomsky (1965) in dealing with the question of correspondence. At the suggestion of McCawley (1968b), Peters and Ritchie have proven (1969b) that such a use of context-sensitive production rules does not augment the weak generative power over that of context-free grammars. The advantage lies exclusively in the augmentation of descriptive adequacy.

Coordination and ambiguities yield the same problems for context-sensitive grammars as for context-free grammars (see the preceding paragraph). Some relationships among sentences, such as active/passive or declarative/interrogative, can to a certain extent be handled by context-sensitive grammars, namely, those

concerning permutations and the addition of new elements. But just as in the example given in Figure 2.5, this leads to phrase markers which are in conflict with linguistic intuition.

To resume, nothing is known of the weak generative power of context-sensitive grammars in connection with natural languages, but the descriptive adequacy of context-sensitive grammars is hardly higher than that of context-free grammars. It seems justified to conclude that natural languages fall outside the class of context-sensitive languages, and that type-0 description is required.

2.5. RECURSIVE ENUMERABILITY AND DECIDABILITY OF NATURAL LANGUAGES

The step toward type-0 models for natural languages must not be taken lightly. The most important reason for caution is the decidability or recursiveness of natural languages. In Volume I, Chapter 7 we showed that the class of type-0 languages is equivalent to the class of sets accepted by Turing machines. Thanks to this equivalence, it was possible to show that type-0 languages are recursively enumerable sets (Theorem 7.3 in Volume I). A recursively enumerable language is a language for which a procedure exists to enumerate the sentences of that language, each in a finite number of steps. We have seen, however, that the complement of a type-0 language is not always recursively enumerable, and that consequently it is not generally true that type-0 languages are *decidable* (recursive). There is no algorithm by which a decision may be made, for every string, as to whether or not it belongs to the language. Such algorithms do exist for languages of types 1, 2, and 3.

With the introduction of type-0 grammars, therefore, we run the risk of generating undecidable languages. This, from a linguistic as well as from a psycholinguistic point of view, is a rather unattractive situation. We shall give three reasons for choosing a theory of natural languages in such a way that the languages generated are not only recursively enumerable, but also decidable.

(1) Native speakers will in general be as capable of judging that a sentence belongs to their language, as of judging that that is not the case. In other words, native speakers have an intuitive algorithm for the *recognition* of their language, and not only for *accepting* it. The formalization of this intuitive datum requires that the natural language be decidable in the model. One may object that there are also many unclear cases, for which, in this respect, there are no strong intuitions. But, as was said earlier, it is more elegant to ascribe this to psychological circumstances. The statement does not alter the intuitive fact that a judgment of ungrammaticality is just as direct as a judgment of grammaticality. If on the ground of this objection we drop the recursive enumerability of the complement of the language (the ungrammatical strings), on the ground of the same objection we must also drop the recursive enumerability, and therefore the type-0 character, of the language itself. It is also the case that intuitions of ungrammaticality are *strong*, i.e. the native speaker can often say what makes the string ungrammatical.

(2) A non-decidable language is unlearnable, even if the learner benefits from an informant. For the precise meaning of "learnability" and "informant" we refer the reader to the discussion in Volume I, Chapter 8, paragraph 3. In short this means that there is no algorithm by which an (observationally) adequate grammar can be derived from a sequence of strings marked "grammatical" and "ungrammatical". If there is no learnability in terms of an algorithm, there is certainly no learnability in terms of human cognitive capacities, given the finite character of the latter. The incontrovertible learnability of natural languages pleads that natural languages be considered as decidable sets.

(3) There remains the methodological principle, discussed in paragraph 2.1, that the strongest possible model must be chosen for a natural language. On the basis of this principle, the first step after the rejection of context-sensitive models is the decidable subset of type-0 languages. This is all the more urgent, since, as we have seen, the rejection of recursiveness in natural languages goes hand in hand with the rejection of recursive enumerability.

But to do so would mean to renounce the possibility of writing a generative grammar for the language, and therefore also the possibility of providing every sentence with an explicit parsing. This would come very near abandoning linguistics as a science.

Therefore the rules of the grammar should be chosen in such a way that the decidability of the language is maintained. This limits the choice considerably, and is not easily realized, as we shall see in Chapter 5. Furthermore, in setting phrase-structure grammars aside, we should take care not to "throw the baby away with the bath". The linguistic advantages of such grammars still hold (cf. 2.3.1), and ought, as far as possible, to be taken over into a more adequate theory of natural languages.

3

MIXED MODELS I: THE TRANSFORMATIONAL GRAMMAR IN *ASPECTS*

A transformational grammar is a pair $TG = (B, T)$, in which B is a *base grammar* and T is a set of *transformations*. In general B is a context-free grammar. Transformations are rules which have tree-diagrams as their input and output; when used in conjunction with the base grammar, they can raise the generative power to type-0 level.

Arguments of various kinds are advanced to support the use of this form of grammar in the description of natural languages. We shall mention a few of these arguments. By way of the B-component, the advantages of phrase structure grammars are simply taken up into a more complete linguistic theory. The transformational component T, on the other hand, makes it possible to assign more than one phrase marker to a sentence, and as we have seen in section 2.3.3, there is considerable need of such a possibility. Moreover, the type-0 character of the grammar is thus limited to the replacement of tree-diagrams with other tree-diagrams, allowing the recursiveness of the grammar to be kept under control. Semantic considerations also support the division of a grammar into two components. The semantic interpretation is determined entirely, or at least for the greater part, by the BASE STRUCTURE or DEEP STRUCTURE, that is, the phrase marker generated by B; the morphology of the sentence, on the other hand, can be described better in terms of the output of T, the SURFACE STRUCTURE. Still another argument is provided by the expectation, based on general language theory, that languages will tend to differ with respect to T, and to agree with respect to B, which would be

considered the proper mechanism for the description of UNIVERSALS (the validity of this expectation is the subject of paragraph 3 in Chapter 5).

Transformational grammars differ quite considerably, however, in (i) the choice of base grammar, (ii) the choice of transformations, (iii) the distinction between *B* and *T*, i.e. the degree to which base and transformation rules may be applied "pell mell", (iv) the ratio between the size of *B* and that of *T*: few base rules may call for compensation in many transformation rules, and, within certain limits, vice-versa, and (v) the importance of *B* or *T* for semantic interpretation.

The diversity of transformational grammars, however, does not alter the fact that all of them are *mixed models*, that is, models in which a grammar of limited generative power (not more than type-1) is coupled with a limited set of rules for changing *P*-markers.

Most transformational grammars have evolved from Chomsky's formulation in *Aspects of the Theory of Syntax* (1965) (from this point we shall simply refer to the work as *Aspects*). In the present chapter we shall discuss the model presented in *Aspects*, first informally (3.1), then with a formal treatment of the structure of transformations (3.2). In the final paragraph of the chapter (3.3), we shall briefly discuss how certain considerations, principally semantic in nature, have led to changes in the original model. The changes proposed fall primarily into categories (iii), (iv) and (v) mentioned above. As the results of this are still very temporary, and as this book deals primarily with matters of syntax, our discussion of these points will not be very extensive. In Chapter 4 we shall treat a few alternative proposals concerning (i) and (ii). Those transformational grammars are in many respects very different from the *Aspects* model.

3.1. THE ASPECTS MODEL, AN INFORMAL DISCUSSION

In *Aspects*, a grammar consists of three components, a *syntactic* component, a *phonological* component, and a *semantic* component.

The syntactic component has the recursive qualities necessary for the generation of an infinite set of sentences. The phonological and semantic components describe respectively the aspects of sound and meaning of the structure generated by the syntactic component. Notice that the word "grammar" is used in *Aspects* in the wider sense (see Chapter 1, paragraph 1), including phonology and semantics. Grammar in the narrower sense, the subject of this book, correspond largely to that which is called the syntactic component in *Aspects*; there is complete correspondence when we do not consider morphology.

In this sense, the grammar in *Aspects* is a pair (B, T) of base grammar and transformations. We shall now discuss its principal properties in an informal way.

3.1.1. *The Base Grammar*

The productions of the base grammar are of two kinds: CATEGORIAL RULES and LEXICAL RULES. The categorial component is composed of context-free rewrite rules. They form a grammar with category symbols (S, *NP*, *Pred P*, *VP*, V, N, etc.) as the nonterminal vocabulary, and with grammatical formatives (*sg*, *pl*, *past t*, etc.), a so-called DUMMY SYMBOL Δ, and the boundary symbol $\#$ as the terminal vocabulary. For reasons which will become clear later, every derivation begins with $\#S\#$ instead of simply with S, but as long as there is no chance of confusion we shall omit the boundary symbols. The categorial rules, moreover, have the following two properties: (1) *Recursivity*: S (or actually $\#S\#$) appears in one or more productions to the right of the arrow, so that S can again be introduced into a derivation; there are no other recursive rules in B.[1] (2) The rules of the categorial component are applied in a certain *order*. This is done cyclically: when one arrives at the end of the list, one must start again at the beginning, and if there is an S which has not yet been rewritten, it is first to be

[1] This means that if, for a certain element X_n, the categorial component allows the following derivation $X_n \Rightarrow \omega_1 X_1 \psi_1 \Rightarrow \ldots \Rightarrow \omega_1 \ldots \omega_n X_n \psi_n \ldots \psi_1$, it holds necessarily that $X_i = S$ for some i, $i = 1, \ldots, n$.

dealt with. This ordering is inspired by the so-called "sequential grammars" of Ginsburg and Rice (1962). The restriction on the order of application is formulated in *Aspects*, but not used. Peters (1966) showed that sequential context-free grammars are weakly equivalent to unordered ("ordinary") context-free grammars. We shall ignore this restriction in the further discussion.

Example 3.1 gives part of a base grammar. Like the other examples in this chapter, it is meant only as an illustration. These examples are given to clarify certain properties, and not as serious proposals for a transformational grammar of English.

EXAMPLE 3.1. The base grammar contains nine productions, the nonterminal vocabulary consists of the following elements, *S*, *NP*, *VP*, *V*, *N*, *D*, *Num*, and the terminal vocabulary is made up of the following elements, *it*, *sg*, *pl*, Δ, and *Q* (a "question" morpheme). The productions are:

1. $S \rightarrow (Q)+NP+VP$
2. $VP \rightarrow V+(NP)$
3. $NP \rightarrow (D)+N+Num$
4. $NP \rightarrow it+S$
5. $NP \rightarrow \Delta$
6. $Num \rightarrow \{sg, pl\}$
7. $V \rightarrow \Delta$
8. $N \rightarrow \Delta$
9. $D \rightarrow \Delta$

By these production rules the tree-diagram in Figure 3.1 can be generated. Between parentheses in the diagram morphemes are given which can replace the dummy symbols. The way in which this is done is determined by the lexical rules, which will be discussed later. Let us suppose for the moment that the replacement has

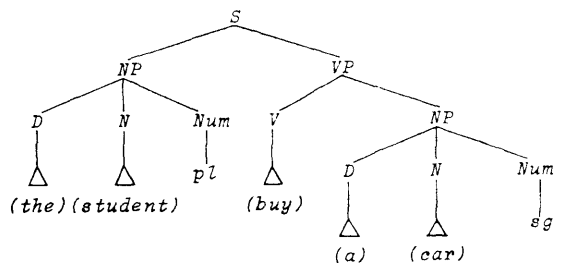


Fig. 3.1. Tree-diagram generated by the categorial rules in Example 3.1.

already taken place. Let us also suppose that the transformational component, applied to this diagram, successfully gives a terminal derivation (see paragraph 1.2 of this chapter). We can then call Figure 3.1 the DEEP STRUCTURE of the sentence *the students buy a car*. (It should be mentioned that no rules are given in this example for dealing with tense. When this enters the discussion in *Aspects*, Chomsky rewrites S to $NP + PredP$, where $PredP$ stands for *predicate phrase*. This latter can in turn be rewritten as $Aux + VP$, and Aux as *pres t*, *past t*, etc., or it can be replaced by an auxiliary verb. The place of an indication of tense in the phrase marker, however, is still very arbitrary.)

In *Aspects* functional relations such as SUBJECT OF, PREDICATE OF, and DIRECT OBJECT OF are defined in terms of categorial properties of deep structures. For this definitions of DIRECT DOMINANCE and GRAMMATICL RELATION are necessary. Let $A \rightarrow \omega B \psi$ be a categorial rule in the base grammar (A and B are category symbols, and ω and ψ are possibly empty strings of terminal and/or nonterminal elements). Suppose that the base rules allow the derivations $\omega \overset{*}{\Rightarrow} \gamma$, $\psi \overset{*}{\Rightarrow} \delta$, and $B \overset{*}{\Rightarrow} \beta$, in which β is a non-empty string of terminal elements and γ and δ are possibly empty strings of terminal elements. In this case $A \overset{*}{\Rightarrow} \gamma\beta\delta = \alpha$ is a terminal derivation. It may be said then that (1) in this derivation A DIRECTLY DOMINATES $\omega\beta\psi$, because $\omega\beta\psi$ is derived from A in only one rewrite, and that (2) β has the GRAMMATICAL RELATION $[B, A]$ to α. In the example given in Figure 3.1, *student* has the grammatical relation $[N, NP]$ to *the student pl*, but *car* has no grammatical relation to *buy a car sg*, for there is no production $VP \rightarrow \omega N \psi$ in the grammar. Chomsky gives the following functional definitions. The grammatical relation $[NP, S]$ is "subject of". In the example, the noun phrase *the student pl (the students)* is the subject of the sentence *the student pl buy a car sg (the students buy a car)*. The relation $[VP, S]$ is "predicate of". Thus in the example, *buy a car* is the predicate of the sentence *the students buy a car*. The relation $[NP, VP]$ is "direct object of". Thus in the example, *a car* is the direct object of *buy a car*. Finally, the relation $[V, VP]$ is "main verb of". In the example, *buy* is the main verb of *buy a car*.

In paragraph 3.3. of the preceding chapter we met the ambiguities concerned precisely with such grammatical relations. *John watched the eating of shrimps*, for example, was ambiguous because *shrimps* could be taken either as the subject or as the direct object. It is possible on the basis of the just given definitions to express these two interpretations. The grammar in Example 3.1 can generate the phrase markers shown in Figure 3.2; they show two different deep structures for *John watched the eating of shrimps*. In Figure 3.2a, *shrimps* is the *subject* of *shrimps eat* according to the definitions, given the relation [*NP*, *S*] within the embedded clause. In Figure 3.2b, *shrimps* is the *direct object* of the embedded clause, given the relation [*NP*, *VP*]. Furthermore, quite in agreement with the intuition, the main clause has *John* as subject and *watched* as main verb, while the noun phrase which contains the subordinate clause is the direct object of the main clause. This representation is

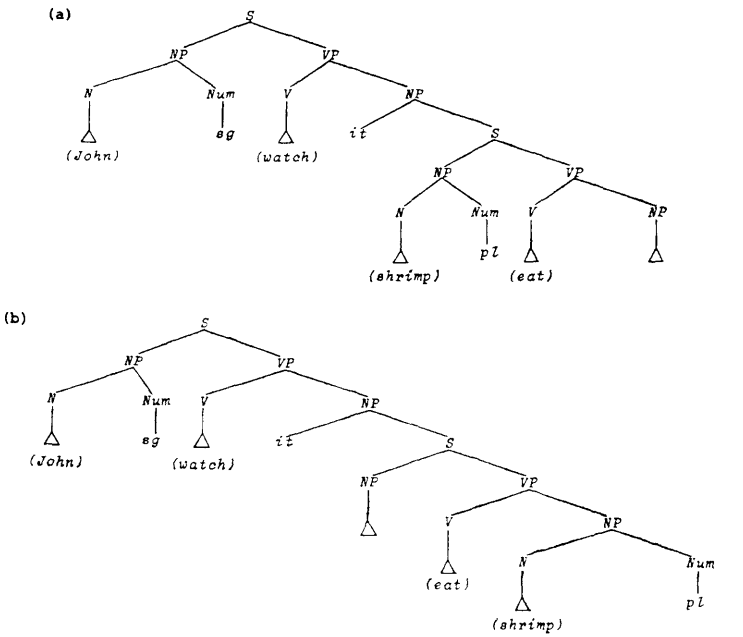


Fig. 3.2. Two deep structures for *John watched the eating of shrimps*.

satisfying up to this point, though it must, of course, be complemented by such transformations of the deep structures that ultimately the same terminal string, *John sg watch sg the eating of shrimp pl*, will be derived for both deep structures. But before going on to the discussion of transformations, we must still treat the lexical rules.

The *lexical rules* replace the dummy symbols with lexical formatives. A lexical formative consists of three "parts": (i) a phonological part, in which the sound properties of the formative are established; for the sake of simplicity we indicate this by *spelling* the morpheme: *shrimp*, *eat*, etc.; (ii) a syntactic part or set of syntactic features to which we return presently; (iii) a semantic part or set of semantic features, which will not be discussed here.

The conditions for replacing the dummy symbol Δ with a given lexical formative are couched in the syntactic features of that formative. When the tree-diagram satisfies these conditions the replacement may be performed.

A first condition for the replacement of a dummy symbol by a formative is that the formative be of the correct lexical category. Consider the sentence *the students buy a car* from Example 3.1, and notice the insertion of the formative *buy*. A condition for the insertion of a formative in that place is that it must be of the category *V*. Thus the dummy symbol in question cannot be replaced by a formative such as *magazine*. In order to exclude such strings as **the students magazine a car* while maintaining the possibility of a sentence like *the students buy a magazine*, the lexicon specifies that *magazine* has the category feature $[+N]$, and *buy* the category feature $[+V]$.

However not all the lexical formatives with the characteristic $[+V]$ may replace the dummy symbol. Thus in the example, the verb *laugh* is excluded, as we see in the ungrammatical string **the students laugh a car*. Obviously *buy* has a characteristic which *laugh* has not: *buy* is a transitive verb, while *laugh* is intransitive. Thus Chomsky distinguishes SUBCATEGORIES within a category; in this case the subcategories are those of transitive and intransitive verbs. Transitiveness and intransitiveness are syntactic features

called (STRICT) SUBCATEGORIZATION FEATURES. Transitiveness can simply and efficiently be denoted as follows [+—*NP*]. This means that a formative with this feature can (+) appear in the place (—) immediately before a noun phrase (*NP*) in the deep structure. It is clear that the dummy symbol above *buy* in Figure 3.1 is in just such a place, and in this respect, therefore, may be replaced by *buy*.

But this still is not sufficient. Beside category and subcategory features, lexical formatives also need SELECTIONAL FEATURES. The verb *doubt*, just as *buy*, has the features [+*V*] and [+—*NP*], but the string **the students doubt a car* is nevertheless ungrammatical. The nature of the object obviously determines the kind of transitive verb which may be selected. Thus *doubt* may not be followed by a physical object like *car*. This may be expressed formally by assigning the selectional feature [− —[+phys.obj.]] to *doubt*. This means that *doubt* cannot (−) occur in the place (—) directly before a phrase which has (+) the property "physical object". The verb *buy* is positive with respect to the same selectional feature.

Thus in the *Aspects* model every lexical formative receives a string of three kinds of features: category, subcategory and selectional features. For *buy*, for example, the string is as follows:

buy: [+*V*], [+—*NP*], [+—[+phys. obj.]], ...

The set of features of a lexical element is called the COMPLEX SYMBOL in *Aspects*, and abbreviated as *C*. The complex symbol of a lexical element contains the conditions under which that element may replace a given dummy symbol.

By way of a number of general rules, the so-called LEXICAL REDUNDANCY RULES, complex symbols can be simplified. Thus a formative with the property [+phys. obj.] is also an *N*. It is sufficient to take only the feature [+phys. obj.] into the complex symbol. A general lexical redundancy rule specifies that all formatives with this feature are at the same time [+*N*]. Much attention is paid to lexical structure in *Aspects*; redundancy rules of various kinds are treated, but we shall not deal with them here.

If lexical insertion is not performed by means of context-free rewrite rules, what kind of grammar is the base grammar B? Chomsky calls lexical insertion a transformation, and thus stated, B is already a transformational grammar. The reason for calling lexical insertion a transformation is that a phrase marker with certain features (specified in the complex symbol) is replaced by another phrase marker (in which Δ is replaced by a lexical formative). Such substitution transformations, however, are completely local operations on the phrase marker. In fact they do not take the weak generative power of the grammar beyond the reach of a context-free grammar. In other words, lexical insertion could also be realized by means of complicated context-free rules (cf. Peters and Ritchie, 1973). We have also seen that the other modification with respect to the ordinary context-free form, namely the ordering of rules, likewise does not lead to raising the generative power of context-free grammars. It holds, therefore, that the base grammar B is weakly equivalent to a context-free grammar; as for the categorial part of the grammar, moreover, there is a high degree of strong equivalence. The output of B consists of tree-diagrams with category symbols as nonterminal nodes and lexical formatives as terminal elements, as well as the special boundary symbol $\#$ and the dummy symbol Δ (not all dummy symbols need be replaced by lexical formatives; remaining dummy symbols can later be transformationally removed). If the transformation rules do not block when such a diagram is presented as input, we call the diagram a DEEP STRUCTURE of the sentence which will later be derived transformationally. The "language" generated by B has the usual notation $L(B)$, and the analyzed language generated by B, i.e. the set of phrase markers, is denoted by $\Lambda(B)$.

3.1.2. *The Transformational Component*

The function of this component is the transformation of deep structures, by way of derived structures, into SURFACE STRUCTURES. Surface structures are tree-diagrams with terminal strings from which sentences of the language can be derived by morphonological

operations. We shall freely call such surface strings *sentences*. It is quite clear that a good deal will be necessary to derive the sentence *John watched the eating of shrimps* from the diagrams in Figure 3.2. Some of the structures generated by B even resist operation by the transformational component, and the transformations are said to *block*. At the end of this paragraph we shall give a more exact description of the conditions under which this occurs. The subset of $A(B)$ for which the transformations do not block is the set of deep structures generated by the transformational grammar. The transformational component, thus, also has the function of *filter*.

The transformational component is a finite ordered sequence of transformations: $T = (T_1, T_2, \ldots, T_k)$. Each transformation T_i consists of two parts: (1) a STRUCTURAL CONDITION which indicates the domain of the transformation. It defines the conditions which the tree-diagram must satisfy if the transformation is to be applied. In particular, one may find in the structural condition the way in which the tree-diagram will have to be subdivided into terms or *factors* (these are parts of the tree-diagram which will be further defined below). As we shall see, the structural conditions also establish other conditions. (2) A *set of* ELEMENTARY TRANSFORMATIONS. Three types of elementary transformation are described in *Aspects*, the elementary *adjunction*, *substitution* and *deletion* of a factor or string of factors. The transformation consists of the simultaneous performance of such elementary operations, once the tree-diagram has been factorized according to (1). The substitution or deletion of a factor is limited by the PRINCIPLE OF RECOVERABILITY OF DELETIONS: when a string of factors disappears, some trace of it must be left behind. This can happen in two ways. One possibility is that a replica of the string of factors is present elsewhere in the derived tree-diagram. Another possibility is that every grammatical category has a finite number of deletable terminal strings, determined in advance; deletion will therefore cause no complete loss of information. For the moment we shall not discuss this principle, but will return to it in the formal treatment of transformations in paragraph 2 of the present chapter, and in Chapter 5.

The principle is of essential importance in determining the generative power of a transformational grammar.

The transformations are applied in order. We speak of a TRANSFORMATIONAL CYCLE as the operation of going through the list of transformations once. The cycle begins with the subsentences most deeply embedded in the deep structure. These are the parts of the tree-diagram which themselves are tree-diagrams with *S* as root, but in which no further *S* occurs. For every "subtree" with *S* as root, the cycle may be applied only if it has already been applied to every subsentence of *S*. The final cycle deals with the "top *S*" of the deep structure. It is therefore said that a transformational derivation works "cyclically from the bottom up".

A very informal example of such a cyclic application is the derivation of *John watched the eating of shrimps* from the deep structure of Figure 3.2b. The first transformational cycle begins with the subtree for Δ *eat shrimp pl*. In going through the list, we remove the Δ, and nominalize *eat shrimp pl* as *eating of shrimp pl*. In the second cycle, which deals with the main clause, *the* is substituted for *it*, and *sg* is adjoined to *V*. The final surface structure is shown in Figure 3.3.

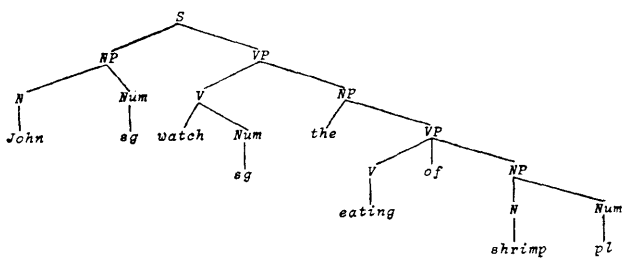


Fig. 3.3. Surface structure for *John watched the eating of shrimps.*

In more (but still in many ways incomplete) detail, we shall now discuss how a transformational grammar might handle a Dutch or German interrogative sentence. In Dutch and German the interrogative is formed by exchanging the positions of subject and (auxiliary) verb. Thus the declarative sentence *de aannemer*

bouwt het huis (*the contractor builds the house*; German: *der Bauunternehmer baut das Haus*) becomes *bouwt de aannemer het huis?* in the interrogative (*does the contractor build the house?*; German: *baut der Bauunternehmer das Haus?*). The Dutch and German interrogative form is especially suitable for explaining some notions which will be needed in the formal analysis of transformations (paragraph 2 of the present chapter).

The base grammar in Example 3.1 can generate a deep structure for the interrogative sentence *bouwt de aannemer het huis?* If we do not take number and congruence of number into consideration, we can accept Figure 3.4 as a representation of this deep structure.

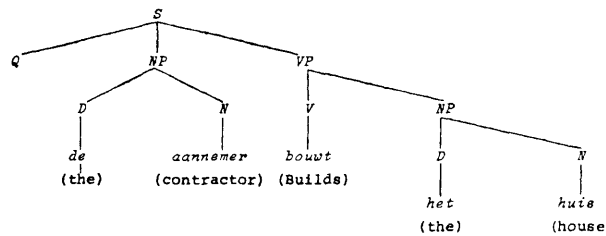


Fig. 3.4. Deep structure of *bouwt de aannemer het huis?* (abbreviated)

The Dutch question transformation T_Q has the factorization Q_1 - NP_2 - V_3 in its structural condition; Q_1, NP_2, and V_3 are single numbered factors. Does the deep structure of Figure 3.4 satisfy this condition? The question is whether the tree-diagram can be subdivided into subtrees in such a way that Q is the root of one subtree, NP is the root of the next subtree to the right, and V is the root of the subtree to the right of that. This is indeed possible; the factorization is represented in Figure 3.5. The tree thus satisfies the structural condition and is correspondingly factorized.

The elementary transformations can now be applied. There is only one of these in our question transformation, the substitution of the V-factor for the Q-factor, or in other words, the third factor comes to take the place of the first. The elementary substitution transformation T_s thus concerns the pair of factors (1,3) for a

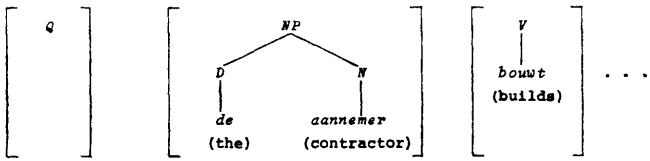


Fig. 3.5. Factorization of the deep structure in Figure 3.4, according to the structural condition of the question transformation.

question transformation. The two parts of the question transformation, the structural condition and the set of elementary transformations, can be summarized in the following notation: $T_Q = (Q_1 - NP_2 - V_3, T_s\ (1,3))$. The regrouping of the factors yields the tree-diagram in Figure 3.6. If no more transformations remain to be performed, this tree-diagram is the surface structure of the sentence.

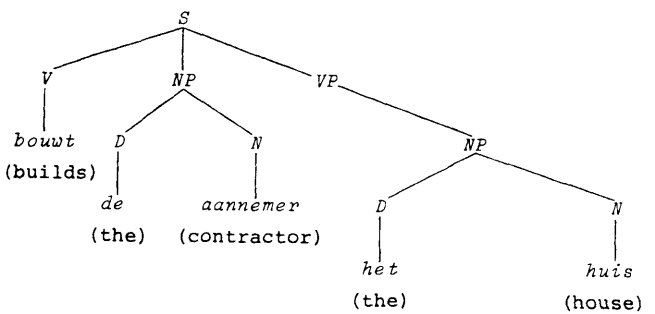


Fig. 3.6. Surface structure for *bouwt de aannemer het huis?*

A complete question transformation for Dutch or German is, of course, more complicated than outlined here. If there is an auxiliary verb, in effect, it is not the main verb, but the auxiliary verb which changes places with the subject; thus the declarative sentence *de aannemer heeft het huis gebouwd (the contractor has built the house)* becomes *heeft de aannemer het huis gebouwd? (has the contractor built the house?)* in the interrogative. There are also other conditions for the question transformation, more difficult to define, not only for Dutch but also for English. Take the dubiously

grammatical sentence, for example, **are you undoubtedly ill?* (the Dutch equivalent, **bent u ongetwijfeld ziek?*, mentioned in Kraak and Klooster (1968), has the same difficulties as the English). Differences of opinion might exist on the advisability of the path $S - VP - NP$ in the diagram in Figure 3.6; one would prefer to eliminate the node with *VP*. Such an operation would be called TREE PRUNING, and can be accomplished, as we shall see in paragraph 2.2. of this chapter, by more formal means than those treated in the present paragraph.

Does the transformation satisfy the principle of recoverability? It does in fact. The *Q* disappears from the tree-diagram, but *Q* is the only element in its category. This case shows clearly what recoverability actually takes in. It means that if the transformation of which a given structure is the output is known (Figure 3.6, for example, is the result of a question transformation), then the input structure (Figure 3.4) can be reconstructed.

A distinction is made between OPTIONAL and OBLIGATORY TRANSFORMATIONS. Obligatory transformations *must* be applied, if at a given point in the cycle its structural conditions are fulfilled. Optional transformations *may* be applied under such circumstances.

We have mentioned above that transformations may act as filters. An example of this is the derivation of a relative clause. Consider the sentence *the postman who brought the letter asked for a signature.* This sentence is derived from (a) *the postman asked for a signature* and (b) *the postman brought the letter.* For the purposes of demonstration it is not very important whether (a) and (b) occur in the deep structure of the sentence in conjunction (linked by *and*) or in the form of an embedded constituent. We opt for the latter possibility, and will proceed to illustrate it. We suppose that the sentence is derived from a deep structure with the following terminal string (irrelevant details are overlooked): *the postman # the postman brought the letter # asked for a signature.* The two boundary symbols occur here because of the rewriting of $\#S\#$ for the embedded sentence; they are mentioned here explicitly because they play a role in the transformation. The structural condition for this relative clause transformation is NP_1 —

$\# - NP_2 - V_3 - NP_4 - \#$, $NP_1 = NP_2$. This means that the tree-diagram must be able to be factorized as indicated, and moreover that the terminal strings of NP_1 and NP_2 are identical. The transformational modification now consists of a number of elementary transformations which yield the following facorization: NP_1 — *who* — V_3 — NP_4, *the postman who brought the letter.* However, there is nothing in the base grammar to prevent the generation of the following terminal string: *the postman # the dustman brought the letter # asked for a signature*, for every grammatical sentence can also be generated as an embedded sentence. This structure, however, is transformationally blocked, because of the identity condition $NP_1 = NP_2$ in the structural condition for the relative clause transformation. If NP_1 = *the postman* and NP_2 = *the dustman*, this condition is not satisfied. A transformational derivation is said to block when there is still one or more boundary symbol in the terminal string at the end of the last transformation cycle. This would be the case with this last example, as the complement transformation would fail. The input structure is "filtered out"; it is not a deep structure.

3.1.3. *Schematic Summary*

Figure 3.7 shows a diagram of the grammar in *Aspects*. It shows that the model generates a deep structure and a surface structure for every sentence in the language. The deep structure contains syntactic information which is necessary and sufficient for a complete semantic interpretation of the sentence. The surface structure gives all syntactic information which is needed for the determination of the morphological and phonological form of the sentence. In the *Aspects* model, these two structures are derived for every sentence in the language, as are all intermediary diagrams which occur in the transformation cycle. The STRUCTURAL DESCRIPTION Σ of a sentence is defined in this model as the pair (δ, ω), where δ is the deep structure, and ω is the surface structure. If, for a given sentence, two or more δ exist, but only one ω, the sentence is said to be DEEP STRUCTURE AMBIGUOUS. An example

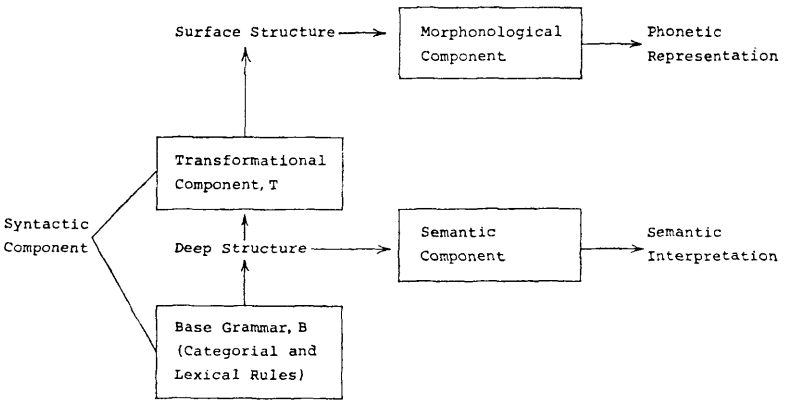


Fig. 3.7. Schema of the *Aspects* model.

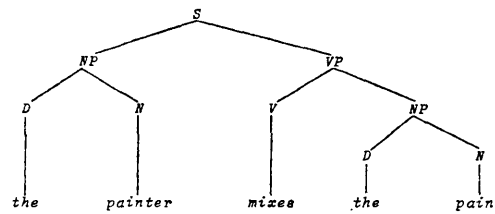
of this is *John watched the eating of shrimps*, which has two deep structures in Figure 3.2, and one surface structure in Figure 3.3. The examples in section 2.3.3 under (3), in which context-free grammars failed to represent ambiguity, are deep structure ambiguous; they could be treated adequately by a transformational grammar. If, for a given sentence there are more than one δ, and also two or more ω, the sentence is said to be SURBACE STRUCTURE AMBIGUOUS. The sentence *malicious boys and girls tease the little children* is an example of this.

3.2. TRANSFORMATIONS, FORMAL TREATMENT

3.2.1. *The Labelled Bracketing Notation*

The input and output of transformations are tree-diagrams. The visual advantage of a two-dimensional tree-diagram is a technical disadvantage when it must figure in a written transformation rule. We would prefer to symbolize transformations, like the production rules of a phrase structure grammar, as rewrites of strings. Consequently, we need a string notation which is isomorphous with the tree notation. The common system for this uses "labelled brackets"

for the representation of tree-diagrams; the notation is therefore called LABELLED BRACKETING NOTATION. An example of a labelled bracketing is given in Figure 3.8. For every constituent of the tree-



$(_S(_{NP}(_D the)_D(_N painter)_N)_{NP}(_{VP}(_V mixes)_V(_{NP}(_D the)_D(_N paint)_N)_{NP})_{VP})_S$

Fig. 3.8 Tree-diagram and labelled bracketing for the sentence *the painter mixes the paint*

diagram there is a pair of brackets, left and right, each of which is labelled according to the syntactic category of the constituent concerned. The representation of a sentence in labelled bracketing notation is called a LABELLED BRACKETING. But not every labelled bracketing concerns a sentence. In the following we wish to use the notion in a very general sense. We define it, therefore, as follows: For a grammar a LABELLED BRACKETING is every finite string of elements from $V_T \cup V_N \cup L \cup R$, where L is the set of labelled left brackets, $L = \{(_A, A \in V_N\}$, and R is the set of labelled right brackets, $R = \{)_A, A \in V_N\}$. We state without proof that for every tree-diagram in a grammar (see definition in Volume I, section 2.2), there is one and only one labelled bracketing. The inverse does not hold. Every labelled bracketing which corresponds to a tree-diagram is WELL-FORMED. Labelled bracketings which are not well-formed would be, for example, $(_S a)_V$, $(_S a(_S$, $(_S(_V a)_V$, and so forth. For grammars one can also define the concept directly as follows. (In the rest of this chapter we shall number the most important definitions to facilitate reference.)

DEFINITION 3.1. A WELL-FORMED LABELLED BRACKETING is every string ω over $V_N \cup V_T \cup L \cup R$, for which either

(1) $\omega \in V_N \cup V_T$ or,
(2) $\omega = ({}_A\psi)_A$ or,
(3) $\omega = \psi\varphi$

where ψ and φ are well-formed labelled bracketings.

(This is called a *recursive definition*; note that although the concept itself is used in the definition, the latter is not tautological.)

A well-formed labelled bracketing is said to be CONNECTED in cases (1) and (2). Thus $({}_S a)_S({}_{NP}({}_N a)_N)_{NP}$ is a well-formed labelled bracketing which is not connected, while $({}_{NP}({}_D a)_D({}_N a)_N)_{NP}$ is connected and consequently also well-formed. A TERMINAL LABELLED BRACKETING is a labelled bracketing with elements exclusively from $V_T \cup L \cup R$.

In order to speak of the terminal string of a tree-diagram, we must be able to remove the brackets. We must, therefore, define the *debracketing function.*

DEFIUITION 3.2. The DEBRACKETIZATION $d[\omega]$ of the labelled bracketing ω is the string which remains when all labelled brackets are removed from ω;

Thus $d[({}_{NP}({}_D a)_D({}_N a)_N)_{NP}] = aa$.

3.2.2. *A General Definition of Transformations*

The replacement of tree-diagram with tree-diagram in diagram notation becomes the replacement of connected well-formed labelled bracketing with connected well-formed labelled bracketing in labelled bracketing notation. For the general definition of transformations, which is much broader than the definition given in *Aspects* which will be formalized in paragraph 3.2.4 of this chapter, we shall deal only with the rewriting of terminal labelled bracketings. This is in complete agreement with the linguistic use of transformations. (Notice that the deep structure of a sentence corresponds to a terminal labelled bracketing.)

Before presenting the definition, we must first treat two questions. In the first place we must realize that transformations are

not ordinary rewrite rules, but *rule schemas*. We have seen rule schemas already, such as $NP \rightarrow NP^n + and + NP$, $n > 0$, in section 2.3.3. A rule schema stands for a possibly infinite set of rewrite rules. Many structural conditions are of this sort. For the Dutch question transformation, we found the condition (much simplified) $Q_1 - NP_2 - V_3$. Every tree-diagram which fulfills this condition lies in the domain of the question transformation. If the grammar generates an indefinite number of noun phrases, there is an indefinite number of tree-diagrams which satisfy this condition. The question transformation is a summary of an infinity of rewrite rules over terminal labelled bracketings. A transformation, thus, indicates how a *set* of terminal labelled bracketings can be rewritten. Let us call such a set a TREE TYPE. The definition of transformations must therefore show that tree types are rewritten as tree types. The fact that transformations are rule schemas is a direct consequence of the linguistic usage of applying transformations to *complete* tree-diagrams. If it were permitted to apply transformations to incomplete tree-diagrams (cf. Figure 2.2 in Volume I), that is, before a terminal derivation is obtained, it would not be necessary to define transformations over *terminal* labelled bracketings. As the tree can usually be completed in various ways, transformations must be schemas.

In the second place, it can occur in linguistics that a particular transformation is applicable in more than one place in the tree-diagram. Suppose that we have a structure which can be factorized as $A-B-A-B-A$, and a transformation whose structural condition is the factorization $A-B-A$. In such a case the transformation could be applied either to the first three factors or to the last three, possibly with differing results. This, however, will rarely be the case in linguistics, especially since every transformational cycle concerns only a strongly limited domain in the tree-diagram. On the other hand it does happen that the structural condition is satisfied in two different places in the tree-diagram, without overlapping (in practice this occurs particularly in phonology; cf. Chomsky and Halle (1968)). With the condition $A-B-A$, we see this in a factorization such as $A-B-A-X-A-B-A$, where X is an

arbitrary string of factors. In general, then, a transformation is a *nondeterministic rule*. It transforms a given tree type into a finite set of tree types. This is entirely analogous to the transition rules of non-deterministic automata (cf. Volume I, sections 4.2, 5.2, and 6.1).

Suppose that $W(V_N, V_T)$ is the set of terminal connected well-formed labelled bracketings over nonterminal vocabulary V_N and terminal vocabulary V_T. W is then a set of terminal (complete) tree-diagrams. Let w stand for a possibly infinite subset of W; thus $w \subset W$, and w is a tree type. Let us indicate any finite set of tree types by f. The output of a transformation, as we have just seen, must be such a finite set. The entire set of such finite sets f over $W(V_N, V_T)$ is noted as $F(W(V_N, V_T))$, or simply $F(W)$. This represents "the set of finite sets of tree types". Transformations, then, can be defined as follows:

DEFINITION 3.3. A TRANSFORMATION over (V_N, V_T) is a pair (w, f), where w is a subset of $W(V_N, V_T)$, and f is a subset of $F(W)$.

Equivalent formulations of this are: A transformation maps a subset of W into the subsets of F, and: A transformation T is a subset of the cartesian product of W and F, $T \subset W \times F$.

One way to write a transformation is in the form $w \rightarrow f$, just like the notation for production rules. (Notice that this notation differs from the informal notation given in the preceding paragraph. We shall return to his subject in paragraph 2.4 of this chapter.) Thus, the Dutch-German question transformation can be written as:

$$T_Q\colon ({}_S Q({}_{NP} X)_{NP}({}_{VP}({}_V Y)_V R)_{VP} U)_S \rightarrow \{({}_S({}_V Y)_V({}_{NP} X)_{NP} R U)_S\}$$

The subset of W appears before the arrow. The variables X, Y, R, and U stand for well-formed labelled bracketings, and R and U are possibly empty. Because in principle, for each of these variables, an infinity of terminal labelled bracketings can be chosen, the terms to the left of the arrow stand for an infinite set of terminal trees. Notice also that the term to the left is connected: if U is well-formed, then ${}_S$ necessarily corresponds to $)_S$. The variables them-

selves need not be connected. Thus X can stand for $(_N John)_N and$ $(_N Peter)_N$, where the leftmost $(_N$ does not correspond to the rightmost $)_N$. The term to the right of the arrow is a finite set with only one element. That element stands for a tree *type*, and thus for a (possibly infinite) set of terminal trees. Its variables (Y, X, R, and U) mean that if a given terminal labelled bracketing is chosen for the term at the left, the same labelled bracketing must be chosen for the same variable in the term at the right.

Let us show that T_Q is applicable to the terminal connected labelled bracketing

$$(_S Q(_{NP}(_D de)_D(_N aannemer)_N)_{NP}(_{VP}(_V bouwt)_V(_{NP}(_D het)_D(_N huis)_N)_{NP})_{VP})_S$$

The variables here have the following values: $X = (_D de)_D(_N aannemer)_N$, $Y = bouwt$, $R = (_{NP}(_D het)_D(_N huis)_N)_{NP}$, and $U = \lambda$. The transformation changes the labelled bracketing to

$$(_S(_V bouwt)_V(_{NP}(_D de)_D(_N aannemer)_N)_{NP}(_{NP}(_D het)_D(_N huis)_N)_{NP})_S.$$

By drawing the tree-diagram for this, the reader will see that tree pruning has taken place, that is, the superfluous VP node in Figure 3.6 has been removed. If we stipulate in the grammar that auxiliaries belong to category V, then the question transformation given here will also provide the correct solution for Dutch and German sentences with auxiliary verbs. The main verb will be found in factor R, and will remain in place during the transformation. Finally, let us point out that the debracketization of this labelled bracketing is precisely the sentence *bouwt de aannemer het huis?*

3.2.3. *The Interfacing of Context-free Grammars and Transformations*

Before returning to the formal facets of transformations in the *Aspects* model (paragraph 3.2.4), we shall first show that it is possible, by the use of the debracketing function d, to give a very simple representation of a transformational grammar.

Let G be a context-free grammar. The language generated by G

is $L(G)$, and the analyzed language is $\Lambda(G)$. It is obvious that $L(G)$ is obtained by debracketing the elements of $\Lambda(G)$.

It is possible to write a grammar G' in such a way that $L(G') = \Lambda(G)$. The sentences of $L(G')$ will be precisely the structural descriptions of the sentences generated by G. This may be seen in the following. Take $G = (V_N, V_T, S, P)$. $G' = (V_N, V'_T, S, P')$ is constructed as follows.

(i) $V'_T = V_T \cup L \cup R$, in which $L = \{(_A \mid A \in V_N\}$ and $R = \{)_A \mid A \in V_N\}$. Thus the sets of labelled left and right brackets are added.

(ii) For every production $A \to \alpha$ in P, P' will contain a production $A \to (_A\alpha)_A$.

We shall illustrate, by way of an example, and without proof, that if G' is thus constructed, $L(G') = \Lambda(G)$.

EXAMPLE 3.2. Let G have the productions listed below in column (1), and G' the productions listed in column (2).

(1)	(2)
$S \to NP+VP$	$S \to (_S NP+VP)_S$
$VP \to V+NP$	$VP \to (_{VP}V+NP)_{VP}$
$NP \to D+N$	$NP \to (_{NP}D+N)_{NP}$
$D \to$ *the*	$D \to (_D$*the*$)_D$
$N \to \{$*people, animals*$\}$	$N \to \{(_N$*people*$)_N$, $(_N$*animals*$)_N\}$
$V \to$ *help*	$V \to (_V$*help*$)_V$

It is not difficult to derive the sentence *the people help the animals* from grammar G. If the corresponding production rules of G' are applied in the same order, we obtain

$$(_S(_{NP}(_D\textit{the})_D(_N\textit{people})_N)_{NP}(_{VP}(_V\textit{help})_V(_{NP}(_D\textit{the})_D(_N\textit{animals})_N)_{NP})_{VP})_S$$

as may easily be verified. This sentence in $L(G')$ is precisely the structural description of *the people help the animals* in $L(G)$. If x' is a sentence in $L(G')$, then $x = d(x')$, the debracketization of x', is a sentence in $L(G)$.

In this way, a transformational grammar $TG = (B, T)$ such as in *Aspects*, with a context-free base grammar, can now simply be considered as a triad (B', T, d), in which B' is the context-free grammar which generates as its sentences the structural descrip-

tions of the sentences generated by B. The transformational component T, will then indicate how such *sentences* (and not tree-diagrams) are to be rewritten. In this case, transformations will replace strings with strings. If a transformation replaces a sentence with a shorter string, we are dealing with a type-0 rule (which is neither of type-1 nor of type-2). Finally, the debracketing function, d, acts to remove the brackets after application of the transformations. It still holds, however, even for this (B', T, d) model, that the transformations are not ordinary type-0 rules, but rule schemas. Unfortunately, little is known of the generative power of rule schemas, and of their place (or lack of it) in the hierarchy of grammars.

3.2.4. *The Structure of Transformations in Aspects*

The general definition of transformations (Definition 3.3) includes much more than what is used in *Aspects*, and more than is necessary on empirical linguistic grounds. Every substitution of a tree-diagram for a tree-diagram is included in the general definition of transformation, but in paragraph 3.1.2 we saw that in *Aspects* only three elementary transformations were admitted: adjunction, substitution, and deletion of a factor or string of factors, and this within the limitations of the principle of recoverability. The formulation of this in *Aspects* is quite informal, however, and it is impossible to see precisely what can be done with transformations as long as a much more precise definition is not given. Peters and Ritchie (1973) were the first to perform a formalization of the *Aspects* model, and the results they obtained were surprising, as we shall see in Chapter 5. Without attempting to be exhaustive, we shall present the essence of this formalization. In order to be clear and concise in this, we shall first introduce the concept of *elementary factorization*, which was not used by Peters and Ritchie.

DEFINITION 3.4. The ELEMENTARY FACTORIZATION of a terminal labelled bracketing φ is the ordered set of p elementary factors $[\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_p]$, such that $\varphi = \varepsilon_1\ \varepsilon_2 \ldots \varepsilon_p$, where

(i) φ is a component of a connected terminal labelled bracketing (cf. Definition 3.1), and where ELEMENTARY FACTOR is defined in (ii) and (iii):
(ii) ε_i contains one and only one terminal element;
(iii) the leftmost symbol of ε_i is not a right bracket, and the rightmost symbol of ε_i is not a left bracket.

In this way, φ is divided into the smallest possible "terminal" factors, and the boundaries between factors are precisely the phrase boundaries. Thus, the elementary factorization of $\varphi = (_S(_{NP}(_D\mathit{the})_D (_N\mathit{people})_N)_{NP}(_{VP}(_V\mathit{help})_V(_{NP}(_D\mathit{the})_D(_N\mathit{animals})_N)_{NP})_{VP})_S$ is $[(_S(_{NP}(_D\mathit{the})_D, (_N\mathit{people})_N)_{NP}, (_{VP}(_V\mathit{help})_V, (_{NP}(_D\mathit{the})_D, (_N\mathit{animals})_N)_{NP})_{VP})_S]$. In this example, $\varepsilon_2 = (_N\mathit{people})_N)_{NP}$. Notice that not every labelled bracketing has an elementary factorization. This is the case, for example, for labelled bracketings which contain no terminal elements.

DEFINITION 3.5. A FACTORIZATION of a terminal labelled bracketing φ is defined if φ has an elementary factorization $[\varepsilon_1, \varepsilon_2 \ldots, \varepsilon_n]$. A factorization then is an ordered set of m FACTORS $[\psi_1, \psi_2, \ldots, \psi_m]$, such that $\varphi = \psi_1\psi_2 \ldots \psi_m$, in which $\psi_1 = \varepsilon_1\varepsilon_2 \ldots \varepsilon_i$, $\psi_2 = \varepsilon_{i+1}\varepsilon_{i+2} \ldots \varepsilon_j, \ldots, \psi_m = \varepsilon_k\varepsilon_{k+1} \ldots \varepsilon_{n-1}\varepsilon_n$. In other words, a factorization is a partition of an elementary factorization.

The example given with the preceding definition allows the following factorization, inter alia:

$[(_S(_{NP}(_D\mathit{the})_D, (_N\mathit{people})_N)_{NP}(_{VP}(_V\mathit{help})_V,$
$(_{NP}(_D\mathit{the})_D(_N\mathit{animals})_N)_{NP})_{VP})_S]$

In this factorization, $\psi_2 = \varepsilon_2\varepsilon_3 = (_N\mathit{people})_N)_{NP}(_{VP}(_V\mathit{help})_V$. Another factorization of the same labelled bracketing is

$[(_S(_{NP}(_D\mathit{the})_D(_N\mathit{people})_N)_{NP}(_{VP}(_V\mathit{help})_V,$
$(_{NP}(_D\mathit{the})_D(_N\mathit{animals})_N)_{NP})_{VP})_S]$

Here $\psi_2 = \varepsilon_4\ \varepsilon_5 = (_{NP}(_D\mathit{the})_D(_N\mathit{animals})_N)_{NP})_{VP})_S$.

We can now try to define a very special factorization of a terminal labelled bracketing φ. The factorization should on the one hand not cut through connected well-formed substrings in φ. It should be remembered that these are strings which are either surrounded by corresponding brackets, or strings consisting of just one terminal element (cf. Definition 3.1). This condition means that each connected substring of φ is in its entirety part of a factor in the factorization. On the other hand, the factorization should be as fine as possible, i.e. contain as many factors as possible. As an example, let us consider the case where $\varphi = (_{NP}(_{D}the)_{D}(_{N}people)_{N})_{NP}(_{VP}(_{V}help)_{V}(_{NP}(_{D}the)_{D}(_{N}animals)_{N})_{NP}$. This φ has an elementary factorization since φ is a part of a well-formed terminal labelled bracketing. The elementary factorization is (with numbering): $[\varepsilon_1 = (_{NP}(_{D}the)_{D}$, $\varepsilon_2 = (_{N}people)_{N})_{NP}$, $\varepsilon_3 = (_{VP}(_{V}help)_{V}$, $\varepsilon_4 = (_{NP}(_{D}the)_{D}$, $\varepsilon_5 = (_{N}animals)_{N})_{NP}]$. There is only one way to factorize φ in such a way that, on the one hand, each connected labelled bracketing, also the largest, is part of a factor, and, on the other hand, there is a maximum number of such factors. That is the factorization $[\psi_1, \psi_2, \psi_3]$ in which $\psi_1 = \varepsilon_1\varepsilon_2$, $\psi_2 = \varepsilon_3$, $\psi_3 = \varepsilon_4\varepsilon_5$: $[\psi_1 = (_{NP}(_{D}the)_{D}(_{N}people)_{N})_{NP}$, $\psi_2 = (_{VP}(_{V}help)_{V}$, $\psi_3 = (_{NP}(_{D}the)_{D}(_{N}animals)_{N})_{NP}]$. Such a factorization of φ is called the *unique factorization* of φ. (See the more detailed treatment of the notion "standard factorization" in Peters and Ritchie, 1973.) A broad definition of this will be sufficient here.

DEFINITION 3.6. The UNIQUE FACTORIZATION of a terminal labelled bracketing φ (defined if φ has an elementary labelled bracketing) is the factorization in which
(i) every substring of φ which is a connected well-formed labelled bracketing is as a whole a part of a factor;
(ii) the factorization is the most minute for which (i) holds, i.e. of all factorizations which fall under (i) the unique factorization counts the largest number of factors.

We offer a few examples of unique factorizations (2) of labelled bracketings (1).

	(1) Labelled Bracketing	(2) Unique Factorization
φ_1 =	$(_{NP}(_D the)_D(_N people)_N$	$[(_{NP}(_D the)_D, (_N people)_N]$
φ_2 =	$(_{NP}(_D the)_D(_N people)_N)_{NP}$	$[(_{NP}(_D the)_D(_N people)_N)_{NP}]$
φ_3 =	$(_{VP}(_V help)_V(_{NP}(_D the)_D(_N animals)_N)_{NP})_{VP}$	$[(_{VP}(_V help)_V(_{NP}(_D the)_D(_N animals)_N)_{NP})_{VP}]$
φ_4 =	$(_{VP}(_V help)_V(_{NP}(_D the)_D(_N animals)_N)_{NP}$	$[(_{VP}(_V help)_V, (_{NP}(_D the)_D(_N animals)_N)_{NP}]$
φ_5 =	$(_{VP}(_V help)_V(_{NP}(_D the)_D(_N animals)_N$	$[(_{VP}(_V help)_V, (_{NP}(_D the)_D, (_N animals)_N]$

In the unique factorization of φ, as we have pointed out, every connected labelled bracketing in φ is part of a factor. The first example in column (1), $\varphi_1 = (_{NP}(_D the)_D(_N people)_N$, has the following connected parts: *the*, $(_D the)_D$, *people*, and $(_N people)_N$. Such a connected part is the same as that which we have called a subtree in paragraph 3.1.2 of this chapter. Each of these parts appears uncut in one of the factors in column (2). The *interior* of a factor is defined as the largest connected part of that factor.

DEFINITION 3.7. The INTERIOR $I(\psi)$ of a factor ψ in a unique factorization is the largest connected labelled bracketing in that factor.

The interior of $(_{NP}(_D the)_D$ is not *the*, but $(_D the)_D$; that of $\psi = (_N people)_N$ is not *people*, but $I(\psi) = (_N people)_N$. The interior of φ_2 in column (1) is the labelled bracketing itself. Notice that every factor of a unique factorization has an interior, for every factor contains at least one terminal element. If there is no greater connected unity, that element is the interior. This definition leads directly to the following.

DEFINITION 3.8. The LEFT-HAND EXTERIOR $E_l(\psi)$ of a factor in a unique factorization is the part of the factor to the left of the interior; the RIGHT-HAND EXTERIOR $E_r(\psi)$ of a factor in a unique factorization is the part of the factor to the right of the interior.

The left-hand exterior $E_l(\psi)$ of $(_{NP}(_D the)_D$ is $(_{NP}$, the righthand exterior $E_r(\psi)$ is λ, because the interior is $(_D the)_D$. The lefthand exterior of a factor such as $animals)_N)_{NP})_{VP})_S$ is λ and the right-

hand exterior is $)_N)_{NP})_{VP})_S$, because *animals* is the interior. The exterior thus consists of the labelled brackets which remain after the interior is removed.

We have just seen that for $\psi = (_N people)_N$, $I(\psi) = (_N people)_N$. This interior has the general form $(_{A_1}(_{A_2}\ldots(_{A_m}\omega)_{A_m}\ldots)_{A_2})_{A_1}$, where ω contains no corresponding exterior brackets. In this example $m = 1$, $A_1 = N$, and $\omega = people$. We call ω the KERNEL of $I(\psi)$, denotated by $K(\psi)$. The kernel of $(_{NP}(_D the)_D(_N people)_N)_{NP}$ is $(_D the)_D$ $(_N people)_N$, in which $(_D$ and $)_N$ are not corresponding brackets. The kernel of $(_{NP}(_N people)_N)_{NP}$ is *people*. If the kernel is removed, that which remains to the left and to the right of it will be called respectively $U_l(\psi)$ and $U_r(\psi)$. Thus for $\psi = (_{NP}(_D the)_D(_N people)_N)_{NP}$, $I(\psi) = \psi$, $K(\psi) = (_D the)_D$ $(_N people)_N$, $U_l(\psi) = (_{NP}$ and $U_r(\psi) =)_{NP}$. For $(_{NP}(_N people)_N)_{NP}$, $U_l(\psi) = (_{NP}(_N$ and $U_r(\psi) =)_N)_{NP}$. U_l and U_r always form a symmetric pair. Summing up:

$$\psi = E_l(\psi)I(\psi)E_r(\psi) = E_l(\psi)U_l(\psi)K(\psi)U_r(\psi)E_r(\psi).$$

We shall now define the *content* of a unique factorization as the string of interiors of the factors.

DEFINITION 3.9. The CONTENT $C(\varphi)$ of φ, given the unique factorization $[\psi_1,\ldots,\psi_n]$ of φ, is the string $I(\psi_1)$ $I(\psi_2)\ldots I(\psi_n)$, where $I(\psi_i)$ is the interior of ψ_i.
The content is thus defined only if φ has a unique factorization.

Once again our examples are taken from the labelled bracketings in column (1) on page 69. Their contents are given in column (3).

(3) Content
$C(\varphi_1) = (_D the)_D(_N people)_N$
$C(\varphi_2) = (_{NP}(_D the)_D(_N people)_N)_{NP}$
$C(\varphi_3) = (_{VP}(_V help)_V(_{NP}(_D the)_D(_N animals)_N)_{NP})_{VP}$
$C(\varphi_4) = (_V help)_V(_{NP}(_D the)_D(_N animals)_N)_{NP}$
$C(\varphi_5) = (_V help)_V(_D the)_D(_N animals)_N$

The content of a connected labelled bracketing is the labelled bracketing itself, as is the case for φ_2 and φ_3 in column (3).

Just as the content is defined as a string of interiors, we define the REST, $R(\varphi)$, as the string of exteriors of a unique factorization

which remains after the interiors have been removed; thus $R(\varphi_1) = (_{NP}$, $R(\varphi_2) = \lambda$, $R(\varphi_3) = \lambda$, $R(\varphi_4) = (_{VP}$, and $R(\varphi_5) = (_{VP}(_{NP}$.

We are now able to define the elementary transformations of deletion, substitution and adjunction.

DEFINITION 3.10. The ELEMENTARY DELETION of a labelled bracketing φ, $T_d(\varphi)$, is defined as $R(\varphi)$.

The deletion of φ is thus that which remains after the content of φ has been removed. $T_d(\varphi)$, then, can only be defined if $T_d(\varphi)$ has a content. Examples of this (with reference to column (1)) are: $T_d(\varphi_1) = R(\varphi_1) = (_{NP}$, $T_d(\varphi_2) = R(\varphi_2) = \lambda$, and so forth.

DEFINITION 3.11. The ELEMENTARY SUBSTITUTION $T_s(\psi, \varphi)$ is the replacement of the interior of ψ with the content of φ, thus $T_s(\psi, \varphi) = E_l(\psi)C(\varphi)E_r(\psi)$.
Substitution is defined only if ψ has an interior, that is, if it is a factor of the unique factorization of a labelled bracketing, and if φ has a content, that is, if it itself has a unique factorization.

Take, for example, $\psi = (_N people)_N)_{NP}$ and $\varphi = (_{VP}(_V help)_V(_{NP}(_D the)_D\ (_N animals)_N$. Here $E_l(\psi) = \lambda$, $E_r(\psi) =)_{NP}$, and $C(\varphi) = (_V help)_V(_D the)_D(_N animals)_N$. Therefore $T_s(\psi,\varphi) = (_V help)_V(_D the)_D\ (_N animals)_N)_{NP}$.

DEFINITION 3.12. The ELEMENTARY LEFT-ADJUNCTION $T_l(\psi, \varphi)$ is defined as $E_l(\psi)U_l(\psi)C(\varphi)K(\psi)U_r(\psi)E_r(\psi)$. The ELEMENTARY RIGHT-ADJUNCTION $T_r(\psi, \varphi)$ is defined as $E_l(\psi)U_l(\psi)K(\psi)C(\varphi)U_r(\psi)E_r(\psi)$. The conditions on ψ and φ for T_l and T_r are the same as in the preceding definition.

As an example of *elementary right-adjunction* we construct the following. Let $\varphi = (_{PP}(_{Prep} in)_{Prep}(_{NP}(_N Norway)_N)_{NP})_{PP})_{NP}$, in which *PP* stands for "prepositional phrase" and *Prep* for "preposition", and $\psi = (_S(_{NP}(_D the)_D(_N people)_N)_{NP}$. We then have the following values for the various terms of the transformation: $E_l(\psi) = (_S$, $E_r(\psi) = \lambda$, $U_l(\psi) = (_{NP}$, $U_r(\psi) =)_{NP}$, $K(\psi) = (_D the)_D(_N people)_N$, and $C(\varphi) = (_{PP}(_{Prep} in)_{Prep}(_{NP}(_N Norway)_N)_{NP})_{PP}$. Then $T_r(\psi, \varphi) = (_S(_{NP}(_D the)_D(_N people)_N(_{PP}(_{Prep} in)_{Prep}(_{NP}(_N Norway)_N)_{NP})_{PP})_{NP}$.

Finally, a convention introduced by Peters and Ritchie, the REDUCTION CONVENTION, should also be mentioned in this connection. One of the two following cases can occur as part of a labelled bracketing, either through peculiarities of the base grammar, or through the transformations.

(i) $(_A\lambda)_A$, where λ is the null-string. This could occur, for example, through a deletion transformation.

(ii) $(_{A_1}(_{A_2}\cdots(_{A_n}(_{A_1}\omega)_{A_1})_{A_n}\cdots)_{A_2})_{A_1}$, where ω is a well-formed labelled bracketing. This is called the *nesting* of A_1 in A_1.

In (i) a labelled bracketing is obtained which is not well-formed (cf. Definition 3.1), and in (ii) the labelled bracketing is redundant, because it is said twice that ω belongs to category A_1. The reduction convention states that substrings of type (i) are to be removed as soon as they occur, and that the interior pair of brackets $(_{A_1},)_{A_1}$ are to be removed when cases of type (ii) occur. Since this is a general convention concerning labelled bracketings, we shall not specifically write "reduced labelled bracketing" when the reduction has taken place. We shall omit the adjective "reduced", for by convention every labelled bracketing is reduced.

Every transformation, such as the question transformation and the complement transformation, is presented as a combination of elementary transformations. For a complete definition of transformations according to the *Aspects* model, two matters must still be worked out: in the first place the manner in which elementary transformations are combined into such a transformation for a given labelled bracketing, and in the second place, the conditions on which the transformation may be applied, that is, the structural condition which the labelled bracketing must satisfy and the general principle of recoverability.

The combination of elementary transformations $\{T_{el_1}, T_{el_2}, \ldots T_{el_p}\}$ for a given labelled bracketing φ can be further defined by indicating the factors which these elementary transformations concern. They may have to do with only one factor when T_{el} is a deletion, or they may concern two factors in cases of substitution and adjunction. For an elementary substitution, for example, $T_s(\psi, \chi)$, the factors to which ψ and χ correspond must be indicated.

This may be done most clearly on the basis of the elementary factorization of φ. For the *Aspects* model, moreover, the whole discussion can be limited to labelled bracketings which are *connected.* Let $[\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n]$ be the elementary factorization of the connected labelled bracketing φ. The elementary transformations are notated as follows.

(i) *Deletion.* $T_d(\varepsilon_{h\to t})$. This means that the factor which consists of the series of elementary factors $\varepsilon_h, \varepsilon_{h+1}, \ldots, \varepsilon_t$ is deleted (if deletion is defined for that factor).

Let $\varphi = (_S(_{NP}(_D\textit{the})_D(_N\textit{people})_N)_{NP}(_{VP}(_V\textit{help})_V(_{NP}(_D\textit{the})_D(_N\textit{animals})_N)_{NP})_{VP})_S$, with elementary factorization $[\varepsilon_1 = (_S(_{NP}(_D\textit{the})_D, \varepsilon_2 = (_N\textit{people})_N)_{NP}, \varepsilon_3 = (_{VP}(_V\textit{help})_V, \varepsilon_4 = (_{NP}(_D\textit{the})_D, \varepsilon_5 = (_N\textit{animals})_N)_{NP})_{VP})_S]$. This means that $T_d(\varepsilon_{1\to 2})$ is the deletion of the factor $\varepsilon_1\varepsilon_2$, or $(_S(_{NP}(_D\textit{the})_D(_N\textit{people})_N)_{NP}$. The interior of this factor is $(_{NP}(_D\textit{the})_D(_N\textit{people})_N)_{NP}$, and therefore $T_d(\varepsilon_{1\to 2}) = (_S$. The effect of $T_d(\varepsilon_{1\to 2})$ on the original labelled bracketing is thus: $(_S(_{VP}(_V\textit{help})_V(_{NP}(_D\textit{the})_D(_N\textit{animals})_N)_{NP})_{VP})_S$. For the same φ, we see that $T_d(\varepsilon_{2\to 3})$ is not defined. The factor $\varepsilon_2\varepsilon_3$ is $(_N\textit{people})_N)_{NP}$ $(_{VP}(_V\textit{help})_V$; it has no interior because it is not a factor in a unique factorization.

(ii) *Substitution*: $T_s(\varepsilon_{h\to t}, \varepsilon_{j\to k})$. This indicates the replacement of the interior of the factor $\varepsilon_h\varepsilon_{h+1}\ldots\varepsilon_t$ with the content of the factor $\varepsilon_j\varepsilon_{j+1}\ldots\varepsilon_k$, if defined.

For φ in our example, $T_s(\varepsilon_{1\to 2}, \varepsilon_{4\to 5})$ means the substitution of $(_{NP}(_D\textit{the})_D(_N\textit{people})_N)_{NP}$, i.e. the interior of $\varepsilon_1\varepsilon_2 = (_S(_{NP}(_D\textit{the})_D$ $(_N\textit{people})_N)_{NP}$, by $(_{NP}(_D\textit{the})_D(_N\textit{animals})_N)_{NP}$, i.e. the content of $\varepsilon_4\varepsilon_5 = (_{NP}(_D\textit{the})_D(_N\textit{animals})_N)_{NP})_{VP})_S$. This yields $(_S(_{NP}(_D\textit{the})_D$ $(_N\textit{animals})_N)_{NP}(_{VP}(_V\textit{help})_V(_{NP}(_D\textit{the})_D(_N\textit{animals})_N)_{NP})_{VP})_S$.

(iii) *Adjunction*: $T_l(\varepsilon_{h\to t}, \varepsilon_{j\to k})$ or $T_r(\varepsilon_{h\to t}, \varepsilon_{j\to k})$. For T_l (and similarly for T_r), this means the replacement of factor $\varepsilon_h\varepsilon_{h+1}\ldots\varepsilon_i$ by $E_l(\varepsilon_{h\to t})U_l(\varepsilon_{h\to t})C(\varepsilon_{j\to k})K(\varepsilon_{h\to t})U_r(\varepsilon_{h\to t})E_r(\varepsilon_{h\to t})$, where $\varepsilon_{j\to k}$ is the factor $\varepsilon_j\varepsilon_{j+1}\ldots\varepsilon_k$.

For the labelled bracketing φ in the example, $T_l(\varepsilon_{1\to 2}, \varepsilon_{4\to 5})$ means that φ will be replaced by $(_S(_{NP}\ (_{NP}(_D\textit{the})_D\ (_N\textit{animals})_N)_{NP}$ $(_D\textit{the})_D(_N\textit{people})_N)_{NP}(_{VP}(_V\textit{help})_V(_{NP}(_D\textit{the})_D(_N\textit{animals})_N)_{NP})_{VP})_S$.

Each of the elementary transformations in $\{T_{el_1}, \ldots, T_{el_p}\}$ is of

one of these three forms. Notice that in substitution and adjunction the new element is already present in the original labelled bracketing. A transformation, therefore, can introduce no new element from outside the labelled bracketing. This is a relatively restrictive formalization of the *Aspects* model.

We must see to it at this point that the elementary transformations do not "clash". This would occur if the factor $\varepsilon_{h \to t}$ of the one elementary transformation is identical with or overlaps the factor $\varepsilon_{j \to k}$ of another elementary transformation. In that case, what would happen if both elementary transformations were applied at the same time is not defined. In *Aspects* the general solution which was given in section 3.2.2. is not followed. It should be remembered that that solution consisted in defining the output of a transformation nondeterministically as a set. For a deterministic solution, a general condition must be placed upon transformations, namely that the factors concerned may not overlap. If, in formal terms, $\varepsilon_{h_m \to t_m}$ is the first factor of an elementary transformation T_{el_m} in the combination T_{el_1}, ..., T_{el_p}, where $m = 1, 2, \ldots, p$, then the NON-OVERLAP CONDITION means that $1 \leqslant h_1 \leqslant i_1 < h_2 \leqslant i_2 < h_3 \leqslant i_3 \ldots < h_p \leqslant i_p \leqslant n$, where n is the number of elementary factors in the labelled bracketing.

DEFINITION 3.13. An ELEMENTARY TRANSFORMATIONAL MAPPING with n terms, $M = \{T_{el_1}, T_{el_2}, \ldots T_{el_p}\}$ for a labelled bracketing φ is defined when

(i) φ has an elementary factorization with n elementary factors;

(ii) each of the elementary transformations T_{el} in M is defined;

(iii) M satisfies the non-overlap condition.

It is the labelled bracketing which is obtained by applying $T_{el_1}, \ldots, T_{el_p}$ to φ at the same time. (This definition is somewhat rough; for one more detailed, see the original article, Peters and Ritchie, 1973.) This labelled bracketing is also called the *value* of the transformational mapping. It is determined by convention that if (ii) does not apply, the value of the transformational mapping $M(\varphi)$ is equal to φ.

The notion of "transformational mapping" can now be extended to every factorization of φ:

DEFINITION 3.14. $M = \{T_{el_1}, T_{el_2}, \ldots T_{el_p}\}$ is an m-term TRANSFORMATIONAL MAPPING for labelled bracketing φ, if there is a factorization $\psi_1, \psi_2, \ldots, \psi_m$ of φ, and an n-term elementary transformational mapping $M' = \{T'_{el_1}, T'_{el_2}, \ldots T'_{el_p}\}$, such that for each pair T_{el_m}, T'_{el_m}, it holds that $\psi_{h_m \to i_m} = \varepsilon_{h'_m \to i'_m}$ (notice that it is not necessary that $h_m = h'_m$ or $i_m = i'_m$), and in substitution and adjunction transformations it is true for every pair T_{el_m}, T'_{el_m} that $\psi_{j_m \to k_m} = \psi'_{j'_m \to k'_m}$ (where it is again not necessary that $j_m = j'_m$ or $k_m = k'_m$).

The value of the m-term transformational mapping for φ is thus equal to that of the n-term elementary transformational mapping for φ; $M(\varphi) = M'(\varphi)$; The elementary transformations are in fact the same in both cases; only the units chosen for the m-term transformations are greater, or in any case not smaller. If one or more of the elementary transformations in M are not applicable to φ, then by convention $M(\varphi) = \varphi$, i.e. M leaves φ unchanged.

As the last step toward the definition of transformation according to the *Aspects* model, we shall now treat the structural condition and the principle of recoverability. In *Aspects* the structural condition consists of three kinds of data which the labelled bracketing must satisfy, (i) the "*is a*" *relation*, (ii) the *content-identity relation*, and (iii) the *debracketization relation*.

Suppose that φ has the (not necessarily elementary) factorization $[\psi_1, \ldots, \psi_n]$. We may then say the following.

(i) $\psi_{h \to i}$ *is an* A, if the interior of the factor $\psi_h \psi_{h+1} \ldots \psi_i$ $(1 \leqslant h \leqslant i \leqslant n)$ can be written as $(_{A_1}(_{A_2} \ldots (_{A_m}\omega)_{A_m} \ldots)_{A_2})_{A_1}$, where it is true of some A_i (i = 1, ..., m) that $A_i = A$, and where ω is well-formed.

Example: $(_{NP}(_D the)_D(_N people)_N)_{NP}$ *is an NP*, $(_{NP}(_N people)_N)_{NP}$ *is an NP*, but also *is an N*.

If φ has n factors, the notation for the fact that $\psi_{h \to i}$ is an A is: $A^n_{h \to i}$.

(ii) $\psi_{h \to i}$ *has the same content as* $\psi_{j \to k}$, if the content of the factor

$\psi_h\psi_{h+1} \dots \psi_i$ is identical to that of the factor $\psi_j\psi_{j+1} \dots \psi_k$, thus $C(\psi_{h\to i}) = C(\psi_{j\to k})$, where $1 \leqslant h \leqslant i \leqslant n$ and $1 \leqslant j \leqslant k \leqslant n$. Example: For $(_S(_{NP}(_D the)_D(_N people)_N)_{NP}(_{VP}(_V help)_V(_{NP}(_D the)_D(_N animals)_N)_{NP})_{VP})_S$, it holds that $C(\varepsilon_{1\to 1}) = C(\varepsilon_{4\to 4}) = (_D the)_D$.

If φ has n factors, the content-identity relation is written $C^n_{h\to i} = C^n_{j\to k}$.

(iii) $\psi_{h\to t}$ *has debracketization* x, if the debracketization of the factor $\psi_h\psi_{h+1}\dots\psi_i$ is the terminal string x, thus $d(\psi_{h\to t}$ $(= x$.

DEFINITION 3.15. A STRUCTURAL CONDITION C for an n-term factorization $[\psi_1, \psi_2, \dots, \psi_m]$ is a combination of n-term properties of types (i), (ii), and (iii).

Finally we shall define the principle of recoverability. This is necessary because we do not wish to call every combination of structural condition C and transformational mapping M a transformation. We wish to speak of transformation only when such a pair (C, M) leaves a "trace" after deletion or substitution. In *Aspects* this is presented in the following form. If the pair (C, M) and the result of the transformational mapping, φ', are given, then there is no more than a finite number of labelled bracketings φ, from which φ' can be derived by means of the mapping (C, M). In the case of more than one φ, we can speak of *structural ambiguity*. The guarantee of recoverability can be given in two ways. This first is that there be a copy in φ' of the string which has been deleted or replaced. The second is that the string which has been deleted or replaced is one of a finite number, determined beforehand, of deletable strings in that syntactic category. In Chapter 5 we shall see that the principle of recoverability is the pivot on which every argument on the generative power of the theory presented in *Aspects*, the theory of the universal base grammar, and the learnability of the language turns. The reason for the introduction of such a principle is to guarantee that an algorithm exists which assigns no more than a finite number of structural descriptions to every sentence in the language.

DEFINITION 3.16. A pair (C, M), in which C is an n-term structural

condition and M is a *n-term* transformational mapping, satisfies the PRINCIPLE OF RECOVERABILITY if for every elementary deletion $T_d(\psi_{h\to i})$ and every elementary substitution $T_s(\psi_{h\to i}, \psi_{j\to k})$ in M, one of the two following conditions is met:

(i) After the application of (C, M), there is a copy left of the content of $\psi_{h\to i}$, i.e. there is a pair of natural numbers t and u such that the following property is an element of the structural C: $C^n_{h\to i} = C^n_{t\to u}$, and that M contains no elementary transformations by which $C^n_{t\to u}$ will come partially or completely to be omitted. That is, if M contains elementary transformation $T_d(\psi_{f\to g})$ or $T_s(\psi_{f\to g}, \psi_{v\to w})$ with $t \leqslant f \leqslant u$ or $t \leqslant g \leqslant u$ (and $\psi_{f\to g}$ thus overlaps $\psi_{t\to u}$), then M also contains elementary transformations $T_s(\psi_{y\to z}, \psi_{p\to q})$, $T_l(\psi_{y\to z}, \psi_{p\to q})$ or $T_r(\psi_{y\to z}, \psi_{p\to q})$ such that $p \leqslant t \leqslant u \leqslant q$ (i.e. $\psi_{t\to u}$ is contained in $\psi_{p\to q}$). This guarantees that the content of $\psi_{t\to u}$ nevertheless remains somewhere in the transformational mapping.

(ii) The structural condition C states that $d(\psi_{h\to i})$ is one of a finite number of terminal strings $x_1, \ldots, x_m$.

A transformation according to the theory presented in *Aspects* can now be defined as follows:

DEFINITION 3.17. A TRANSFORMATION is a pair (C, M), in which C is an n-term structural condition (cf. Definition 3.15), and M is an n-term transformational mapping (cf. Definition 3.14), which fulfills the principle of recoverability (Definition 3.16).

A factorization $[\psi_1, \psi_2 \ldots, \psi_n]$ is a PROPER ANALYSIS for the transformation (C, M) if each of the n-term properties of C holds for $[\psi_1, \ldots, \psi_n]$ and if the factorization satisfies the structural conditions specified in Definition 3.15. In this we allow that a factor may be empty. If φ does not have a proper factorization for the transformation $T = (C, M)$, then, by convention, $T(\varphi) = \varphi$, i.e. the transformation leaves φ unchanged. The value of an OPTIONAL transformation of the labelled bracketing φ is the two-term set $\{\varphi, \varphi'\}$ if φ has a proper factorization, and φ if that is not the case. In the former case φ may be changed "at will" to φ', or left unchanged.

In the following example we present a transformation according to the *Aspects* model. It is the passive transformation (*Aspects*, p. 104).

EXAMPLE 3.3. The English passive transformation is a nine-term pair, $T_p = (C, M)$. In other words, a proper factorization for T_p contains nine factors. We shall first give a rough characterization of T_p; the formal discussion will follow.

The nine factors are the following: U_1, NP_2, Aux_3, V_4, W_5, NP_6, X_7, $Pass_8$, Y_9, where U, W, X and Y are more or less arbitrary. T_p changes this string of factors to the string $U_1+NP_6+Aux_3+Pass_8+V_4+W_5+X_7+NP_2+Y_9$.

Formally, the structural condition C for the passive transformation is the following set of properties: $\{NP^9_{2\to 2}, Aux^9_{3\to 3}, V^9_{4\to 4}, NP^9_{6\to 6}, Pass^9_{8\to 8}\}$. (A careful reading of *Aspects* would perhaps demand that it be added that W_5 is not an NP, thus, $\sim NP^9_{5\to 5}$.) This means that in the nine-term factorization the second factor is an NP, the third is an *Aux* (for "auxiliary verb" including tense), the fourth is a V, the sixth is an NP, and the eighth is of the category *Pass* ("passive"-formative). The nine-term mapping M consists of the following elementary mappings: $M = T_s(\psi_{2\to 2}, \psi_{6\to 6}), T_r(\psi_{3\to 3}, \psi_{8\to 8}), T_d(\psi_{6\to 6}), T_s(\psi_{8\to 8}, \psi_{2\to 2})$. It is obvious that M satisfies the non-overlap condition, and that it is defined for every nine-term factorization in which ψ_2, ψ_3, ψ_6, and ψ_8 have an interior, and ψ_1, ψ_1, and ψ_2 have a content.

Let us now see if the following labelled bracketing has a proper factorization for T_p. $\varphi = (_S(_{NP}$*the secretary*$)_{NP}(_{PredP}(_{Aux}(_{Tense}$*pt*$)_{Tense}$ $(_{Aspect}$*have en*$)_{Aspect})_{Aux}(_{VP}(_V$*pass*$)_V(_{Prt}$*on*$)_{Prt}(_{NP}$*the mail*$)_{NP}(_{Dir}$ *to the director*$)_{Dir}(_{Man}(_{PP}(_{Prep}$*by*$)_{Prep}(_{Passive}$*be en*$)_{Passive})_{PP})_{Man})_{VP}$ $(_{Time}$*yesterday*$)_{Time})_{PredP})_S$. In this labelled bracketing, *PredP* stands for *predicate phrase*, *pt* for *past tense*, *Prt* for *particle*, *Dir* for *direction*, *Man* for *Manner*, *PP* for *prepositional phrase*. This labelled bracketing obviously supposes a much more extensive base grammar than we have treated here.

There is indeed a proper factorization for φ, namely, in the following nine factors:

$\psi_1 = \lambda$
$\psi_2 = (_S(_{NP}\textit{the secretary})_{NP}$
$\psi_3 = (_{PredP}(_{Aux}(_{Tense}\textit{pt})_{Tense}(_{Aspect}\textit{have en})_{Aspect})_{Aux}$
$\psi_4 = (_{VP}(_V\textit{pass})_V$
$\psi_5 = (_{Prt}\textit{on})_{Prt}$
$\psi_6 = (_{NP}\textit{the mail})_{NP}$
$\psi_7 = (_{Dir}\textit{to the director})_{Dir}(_{Man}(_{PP}(_{Prep}\textit{by})_{Prep}$
$\psi_8 = (_{Passive}\textit{be en})_{passive})_{PP})_{Man})_{VP}$
$\psi_9 = (_{Time}\textit{yesterday})_{Time})_{PredP})_S$

This factorization is a proper analysis because (1) the factorization has the features mentioned under C, namely, it has nine terms, ψ_2 *is an NP*, ψ_3 *is an Aux*, ψ_4 *is a V*, ψ_6 *is an NP*, and ψ_8 *is a Passive*, (2) the factorization allows definition of each of the elementary transformations in M, because ψ_2, ψ_3, ψ_6 and ψ_8 all have interiors, and ψ_6, ψ_8 and ψ_2 have contents.

The transformation $T_p = (C, M)$ gives rise to the following factors:

$\psi_1' = \lambda$ (nothing is said of ψ_1 in M)
$\psi_2' = (_{S}(_{NP}\textit{the mail})_{NP}$ (by $T_s(\psi_{2\to 2}, \psi_{6\to 6})$)
$\psi_3' = (_{PredP}(_{Aux}(_{Tense}\textit{pt})_{Tense}(_{Aspect}\textit{have en})_{Aspect}(_{Passive}\textit{be en})_{Passive})_{Aux}$ (by $T_r(\psi_{3\to 3}, \psi_{8\to 8})$)
$\psi_4' = (_{VP}(_V\textit{pass})_V$ (nothing is said of ψ_4 in M)
$\psi_5' = (_{Prt}\textit{on})_{Prt}$ (nothing is said of ψ_5 in M)
$\psi_6' = \lambda$ (by $T_d(\psi_{6\to 6})$)
$\psi_7' = (_{Dir}\textit{to the director})_{Dir}(_{Man}(_{PP}(_{Prep}\textit{by})_{Prep}$ (nothing is said of ψ_7 in M)
$\psi_8' = (_{NP}\textit{the secretary})_{NP})_{PP})_{Man})_{VP}$ (by $T_s(\psi_{8\to 8}, \psi_{2\to 2})$)
$\psi_9' = (_{Time}\textit{yesterday})_{Time})_{PredP})_S$ (nothing is said of ψ_9 in M)

The output of the transformation ψ' is thus: $(_{S}(_{NP}\textit{the mail})_{NP}$ $(_{PredP}(_{Aux}(_{Tense}\textit{pt})_{Tense}(_{Aspect}\textit{have en})_{Aspect}(_{Passive}\textit{be en})_{Passive})_{Aux}$ $(_{VP}(_V\textit{pass})_V(_{Prt}\textit{on})_{Prt}(_{Dir}\textit{to the director})_{Dir}(_{Man}(_{PP}(_{Prep}\textit{by})_{Prep}(_{NP}\textit{the secretary})_{NP})_{PP})_{Man})_{VP}(_{Time}\textit{yesterday})_{Time})_{PredP})_S$. The following sentence is thence derived: *the mail had been passed on to the director by the secretary yesterday.*

In somewhat less detail Peters and Ritchie also give definitions of *transformational cycle* and of *transformational derivation.*

A transformational cycle supposes an ordered list of transformations. We shall call this list $(T_1, T_2, \ldots T_k)$.

DEFINITION 3.18. A TRANSFORMATIONAL CYCLE with reference to $(T_1, \ldots T_k)$ is an ordered set of labelled bracketings $(\varphi_1, \varphi_2, \ldots, \varphi_{k+1})$ for which $T_i(\varphi_i) = \varphi_{i+1}$, $i = 1, 2, \ldots, k$.

Notice that it is not necessary that $\varphi_i \neq \varphi_{i+1}$. This is not the case, in particular, when φ_i has no proper factorization for T_i.

This definition is insufficient when the list also includes optional transformations. It should be remembered that the value of an optional transformation is a set of two labelled bracketings if φ has a proper factorization: $T(\varphi) = \{\varphi', \varphi\}$. We may maintain the definition, however, by the convention that if T_i is optional and $T_i(\varphi_i) = \{\varphi_i', \varphi_i\}$, then $\varphi_{i+1} \in \{\varphi_i', \varphi_i\}$. This means that if the list $(T_1, \ldots, T_k)$ contains optional transformations, the possibility exists that for a given labelled bracketing φ_1 there is more than one transformational cycle with reference to $(T_1, \ldots, T_k)$.

A transformational derivation is a certain series of transformational cycles. We shall first illustrate this with an example. It was stated in paragraph 3.1.1. of this chapter that every derivation in the base begins with $\#S\#$, and that every new S introduced is also surrounded by two boundary symbols. The following string represents a terminal labelled bracketing, derived from the base grammar:

$$\varphi = \#(_S\alpha_1\#(_S\alpha_2)_S\#\alpha_3\#(_S\alpha_4)_S\#\alpha_5\#(_S\alpha_6\#(_S\alpha_7)_S\#\alpha_8)_S\#\alpha_9)_S\#.$$

Each α in this string is a terminal labelled bracketing which contains neither $(_S$, nor $)_S$, nor $\#$. A transformational derivation is performed as follows. First the right hand brackets $)_S$ in φ are numbered in ascending order from left to right. In the example, this operation has already taken place: $)_{S_1},)_{S_2}, \ldots,)_{S_5}$. Then the corresponding left hand brackets are numbered correspondingly (this has also been done in the example). The first transformational cycle con-

cerns the α between $(_{S_1}$ and $)_{S_1}$, in the present case α_2. The last labelled bracketing in the cycle we call α_2'. The first cycle in the transformational derivation will then have replaced α_2 with α_2'. The second cycle concerns the α between $(_{S_2}$ and $)_{S_2}$, i.e. α_4, which it replaces with α_4'. The third cycle concerns S_3 and replaces α_7 with α_7'. At that moment φ has been changed to

$$\varphi' = \#(_{S_3}\alpha_1\#(_{S_1}\alpha_2')_{S_1}\#\alpha_3\#(_{S_2}\alpha_4')_{S_2}\#\alpha_5\#(_{S_4}\alpha_6\#(_{S_3}\alpha_7')_{S_3}\#\alpha_8)_{S_4}\alpha_9)_{S_5}\#.$$

The following cycle concerns S_4. The first string of this cycle is $\beta_1 = \alpha_6\#(_{S_3}\alpha_7')_{S_3}\#\alpha_8$. Let the result of this cycle be called β_1'. The effect of this cycle is the replacement of φ' by φ'':

$$\varphi'' = \#(_{S_5}\alpha_1\#(_{S_1}\alpha_2')_{S_1}\#\alpha_3\#(_{S_2}\alpha_4')_{S_2}\#\alpha_5\#(_{S_4}\beta_1')_{S_4}\alpha_9)_{S_5}\#.$$

The last cycle concerns S_5. Denote the string between $(_{S_5}$ and $)_{S_5}$ by β_2. β_2 is then the initial string of the cycle; the terminal string is then β_2'. This finally yields $\varphi''' = \#(_{S_5}\beta_2')_{S_5}\#$.

DEFINITION 3.19. (rough definition) The labelled bracketing ω is the result of a TRANSFORMATIONAL DERIVATION from γ with reference to $(T_1, T_2, \ldots, T_k)$, if ω is obtained by applying the list $(T_1, \ldots, T_k)$ first to the subsentence farthest to the left in δ, i.e. the labelled bracketing α for which it holds that $(_S\alpha)_S$ is a component of δ, $)_S$ is the leftmost right hand bracket in φ, and $(_S$ corresponds to $)_S$; the list is then applied to the subsentence which is bordered on the right by the leftmost $)_S$ less one, and so forth until the rightmost $)_S$ is reached.

DEFINITION 3.20. For the transformational grammar $TG = (B, T)$, the labelled bracketings δ and ω are respectively called DEEP STRUCTURE and SURFACE STRUCTURE of the SENTENCE x, if

(i) δ is generated by B,

(ii) ω is the result of a transformational derivation from δ relative to T,

(iii) $\omega = \#(_S\psi)_S\#$, where ψ contains no $\#$,

(iv) $x = d(\psi)$.

The pair $\Sigma = (\delta, \omega)$ is called the STRUCTURAL DESCRIPTION of x. The LANGUAGE generated by TG consists of the strings in V_T^* for

which such a pair (δ, ω) exists. The third condition in fact contains a formalization of the notion of BLOCKING. If at the end of a transformational derivation boundary symbols remain within the outer S-brackets, then neither deep structure nor surface structure nor sentence are defined. An example of such blocking for a relative clause was given in paragraph 3.1.2. When a derivation blocks, the labelled bracketing in question is filtered out.

The filtering function of the transformational component is limited, because only one pair of boundary symbols per subsentence can be removed. However this filtering function can be increased when the base grammar is modified in such a way that the boundary symbol can also be introduced elsewhere than around S. Proposals in this direction were also made in *Aspects* (p. 191).

In paragraph 3.1.1 of this chapter the dummy symbol Δ was presented as an element of V_T. But neither in *Aspects*, nor in Peters and Ritchie (1972), nor in the above do we find guarantees that Δ will not occur in ω. Chomsky suggests that the symbol be removed transformationally, while Peters and Ritchie allow it to appear in ω, supposing, apparently, that the morphological rules will deal with it. We shall leave this as an open question here.

In closing this paragraph, we would make a few general remarks on the formalization which has been presented. *Aspects of the Theory of Syntax* is an informal book which allows very divergent kinds of formalization. It is most unfortunate that efforts to formalize the conceptual framework of that work, perhaps the most widely read and often quoted in modern linguistics, were only made seven years after its first publication. The *Aspects* model is only an outline of a linguistic theory, and it is difficult, if not impossible, to determine whether or not the theory should be further developed in that direction. The aim of Peters and Ritchie was to define the notion of "transformational grammar" as precisely as possible without leaving the framework of *Aspects*. Despite the fact that this leads to formulations which at times are not very graceful from a mathematical point of view, such an undertaking is well founded. In effect, if such an extremely restrictive definition of transformation should show that the theory

is still too strong, i.e. generates too much, there would be good reason to diverge from the given outline. In Chapter 5 we shall show that this is indeed the case.

3.3. LATER DEVELOPMENTS

One of the important principles of the *Aspects* model is that transformations do not change meaning; they are PARAPHRASTIC. In this the theory presented in *Aspects* is clearly different from that of *Syntactic Structures*, in which a transformational syntax was developed which was completely independent of semantic considerations. The criterion for the correctness of syntactic rules lay in the justification of the distinction between grammatical and ungrammatical. In *Aspects*, paraphrase relations come to play an important part. Chomsky does this following a proposal by Katz and Postal (1964): transformations are paraphrastic, that is, meaning-preserving. This is shown in the diagram of Figure 3.7. The semantic interpretation is determined exclusively by an input from the base grammar; deep structures carry all the syntactic information necessary for semantic interpretation, while transformations have no influence on this.

Soon after the publication of *Aspects* this point of view was called into doubt. Let us consider a few of the classic examples responsible for this.

(1) *Reflexive Pronouns.*
The sentence *Nixon voted for himself* goes back to the deep structure *Nixon voted for Nixon.* The relation is a paraphrastic reflexive transformation. But if that is the case for the preceding example, then *everybody voted for himself* must be based on the deep structure *everybody voted for everybody.* This relation, however, is clearly not paraphrastic.

(2) *Relative Constructions.*
The sentence *the postman who brought the letter, asked for a signature* goes back to *the postman brought the letter* and *the postman asked for a signature,* by way of a paraphrastic relative clause

transformation. However it is not the case that the sentence *all postmen who bring letters ask for signatures* is paraphrastically related to the pair *all postmen bring letters* and *all postmen ask for signatures.*

(3) *Coordinations.*
John is both shy and fresh is paraphrastically related to *John is shy* and *John is fresh.* The same coordination transformation, however, is not paraphrastic in the derivation of *no number is both even and odd* from *no number is even* and *no number is odd.*

(4) *Passives.*
The sentence *the target was not hit by the arrows* is based, via a paraphrastic passive transformation, on *the arrows did not hit the target.* The same transformation, however, is not paraphrastic if *the target was not hit by many arrows* is derived from *many arrows did not hit the target* (in one of the two possible readings of this sentence). There is a clear difference in meaning here (to which we shall return in greater detail in Chapter 4, paragraph 3).

The problems occur especially when quantifiers such as *many, all, every* are combined with negations or with a condition of identity of reference, i.e. where the deep structure contains two elements with the same denotation.

It is true that cases (1) to (4) show that some transformations of the *Aspects* model are not meaning-preserving, but they do consistently make the correct prediction concerning grammaticality. The supposed deep structure and corresponding transformations in all cases lead to grammatical sentences, while sometimes a change in meaning takes place, and sometimes not. We seem to return to the principle enunciated in *Syntactic Structures*, that transformations account for the grammaticality of sentences, but that they are not necessarily paraphrastic. This is precisely the conclusion drawn by Chomsky in his publications after *Aspects.* Transformations sometimes change meaning and consequently not only the deep structure is determinant for the semantic interpretation, but also the surface structure. Aside from this, however, the deviations from *Aspects* remained rather minor. There is

still an "independent" syntax which generates the sentence and its structural description, and some aspects of this structural description form the input of the semantic component which gives the semantic interpretation of the sentence. Some change has been made on the question of which aspects of the structural description undergo semantic interpretation. In *Aspects* only the deep structure underwent semantic interpretation, but in Chomsky's later work certain features of the surface structure do also. This new approach is called *interpretative semantics*; it is a question of independently motivated syntactic structures which undergo semantic interpretation. Little is known of the form of such a semantic interpretation, but it is certain that a semantic structure has a different form than a syntactic structure.

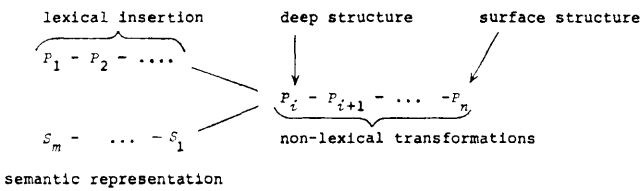
Although examples (1) to (4) show transformational changes of meaning while grammaticality is maintained, cases are known in which the application of transformations of the *Aspects* type causes the loss of grammaticality.

(5) *Each-Hopping.*
This transformation derives *the men each won a prize* from *each of the men won a prize.* Sometimes this leads to changes of meaning, as is the case, for example, when *the men each hate his brothers* is derived from *each of the men hates his brothers.* But the problem here is that when reflexive pronouns are present, this transformation leads to ungrammaticality. From *each of the men shaved himself* it should be possible to derive **the men each shaved himself* (see Hall-Partee (1971b) for a more detailed analysis of this and similar phenomena) and one is led to wonder whether the deep structures generated by the *Aspects* model are really adequate. The doubt is increased by examples of the following kind (Lakoff, 1970):

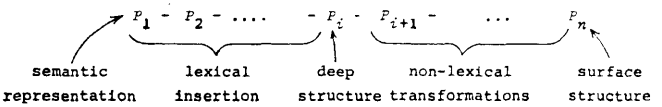
(6) *Ambiguities.*
The sentence *two dogs followed a hundred sheep* is ambiguous. It can mean that the two dogs followed a hundred sheep each, that each of the hundred sheep was followed by two dogs, or that a total of a hundred sheep was followed by a total of two dogs.

This ambiguity is not "lexical" (as is the case in *the tank is filled with water* where the ambiguity is caused by the fact that *tank* can have more than one meaning here). It is not a surface structure ambiguity either, for there is only one possible surface parsing for this sentence (cf. definitions of ambiguity in paragraph 3.1.3 of this chapter). Therefore there must be more than one deep structure for the sentence, i.e. there is more than one transformational derivation. The theory in *Aspects*, however, gives only one transformational derivation for it, i.e. one deep structure.

Examples like (5) and (6) suggest that the notion of "deep structure", as used in the *Aspects* model, is not adequate. Perhaps it is possible to maintain the paraphrastic character of transformations by making the grammar generate more adequate underlying structures. This may be attempted by specifying the range of quantifiers in the underlying structure (cf. Chapter 4, paragraph 3). A more radical approach is also possible, namely, by abandoning the interpretative character of the semantic component, or in other words, by abolishing to a certain extent, the distinction between semantic and syntactic rules. To clarify this, we return briefly to the *Aspects* model. In it, the categorial rules generate a structure P_1, with the dummy symbol and grammatical formatives as terminal elements. The lexical insertion rules transform these gradually into P_t, the deep structure, and the transformational cycles finally transform P_i into P_n, the surface structure. The deep structure P_t is interpreted semantically by means of the semantic rules. In other words, P_t leads successively to structures S_1, S_2, ..., S_m, where S_m is the semantic interpretation. All these "operations" are nothing other than formal relations among structures. They have no direction or temporal order. The *Aspects* schema may therefore be represented as below in (7).



The first proposal is to replace the above schema with (8):



The deep structure in (8) is derived from a sequence of semantic operations, which regulate, among other things, lexical insertion and hierarchical ordering. This approach gave rise to the term *generative semantics* as opposed to interpretative semantics. Much discussion, however, (Chomsky 1971, Katz 1971, Lakoff 1971, Chomsky (1972) has shown that formulations (7) and (8) are notational variants of each other which no empirical test can distinguish.

It is true that that which was left to semantics in the interpretative approach must be recuperated by "syntactical means" in the generative approach, for there is no longer any separate semantic component. On this point the generative semanticists have proposed a number of interesting modifications regarding the *Aspects* theory, concerning, among other things, the mechanism of lexical insertion. In the *Aspects* model, lexical insertion is accomplished by the replacement of a dummy symbol with a lexical element, if the phrase marker satisfies the restrictions defined in the complex symbol of that element. Throughout this process, however, the phrase marker remains unchanged. Generative semanticists, on the other hand, perform lexical insertion by replacing subtrees of a semantic interpretation with lexical elements. The terminal elements of such subtrees are abstract elements, "semantic primitives" which, for simplicity, are denoted by words. The classic example (though now somewhat bypassed, see Fodor 1970) is presented in Figure 3.9. It indicates how the word *kill* is inserted during the generation of *John kills Mary*. At a certain stage of derivation this sentence has the underlying structure shown in Figure 3.9a. This contains an explicit semantic interpretation of *kill*, and the meaning is represented as a nesting of the predicates *cause*, *become*, *not* and *alive*, all of which are semantic primitives. A number of transformations (*predicate*

raising) change this structure through b. and c. to d., and the subtree under *Pred* can then be replaced by *kill*. This yields e. = P_i, the deep structure of *John kills Mary*. The surface structure f. = P_n then follows, details aside, by way of a *subject raising* transformation (more is said on this in Chapter 4, paragraph 3). All transformations here are paraphrastic; thus a. is synonymous with f., and even without the (optional) intermediate transformations the semantic primitives in a. can be replaced directly by the corresponding lexical elements. This, by means of a few obligatory transformations will yield the sentence *John causes Mary to become not alive*, which must then be synonymous with *John kills Mary*.

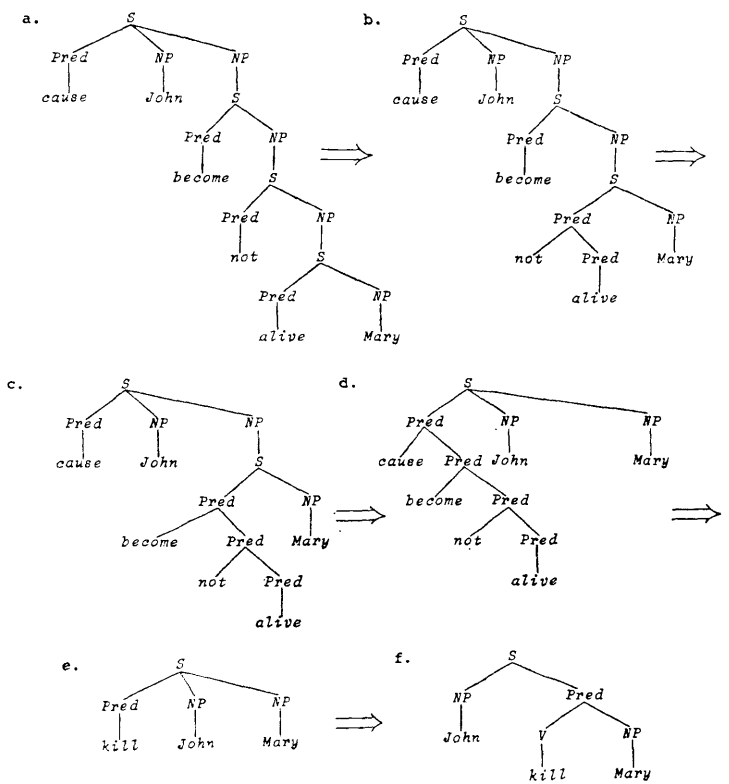


Fig. 3.9. Underlying structures for *John kills Mary*.

The analysis of structures more abstract than deep structures has several other advantages. These have to do with the range of quantifiers in natural languages (cf. Chapter 4, paragraph 3), with presuppositions (the sentence *the man who stole the money lives in Canada* presupposes that money was stolen, and *John never works after five o'clock* presupposes that John sometimes does work before five o'clock, although this does not follow logically), with topic-comment relations, and with focus. Topic-comment relations are usually marked by emphasis in the surface structure; thus *the letter has* ARRIVED is said when the listener expects comment on the letter, while the topic of *the* LETTER *has arrived* is *arrived.* Focus is that which the speaker himself thinks important in the sentence and which is in English usually marked by word order. The active/passive distinction is often a matter of focus; compare, for example, *the mayor opened the council meeting at eight o'clock* and *the council meeting was opened by the mayor at eight o'clock*, with *the mayor* and *the council meeting* as respective focuses.

All of these matters would fall under the semantic component in the *Aspects* model, if the difference between interpretative and generative semantics were limited to the difference between (7) and (8). But generative semanticists hold that the differences are greater. They argue that schema (8) is also unsatisfactory, for the rules of lexical insertion are not applied *en bloc*, but rather some lexical elements are inserted only after one or more non-lexical transformational cycles. If this proves to be the case, it will mean that the notion of "deep structure" as a distinct phrase marker in the derivation of a sentence will no longer be tenable.

Interpretative semanticists have a clear and detailed syntactic theory, but they are not very specific on the structure of semantic interpretation, though there is some tendency to correct this (see, for example, Jackendoff 1969). Generative semanticists, on the other hand, have enriched linguistics with many new semantic insights, but they are not very explicit on the syntactic mechanisms which would transform their underlying structures into sentences. What, for example, are the limitations on the alternation of lexical

and other transformations? One new concept in this connection is that of DERIVATIONAL CONSTRAINT. A derivational constraint is a condition on the well-formedness of a transformational derivation as a whole, apart from the correctness of each individual transformational step. There is only one example of this in *Aspects*, the condition that transformations be applied cyclically and in a given order. At present, many other derivational constraints are being added to this, they are essentially conditions on pairs of (not necessarily directly consecutive) tree-diagrams in the transformational derivation. An example of a derivational constraint is the reduction of stress on the auxiliary verb; *Sam is happy*, for example, has the variant *Sam's happy* where the stress on *is* has been reduced. This optional transformation, however, may not be applied, if, somewhere in the transformational derivation, the element which follows the verb has been deleted. An example of this is *Max is happier than Sam is these days*, for which there is no stressless variant *Max is happier than Sam's these days*. This is therefore a condition on which the well-formedness of the entire derivation will depend. Lakoff (1971) remarks that in this respect generative semantics far outstrips the *Aspects* theory. This is indeed the case, and this new syntactic concept might well be justified from a linguistic point of view (although there is scarcely any agreement on the matter. cf. Chomsky (1972). But this new theory has in fact only *removed* limitations. A whole arsenal of new blocking mechanisms has been added to the filtering function of transformations in the *Aspects* model; with these new mechanisms, any enumerable set of sentences whatsoever can, in principle, be defined by a transformational grammar. Derivational constraints only raise the generative power of transformational grammars, and, as we shall see in Chapter 5, there is decidedly no need of that. Just as interpretative semantics is in need of more specific semantic rules, generative semantics needs a much more restricted syntax.

To summarize, we can state that it has proven impossible to maintain both the notion of "deep structure" as presented in *Aspects*, and the principle of paraphrastic transformations. When the former is set aside, generative semantics results, and when the

latter is abandoned, interpretative semantics results. The question is to what extent the two trends may be variants from a formal point of view. But most of the syntactic modifications within the generative semantics group are enlargements with respect to the *Aspects* model, with all the serious disadvantages to be discussed in Chapter 5. As far as content is concerned, however, a short time of generative semantics has seen the growth of important insights into lexical structure, presuppositions, focus, and topic-comment relations.

4

MIXED MODELS II: OTHER TRANSFORMATIONAL GRAMMARS

4.1. REASONS FOR FINDING ALTERNATIVE MODELS

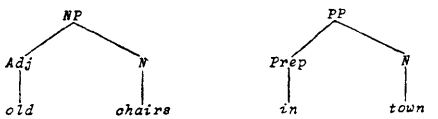
The form of the transformations in a mixed model is largely determined by the nature of the base grammar. In the *Aspects* model the base grammar is a phrase structure grammar, and also after the publication of *Aspects*, transformational linguistics has tended to use phrase structure grammars as base grammars, and consequently transformations have retained the essentials of the originally indicated form. In Chapter 2, paragraph 5 it was mentioned that through the use of phrase structure grammars as base grammars the traditional advantages of phrase structure grammars could be taken into a more complete theory of natural languages. At the same time, many of the weaknesses of such grammars could be met by means of transformation rules. It was noticed that a number of the problems with phrase structure grammars are due to the impossibility of assigning more than one tree-diagram or phrase marker to a sentence at a time; in principle transformational grammars can solve this and many other problems.

But the formalism of phrase structure grammars, even within the framework of transformational grammars, still has a number of unattractive points, and this has led linguists to seek other bases which might be able to represent certain linguistic insights in a more natural way. This in turn has resulted in several alternative proposals concerning the structure of the transformational component of the grammar. To give some impression of those unattractive points, we shall mention a few linguistic notions which could

not be built into a transformational grammar with a phrase structure grammar as base, unless accompanied by the necessary auxiliary constructions.

(1) *Endocentric versus Exocentric Constructions.*
These notions, first introduced by Bloomfield, are closely connected with that of "distribution". A construction is called ENDOCENTRIC if it contains a part which has the same distribution as the construction itself; the part can always take the place of the entire construction. Nearly any sentence in which *old chairs* occurs corresponds to an equally acceptable sentence in which only *chairs* occurs. Consider, for example, *take all the old chairs outside* and *take all the chairs outside*, or *old chairs creak* and *chairs creak*. *Old chairs* is an endocentric construction, the *head* of which is *chairs*. Some endocentric constructions have more than one head, as, for example, in *old chairs and tables*, where both *chairs* and *tables* are heads. All constructions which are not endocentric are exocentric. *In town* is an exocentric construction, because *John lives in town* corresponds to no sentence **John lives in* or **John lives town*.

A phrase structure grammar can express such relations only with difficulty. There is no natural distinction between tree-diagrams such as the following:



We must therefore establish a convention according to which *N* is always the head of the *NP* by which it is directly dominated, whereas the same does not hold for *Prep* and *PP*, or for *N* and *PP*. Such conventions are not superfluous; they are explicitly required for the correct representation of the structural conditions of certain transformations. It is, for instance, a condition for *tree pruning* (i.e. the removal of superfluous nodes in the tree-diagram, or of superfluous brackets $(_A$, $)_A$ in the labelled bracketing) that the head of the syntactic category *A* has been transformationally

deleted. But then the head needs an independent definition for each possible constituent.

(2) *Dependency.*
Closely related to the preceding point is the fact that phrase structure grammars cannot give a simple representation of syntactic dependencies. Although *in town* is an exocentric construction, *town* is in a certain respect dependent on *in*, because it is connected with the rest of the sentence by means of the preposition. In *John lives in town*, *town* is related to *lives* by way of *in*, just as in the relative construction *the postman who brought the letter*, *brought the letter* is dependent on *who* in its relation to *the postman*. Such intuitive dependencies may be found in nearly every construction. It is not among the most difficult linguistic judgments to indicate the element through which a phrase is related to the rest of the sentence. The notion of "dependency" is extremely important to a number of linguistic theories, such as those of Harris and of Tesnière. Phrase structure grammars are remarkably unsuited for representing dependencies. They are designed for categorizing phrases hierarchically, and are good systems for expressing the "*is a* relation" (*old chairs* "*is a*" noun phrase, etc.; cf. Chapter 3, paragraph 2.4), not for representing dependencies.

(3) *The Sentence as Modifier and as Complement.*
In the relative construction mentioned above, *who brought the letter* is dependent on *the postman*. In the *Aspects* model, this construction is generated in the base grammar by means of sentence embedding, the recursive introduction of the symbol *S*. Precisely the same mechanism is used for the derivation of a sentence such as *I know that the postman brought the letter*. But in the first case, *the postman brought the letter* is an adjunction or modifier of *postman*, while in the second case it is the object-complement of the sentence. Intuitively there is a great difference between the addition of a modifier to a given sentence structure (Wundt calls this an *associative* relationship; cf. Chapter 2, paragraph 3.1) and the elaboration of part of that sentence structure, such as the object (Wundt calls this an *apperceptive* relationship). This distinction

is completely neglected when a phrase structure grammar is used as the base grammar. In both cases, sentence embedding is used as the generative mechanism.

(4) *Functional Relations.*
Dependencies indicate the general lines of the functional relations within the sentence. In Chapter 3, paragraph 1.1, definitions of such functional relations were given as "subject of", "object of", etc. It should be remembered that functional relations are defined on the basis of the production rules, through the notion of "direct dominance". Such definitions are not only very indirect from the point of view of intuition, but they become impossible when a serious effort is made to express the *case relations* within the sentence in that way. An example should make this clearer. In the sentence *John gave the boy the money*, there are two noun phrases within the verb phrase, *the boy* and *the money*. Which of these is the "direct object"? For which of the two does the relation [*NP*, *VP*] hold? In the given definition it is implicitly supposed that there is only one noun phrase having the relation [*NP*, *VP*]. This may perhaps be the case, and the sentence may have a deep structure such as *(John (gave ((the money) (to the boy))))*, where *the boy* is no longer in the relation [*NP*, *VP*]. But if we wish to use such relational definitions for all case relations, and not only for the direct object ("objective case") and the indirect object ("dative case"), we reach an impasse. In the sentence *John went by train to Amsterdam*, *by train* is an "instrumental" case, and *to Amsterdam* is a "locative". But these are two coordinated prepositional phrases, both of which proceed from the rewriting of the verb phrase *VP*, according to no intrinsic order. Therefore relational definitions in terms of direct dominance can make no distinction between locative and instrument. Other problems with case relations such as "agent" and "dative" also occur in this connection. But case must be explicitly marked for the various parts of a structural description, because the semantic interpretation of the sentence is based precisely on this information. It is possible to realize this in the lexical insertion rules, by adding special syntactic

case markers to the complex symbol of a word. But then either an ambiguous situation will result in which some relations (such as "main verb of") will be defined configurationally and others (such as case relations) lexically, or the situation will be such that *all* relations will be defined lexically. From this we may conclude that a phrase structure grammar is not a natural means for expressing functional relations (we shall return to this in paragraph 5 of this chapter).

(5) *Hierarchy.*
In section 2.3.3 we pointed out that too much hierarchy may result from coordination. We showed that phrase structure grammars had to be extended with rule schemas (of the form $A \rightarrow B^n$) in order to avoid giving a pseudo-hierarchical description to a construction which is intuitively coordinative. In a transformational grammar with a phrase structure grammar as base this problem still remains unsolved (cf. Dik, 1968); extra mechanisms such as rule schemas are again required there. More generally, the use of phrase structure grammars easily leads to spurious hierarchy in linguistic descriptions. Every linguistic refinement leads either to the introduction of new nonterminal elements which are more or less "intuitive", or to an elaboration of the hierarchy by recursive sentence embedding (such as in Figure 3.9), in which case there is intuitively no longer a relationship between the length of the sentence and the extent of the hierarchy. Both options are unattractive from a linguistic point of view (we shall return to this in Chapter 5), but they are also unattractive from a psycholinguistic point of view. Many extremely subtle distinctions in the nonterminal vocabulary correspond to no "psychological reality" whatsoever (see Volume 3, section 3.1.), and there is no evidence that the native speaker, in understanding or producing sentences, constantly uses complicated hierarchies. The psycholinguist will have more use for a grammar in which all syntactic information is stored in the terminal vocabulary, than for a grammar which consists principally of the rewrite relations among highly abstract syntactic categories. The native speaker can then be described from

the point of view of a detailed lexicon which gives for every word the way in which it may be combined with other words as well as the functional relations which can be expressed with it. A model with an excess of hierarchy is psychologically unattractive.

In this chapter we shall discuss a number of alternative base grammars which, in varying degrees, avoid the difficulties mentioned in (1) to (5). A real comparison of the advantages and disadvantages of the various formulations is not possible at the present stage. The reason for this is that for most base grammars the form of the corresponding transformational component has at best only partially been elaborated. But even when the respective transformational components have been completely formalized, the decisive comparison must be based on the way in which the various transformational grammars can treat a number of "representative" linguistic problems in detail. On this point information is still scarce for all the alternative models.

Until a convincing comparison of the models can be made, the choice among them should be determined by the aims of the investigator. The practicing linguist will be inclined to use phrase structure grammars as base grammars because a great many problems and solutions in modern linguistics have been formulated within that framework. For the psycholinguist, however, such considerations are much less pressing and it might be more fruitful for him to use other types of grammars, more closely related to models of human linguistic behavior. The ideal situation would be one in which the mathematical relations among the various formulations were known in detail. The most workable formalization for a given linguistic or psycholinguistic problem could then be chosen without loss of contact with other formulations. In some cases such relations are already known, as we shall see in the following paragraph.

4.2. CATEGORIAL GRAMMARS

The history of these grammars goes back to the work of the Polish logicians Leśniewski and Ajdukiewicz, who developed a

"theory of semantic categories", not for natural languages, but for artificial languages of logic, especially connected with the "Polish notation" in logic. Later categorial grammars came to be used for the description of natural languages, particularly through the work of Bar-Hillel. The predominating developments in the grammars discussed in the preceding chapter drew attention away from categorial grammars, and it is only since a relatively short time that they are seriously presented as bases for transformational grammars (by Lyons 1968, Miller 1968, Geach 1970, Lewis 1970, and others).

A categorial grammar *CG* is characterized by a finite VOCABULARY *V*, a small (finite) set of PRIMITIVE CATEGORIES *Cp*, including a special element *S*, a RULE or RULES *R* which indicate how COMPLEX CATEGORIES can be derived from primitive categories, and, finally, a LEXICAL (ASSIGNMENT) FUNCTION *A*, which indicates the categories to which vocabulary elements belong. We shall first offer an example of this. Suppose that we have two vocabulary elements, *John* and *eats*, and two primitive categories, *S* and *N*. The following rule R_1 is then introduced; by it complex categories can be derived:

R_1: If C_1 and C_2 are categories, then $C_1\backslash C_2$ is also a category.

Because *S* and *N* are categories, $S\backslash N$ is also a category, and because $S\backslash N$ is a category, $(S\backslash N)\backslash N$, $N\backslash(S\backslash N)$, $(S\backslash N)\backslash S$, $S\backslash(S\backslash N)$, etc are also categories. Each of the words in the vocabulary is assigned one or more of these "category names" by the function *A*, for example, *John* is an *N*, and *eats* is an $N\backslash S$. We define the general LEFT CANCELLATION RULE as follows:

α reduces to β if $\alpha = C_1 + (C_1\backslash C_2)$ and $\beta = C_2$, where α and β are strings of categories.

In the example, *John eats* corresponds to the string $N + N\backslash S$, and the cancellation rule states that that string reduces to *S*. Therefore the string *John eats* belongs to category *S*. Given the categories of the vocabulary elements and the cancellation rule, we can determine the category of a phrase. If the string reduces to *S*, the phrase is said to be a SENTENCE in *L*(*CG*).

If we wish to give a simple description of the sentence *John eats apples*, it must be possible for us to make additions also to the *right* of *eat*. For this we introduce rule R_2.

R_2: If C_1 and C_2 are categories, then C_1/C_2 is also a category.

The RIGHT CANCELLATION RULE, belonging to R_2, is defined as follows:

α reduces to β if $\alpha = (C_1/C_2) + C_2$, and $\beta = C_1$.

If both R_1 and R_2 hold, then complex categories such as the following may be formed: $N\backslash S$ (by R_1), $(N/S)/(N/S)$ (by R_2), $(N\backslash S)/N$ (by R_1 and R_2), and so forth.

Suppose that the function A assigns to *eats* both the above mentioned category $N\backslash S$ and the category $(N\backslash S)/N$. Let *apples* belong to the category N. Does the string *John eats apples* belong to $L(CG)$? This holds by definition if the categories of the string reduce to S. Figure 4.1 represents the reduction of this sentence to S. The dotted lines show how the reductions take place, and it is not difficult to see in this derivation the reflexion of a derivation in a context-free grammar; we shall return to this point later.

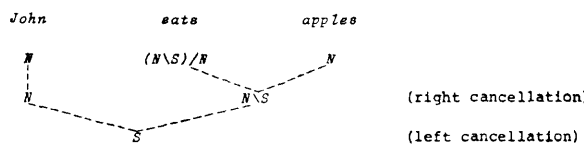


Fig. 4.1. Categorial reduction for the sentence *John eats apples*.

We can assign the category $(N\backslash S)/N$ to all transitive verbs, and the category $N\backslash S$ to all intransitive verbs. Verbs such as *eat* which can be both transitive and intransitive are assigned both categories. The notation for this is as follows: $A(eat) = \{N\backslash S, (N\backslash S)/N\}$. We can go on with other kinds of words; adjectives such as *fat* are N/N, adverbs such as *much* are $(N\backslash S)\backslash(N\backslash S)$. It would be an instructive exercise to reduce the sentence *fat John eats much* with these categories.

If a categorial grammar has only R_1 and the corresponding left cancellation rule, or only R_2 and the right cancellation rule, it is

called UNIDIRECTIONAL; if it has both rules, it is called BIDIRECTIONAL. When used without further indication, *CG* will stand for a bidirectional categorial grammar; *UCG* will be used when express reference is made to a unidirectional categorial grammar. Bar-Hillel (1964), however, has proven that bidirectional and unidirectional categorial grammars are weakly equivalent.

At this point we can define categorial grammars formally.

A CATEGORIAL GRAMMAR is a system $CG = (V, C, R, S, A)$, in which V is a finite vocabulary, C is a finite set of primitive categories, R is a set of rules for the generation of categories, $S \in C$, and A is a function which assigns a set of categories (primitive or derived) to each of the elements of V.

A categorial grammar is UNIDIRECTIONAL if R contains one rule (R_1 or R_2), and BIDIRECTIONAL if R contains both R_1 and R_2. A string $x = a_1a_2 \ldots a_n$ in V^* belongs to category Y if there is a string of categories $C_1, C_2, \ldots, C_n$ such that (i) $C_i \in A(a_i)$ (i.e., C_i is an element of the set of categories which the function A has assigned to the vocabulary element a_i. Thus, for example, $N\backslash S \in A$ *(eat)*, because $A(eat) = \{N\backslash S, (N\backslash S)\backslash N\}$), and (ii) the string $C_1C_2 \ldots C_n$ reduces to Y. It is said that $x \in V^*$ is a SENTENCE if x belongs to category S. The LANGUAGE $L(CG)$ accepted or generated by CG is the set of sentences accepted or generated by CG. With categorial grammars, just as with automata, we speak rather of "accepting" than of "generating". When a sentence is presented as input, the categorial grammar passes through a series of reductions until the "final state" S is reached, just as an automaton reaches a state at which the sentence is accepted.

Bar-Hillel (1964), together with Gaifman and Shamir, has proven that categorial grammars are weakly equivalent to context-free grammars. We shall not give the proof here, we shall only show by means of an example how a weakly equivalent context-free grammar can be constructed for a given categorial grammar.

EXAMPLE 4.1. Take categorial grammar $CG = (V, C, R, S, A)$. An equivalent context-free grammar $CFG = (V_N, V_T, P, S)$ is constructed as follows. $V_T = V$, $V_N = C \cup W$, where W is the set of

all categories which are assigned by the function A to the elements of V. W is, of course, finite. The productions in P are composed as follows:

(i) If $C_1\backslash C_2$ is a complex category in W, then $C_2 \rightarrow C_1 + (C_1\backslash C_2)$ is a production in P.

(ii) If C_1/C_2 is a complex category in W, then $C_1 \rightarrow (C_1/C_2) + C_2$ is a production in P.

(iii) If C_i is a (possibly complex) category in W, assigned to vocabulary element a_i in V, then $C_i \rightarrow a_i$ is a production in P, for every a_i in V.

For the above example, let $CG = (V, C, R, S, A)$, with $V = \{John, eats, apples\}$, $C = \{S, N\}$, $R = R_1 \cup R_2$, and A as follows:

$A(John) = \{N\}$

$A(eats) = \{(N\backslash S)/N, N\backslash S\}$

$A(apples) = \{N\}$

Then the equivalent context-free grammar is $CFG = (V_N, V_T, P, S)$, with $V_N = \{S, N, (N\backslash S)/N, N\backslash S\}$, $V_T = \{John, eats, apples\}$, and the following productions in P:

$S \rightarrow N+(N\backslash S)$	(according to (i))
$N\backslash S \rightarrow (N\backslash S)/N+N$	(according to (ii))
$N \rightarrow John$	(according to (iii))
$N \rightarrow apples$	(according to (iii))
$(N\backslash S)/N \rightarrow eats$	(according to (iii))
$N\backslash S \rightarrow eats$	(according to (iii))

The reader can verify that the phrase marker in Figure 4.1 may be derived by this context-free grammar.

A categorial grammar is an ideal means for expressing endocentricity. It is not difficult to arrange a categorial grammar in such a way that an endocentric phrase has the same category as its head. Let us return to the example *old chairs* (from 4.1, under (1)). If we assign the category N/N to all the adjectives in the categorial grammar, *old chairs* is $(N/N)+N$, which reduces to N. Similarly, an adverb can be assigned the category $(N/N)/(N/N)$, and consequently the phrase *very old* will have the category string $((N/N)/(N/N))+(N/N)$, which reduces to N/N, the category of *old*.

But this advantage does not simply extend to dependencies in

general. In the example given, it is not the case that a verb phrase is of the same category as the transitive main verb (the verb phrase is $N\backslash S$, and the main verb is $(N\backslash S)/N$). This is indeed in agreement with the fact that the verb phrase is not endocentric but exocentric; yet it would be preferable to have the dependent phrase in the more complex category, as is the case with endocentric constructions (Lyons (1968) also makes this proposal). But that does not hold here; the more complex category is that of the verb, while the dependent noun is of a primitive category. There are also arguments for a reversed approach; if a word has the function of "link" between two other words or phrases, as is the case with transitive verbs, prepositions or relative pronouns, it should have the more complex category, so that the dependent categories to the left and to the right of it might lend themselves to reduction.[1] If, however, the simpler category is assigned in general to the dependent element, the natural advantage of categorial grammars in the description of endocentric constructions is lost, and the head of the construction (or the independent element) no longer receives the same simple category as the entire construction. A categorial grammar can thus give adequate representation of either endocentricity or dependence, but not both at the same time.

There are many imaginable variations on the theme of "categorial grammar". Notice that a categorial grammar, as defined above, unites precisely *two* categories with every reduction. Such a grammar is strongly equivalent to a context-free grammar in Chomsky normal-form (cf. Volume I, Chapter 2, paragraph 3.1), as may easily be verified on the basis of the construction in Example 4.1. One may also seek strong equivalence with other types of context-free grammars, for example, grammars in Greibach normal-form (cf. Volume I, Chapter 2, paragraph 3.2). To do this, the following rule must be introduced to replace rules R_1 and R_2.

R_3: If C, C_1, C_2, ..., C_n are primitive categories, then $C/C_1C_2...C_n$ is also a category.

[1] Such words resemble the *functors* of formal logic (cf. Curry 1961); there the dependent elements are the arguments.

The corresponding cancellation rule is the following:

α reduces to β if $\alpha = (C/C_1C_2...C_n) + C_1 + C_2 + ... + C_n$ and $\beta = C$, where C, C_1, ..., C_n are primitive categories.

EXAMPLE 4.2. Suppose that we have the following context-free grammar in Greibach normal-form: $CFG = (V_N, V_T, P, S)$, with $V_N = \{S, V, P, A, B, D, N\}$, $V_T = \{fast, in, John, park, runs, the, very\}$, and the following productions in P:

$S \rightarrow John + V + P$ $\qquad$ $B \rightarrow fast$
$V \rightarrow runs + A + B$ $\qquad$ $D \rightarrow the$
$P \rightarrow in + D + N$ $\qquad$ $N \rightarrow park$
$A \rightarrow very$

All the productions are of the form $A \rightarrow a\alpha$, with $a \in V_T$ and $\alpha \in V_N^*$. This grammar generates the sentence *John runs very fast in the park*. A strongly equivalent categorial grammar is $CG = (V_T, C, R_3, S, A)$, with $C = \{S, V, P, A, B, D, N\}$, and A as follows:

$A(fast) = \{B\}$ $\qquad$ $A(in) = \{P/DN\}$
$A(very) = \{A\}$ $\qquad$ $A(John) = \{S/VP\}$
$A(the) = \{D\}$ $\qquad$ $A(park) = \{N\}$
$A(runs) = \{V/AB\}$

Figure 4.2 shows the reduction of the sentence *John runs very fast in the park*. The dotted lines show the isomorphism between this diagram and a phrase marker for the derivation with a grammar in Greibach normal-form.

It follows from the fact that grammars in Chomsky normal-form are weakly equivalent to grammars in Greibach normal-form, that

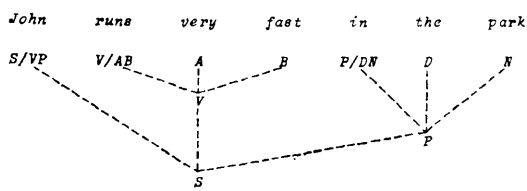


Fig. 4.2. Categorial reduction of the sentence *John runs very fast in the park* (Example 4.2).

the corresponding categorial grammars are weakly equivalent. This means that R_3 adds nothing to the weak generative power of categorial grammars. Because unidirectional and bidirectional categorial grammars are weakly equivalent, we can extend R_3 to a bidirectional rule R_4, without losing equivalence with context-free grammars:

R_4: If C, A_i, and B_j ($i = 1,\ldots, n$, $j = 1,\ldots, m$) are primitive categories, then $A_1A_2\ldots A_n\backslash C/B_1B_2 \ldots B_m$ is also a category.

The corresponding cancellation rule is:

α reduces to β if $\alpha = A_1 + A_2 + \ldots + A_n + (A_1A_2\ldots A_n\backslash C/B_1B_2\ldots B_m) + B_1 + B_2 + \ldots + B_m$, and $\beta = C$.

In this way entire strings of categories to the left and to the right of the complex category can be eliminated.

Because of the weak equivalence of all these grammars, the choice among these possibilities is determined exclusively by consideration of the descriptive adequacy of the grammar. Thus bidirectional grammars have the advantage over unidirectional grammars that a natural representation of both left and right adjunctions is possible. Compare, for example, *the old chairs* and *the chairs here*. A left adjunction, such as *old*, to a noun, receives the category N/N; a right adjunction, such as *here*, receives the category $N\backslash N$. In both cases the categories related to the noun (N) *chairs* reduce to N. The linguist might prefer to derive these adjunctions transformationally (from *the chairs are old* and *the chairs are in this place* respectively) but it may not be taken for granted that every distinction between left and right adjunction can be expressed in the most satisfactory way by means of transformational derivation. Consider, for example, adverbial phrases of place and time which may occur in various places in the sentence, or tense morphemes which sometimes appear to the left, sometimes to the right of the verb: *John* will *come* and *John come*-s. The cancellation of entire strings of categories is an attractive point with respect to the base grammar, for by it various types of verbs can be characterized

very well. The verbs in Table 4.1 give a (unidirectional) example of this.

TABLE 4.1. Complex Categories for Verbs.

Category	Verb	Basic Form	Sentence
S/N	*walk, sit, eat*	*walk (John)*	*John walks*
S/NN	*kill, eat*	*eat (John, apples)*	*John eats apples*
S/NNN	*give, send, tell*	*give (John, apples, children)*	*John gives apples to the children*
S/S	*continue, begin*	*begin (the bell rings)*	*the ringing of the bell begins*
S/NS	*say, think, know*	*say (John, it is raining)*	*John says that it is raining*
S/SN	*amaze, enjoy*	*amaze (the sun is shining, me)*	*that the sun is shining amazes me*
S/NNS	*tell, ask*	*ask (John, Peter, it is raining)*	*John asks Peter if it is raining*
S/NSS	*explain, relate to*	*relate to (I, John is hungry, John is growing)*	*I relate John's hunger to his growing*

It is possible in this way to express various functional relations, especially case relations. In Chapter 8, paragraph 1 we mentioned that in a phrase structure grammar such information can only be given in the lexicon. As a categorial grammar is essentially nothing more than a lexicon, it is not at all surprising that case relations can be formulated very naturally in it.

Some of the problems raised by phrase structure grammars (cf. Chapter 2, paragraph 3.3) were solved transformationally (cf. Chapter 3). One of those problems was that of discontinuous constituents (e.g. *I saw the man yesterday whom you told me about*). Context-sensitive grammars were able to deal with this, but not in a very convincing way, namely, by exchanging the grammatical categories of the various elements (cf. Chapter 2, paragraph 4). One might imagine a generalization of categorial grammars in which discontinuities could be formulated simply without changes of category. The generalization exists in that the *continuity restriction* implicit in the cancellation rule of R_3 is dropped. This means that $C/C_1C_2...C_n$ may be reduced to C if this complex category

occurs in a string in which C_1, C_2, ..., C_n occur in this order, but not necessarily without interruption. More precisely, while retaining R_3 one can make the convention that reduction can take place if the string is of the form $\alpha_1 C_1 \alpha_2 C_2 \ldots \alpha_n C_n \alpha_{n+1}$ where α, α_2, ..., α_{n+1} are strings of zero or more categories, and the complex category occurs in one of the α's. The place of the complex category with respect to the primitive categories is therefore no longer important. For the sentence *I saw the man yesterday, whom you told me about*, the reduction of *I saw the man whom you told me about* is not hampered by the fact that *yesterday* breaks the sequence *man, whom*. The categories of these elements remain nevertheless unchanged. If the continuity restriction is abandoned, the argument for a bidirectional categorial grammar loses its value, for adjunction can then take place freely to the left and to the right. Even the interruption caused by the verb itself (e.g. in *John kills Peter*, $N + S/NN + N$) does not block the reduction. A number of typical transformational phenomena can thus be treated in this way. By dropping the continuity restriction, however, we raise the weak generative power of the grammar above that of a context-free grammar, but it is not known to what degree.

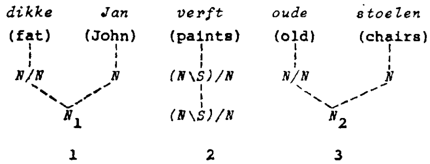
The power can also be raised by dropping the *ordering restriction* in the cancellation rule. In that case, one would allow that $C/C_1C_2 \ldots C_2 \ldots C_n$ reduce to C, when the complex category is followed by some permutation of C_1, C_2, ..., C_n. Interchanges, such as of the positions of particle and object *(John put on his coat, John put his coat on)*, can be expressed categorially in this way. In this case also, the degree to which the generative power is increased is unknown. If both the continuity restriction and the ordering restriction are dropped, the grammar is called an UNRESTRICTED categorial grammar. In such a grammar, only the elements which can occur in a sentence at the same time are specified; the order in which they may occur is not specified. In other words, every permutation of elements yields a new sentence. Unrestricted categorial grammars are equivalent to the systems called *set-systems* in connection with context-free grammars. A set-system has rewrite rules, the output of which does

not consist of strings, but of *unordered* sets of elements; their formal structure was studied by Curry (1961) and by Šaumjan and Soboleva (1963). Obviously the generative power of such grammars is considerably greater than that of restricted categorial grammars and that of context-free grammars. Although they solve a number of problems (discontinuity, interchanges), they also raise new problems, such as the way to deal with restrictions on word order.

Linguistic literature offers no serious attempt whatsoever to define a transformational component for a categorial base grammar. If the base is restricted in the usual way, the transformational component will tend to function in the same way as that of the *Aspects* model, that is, in the adjunction, deletion, and substitution of subtrees. In terms of categorial grammars, a subtree is a category, primitive or complex, and consequently transformations will consist of the rewriting of strings of categories as strings of categories. The structural condition of a transformation will then specify whether a given string of elements is appropriate for the transformation. It therefore contains a string of categories; the string to be transformed has a proper analysis for the transformation if it can be reduced to that string of categories. Transformations thus consist of substitutions, adjunction and deletions, as well as *categorial changes*. The following example should make this more clear; it has no linguistic pretensions, however.

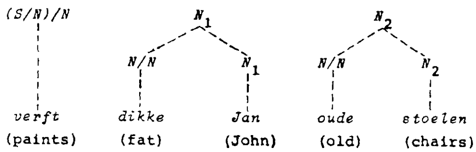
EXAMPLE 4.3. Suppose that the structural condition for the German or Dutch question transformation T_Q (cf. Chapter 2, paragraph 2.2) is $\underset{1}{N} + \underset{2}{(N\backslash S)/X} + \underset{3}{X} + \underset{4}{Y}$, where X and Y may be empty.

Does the Dutch string *dikke Jan verft oude stoelen (fat John paints old chairs)* fall into the domain of T_Q? Let us suppose that the base grammar assigns the following categories to the string: *dikke (fat)* $= N/N$, *Jan* (John) $= N$, *verft (paints)* $= (N\backslash S)/N$, *oude (old)* $= N/N$, *stoelen (chairs)* $= N$, and that the grammar is based on R_1 and R_2 with the corresponding cancellation rules. In that case the sentence can indeed be reduced to the structural condition of T_Q, as follows:



in which $X = N$, and $Y = \lambda$.

The transformation then changes the places of 1 and 2, and alters the category of 2 as follows: $(N\backslash S)/X \rightarrow (S/X)/N$. In general this will yield the new string $\underset{2}{(S/X)/N} + \underset{1}{N} + \underset{3}{X} + \underset{4}{Y}$. Applied to the example, this will yield:



The change of category is necessary to retain the possibility of reduction to S after the transformation. Concerning the structure of transformations we can state that on the one hand it is very easy to indicate the structural condition for a transformation: the domain of a transformation can be given as a string of (possibly complex) categories. But on the other hand it is sometimes necessary to make category changes which are hardly natural or attractive. This occurs especially with deletion transformations, where rather arbitrary category changes are needed.

To close this paragraph, we shall summarize the advantages and disadvantages of the use of categorial grammars for natural languages. Among the advantages, categorial grammars are particularly well suited for representing word order, and for the hierarchical description of phrase structure. Categorial grammars can give a satisfactory formulation of the concept of "endocentric construction". The nonterminal vocabulary is very limited, but functional relations can nevertheless be expressed simply. Unrestricted categorial grammars can easily deal with discontinuities

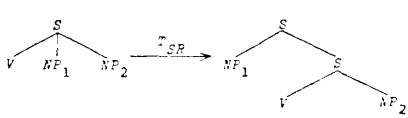
and free word order. Syntax is contained completely in the lexicon. Concerning the transformational component, categorial grammars can formulate structural conditions very elegantly, and in such a way that one can easily determine whether or not a given string satisfies such a condition.

Among the disadvantages, categorial grammars do not represent syntactic dependence satisfactorily, and lexical elements are often assigned multiple or very complex categories. The transformational component presents problems which as yet cannot be treated adequately, such as the arbitrary nature of category changes, especially in cases of deletion.

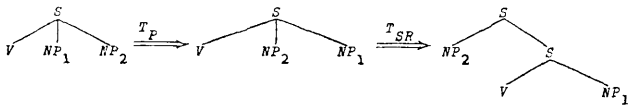
Until now the actual use of categorial grammars has been limited to the solution of problems in other grammars. We shall meet examples of this in Chapter 4, paragraphs 3 and 5. In Chapter 6, paragraph 2 we shall mention a contribution to the development of probabilistic categorial grammars.

4.3. OPERATOR GRAMMARS

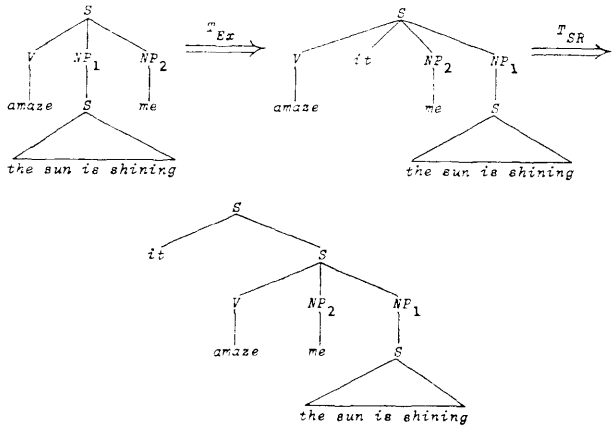
Closely related to categorial grammars, various proposals may be found in linguistic literature to represent base grammars as systems of logic. This implies replacing the subject-and-predicate construction of the *Aspects* theory with constructions of the form predicate-and-arguments, also called operator-and-operands or functor-and-arguments. Some authors take the main verb as the operator, and noun phrases as arguments (Harman 1970); the elementary syntactic rule is thus $S \rightarrow V + NP_1 + NP_2 + \ldots + NP_n$. The role of transformations is essentially the re-ordering of noun phrases. One such substitution is called SUBJECT RAISING (mentioned above in Chapter 3, paragraph 3), T_{SR}. The following diagram shows this transformation for the case where there are two noun phrases:



This gives the usual subject-and-predicate relation. Harman shows that it is in many ways better to place this T_{SR} at the end of the cycle. This can be illustrated with the treatment of the passive transformation. This moves the first *NP* to the end of the string, and when that is done before the T_{SR} takes place, the following transformational sequence is obtained:



A condition for this is that NP_1 contains something like *by* in the underlying form, but Harman gives no details on this. Table 4.1 shows that the verb *amaze* can have a sentence and a noun phrase as its arguments. The basic form for the sentence *that the sun is shining amazes me* is *amaze (the sun is shining, me)*. The same base form can also underlie the sentence *It amazes me that the sun is shining*. Harman shows that this sentence can be obtained by a transformational substitution of arguments followed by *subject raising*. The substitution transformation here is called *extrapolation*, T_{Ex}; it moves NP_1, leaving *it* behind. The cycle is as follows:[1]



[1] The triangles in these diagrams stand for the subtrees the internal structure of which is left unspecified.

Elements other than the main verb, especially *quantifiers* and *negation*, may also be described as operators. The argument of these operators is *S*. Quantifiers also contain *variables*. We shall show that there are important reasons for the introduction of such operators in the description of underlying structures. In Chapter 3, paragraph 3 we noticed that quantifiers (*all*, *every*, *many*, *some*, *a few*, *one*, etc.) lead to problems in the *Aspects* theory. The range or domain of a quantifier can be changed by transformations, and the result of this may be that the transformation is not strictly paraphrastic. In the following example, the passive variant of (1) according to the *Aspect* model is (2), but there is a noticeable difference in meaning between (2) and one of the possible readings of (1).

(1) *many arrows did not hit the target*
(2) *the target was not hit by many arrows*

We would paraphrase (1) with (3), and (2) with (4):

(3) *there are many (arrows which did not hit the target)*
(4) *it is not the case that (many arrows hit the target)*

In these paraphrases, we have placed the range of the operator (respectively *many* and *not*) between parentheses. From this we can see the difference immediately: in (3), and therefore also in (1), the operator *not* lies within the range of the operator *many*, whereas in (4), and therefore also in (2), the operator *many* lies within the range of the operator *not*. Chomsky pointed out this difference in *Aspects* (p. 224), but did not account for it in terms of different deep structures for (1) and (2).

Harman suggests the deep structures (5) and (6) (cf. Figure 4.3) for sentences (1) and (2) respectively, introducing the quantifier with the rule $S \rightarrow QS$, and limiting the range of the quantifier to the following *S*. The variables in these constructions make it possible to indicate the identity of certain elements in the deep structure. This is a very general need in linguistics, and is not limited to the treatment of quantifiers. Another example is pronominalization: *John washes John* is not synonymous with *John*

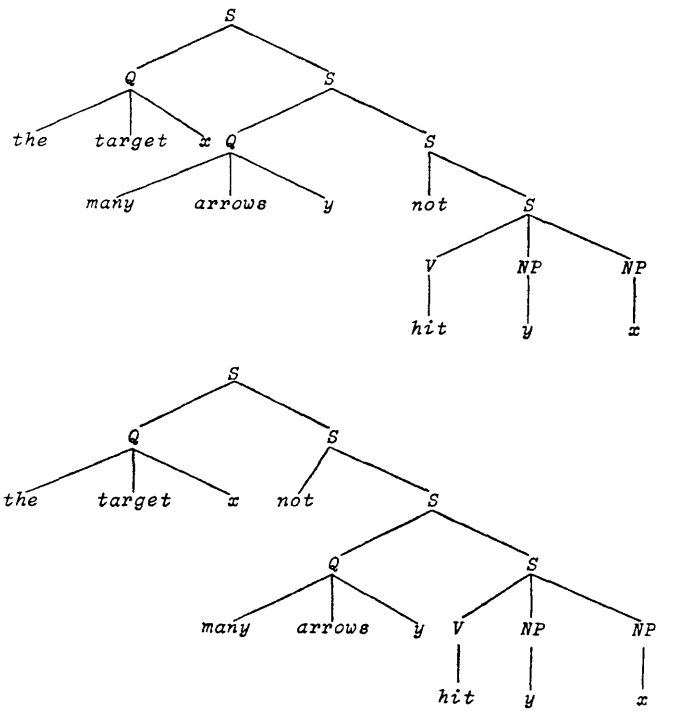


Fig. 4.3. Deep structures for *many arrows did not hit the target* (5), and *the target was not hit by many arrows* (6).

washes himself. The first sentence may be changed to the second only if the first *John* in *John washes John* is identical with the second *John*. This can be indicated in the deep structure by means of variables. Still another example is coordination. In *Clara takes her book and goes to school*, the subject of *goes* is *Clara*. Deletion of a second mention of *Clara* has taken place on the basis of its identity with the first mention. This also may be represented in the deep structure by means of variables, as is the case in (5) and in (6).

Beside verbs, quantifiers, and negation, several other linguistic categories also lend themselves to treatment in terms of operators. Seuren (1969) made one of the first proposals in this field. He

treated not only quantifiers but also *qualifiers* as operators. The latter category takes in the already mentioned *negation*, but also *question* ("I ask if *S*"), *imperative* ("I request that *S*"), *assertion* ("I assert that *S*"), *suggestion* ("I suggest that *S*"), as well as *tense*, sometimes in combination with modal verbs (*can*, *must*, etc.). Nesting of operators is possible in Seuren's base grammar: the result of one operator is the argument of another operator. Thus negation may lie within the range of the question operator and not vice versa. Seuren calls the smallest non-nested element the NUCLEUS; in essence this is a string of subject, main verb, objects, and prepositional phrases. Thus Seuren omits Harman's very starting point, namely the definition of the main verb as an operator and the definition of other phrases as arguments. In this respect Seuren's position is quite remarkable, because in dealing with the nucleus he discusses some of its properties which would justify formulation in terms of operators in that case as well. They are, in particular, the relative lack of importance of the order of elements in the nucleus, and the dominant role of the main verb in the selection of the various phrases within the nucleus. In fact Seuren's nucleus grammar could easily be described as a categorial grammar without ordering restriction (cf. Chapter 4, paragraph 2), in which the main verb has a complex category as in Table 4.1, and the restrictions on the other phrases in the nucleus are stated in the category of the main verb. Categorial grammars were developed precisely for the representation of operator-operand constructions.

Harman's analysis only shows *how* operators can be used in a base grammar. Seuren develops the operator approach in much greater detail, but his is a hybrid system in which operators occur outside the nucleus, but not inside it. Neither author gives a systematic treatment of a transformational component.

The recent operator grammar presented by Harris (1970a) is much more comprehensive. It shows a striking degree of agreement with the work of the generative semanticists, although, due to historical and terminological circumstances, there is no question of any interaction. This is extremely unfortunate since Harris supports his system with an abundance of linguistic analyses which are also

very essential to the generative-semantics point of view, and which are often of the same tenor as the arguments advanced in that camp.

Some difficulty in reading Harris' work is caused by the distributional framework from which he works. His method consists of the isolation of certain distributional dependences among syntactic elements, followed by the systematic description of them. This method of "working back" from the surface contrasts with the generative method in which the grammar is considered as a sentence-generating system. But Harris' operator grammar can also be represented as a generative grammar. In consonance with the general approach of this book, we shall attempt to give a generative summary of the Harris model. For further detail from the linguistic point of view, we refer to the original publication (Harris 1970a).

We shall begin the description with the construction of a KERNEL SENTENCE. This is a very simple sentence, with a minimum of operators; for the moment we shall limit the number of operators to one. The verb is an important operator, and we refer the reader again to Table 4.1, which was composed on the basis of Harris' survey of verb types. In Harris' notation, a verb which is a predicate over two noun phrases is of the category V_{NN}; it should be followed in the base by two noun elements, $V_{NN}+N+N$. This may also be expressed in a rewrite rule, $S \rightarrow V_{NN}+N+N$. Lexical insertion might yield, for example, *stroke*+*John*+*the dog*. Only one transformation is performed on such a string. Harris calls this transformation GLOBAL PROJECTION; it is identical with *subject raising*, in which the first argument changes places with the operator. This will yield a PROTO-SENTENCE, such as *John*+*strokes*+*the dog*. There is a system of morphophonemic rules by which the protosentence is given a morphemic realization, in this case, the kernel sentence *John strokes the dog*.

The kernel sentences of a language are finite in number. The generative power of the grammar resides in two groups of transformations, PARAPHRASTIC TRANSFORMATIONS, and EXPANSIONS (*incremental transformations* in Harris' terminology). Paraphrastic transformations operate on proto-sentences, and, as their name

indicates, do not lead to changes in meaning. At most they lead to changes in the relationship between the sentence and the speaker or the hearer (as in *topic-comment* and *focus* relations). An example of this is the passive transformation, which, operating on the proto-sentence *John strokes the dog*, yields *the dog is stroked by John*. This proto-sentence can in turn be realized as a sentence by means of the morphophonemic rules. But this sentence is not a kernel sentence.

Expansions do not operate on proto-sentences, but on strings in the base. Expansions are obtained by taking one operator as the argument of another operator; this can, in its turn, have other operators as arguments and so forth. This embedding process can continue indefinitely. It is in this way that the hierarchic nesting of predicates comes about. Take the operator *relate to*, for example. It belongs to the category V_{NVV}, and we substitute *I* for *N*, *being hungry* for *V*, and *growing* for *V*. This gives us the basic form *relate to* (*I, being hungry, growing*). By the global projection (subject raising), this form can be transformed into a proto-sentence without further expansion: *I+relate to+being hungry+growing*, which the morphophonemic rules make into *I relate being hungry to growing*. It is possible, however, further to expand the operators *being hungry* and *growing*, both of which are of the type V_N, for example, as follows: *being hungry (John)* and *growing (John)*. The nested basic form will then be:

relate to (I, being hungry (John), growing (John)).

The global projection is then successively applied to each operator from the inside out. This yields the proto-sentence:

I relate to ((John being hungry) (John growing))

Morphophonemic rules transform the proto-sentence into:

I relate John's being hungry to John's growing.

The proto-sentence can undergo more paraphrastic transformations, and other proto-sentences can be constructed which lead to other morphophonemic realizations, such as, for example, *I relate*

John's being hungry to the fact that he is growing or *I relate John's hunger to his growing*. But there is only one sentence which can be derived from a base structure without paraphrastic transformations. If the base structure has undergone no expansion, that sentence is the kernel sentence; if expansion has taken place, the sentence has a special status in Harris' grammar: it is an element of the REPORT LANGUAGE. The report language consists of the sentences which are generated without paraphrastic transformations. Kernel sentences are the simplest sentences in the report language; all the other sentences in it have one or more expansion in their generation history. The idea of a report language is rather surprising in itself. It means that a natural language contains a sub-language in which precisely the same things can be said as in the language as a whole, as paraphrastic transformations retain meaning. The sentences of this sub-language, abstraction made of morphophonemic variations, consist of nestings of predicates.

Lexical insertion takes place first in the base. But paraphrastic transformations and other morphophonemic rules can replace the lexical elements of the proto-sentence,[1] as we have seen in the examples. Just as in generative semantics, lexical insertion need not precede transformations.

The outline given to this point of the formal framework of the Harris theory is summarized in Figure 4.4. The schema in the figure has been simplified in that diagonal lines have been omitted. The morphophonemic rules, and especially the paraphrastic transformations, can yield the same phonemic form for different proto-sentences; when this occurs, the sentences are ambiguous. Such is the framework of the Harris operator grammar. Two remarks will now be made, on the language and on the report language generated by that grammar.

(i) The grammar generates more than the sentences usually called "grammatical", for there is no lexical or transformational mechan-

[1] Harris describes the paraphrastic transformations as a part of the morphophonemic system. It is not clear to what extent it is more desirable to have the paraphrastic transformations precede other morphophonemic rules than to mix them.

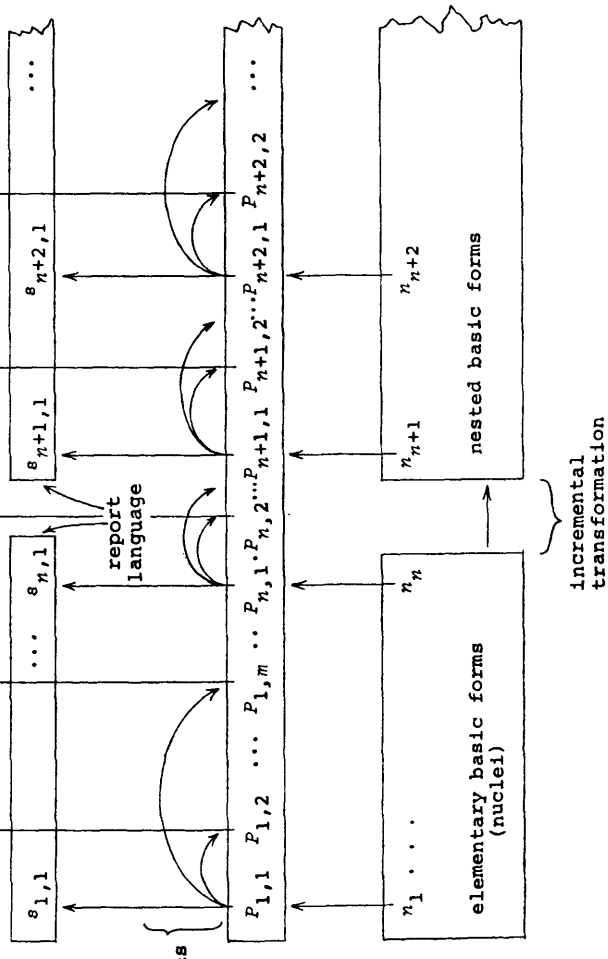


Fig. 4.4. Schema of the Harris Operator Grammar.

ism to block such sentences as *the apple eats the boy* or *I am knowing*. It lacks the selectional features mentioned in Chapter 3, paragraph 1.1. For Harris, however, this is anything but a disadvantage. In his opinion, restrictions of this sort are made by the *universe of discourse*, the state of affairs of the world of which we speak, or at least our knowledge of it. Suppose, for example, that neurophysiologists discovered that "knowing" is a neurological process; for them, then, a sentence like *I am knowing* could be completely acceptable. As for *the apple eats the boy*, there could be huge apples on the other side of the moon, with skin pores so enormous that an astronaut could easily disappear into them. The point Harris wishes to make is that language must be communicative in *every* domain of discourse. Only when the linguist takes a particular domain (the language of weather reports, the language of chemistry, etc.) for further analysis can such selectional features be introduced meaningfully. They can then be integrated into the system of paraphrastic transformations, and can lead to the blocking of some sentence forms.

(ii) The report language is minimal in more than one respect. While all information can be expressed in the report language, it will often be in a very "meager" form. The syntactic structure of the report language is very simple, as we have seen, but the language is also minimal from a lexical point of view, as the number of operators in the base is held to a minimum. Harris wishes the report language to have the smallest possible set of operators which are as general as possible. All other operators are derived inasmuch as they are introduced by means of the paraphrastic transformations for the replacement of a nesting of elementary operators. An example might be the following: adverbs of measure can generally modify verbs as well as adjectives: *he works a little* and *this bike is a little clumsy*. Some adverbs of measure, however, cannot easily be combined with verbs. An example of this is *very* in **the dog limps very*. This is a reason to exclude the adverb *very* from the set of primitive operators. In its place we can have *to a great extent*, which has the same meaning, but not the same distributional limitation as *very*. We can say *the dog limps to a great*

extent, although this is hardly a very elegant sentence. The word *very* will therefore not appear in the report language; it can be inserted by a paraphrastic transformation if *to a great extent* has an adjective in its range, such as in a basic form like *to a great extent (is large (the house))*, where *is large* is an adjectival operator. After global projection, a paraphrastic transformation yields *the house is very large.* This transformation is not applicable to *to a great extent (limps (the dog))*, where *limps* is a verbal operator. Various restrictions on the occurrence of lexical elements in certain contexts, treated in the *Aspects* theory by means of subcategorization and selectional features, can therefore also be handled in the Harris system, though not in the base. Thus remark (*i*) should be refined: only in the report language are there no selection restrictions on the combination of lexical elements.

This whole approach very much resembles that of the generative semanticists. The information carried by a sentence can be described by means of a hierarchy of elementary predicates. Parts can be replaced transformationally by "derived" lexical elements. However the information contained in such a derived element is completely contained in the hierarchy of predicates which it replaces. The derivation of *John kills Mary* in Figure 3.9 would fit very well, as far as form is concerned, into the Harris operator grammar. The only difference is that in the figure, the terminal elements other than *John*, *kill*, and *Mary* are abstract semantic primitives, while in the Harris model they would represent ordinary morphemes. It is noteworthy, however, that in generative semantics an appropriate verbal form can always be found for a semantic primitive, and this strengthens the impression that such a limited *report language* does indeed exist. It would be very interesting to see research done on the extent to which transformational lexical insertion, such as in Figure 3.9, is *optional.* If that extent proved to be great, a report language could also be defined on the basis of generative semantics, and all information could be expressed in it with a minimal vocabulary and an extremely limited syntax.

Finally, we shall discuss a few classes of operators in the Harris grammar. *Verbs* are operators, as shown in Table 4.1. Some verbs

are composed of more than one word *(relate to)*, but little linguistic analysis is available on this class of operators. One might wonder if Fillmore's verb analyses (cf. Fillmore 1969, et al.) satisfy the distributional restrictions required by Harris. For the present it is not known which verbs the report language should contain.

The progressive form has an elementary operator, *be in the process of*, which is the basis of such sentences as *I am writing*. We have already mentioned *to a great extent* as a measure operator. Adverbs of manner have the form *be of x manner*, in which *x* is *slow*, *quick*, etc. The subjunctive mood is generated by means of a *demand* operator. Time relations among subsentences are all based on the operator *be before*: the sentence *John comes after Peter* is based on *be before (come (Peter), come (John))*. Similarly, the operator for comparatives is *be more than*. Reference-identity, which is a condition for certain deletion transformations, such as *Clara takes the book and (Clara) goes to school*, are performed by means of a *sameness operator*, which is also important in the derivation of adjectives and relative clauses. The operator *and* is also involved in these derivations. The operators *and* and *or* are the only ones in the Harris system which are not predicates. These operators can be demonstrated together in the following adjective derivation:

Nested basic form: *Same* (N_1, N_2) (*and*(($V_N(N_1)$, $V_N(N_2)$)))
Example: *Same* (dog_1, dog_2) (*and*((*limps*(dog_1), *be old* (dog_2)))
After global projection: N_1V_N *and* N_2V_N *and* N_1 *is the same as* N_2
Example: dog_1 *limps and* dog_2 *is old and* dog_1 *is the same as* dog_2

This is report language; a paraphrastic relative transformation yields: *the dog which is old limps*, which, after an adjective transformation, yields: *the old dog limps*.

In the Harris grammar, adjectives are essentially subcategories of verbs. This also holds for plural morphemes, and some conjunctions are also taken as verbs. Thus, *because* is an operator of the form V_{VV}, a predicate with two arguments. Adverbs and negation can likewise be treated as verbs. We can thus characterize operators in

the Harris model concisely as consisting of predicative verbs with nouns or verbs as arguments, as well as of *and* and *or* which are operators but not predicates. Finally, *tense* is also an operator;[1] it is perhaps the last or one of the last to be applied to a basic form. Harris is not very explicit about the ordering of operators, however, and in general there are many formal questions in his model which remain unanswered.

The most detailed treatment is still that of the base grammar. The other subsystem, the morphophonemic rules including paraphrastic transformations are given little attention, not only from a linguistic, but also from a formal point of view. The manner in which the domain of a transformation is defined remains an open question, as does the precise form of a morphophonemic rule. The restrictions on alternation of transformations which introduce lexical elements and other transformations are also unknown. Such restrictions decidedly do exist, but they are not formulated by Harris any more than they are in generative semantics (for a similar criticism of the latter, see Fodor 1970). The base of the Harris grammar is beyond doubt a context-free grammar, but little can be said of the generative power of this operator grammar without a more precise definition of the morphophonemic system.

The advantages of operator grammars might therefore be summarized as follows. They are very well suited for detailed representation of functional relations, for the argument on which a given predicate operates is always explicitly stated. Moreover all these predicates are elementary operations, and consequently, in semantic analysis, no further atomization of these predicative relations is necessary. The far reaching distinction between a base with expansions as recursive rules, and a paraphrastic morphophonemic component, is also attractive, at least as an empirical challenge. Can one indeed distinguish a report language with sentences in a simple logical form, on which the whole language is based? The nonterminal vocabulary in an operator grammar is very limited, and the base is extremely simple; various divergent

[1] It is by *tense* that a kernel sentence can be based on more than one operator.

phenomena such as verbs, quantifiers and tense are all treated uniformly.

Many of the disadvantages of operator grammars can be traced to the fact that the transformational component has undergone only a rudimentary elaboration. Both Harman and Seuren give only incidental information on the subject. In the Harris system the morphophonemic rules, including the paraphrastic transformations, are a closed book. We do not know how word order is determined (abstraction made of the "global projection"), nor do we know how words form groups and subgroups, how discontinuous constituents come into being, or what the internal structure is of operators which consist of more than one morph. There is still no formal basis either for the distinction between exocentric and endocentric constructions, nor for a separate treatment of sentence adjunction. However it does seem that in the Harris grammar this latter, as well as the endocentric relation, is characterized by a derivation in which the operator *and* plays a role (as we have seen in the derivation of *old dog*). The Harris grammar, moreover, is very limited in the treatment of selection restrictions, and it has not been shown that lexical insertion can be handled adequately within this system without great formal difficulties.

4.4. ADJUNCTION GRAMMARS

An adjunction grammar might be called a "grammar of modifiers". The idea is that the sentence has a very simple frame which can be made more complicated only by the addition of modifiers or adjuncts. Inversely, one can successively cancel the adjuncts of a given sentence, without losing the status of "sentence". In *crowds from the whole countryside demanding their rights surrounded the palace*, we can first cancel *whole* and *their*, then *from the countryside* and *demanding rights*, leaving finally *crowds surrounded the palace* With every cancellation the string still remained a sentence. One could also say that a sentence contains endocentric constructions, and the modifiers of all those constructions can be cancelled, so that only a grammatical string of heads remains. The adjunct

which is cancelled is either a single word (*whole, their*) or a string (*from the countryside, demanding rights*). The latter are exocentric phrases, as is the remaining "sentence frame". We call such sentence frames CENTER STRINGS (this corresponds roughly to Seuren's *nuclei*); the modifiers are called ADJUNCTS. In natural languages we see that adjuncts are not only added to elements of the center string or its expansions, but also to the center string as a whole (in such a case they are called SENTENCE ADJUNCTS). Is it possible to characterize a language completely, with a finite set of center strings, a finite set of adjuncts, and a system of rules which regulates the way in which adjuncts and center strings are joined?

This thought was developed in very much linguistic detail by Harris (1968), and may be seen as the beginning of his operator grammar, which in fact contains an elaboration in detail of the internal relational structure of center strings and adjuncts. Harris' work since 1959 in *string analysis* (cf. Harris 1970b) provides a good deal of linguistic justification for adjunct grammars, and moreover, there is a detailed formal treatment of them. It was Joshi who developed a formal theory for adjunct grammars, much as Peters and Ritchie formalized the *Aspects* theory. Joshi also expanded Harris' original work on a number of important linguistic points. The Joshi adjunct grammar and the *Aspects* model are the only linguistically interesting mixed models, the transformational components of which have been formalized, and the generative powers of which are known. The Joshi grammar is usually called MIXED ADJUNCT GRAMMAR: *MAG*. One of the best and most extensive computer programs for syntactic analysis (Sager 1967) is based on an adjunction agrammar.

From a linguistic point of view, one of the most important contributions made by Joshi (Joshi et al 1972a, b; Joshi 1972) is the addition of a new category of segments, the REPLACERS, to the existing classification of center strings and adjuncts. Replacers are themselves center strings, but they may also replace an element in another center string. This is a form of sentence embedding. In the sentence *John tells that his bike has been stolen, that his bike has been stolen* is the replacer for *S* in *John tells S*.

However, Joshi's center strings, adjuncts and replacers are not strings of words, but rather of grammatical categories. A few examples of center strings are the following: *NtV* (*John will walk*; *t* stands for *tense*), *NtVN* (*John eats apples*), *NtVNPN* (*John gave the book to Charles*; *P* stands for "preposition"). These are ordinary elementary sentence forms, much like those in Table 4.1 and those in Seuren's grammar, with the exception that tense here plays a role in the center string, while this was not the case for Seuren's nuclei.

A MIXED ADJUNCT GRAMMAR *MAG* has a base which contains an adjunct grammar *AG*, and a transformational component *T*. We shall begin with a description of the base.

There are various proposals for the base. The differences reside in the characterization of the three types of segments: center string, adjuncts, and replacers. We shall follow the simplest and most graceful proposal, namely, to have the three sets coincide; adjuncts and replacers are also center strings, and every center string can, in principle, act as an adjunct or as a replacer.

The center strings which figure in a given grammar, and the conditions under which they may be adjuncts or replacers are established in the JUNCTION RULES and the REPLACING RULES, which comprise the categorial component of the base grammar. We define as follows. The categorial component of the base grammar is an ADJUNCT GRAMMAR $AG = (\Sigma, J, R)$. Σ here represents a finite set of center strings; each center string is a finite string of elements over a vocabulary C of categories, of which S is an element. J represents the finite set of JUNCTION RULES, and R represents a finite set of REPLACING RULES.

A JUNCTION RULE indicates (i) which center string is the "host", (ii) which center string is the adjunct, (iii) to which element of the host the adjunct is adjoined, and whether this occurs to the right or to the left of that element. In formal terms, a junction rule is a triad $u = (\sigma_i, \sigma_j, l_k)$, or $u = (\sigma_i, \sigma_j, r_k)$, where $\sigma_i, \sigma_j \in \Sigma$, and σ_i is the host, σ_j is the adjunct, and l_k and r_k indicate respectively that the adjunct is added to the left or to the right of the k^{th} element of σ_i.

Suppose, for example, that $u = (NtVN, NtV, l_4)$. This adjunction will then yield the following compound string: $NtV((NtV)N)$.

If $u' = (NtVN, NtV, r_4)$, then the string $NtV(N(NtV))$ follows, and if $u'' = (NtVN, NtV, r_3)$, we have $Nt(V(NtV))N$. The parentheses indicate the element of the host to which the segment is adjoined.

A junction rule is not only applicable to the center string indicated, but it may also be applied to all strings derived from the center string. The center strings of u and u' are the same, $NtVN$. We first apply u to $\sigma = NtVN$, then derive $\sigma' = NtV((NtV)N)$. We can now apply u' to σ', because it is derived from the correct center string. When u' is applied, NtV must be added to the fourth element of the original center string, thus to N. The result is $\sigma'' = NtV((NtV)N(NtV))$. The fourth element has now received adjuncts both to the left and to the right. If successive adjuncts are added at the same side of the element, by convention they are always inserted directly next to the element. Other adjuncts already present move one place over. A special case of adjunction, sentence adjunction, will be discussed shortly, in the treatment of the replacing rules.

A REPLACING RULE indicates (i) which center string is the host (that center string containing at least one S) (ii) which nucleus is the replacer of S, (iii) which S is replaced, if there is more than one S. In formal terms, a replacing rule is a triad $r = (\sigma_i, \sigma_j, k)$, which means that σ_j replaces the k^{th} S-element. As in practice there is often only one S in the center string, the k $(= 1)$ may be omitted. Thus $r = (\sigma_i, \sigma_j)$ means that σ_j replaces the only S in σ_i.

If, for example, $r = (NtVSPN, NtV)$, this will yield the string $NtV[NtV]PN$. The brackets indicate that the segment is inserted by replacement. As was the case with the junction rules, it holds that the replacing rule is applicable not only to the indicated center string ($NtVSPN$ in the example), but also to all its derivations, regardless of whether these are the result of adjunction or replacement. A condition for the application of this is, of course, that the S with the correct number k in the original center string be maintained. No further replacing rule may therefore be applied to the string $NtV[NtV]PN$, because no further S from the original center string is available. However this string may still undergo further adjunctions.

Until now, we have only spoken of adjuncts as additions to an element in the center string. But there are very good linguistic arguments for the admission of another type of adjunct, the SENTENCE ADJUNCTION. Compare, for example, the sentences *John works hard* and *John works sometimes. Hard* refers here to *work*, but *sometimes* refers to the whole sentence *John works*. Therefore the sentence *sometimes John works* is more acceptable than *hard John works*. The first sentence, moreover, can be paraphrased as *John is a hard worker*, but the second cannot be paraphrased as **John is a sometimes worker*. Thus *sometimes* is a sentence adjunction. The adjunct grammar contains a special rule for sentence adjunction according to which an adjunct is assigned not to an element, but to a whole string. The rule has the form $(\sigma_i, \sigma_j, l_s)$ or $(\sigma_i, \sigma_j, r_s)$, which means that the adjunct σ_j is added to the center string σ_i as a whole, either on the left or on the right. One might wonder why a sentence adjunction cannot be accomplished by means of a standard junction rule, in which an adjunct is assigned to an element S in the center string. The answer is simply that it would be impossible in that way to add sentence adjunctions to the original center strings in Σ, for these are not the result of rewrites of S, as was the case in phrase structure grammars. Given the sentence adjunction rule, moreover, it is not necessary to make adjunction to an S element possible by means of the standard adjunction rule. This is excluded by the following convention: the elements S are not numbered with the other elements of a center string; they are numbered separately for the use in replacing rules only.

In an adjunct grammar, it holds for replacers and adjuncts in the same way as for hosts that they need not be the center string indicated in the rule, but only that they be derived from it. If, for example, NtV can be a replacer for the S in $NtVSPN$, then every derivation of NtV can also play the role of replacer. Suppose, for example, that there is a junction rule $(NtV, NtVN, r_1)$; in that case not only NtV, but also $(N(NtVN))tV$ can replace the specified S, and this would yield the string $NtV[(N(NtVN))tV]PN$.

Thus the two types of rules form a recursive system over the set

of center strings Σ. A string is said to be *derived from a center string* in Σ, if it is obtained from that center string by the application of zero or more adjunctions and/or replacements. Although this is not essential, for the rest of the discussion a derivation can best be imagined as going "from the bottom up" and not "from the top down", as was the case in phrase structure grammars. In other words, a string which is inserted as an adjunct or a replacer may not be altered in further derivations. The very last phrase of the derivation is therefore the insertion of all the "prefabricated" strings in the "matrix center string".

The LANGUAGE generated by an adjunct grammar is the set of strings in $(C-S)^*$, i.e. strings of categories in which S does not occur, and which can be derived from the center strings in Σ by means of R and J. This language is *preterminal* because the strings contain no lexical elements. Obviously the center strings in which S does not occur are also sentences of this preterminal language.

A system of lexical insertion rules is appended to this categorial adjunct grammar, much as was the case in the *Aspects* theory. These rules replace the preterminal category symbols with terminal lexical elements. A lexical element is "attached" to the category elements of the center string at the moment the latter is inserted into another string. The last operation of insertion applies, of course to the "matrix" center string itself. The great advantage of this method is that it makes it possible in a simple way to indicate the selection restrictions which concern (i) the relations within a center string or nucleus (quite in agreement with Seuren's approach in that regard), (ii) the relations among elements in the host or the adjuncts already present in it, and elements of the inserted center string. The form of the lexical rules in the Joshi grammar, however, has not been further elaborated, and we shall leave it out of the following discussion. We will now round up the discussion of the base grammar by another example.

EXAMPLE 4.4. Let $AG = (\Sigma, J, R)$, with

$\Sigma = \{\sigma_1 = NtVN, \sigma_2 = NtVS\}$

$J = \{u_1 = (NtVN, NtVN, r_4)\}$

$R = \{r_1 = (NtVS, NtVN)\}$

We use u_1 first. This may be done only if we fill in the lexical elements for the two center strings *NtVN* at the same time. Let us suppose that the lexical rules allow that u_1 concerns the following terminal strings:

$u_1 = ($ *N t V N* , *N t V N* , r_4),

John inf read article Charles pt write article

where *inf* stands for "infinitive" and *pt* for "past tense". This yields:

N t V (N (N t V N))

John inf read (article (Charles pt write article))

To this we apply replacing rule r_1, i.e. we use this result, which is a derivation of *NtVN*, for the replacement of the *S* in *NtVS*. To do so, however, we must first insert the lexical elements into *NtVS*. Suppose that the lexical rules allow the following insertion:

N t V S.

John pt try

The replacing rule r_1 will then yield:

N t V [N t V (N (N t V N))]

John pt try [John inf read (article (Charles pt write article))]

This string, which no longer contains an *S*, comprises the deep structure of the sentence *John tried to read the article that Charles had written.* It is obvious that a complete transformational procedure will be necessary to derive this sentence. But before treating the transformational component, we shall first make a remark on the relation between adjunct grammars and phrase structure grammars.

In the form described here, adjunct grammars are equivalent to a subset of context-free grammars. This is easy to see if we consider adjunct grammars with replacing rules only, and no junction

rules. Replacing rules replace an S in a center string with another center string which in turn can contain an S. The corresponding context-free rule is $S \rightarrow \sigma$, where σ stands for a string in Σ (such as *NtV*, *NtVSPN*, *StV*, etc.). If for every σ in Σ a rule $S' \rightarrow \sigma$ is added, with S' as the start symbol of a context-free grammar, and $V_N = \{S, S'\}$, then that context-free grammar is equivalent to the adjunct grammar. The context-free grammar will then generate phrase markers as in Figure 4.5. It is obvious that the

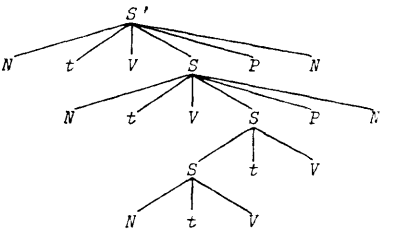


Fig. 4.5. Phrase structure representation of an adjunct grammar structure.

recursive element S is always *directly* derived from S in such a grammar. This is called a context-free grammar with *depth* 1. Such grammars are a strict subset of the set of context-free grammars. Not all context-free languages can be generated by such context-free grammars. But Joshi (to be published) shows that a very minor change in context-free grammars of depth 1 is sufficient to allow them to generate the entire set of context-free languages. It is more important here, however, to know if there is linguistic need of recursive hierarchies of greater depth. This remains an open question, but it is indeed noteworthy that the phrase markers to be found in generative semantic literature generally show no greater depth than 1 (i.e. the recursive element is always introduced by a direct rewrite of itself), or they can easily be reduced to that depth. In Figure 3.9a, for example, we are actually dealing with depth 1, if we replace a pair of rules such as $S \rightarrow Pred + NP$, $NP \rightarrow S$ with the pair $S \rightarrow Pred + NP$, $S \rightarrow Pred + S$. The linguistic need for more hierarchic structure than in sentence embedding is perhaps not very great.

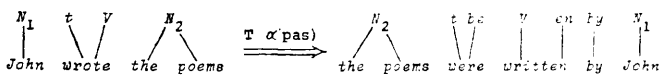
The relations between context-free grammars and adjunct grammars do not change when we also take junction rules into consideration. We can give a junction rule the form of a context-free rewrite rule by the introduction of *dummy* category symbols which will be found in the correct place in the host. For example, the junction rule $u = (NtVN, NtV, l_4)$ can be simulated by the pair $S \rightarrow N + t + V + (S') + N$, $S \rightarrow N + t + V$. The dummy category elements (S' or S'' etc.) are always directly dominated in the tree-diagram by another dummy element or by S. It is indeed the case that the adjunct grammar in this respect is much more elegant than the context-free grammar. Not only does an adjunct grammar clearly show the element to which an adjunct is added, which is not the case for a phrase structure grammar, but it also does not cause senseless multiplication of hierarchic relations when more than one adjunct is added to a single element. This was already mentioned in Chapter 2, paragraph 3.3, and in Chapter 4, paragraph 1, where it was shown that coordination without the use of rule schemas leads to a spurious hierarchic structure. A junction rule can be applied repeatedly without complicating the hierarchy.

There are also various formulations of the *transformational component* of a mixed adjunct grammar. In its simplest form, this component contains two sorts of transformations, α transformations and β transformations.

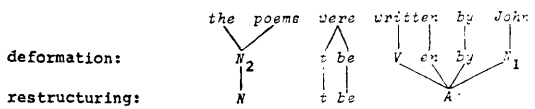
The α transformations are applicable only to elements of Σ (center strings) or strings derived from center strings by means of α transformations. These transformations can delete, adjoin, or substitute elements. Substitution and adjunction can be performed only with elements which are already present and with elements of a finite set given beforehand. The result of a transformation of a string is called the DEFORMATION of that string. The adjunct grammar corresponds closely to the *Aspects* model as far as α transformations are concerned, but it is easier to determine the domain of a transformation with an adjunct grammar. In dealing with the *Aspects* model, Peters and Ritchie needed a very complicated

formulation to establish that a labelled bracketing had a proper analysis for a given transformation. It is a much simpler matter with Joshi's α transformations. An α transformation is defined for a given center string. It indicates the deformation of the center string and its RESTRUCTURING. This latter means simply that by convention the deformation should be considered in the following as some specified center string. An example should illustrate this.

EXAMPLE 4.5. The passive α transformation replaces the center string $N_1 t V N_2$ with the string $N_2 t$ *be* V *en by* N_1. Thus, for example



The result of this transformation is not a center string, but it is established in the definition of passive transformation that the resulting string may be considered to be the center string N *t be* A, that is, the preterminal string for sentences such as *the poems were ugly*. The restructuring is as follows:

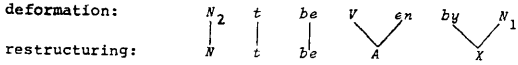


The restructuring also shows the further transformations which may be performed on the string; in this case, it is any transformation which is applicable to the center string N *t be* A. An example of this is the relative clause transformation, which changes *the poems were ugly* to *the poems which were ugly*. In precisely the same way, this transformation changes *the poems were written by John* to *the poems which were written by John*.[1]

Restructuring is not necessarily limited to one center string, and also only a *part* of the deformation may be brought under a new center string. Thus, another restructuring for the deformation

[1] The restructing resulting from the passive transformation demands more detail than we give here: we can derive *the ugly poems*, but not **the written by John poems*. The linguistic elaboration of the transformations in a mixed adjunct grammar is still at its beginnings.

of the passive transformation is the following:

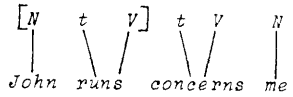


Here *N t be A* is a center string, and *X* is that which remains of the deformation after restructuring. Further transformations, then, only concern the center string. There are certain restrictions on restructuring. Thus the element *t* can never be replaced by *N* or *V*, but only by *t*. The element *V* can only be restructured as *A*, if it is followed by *en* (as in the example) or *ing* (as in *John is writing*) in the deformation. But there is little known about such empirical limitations on restructuring. Attempts have been made to describe the possible restructurings of a deformation by means of a categorial grammar (Hirschman, 1971), in the same way as the *domain* of a transformation is treated in Chapter 4, paragraph 2.

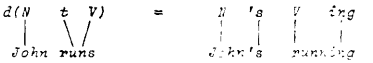
To summarize, we can say that α transformations consist of a center string, a deformation, and one or more restructurings of the deformation. The domain of an α transformation is the center string, or a deformation the restructuring of which is that center string.

The function of β transformations is the transposition of center strings, deformed or not, in the preterminal string. Transposition of a segment means that the point of adjunction of the segment is changed. The points are established in the *R* and *J* rules of the base grammar. A β transformation can therefore be considered as a rule which alters the base rule. It may be said that a β transformation replaces a base rule with a *pseudo rule*. When this occurs, the host is never changed, but when the point of adjunction is changed, the adjunct or replacer may be deformed.

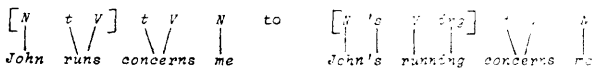
EXAMPLE 4.6. We derive the sentence *John's running concerns me.* The sentence is based on the center strings *NtV (John runs)* and *StVN (S concerns me)*. The insertion of the former into the latter, however, requires the nominalization of *John runs* as *John's running*. We are dealing here with the following replacing rule: $r = (StVN, NtV)$. This yields the string



The β transformation now replaces r with $r' = (StVN, d(NtV))$, where



The change from r to r' thus changes



In this example the point of adjunction is not changed, but the replacer is deformed. When junction rules are applied, the point of adjunction often changes also, for example in *the poem which was ugly* → *the ugly poem*. There are also cases in which an adjunct is divided into segments with different points of adjunction. Take the center string *the proof was concise*, and suppose that we wish to add the adjunct *John proved the theorem* to *the proof*. We then need a β transformation which changes *the proof (John proved the theorem) was concise* to *(John's) proof (of the theorem) was concise*, where the adjunct is divided into two segments with different points of adjunction.

For further details on the transformational component, we refer the reader to the original publications. We might point out that a remark on the ordering of transformations may be found there: the α transformations work cyclically "from bottom to top" just as in *Aspects*, while the β transformations have no extrinsic order.

We wish to close this short discussion of mixed adjunct grammars with a few remarks on a general condition for transformations which Joshi calls the TRACE CONDITION. This condition resembles Chomsky's principle of recoverability of deletions (cf. Chapter 3, page 74, Definition 3.16), but differs from the latter from both linguistic and formal points of view. The trace condition defines a characteristic trace for every transformation. It thus holds that

that traçe cannot be deleted in further transformations. What the trace of a transformation actually is is an empirical question which must be answered separately for each transformation (it is, by the way, also an empirical question whether every transformation does indeed have a trace, but even if this should not hold for some transformations, the mathematical results remain valid). For the sake of example, let us see what the trace of an English passive transformation is. To do this, we begin with the center string *John wrote the poem*, and we use the passive transformation for the derivation *the poem was written by John*. We then use the string thus obtained as a relative clause in the sentence *the poem was ugly*. A β transformation yields the string *the poem (which was written by John) was ugly*. The part between parentheses is that which remains of the passive deformation; it can still further be reduced. The *which deletion* transformation makes this into *the poem (written by John) was ugly*. But this *written by John* is still not a minimal trace of the original passive sentence. Suppose, for example, that we coordinate the sentence obtained with *the poem recited by John was ugly*. Transformationally this yields: *the poem (written by John) was ugly and the poem recited by John was ugly*. The "conjunction reduction" transformation gives *the poem (written) and recited by John was ugly*. We see now that the only thing which remains of the original passive deformation is the word *written*, or, in preterminal terms, $V + en$. If it is impossible to remove this transformationally (and that seems to be the case), we may say that the trace of the passive transformation is $V + en$ At first sight the trace condition seems, like the principle of recoverability, to be bound to the separate transformations. This, however, is not the case. The trace condition regards complete transformational derivations: the trace of a transformation may not be eliminated in the entire further derivation. If each transformation leaves at least one morpheme behind, it is obvious that there is an upper limit to the length of the deep structure of a sentence of a given length, for no more transformations can have taken place than the number of morphemes in the surface structure allows. In the following chapter we shall see that this, as op-

posed to the principle of recoverability, is the guarantee of recursiveness of the transformational grammar.

What, in summary, are the most important advantages and disadvantages of adjunct grammars? Their most noticeable advantage is the explicit distinction of head, adjunct and sentence complement, in contrast with phrase structure grammars in which the distinction between head and adjunct has no natural representation, and no distinction is made between the adjunction (of modifiers) and the replacement (in the form of sentence complements). The constructions which result from adjunction are all endocentric, and all others are exocentric. The mixed adjunct grammar is the only grammar in which this distinction is completely accounted for. The mixed adjunct grammar also offers a strikingly simple solution for the unrestricted coordination of adjuncts; this does not lead to false hierarchy, as is the case with phrase structure grammars. At the same time the amount of hierarchy in these grammars is kept to a minimum, as is the nonterminal vocabulary. This is attractive for theories on the native speaker. In particular, the idea of a small set of center strings or minimal sentence frames which are joined in series in speech is a challenge which psychologists have not yet answered. The formal properties of these grammars are known rather precisely, and, especially in Harris' and in Sager's work, there is a good deal of detailed linguistic "filling".

A mixed adjunct grammar works with rather large units, the center strings. The relations within the center string, consequently, receive very little attention. Functional relations among the elements of the center string, such as dependencies and case relations, can indeed be defined *ad hoc*, but they fit less naturally into the total formal system; this is precisely what Harris sought to work out in his operator grammar. Very little linguistic elaboration has, as yet, been accomplished on the transformational component, and less still on the morphological rules.

4.5. DEPENDENCY GRAMMARS

A phrase structure grammar is not very well suited for describing dependency relations among the elements of a sentence. This becomes very obvious in the treatment of endocentric constructions: a tree-diagram can distinguish neither head nor modifier. Categorial grammars are somewhat more successful in this, as we have seen; in them the head has the same category as the entire constituent. Adjunct grammars were developed especially for the description of head/modifier relations. Endocentric constructions, however, are not the only ones in which linguistic dependency among elements occurs; it can also very well appear in exocentric constructions. In the nuclei of the Seuren model, for example, nominal elements are dependent on the main verb: according to Seuren, it is the main verb which determines the selection restriction for the nominal elements in the nucleus, and not the inverse. Another example is the prepositional phrase, where the noun phrase is dependent on the preposition. Dependency is actually a distributive notion: the syntactic surroundings in which a word group can occur as a whole are determined principally by the independent element, the head of the word group, and the other words contribute very little in this regard. This holds for both endocentric and exocentric constructions. We have seen that the endocentric phrase *old chairs* can occur nearly everywhere *chairs* can occur alone. The word *old* scarcely limits that distribution at all. Correspondingly, the syntactic surroundings in which a prepositional phrase (such as *over the house*) can occur are much more limited by the preposition *(over)* than by the noun phrase *(the house)*; the preposition is the head of the construction.

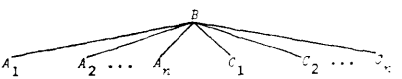
Categorial grammars are suited for expressing only one type of dependency, either the endocentric or the exocentric type. Operator grammars offer a good representation of exocentric dependency, especially the dependency between the main verb and noun groups in the sentence. But endocentric dependencies are represented only indirectly; they go back transformationally to exocentric constructions in the base. Adjunct grammars, finally, give no detailed

analysis of the relations within the center string or nucleus, and in this sense they fail to deal with exocentric dependencies.

DEPENDENCY GRAMMARS were developed especially to express such syntactic dependencies. Like all the other grammars in this chapter, they have the advantage of a very limited nonterminal vocabulary; it consists here only of preterminal syntactic categories, each of which can be replaced only by terminal elements.

A DEPENDENCY GRAMMAR $DG = (V_N, V_T, D, L, T)$ is characterized by a finite NONTERMINAL VOCABULARY V_N, a finite TERMINAL VOCABULARY V_T, a finite set of DEPENDENCY RULES D, a finite set of LEXICAL RULES L, and a set of START SYMBOLS. In the following discussion we shall suppose, for reasons which will be indicated later, that the set of start symbols contains only one element, T.

The DEPENDENCY RULES D indicate for each category in V_N which categories are dependent on it and in which relative position. The rule $B(A_1\ A_2\ \ldots\ A_n\ *\ C_1\ C_2\ \ldots\ C_m)$ means that $A_1, \ldots, A_n$, $C_1, \ldots, C_m$ are dependent on B in the indicated sequence, with B in the place of *. This can be represented graphically by placing B above the string and connected with the dependent elements as follows



The number of dependent elements in a dependency rule is equal to or greater than zero. If it is equal to zero, the rule is as follows: $B(*)$, which means that the element B can occur without dependent elements.

The LEXICAL RULES L are simple rewrite rules of the form $A \rightarrow a$, in which $A \in V_N$ and $a \in V_T$. Although one might expect that an adequate dependency grammar would need a more complicated form of lexical insertion, as was the case in *Aspects*, little is known on the subject. We shall return to this subject, but for the moment we retain this simple rewrite form for the lexical rules.

START SYMBOLS are categories which need not be dependent on another category; they can start a derivation. We do not use the

symbol S for this, however, because all the elements in V_N are preterminal, and we prefer not to have lexical rules of the form $S \rightarrow a$. We suppose that there is only one start symbol, T, with the intuitive meaning of *sentence type* (interrogative, imperative, etc.).

EXAMPLE 4.7. $DG = (V_N, V_T, D, L, T)$, with $V_N = \{D, N, V, T, P\}$, $V_T = \{a, ass, boy, child, gave, ice\ cream, the, to\}$, with the dependency rules D and the lexical rules L as follows:

1. $T(*V)$
2. $V(N*NP)$
3. $P(*N)$
4. $N(D*)$
5. $T \rightarrow ass$
6. $V \rightarrow gave$
7. $N \rightarrow \{boy, child, ice\text{-}cream\}$
8. $P \rightarrow to$
9. $D \rightarrow \{the, a\}$

With this grammar we derive the sentence *the boy gave the ice-cream to a child.* The start symbol occurs only in rule 1; this gives the string TV. By rule 2 the dependents of V are inserted, yielding $TNVNP$, and by rule 3 the dependents of P are introduced, yielding, $TNVNPN$. By applying rule 4 three times, we get $TDNVDNPDN$, the preterminal string from which the sentence desired can be derived by means of lexical rules 5 to 9. This derivation can be represented in a DEPENDENCY DIAGRAM, as in Figure 4.6.

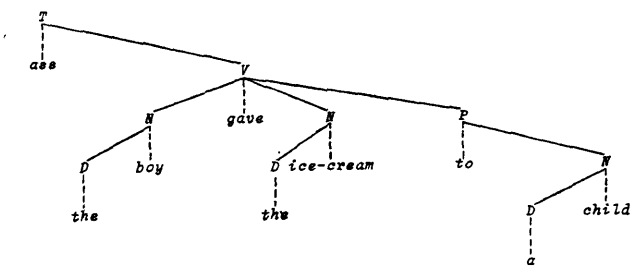


Fig. 4.6. Dependency diagram for the sentence *the boy gave the ice cream to a child.*

In such a diagram we can see the dependency relations from the top of the diagram to the bottom; the category elements in those relations are ordered horizontally according to their position in the

preterminal string. The lexical elements are added, and connected to the diagram by dotted lines. The terminal element *ass* stands for *assertion.*

The DIRECT DEPENDENTS of an element are the elements which are mentioned in the dependency rule. In the example, V is directly dependent on T, and P is directly dependent on V. The INDIRECT DEPENDENTS of an element are the elements which are derived from that element in more than one step. In Figure 4.6 P is indirectly dependent on T. A CONSTITUENT is an element with all its direct and indirect dependents. In the figure, *a, a child, to a child, the boy gave the ice-cream to a child,* etc. are constituents. The HEAD of the constituent is the element which is independent of the other elements in the constituent. Thus *gave* is the head of *the boy gave the ice-cream to a child,* and *to* is the head of *to a child.*

The generative power of a dependency grammar resides in recursive rules, which insert the start symbol T, as, for example, in the rule $N(T^*)$. Gaifman (1965) has proven that dependency grammars are (weakly) equivalent to context-free grammars. The proof, which we shall not treat here, is indirect; it shows the equivalence of dependency grammars and categorial grammars which in turn are equivalent to context-free grammars. It is not difficult to construct an equivalent context-free grammar for a given dependency grammar (the inverse is much more complicated). A context-free grammar equivalent to the dependency grammar in Example 4.7 has the following production rules:

$S \rightarrow T + V'$
$V' \rightarrow N' + V + N' + P'$
$P' \rightarrow P + N'$
$N' \rightarrow D + N$
and rules 5 to 9.

The context-free phrase marker which corresponds to the dependency diagram in Figure 4.6 is given in Figure 4.7. The construction procedure is based on the insertion of an extra nonterminal symbol for each category which can have dependents itself (S for T, N' for N, and P' for P).

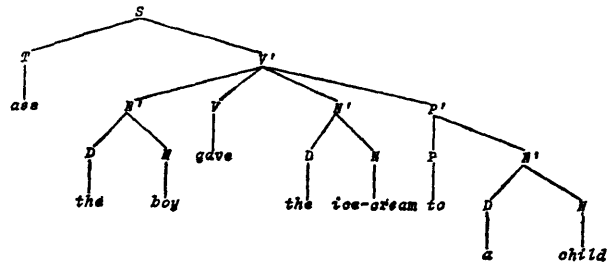


Fig. 4.7. Context-free phrase marker for *the boy gave the ice cream to a child.*

A comparison of Figures 4.6 and 4.7 shows that the former, in contrast to the latter, clearly represents the dependents without excess of hierarchic relations or nonterminal elements. The various relations within the nucleus, the dependency of the various word groups on the main verb or the "direction" of the selection restrictions, are particularly well represented. But on the other hand, the distinction between exocentric and endocentric is lost. The dependency diagram also does not allow one to deduce the type of a constituent. Robinson (1970) gives the following rule: an element with more than one direct dependent is the head of an exocentric construction. Thus in the example, *V* is the head of an exocentric construction. This condition, however, is sufficient but not necessary. The preposition *to* has only one direct dependent, but *to a child* is nevertheless exocentric. The intuitive interpretation of a dependency diagram is rather one of *selection*: the head determines the choice of the dependent elements. Such diagrams, like chemical structures, show the valence of each element (the number of direct dependents which an element can have) and the connected chains in which elements are ordered.

A dependency diagram is perhaps a fitting means for expressing case relations in the base. Caution is necessary in this respect, however, for, despite the work done by Fillmore and others (cf., for example, Fillmore, 1968), research on the formal properties of case relations is still in a very early stage of development. For a thorough linguistic analysis on the subject, in which the formal

system of dependency grammars is used, we refer the reader to Anderson (1971). Without going into much linguistic detail, we shall at this point outline the formal means for the dependency representation of case, following the general lines of the work done by Robinson (1969, 1970).

It is possible to introduce a syntactic category for each case. Such syntactic categories may be rewritten as case morphemes. In English this will generally take the form of a preposition, but it can also take that of a suffix. Let us introduce the following non-terminal symbols, by way of example, without taking position as to the linguistic relevance of the case categories used: *A* for *agent*, where *A* can be rewritten as *by*; *I* for *instrument*, where *I* can be rewritten as *with* (or *by*; we shall return to this later); *Dt* for *dative*, where *Dt* can be rewritten as *to*; *L* for *locative*, where *L* can be rewritten as *in*, *at*; *O* for *objective*, where *O* can be rewritten as *of*. These case categories are introduced into the base as direct dependents of the verb, for example, by a rule such as $V(A^*O\ Dt)$. Each of these categories can then be given an *N* as dependent, by rules such as $A(^*N)$. Thus the underlying structure of a sentence such as *the boy gave the ice-cream to a child* is something like the diagram in Figure 4.8. The lexical elements are inserted for the sake of clarity.

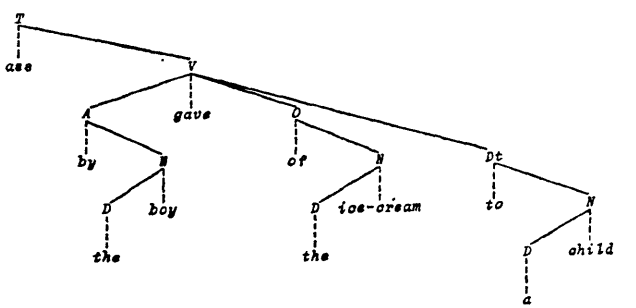


Fig. 4.8. Dependency diagram for *the boy gave the ice cream to a child* with case relations.

Lexical insertion rules, as we have already mentioned, are not simple rewrite rules, and we must now examine this matter more

closely. The verb *give* has a case specification in the lexicon, namely, [*A—O Dt*]. The insertion of the verb is possible only if its syntactic case specification in the lexicon agrees with the structure of the dependency diagram (this is completely analogous to lexical insertion in the *Aspects* model). How does the insertion of a noun take place? It is pointless to give nouns case specifications in the lexicon. Case features are excluded because the majority of nouns can occur in a variety of different case roles. According to Robinson, it is the presence of a "case-related" feature which determines whether or not a given word may fill a given case function. Thus, for the dative and for the agent, it is probably possible only to choose words which are [+ animate]. By a "syntactic redundancy rule", the feature [+ animate] is added to the *N* which is directly dependent on the *A* or *Dt* in the dependency diagram. The two redundancy rules in question are: $N \rightarrow$[+ animate]/ *A*[*—] and $N \rightarrow$ [+ animate]/ *Dt*[*—]. As usual, the surroundings in which the feature is added to *N* are specified to the right of the diagonal "/". Lexical insertion of a given noun may take place only if, according to the lexicon, the noun possesses the feature required. Thus in Figure 4.8 the feature [+animate] is added to the *N* which is directly dependent on *A*, by means of the syntactic redundancy rule. Lexical insertion of *boy* is allowed because in the lexicon, *boy* is specified as [+animate]. Inanimates such as *stone* and *comparison* are therefore excluded as agents. Similar conditions hold for the dative.

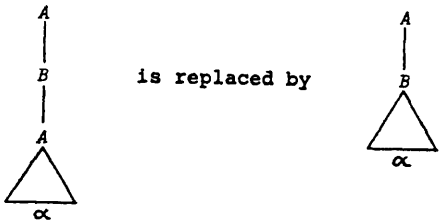
As for prepositions, we suppose that they are specified according to case: *by* has the feature [+*A*], *with* has [+*I*], etc. Their insertion is determined by these case features. But there are also other conditions for the insertion of prepositions. Thus *by* can also be used for *I*, provided that no *A* is specified; compare, for example, *the window was broken by the ladder* and *John broke the window with the ladder*. Prepositions, likewise, are often not realized in lexical insertion. The *by* of the *agent* appears only in passive constructions, and the *to* of the dative is dependent on position; compare *John gave the book to Peter* and *John gave Peter the book*. The objective preposition is realized even less often in lexical

insertion. In this connection we naturally think of transformational mechanisms, but very little is known of their function in the present question.

We close this paragraph with a few remarks on the transformational component of a dependency grammar. Transformations replace dependency diagrams with dependency diagrams. They may be written not only in diagram form, but also in the labelled bracketing notation. For more detail on this, we refer the reader to Robinson (1970). Transformations must be able to delete, adjoin and substitute elements. The deletion of an element presents no problems, if that element has no dependents. The following convention can be introduced for the case in which the element to be deleted does have dependents. If *C* depends directly on *B*, and *B* on *A*, when *B* is deleted, *C* depends directly on *A*. Adjunction makes the element added (possibly together with the constituent dependent on it) dependent on a new head. Substitutions, however, raise all sorts of formal and empirical problems. The matter is still simple when substitution consists only in the interchanging of the dependents of an element, as in the exchange of positions of noun phrases in a passive transformation, for all the elements remain dependent on *V*. It remains an open empirical question, however, if exchanges of roles of head and dependent can take place when the elements are exchanged. In other words, is it possible in the surface structure to reverse such a semantically significant relation? Robinson takes as a working hypothesis that this is not possible.

Like the other grammars in this chapter, dependency grammars have the advantage of a limited nonterminal vocabulary. This, as we have seen, consists entirely of preterminal elements. This offers certain advantages in the transformational component, all the more striking when compared with transformations in the *Aspects* model. In the latter, the output often contains various superfluous category symbols, and *ad hoc* conventions are needed to eliminate them. One such convention, as we have seen, is the reduction convention (cf. Chapter 3, paragraph 2.4): if, after a transformation, such a labelled bracketing as $(_A(_B(_A\alpha)_A)_B)_A$ occurs, the inner

pair of brackets $(_A$, $)_A$ are "automatically" removed. In diagram form,



Another convention is *tree pruning*: remove every embedded *S* from which no more than one branch leads. Thus the path — *A* — *S* — *B* — is simplified to — *A* — *B* —. But also other category symbols often fill no role whatsoever after transformation. Every transformational treatment of the adjective, for example, meets this problem. Robinson (1970) gives the following example of this (after Ross (1967)): Two stages in the transformational derivation of the adjective are given in Figure 4.9.

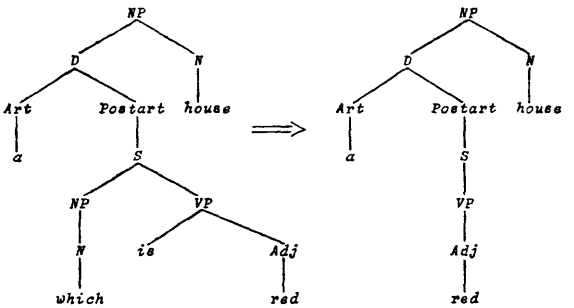


Fig. 4.9. Two stages in the transformational derivation of an adjectival construction (*Aspects* model).

Tree pruning does eliminate *S* from the path of categories which, after transformation, is dominated by *red*, but *Postart* and *VP* remain and have very little intuitive significance. Suitable solutions may, of course, be found for this, but this example shows that the use of an abstract nonterminal vocabulary demands transforma-

tional means with only formal and no intuitive significance. Reduction and tree pruning are pure artifacts of the rule system used, the phrase structure grammar. Robinson (1970) shows how the same two stages look in the dependency system. They are given in Figure 4.10.

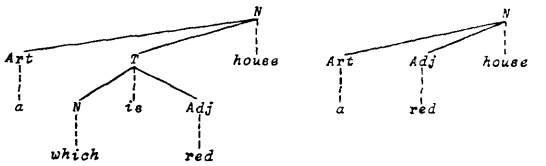


Fig. 4.10. Two stages in the transformational derivation of an adjectival construction (dependency system).

But the intuitive attractiveness of the transformational component can only be judged when we dispose of (a) a complete formalization of the dependency transformations, like the formalizations which already exist for the *Aspects* model and for adjunct transformations, and (b) a detailed analysis of a number of "representative" linguistic cases. At present neither is available. Concerning the formalization, we should point out that for dependency grammars, nothing has yet been done to define a principle of recoverability or a trace condition for the transformations.

To summarize, the advantages of a dependency grammar include the natural representation of distributional dependencies, the limited nonterminal vocabulary, the facility in formulating case relations, and the simple way of accounting for word order and constituent structure. Disadvantages include the limited possibility of distinguishing endocentric constructions from exocentric constructions, and, until now, the uncertainty on the structure of the transformational component and its generative power.

4.6. FINAL REMARKS

The grammars which have been discussed in this chapter differ in many respects. Other mixed models, moreover, are being published

regularly (see, for example, Hudson (1971) and Huddleston (1971) for a formalization of Halliday's *systematic grammar*). The question as to which model is correct is pointless and without answer, for every kind of transformational grammar has its pro and con. The linguist and the language psychologist who seek a model will be guided in their choice by the nature of the phenomenon they wish to study, for some phenomena are naturally representable by one form of grammar, whereas others require a different formalism. All investigators, however, would be served by more detailed data on the weak and strong equivalence of the various transformational grammars. It is usually only for historical reasons that schools of linguistics tend to bind themselves to a particular formal system. If it were shown that different systems were equivalent to a considerable degree, there might be a chance to break through the isolation which is so characteristic of the formation of schools. Where differences are only notational conventions, mathematical linguistics could play an important boundary spanning role by showing how one system might be translated into the other. Where differences concern really substantial questions, only reflection on the possibility of notational translation will allow a judgment on the greater or lesser descriptive adequacy of one or another grammar, or show that the problems in question can be solved more or less independently of the formal system used. Unfortunately at the present stage knowledge about the formal equivalence of the various grammars is still very limited, especially as far as the transformational components are concerned. The practicing linguist has no choice but to acquire some skill in the use of the most important systems, lest he should no longer see the substantial forest because of the formal trees. A decidedly important reason for the use of a transformational mechanism of the *Aspects* type is the simple fact that so many modern linguistic studies are based on the *Aspects* model, and scientific communication is thereby facilitated. But this reason is not sufficient, for that system retains its weak points, and, on the other hand, many important linguistic discoveries have been formulated in other systems.

5

THE GENERATIVE POWER OF TRANSFORMATIONAL GRAMMARS

The conclusion of Chapter 2 stated that the step toward type-0 grammars for the description of natural languages should not be taken lightly. In Chapter 2, paragraph 5 it was argued that for linguistic purposes only grammars should be considered which generate *recursive* sets. In the present chapter we shall discuss the extent to which the *Aspects* model satisfies this condition. We shall also make comparisons with the mixed adjunct grammar, the only other transformational grammar of which the formal structure is known in detail. It will further be shown that the *Aspects* model does indeed generate a type-0 language; the discussion of this in Chapter 3, paragraph 2.3 was not carried out completely, when it appeared that transformations are rule *schemas*.

In the present chapter we shall first show that the *Aspects* theory gives no guarantee of decidability, and moreover, that a transformational grammar of that form, or even of a simpler form, can generate all type-0 languages, that is, all recursively enumerable sets (paragraphs 1 and 2). In paragraph 3 we shall show that this conclusion has serious consequences for linguistics. In paragraph 4, finally, we shall discuss the direction in which solutions to the problem may be sought.

5.1. THE GENERATIVE POWER OF TRANSFORMATIONAL GRAMMARS WITH A CONTEXT-SENSITIVE BASE

Peters and Ritchie (1973) give a strongly restricted definition of "transformation". The *Aspects* model would certainly tolerate

wider definitions. Nevertheless these authors were able to prove that transformational grammars of that form can generate all recursively enumerable sets. In this paragraph we shall follow in some detail the proof that transformational grammars of the *Aspects* type with context-sensitive base grammars can generate all type-0 languages. The proof uses only the elementary deletion transformation, the cyclic character of the transformational component, and the principle of recoverability. Although the base of the *Aspects* theory contains context-sensitive rules (and transformations), we have seen that it is equivalent to a context-free grammar. In the following paragraph we shall follow — in somewhat less detail — the argumentation advanced by Peters and Ritchie that in that case, and even when the base is much more elementary, the generative overcapacity remains. For the latter it will be necessary to use the filtering function of transformations. That property, however, will play no role in the present paragraph.

THEOREM 5.1. Every type-0 language can be generated by a transformational grammar with a context-sensitive base and *Aspects* type transformations.

PROOF. Let $G = (V_N, V_T, P, S)$ be a type-0 grammar. We can suppose, without loss of generality, that all the production rules in P have the form $\chi\alpha\omega \rightarrow \chi\beta\omega$ or $A \rightarrow a$, where χ and ω are strings in V^* (possibly λ), α and β are strings in V_N^* (strings of variables, indefinite in length), $A \in V_N$ and $a \in V_T$. The obvious reason for this is the same as that discussed in the proof of Theorem 2.10 in Volume I.

We first construct a context-sensitive grammar $G' = (V_N', V_T', P', S)$ which has the following relations with G:

(i) $V_T' = V_T \cup b$ (there is one new terminal element b)

(ii) $V_N' = V_N \cup B$ (there is one new nonterminal element B)

(iii) P' is composed as follows:

If $\alpha \rightarrow \beta$ is a production in P and $|\alpha| \leqslant |\beta|$, then $\alpha \rightarrow \beta$ is a production in P' (non-abbreviating productions are taken over unchanged).

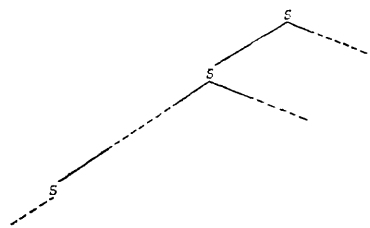
If $\alpha \rightarrow \beta$ is a production in P and $|\alpha| > |\beta|$, then $\alpha \rightarrow \beta B^n$ is

a production in P', where B^n is a string of n successive B's, and $n = |\alpha| - |\beta|$ (β is thus supplemented with B's until the length of α is attained).

For every A in V_N, P' contains a production $BA \rightarrow AB$, and finally P' contains the production $B \rightarrow b$.

G' is thus constructed in such a way that it contains no abbreviating productions; it is therefore a type-1 grammar. The language $L(G')$ generated by it has the following relation to the type-0 language $L(G)$: $L(G)$ contains all and only the strings obtained by the deletion of the terminal element b from the sentences of $L(G')$.

The next step is to construct a context-sensitive base grammar in Kuroda normal-form, and equivalent to G' (cf. Volume I, Chapter 2, paragraph 4.2). Such a grammar B exists (cf. Theorem 2.11 in Volume I) and the reader may remember that in Kuroda normal-form the only productions in which S occurs to the right of the arrow are of the form $S \rightarrow SF$; these are also the only *expanding* productions in the grammar. Because S can never exchange places with another element (in the normal-form a production of the form $AB \rightarrow CD$ never contains an S), the tree-diagram for every sentence $x = a_1 a_2 \ldots a_n$ in $L(B)$ is of the form:



or in (incomplete)[1] labelled bracketing notation, $(_S(_S\ldots(_S a_1)_S a_2)_S\ldots)_S a_n)_S$. It follows from this that each terminal element of sentence x, and in particular the special element b, is the rightmost element of one or another subsentence of x (a subsentence is a string which has the feature "*is an S*" in the labelled bracketing).

[1] "Incomplete" in the sense that each a_i has still other *is a* relations to other nonterminal elements. At least one sequence of direct derivations $S \rightarrow A$, $A \rightarrow a$ must be carried out for the generation of a terminal element. For each a there is therefore at least one extra pair of brackets $(_A$, $)_A$ in a complete labelled bracketing. This, however, is not important to the argument.

The transformational component T of the transformational grammar TG is composed as follows. There is one and only one transformation. This deletes the rightmost element of a subsentence, if that element is b. This means that the transformation is applicable if the bracketing of the subsentence can be divided into two factors, the second of which has the debracketization b. In terms of Definition 3.17, $T = (C, M)$, where C, the structural condition, is $d(\psi^2_{2\to 2}) = b$, and $M = T_d(\psi_{2\to 2})$. The result is the deletion of the interior of the second factor, b with corresponding brackets. Further, we allow the transformational component to work cyclically, according to the *Aspects* theory, first on the most embedded sentence, and thence according to Definition 3.18. Thus each b in the labelled bracketing is successively deleted. This transformation satisfies the principle of recoverability (Definition 3.16), because the structural condition states that $d(\psi^2_{2\to 2})$ is one of a finite number of terminal strings, namely b (see the definition under (ii)). Since the transformation eliminates all the b's from the sentences of $L(B) = L(G')$, it holds that $L(TG) = L(G)$.

The inverse of the proposition also holds, as we see in the following theorem.

THEOREM 5.2. Every transformational grammar with *Aspects* type transformations generates a type-0 language.

PROOF (outline). It follows from the equivalence of type-0 grammars and Turing machines (Theorems 7.1 and 7.2 in Volume I) that it is sufficient to prove that for every transformational grammar with a context-sensitive base there is a Turing machine which accepts $L(TG)$ and only $L(TG)$. In other words, there must be a procedure for the enumeration of the sentences of $L(TG)$ and only the sentences of $L(TG)$. That procedure exists; in its general lines, it is as follows.

Let V_T be the terminal vocabulary of $TG = (B, T)$. Number the strings in V_T^* in the way indicated in Volume I, Chapter 7, paragraph 4. Enumerate the pairs (n, m), where n and m are natural numbers, in the "diagonal" way given in Table 7.1 of Volume I. For every pair (n, m) there is a procedure to determine whether the

string in V_T^* with number m has a deep structure in the transformational grammar TG with no more than n subsentences. Such a procedure exists, for the number of sentences in $L(B)$ with no more than n subsentences is finite (as the rules which introduce S are the only recursive rules in the base grammar (cf. Chapter 3, paragraph 1), and if another context-sensitive base grammar is chosen, there is always an equivalent grammar in Kuroda normal-form which does have this property). That finite number of sentences can be enumerated. In the procedure, T is then applied cyclically to each of those sentences. If the result of this contains the string with number m, the string is accepted and "enumerated". If the procedure yields the string with number m for none of those sentences, it goes on to the following pair (n', m'). In this way the Turing machine generates the sentences of $L(TG)$ and only the sentences of $L(TG)$. Thus, $L(TG) = L(TM)$, and $L(TG)$ is of type-0.

The two theorems in this paragraph show the equivalence of the class of type-0 grammars and the class of transformational grammars with a context-sensitive base and *Aspects* type transformations. We can therefore conclude that such transformational grammars offer no guarantee that the language generated is recursive.

5.2. THE GENERATIVE POWER OF TRANSFORMATIONAL GRAMMARS WITH A SIMPLER BASE

At first glance one might be inclined to attribute the overcapacity of transformational grammars pointed out in the preceding paragraph to the rather strong base, a context-sensitive grammar. But this is not where the difficulty lies. It can be shown, in effect, that the equivalence of transformational grammars of the *Aspects* type and Turing machines also holds when the base is of a simpler form. Proof of this was presented more or less simultaneously and more or less independently by Ginsburg and Hall (1969) for a context-free base, by Kimball (1967) and Salomaa (1972) for a

regular base, and by Peters and Ritchie (1971) for both. We shall follow the formulations given by the last, because it comes closest to that of *Aspects*. Using the filtering function of transformations, they were able to prove a number of theorems, the most important of which we state (without proof):

THEOREM 5.3. Every type-0 language can be generated by a transformational grammar $TG = (B, T)$, where $B = (V_N, V_T, P, S)$, with $V_N = \{S\}$ $V_T = \{a_1, a_2, \ldots, a_n, b, \#\}$, and the following two productions in P:
(i) $S \rightarrow S\,\#$
(ii) $S \rightarrow a_1a_2\ldots a_nb\#$, and where T only contains *Aspects*-type transformations.

Notice here that B is a right-linear grammar, by which a regular language is generated (Theorem 2.1 in Volume I). The language generated is, moreover, of an extremely elementary kind, i.e. $\{a_1a_2\ldots a_nb\#^m | m > 0\}$. Every base sentence consists of the concatenation of the vocabulary, ending with a b followed by a string of boundary symbols of indefinite length. The labelled bracketing for such a sentence in $L(B)$ has the form:

$$(s(s\ldots(s(sa_1a_2\ldots a_nb\#)s\#)s\ldots\#)s\#)s.$$

Peters and Ritchie show that for every type-0 language L there is a series of transformations, as defined in Definition 2.17, by which this trivial regular set can be transformed into L. Every transformation, moreover, satisfies the principle of recoverability.

Even transformational grammars with such degenerate bases generate undecidable sets, if they contain *Aspects*-type transformations. The main reason for that undecidability is that for a given sentence in such a language there is no upper limit to the number of subsentences in the deep structure. A Turing machine for deciding if a given string *does not* belong to the language would be faced with the hopeless task of seeing whether x could be derived transformationally from each of an infinite set of underlying structures. There is therefore no procedure to determine ungrammaticality;

the complement of language L is not recursively enumerable, and L is therefore not recursive.

5.3. LINGUISTIC CONSEQUENCES

The linguistic consequences of the overcapacity of transformational grammars are great. In the first place, the three conclusions of paragraph 5 in Chapter 2 follow directly: (1) the grammar cannot account for intuitions on ungrammaticality, (2) the language is unlearnable, (3) the chance for descriptive adequacy in the grammar is practically lost, and with it, the possibility of an explanatory linguistic theory. We shall illustrate the last point with the following theorem.

THEOREM 5.4. (Universal base). There is a universal base grammar U, from which all natural languages can be derived transformationally.

PROOF. A trivial example of such a grammar is $U = (V_N, V_T, P, S)$, with $V_N = \{S\}$, $V_T = V_{L_1} \cup V_{L_2} \cup \ldots \cup V_{L_n} \cup \{b\} \cup \{\#\}$, where V_{L_i} is the vocabulary of natural language L_i, and P consists of the productions mentioned in Theorem 5.3. With this base, there is a transformational grammar TG for every language L_i.

An important question in general linguistics is whether a universal base can be found for all natural languages (cf. Chapter 1, paragraph 2). A pet idea among transformational linguists is that transformations tend to be peculiar to specific languages, while the base grammars of various languages coincide to a considerable extent. The theorem on the universal base states that such a base exists on purely formal grounds; the statement, in other words, is not an empirical issue, but only a formal triviality.

The base grammar U is indeed universal, but it is clear that it will generate linguistically absurd deep structures. The strong generative power of U is therefore insufficient. The universal base must also be descriptively adequate, and linguists could maintain

that it is very much an empirical question whether a descriptively adequate universal base can be found. But this appears not to be the case. Peters and Ritchie (1969) show that if the class of transformational grammars is limited to those grammars which have an upper limit to the number of subsentences in the deep structure of a sentence (for example, a limit which is a function of the length of the sentence), universal bases exist which have a strong generative power sufficient for linguistic purposes. More specifically, Peters and Ritchie define "sufficient strong generative capacity" as follows. Such transformational grammars can account for intuitions on:

(i) the grammaticality and ungrammaticality of sentences,
(ii) the number of different structural descriptions which an ambiguous sentence should have,
(iii) which sentences are paraphrases of each other in at least one respect.[1]

The introduction of the upper limit means, of course, that we only consider the transformational grammars which generate recursive languages. The Turing machine in the preceding paragraph which was to decide on ungrammaticality is no longer confronted with what we have called a "hopeless task". The number of deep structures it must examine is now limited to a certain number of subsentences, and a decision can be made after a finite number of steps in the procedure. If we make the transformational grammar decidable by building an upper limit into it, then (i) will follow automatically. The number of different deep structures at the base of a given sentence x also becomes decidable, and for the same reason only a finite number of deep structures need be examined in order to determine how many of them lead to a transformational derivation of x. From this, (ii) follows. A similar argument holds for (iii): given sentence x, there is only a finite number of deep structures for x, each of which leads to only a finite number of transformational derivations, one of which is x.

[1] One might wonder if no requirements should be stated on the parsing of the sentences generated.

The other transformational derivations (non-x) of these deep structures are precisely the sentences which are paraphrases of x in at least one respect.

Provided that we suppose that there are transformational grammars for natural languages, with the extra, but extremely reasonable restriction that for every sentence there is a certain upper limit on the size of the deep structure, we can state that there must be a universal base by which such transformational grammars possess the descriptive adequacy specified in (i), (ii) and (iii). Thus also from the point of view of descriptive adequacy, the question as to whether or not a universal base grammar exists is no empirical question. For purely formal reasons, there is a class of bases which satisfy all three requirements, and we shall, thus, never be able to tell which of two universal bases is correct, if both belong to that class.

More serious still, we cannot even decide if a base for a particular natural language is the correct one. If that were possible, we would in principle be able to decide that two natural languages have different bases, which would conflict with both the strong and the weak versions of the theorem on the universal base.

The importance of this impasse for linguistics should not be underestimated. The whole controversy between generative and interpretative semanticists, for example (cf. Chapter 3, paragraph 3), is carried on in transformational terms which do not differ essentially from the formulation used by Peters and Ritchie. Where deviations do occur, namely, by the addition of *derivational constraints* on transformational derivations, this only leads to increases in the generative power of the grammar, and not to reductions of it. As long as both parties work inside the class of universal bases, it will remain impossible to tell who is right. The controversy does not deal with an empirical question, and it is not unlikely that this is, entirely or to a great extent, the case. This appears in the practice of generative semanticists of freely using that property which precisely leads to undecidability, namely, extremely extensive deep structures. One can often see a veritable morbid growth of recursive embedding for the descrip-

tion of a three or four word sentence; this is the case, for example in Figure 3.9a.

To summarize, we can state that the main problems of linguistics are insolvable by the formal means of the *Aspects* type. Such a linguistic theory can make no judgment on observational adequacy, if this is defined as the possibility of deciding whether or not sentence x belongs to language L. Nor can it satisfy descriptive adequacy (no decision can be made as to which grammar offers the "correct" structural description), or explanatory adequacy (no decision can be made as to which is the universal base, and the theory can give no account of the learnability of natural languages).

5.4. SOLUTIONS FOR THE PROBLEM OF TRANSFORMATIONAL OVERCAPACITY

The principal cause of the undecidability of *Aspects* type transformational grammars is the fact that there is no upper limit to the size of the deep structure of a given sentence. As a consequence of this, an infinite number of underlying structures must be examined in order to make the decision "x is not in L". It was precisely the purpose of the principle of recoverability to avoid this. The principle should have guaranteed that a Turing machine (and, in the abstract, the native speaker) be able, for every string of words, either to reconstruct the deep structures, or to state that there are no deep structures for the string. But on reflection, it is striking to notice how poorly the *Aspects* definition (faithfully formalized in Definition 3.16) fulfills that original purpose. The principle guarantees only that if a labelled bracketing and the transformations by which it was derived are given, it will be possible to reconstruct the original labelled bracketing. For the reconstruction of the deep structure, therefore, the Turing machine would also have to dispose of a list of the transformations used, or at least of the maximum number of transformations which can be performed in a derivation. This would guarantee that no more than a finite number of transformational derivations need be reconstructed.

For each of these it could be determined if the first labelled bracketing in that derivation is generated by the base grammar. The principle, however, does not provide such a guarantee. It allows that for every labelled bracketing there is an earlier labelled bracketing, since a cycle of transformations can always have come before. Suppose, for example, that the word "a" is a sentence in $L(TG)$. It follows from the construction in the proof that the sentence can be derived from each of the infinite number of deep structures $(sa)s$, $(s(sa)sb)s$, $(s(s(sa)sb)sb)s$, ...

A step in the direction of a solution would therefore be to set a limit on this unrestricted cyclic capacity of the transformational component. There are two ways in which this might be done. The first is empirically to establish whether or not in current linguistic practice any upper limit to the number of subsentences in the deep structure is implicitly taken into consideration. Peters and Ritchie (1973) suppose that this is indeed the case. They state that a number k can certainly be found for which a sentence x of length $|x|$ has fewer than $k|x|$ subsentences in its deep structure. They show that this is sufficient to guarantee the recursiveness of such transformational grammars. But this is a very non-committed method. What is needed is an argument for that upper limit. The second way is therefore more interesting: is it possible to change the definition of transformation (including the principle of recoverability) in such a way that the upper limit will automatically follow from it? This has not yet been done for *Aspects* transformations. The only mixed model for which it has been done is the Joshi adjunct grammar. In Chapter 4, paragraph 4 we discussed the trace condition in that grammar. The trace condition requires that each transformation leave one or more elements behind, and that those elements (or that element) may no longer be deleted by further transformations. It is obvious that for a sentence of a given length there is an upper limit to the number of transformations which are applied to that sentence in derivation, and Joshi (1972) shows that this does indeed guarantee the recursiveness of his grammar. From an empirical point of view, however, it remains an open question whether the trace condition holds in all cases. If it holds

for the transformations of an adjunct grammar, it need not necessarily hold for the transformations of grammars of the *Aspects* type. Moreover, the trace condition is applied to the transformational component as a whole, and not to individual transformations: the trace of a transformation must remain in every *possible* transformational derivation. It would be a rather heavy empirical task to account for the plausibility of such a condition.

However, more has to be solved than only the problem of decidability. As we have seen, a strong form of the theorem on the universal base is maintained, even if only decidable transformational grammars are taken into consideration. This may be attributed to the filtering function of transformations. Every type-0 language can be derived from a trivial base, by the intensive use of the filtering function of the transformational component.

The filtering possibility should either be eliminated from the model, or at least limited. It would be interesting here to find linguistic arguments for one or another solution, but until now little effort has been made in that direction. An empirically interesting question, for example, is whether a # which occurs within the domain of a particular transformational cycle and is not removed during that cycle can still be eliminated in a later cycle. The *Aspects* theory allows this, but the need of it, from a linguistic point of view, is doubtful. If, for example, a relative clause transformation in a particular cycle fails because the structural condition $NP_1 = NP_2$ is not fulfilled (cf. Chapter 3 paragraph 1.2), it is unlikely that this might be "repaired" in a later cycle. On such ground the filtering function of transformations might be sufficiently limited to give the question of the universal base empirical content.

This all should encourage great reserve concerning the grammatical means used and the range of the results attained. Since the publication of *Aspects*, however, interest in the formal structure of grammars has rather decreased than increased. Very many interesting linguistic phenomena have been discovered and discussed, but their formulations are only details of a theory which as yet

does not exist. Such formulations are always based on implicit or explicit assumptions concerning the theory as a whole, justification is lacking precisely on essential points. The assumption on the universal base, for example, is incorrectly considered empirically verifiable in the present state of theory. History is obviously repeating itself; in 1965 Chomsky wrote:

> The critical problem for a grammatical theory is not a paucity of evidence, but rather the inadequacy of present theories of language to account for masses of evidence that are hardly open to serious question (*Aspects*, 20).

On a different level, this applies as well to the present situation.

6

STATISTICAL INFERENCE IN LINGUISTICS

We have so far been concerned in this volume with linguistic theory, and have not yet treated the interpretation problem (Chapter 1, paragraph 2) from the point of view of formal grammars. Of the three cases in which that problem appears most strikingly, two, the investigation of linguistic intuitions, and the investigation of language acquisition, will be treated in Volume III. In the present chapter we shall deal with a few applications of formal language theory to the third case, statistical inference with respect to the analysis of a corpus. This chapter will not offer a survey of statistical linguistics; the discussion will be limited to two examples which are relevant to psycholinguistics in particular. The aim of the chapter is principally to show that the interpretation problem calls for linguistic methods other than the "usual" ones, and that the widespread opinion that statistical methods are inappropriate in linguistics is not only unfounded, but it is also a hindrance to linguistic research on interpretation. In the first paragraph (6.1) we shall discuss a few aspects of communication theory from the point of view of inference theory; some linguistic applications of communication theory can be considered as statistical inference with respect to regular grammars. In the second paragraph (6.2) we shall show a linguistic application of the material treated in Volume I, Chapter 8, paragraph 2; this will consist of an estimate of parameters for a probabilistic context-free grammar.

6.1. MARKOV-SOURCES AND NATURAL LANGUAGE

In Chapter 2, paragraph 2 we showed that regular grammars are decidedly unsuited for describing natural languages. But there is a class of probabilistic finite automata which has long served as a model in the analysis of natural languages; the class in question is that of *Markov-sources*. Although there is no doubt that such models are inadequate as linguistic theory, it is nevertheless a practical fact that they are often suitable means for the description of rough parameters of verbal communication processes. They are still used as such in applied communication theory. These rough parameters refer to that which is called the *information value* of the verbal message. "Information" in this sense of the word is a quantitative concept, distinct from the content or meaning of the message. Moreover, it is not an absolute, but rather a relative concept. In information theory it is impossible to say how much information an isolated message contains. Information is defined precisely on the basis of the number of alternative messages which the same source could have produced in the same length of time. The information value of a message should indicate, given the source, the probability of that message. The idea is that a message with a probability of 1 contains no information, for the receiver can predict exactly what the source will produce in that length of time. Only when some uncertainty exists, will the message contain information. Information is equal to the amount of uncertainty which the message eliminates.

The nature of the source is determinant for the probabilities of the various messages, and consequently for their information value. If the source is discrete, that is, if it has a finite vocabulary, we can consider it as a probabilistic grammar, a system which generates sentences with particular probabilities (cf. Volume I, Chapter 3). The most important generalizations in communication theory concern right-linear sources (cf. Volume I, Chapter 2, paragraph 3.5), thus sources which generate regular languages. This is not an essential restriction; a context-free probabilistic grammar might also be taken as source, and the definitions of

information, redundancy, etc. would not have to be altered (an example of this has been treated in the discussion of grammar-grammars in Volume I, Chapter 8, paragraph 4). For historical reasons, however, the restriction does exist, and it is carried over into the applications of communication theory to natural languages.

In the simplest case the source of messages is considered as a finite automaton with as many states as vocabulary elements. Each vocabulary element $(a_1, a_2, \ldots, a_n)$ serves as the label of one state $(s_{a_1}, \ldots, s_{a_n})$, and the transition rules are such that the automaton always passes to the state labelled after the element it has just accepted. The state transition function δ thus contains all and only the rules $\delta(s_{a_i}, a_j) = s_{a_j}$ for all s_{a_i}, s_{a_j} in S, and every a in I. It is clear, then, that all the states are connected with each other, and that the automaton is 1-limited. Finally, every state is assigned a probability, normalized on the basis of the *state* (cf. Volume I, Chapter 4, paragraph 4), that is, the total probability of a transition from a given state, over all possible input symbols (the entire vocabulary), is equal to 1. Such an automaton is called a *Markov-source*. Before defining this more formally, we offer the transition diagram for an elementary Markov-source in Figure 6.1. The source has two elements in the

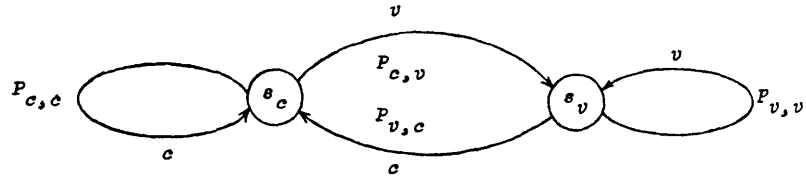


Fig. 6.1. An elementary Markov-source.

input vocabulary, c and v, and therefore two states, s_c and s_v. When the input is v, the automaton passes to state s_v, and when the input is c, to state s_c. The chance that the automaton will go from state s_c to state s_v is indicated in the diagram as $p_{c,v}$ for that transition, of course, the input must be v. The chance for the

opposite transition, from s_v to s_c, is $p_{v,c}$, and the chances for transition to the same state are respectively $p_{c,c}$ and $p_{v,v}$. Normalization means that the total chance of transition from a given state is equal to 1. Thus the total chance of transition form s_c is $p_{c,c} + p_{c,v} = 1$, and the total chance of transition from s_v is $p_{v,c} + p_{v,v} = 1$. We must now determine the state in which the automaton starts, and that in which it stops. In complete analogy with the definition of probabilistic finite automata (Volume I, Chapter 4, paragraph 4), it is customary to consider all the states of a Markov-source as possible initial states, and for each of these states s_i an initial probability $p(s_i)$ is defined. The sum of the initial probabilities is equal to 1. The vector of initial probabilities $(p(s_1), \ldots, p(s_n))$ is called the INITIAL DISTRIBUTION of the Markov-source. It also holds for a Markov-source that every state is a final state. For a certain class of Markov-sources the initial distribution is of little importance; the statistical properties of a long, generated string are, in the limit, independent of the initial state. Markov-sources with this characteristic are called ERGODIC. For them, we can simply suppose that they are generating from infinity, rather than defining an initial distribution. Because every state can be a final state, we can likewise suppose that the source never stops, and generates a string which is infinite both to the left and to the right. Each finite segment of that infinite string is then a sentence (message), generated (accepted) by the Markov-source. As linguistic applications of communication theory always suppose ergodic sources, we shall limit further discussion to this subclass, and omit definition of an initial distribution.

A MARKOV-SOURCE, then, is completely characterized by its finite INPUT VOCABULARY I $\{a_1, \ldots, a_n\}$, and its TRANSITION PROBABILITIES, $p_{i,j}$, where $p_{i,j}$ is defined for all pairs a_i, a_j (with a_i, a_j in I) and stands for the chance that element a_i is followed by element a_j, and in which the probabilities are normalized as follows: $\sum_{j=1}^{n} p_{i,j} = 1$. Because of the one-to-one relation between input vocabulary and state set, it would be redundant to include this last in the characterization of a Markov-source.

Such a Markov-source can quite as well be written as a regular grammar, with rules of the form $A_i \overset{p_{i,j}}{\rightarrow} a_j A_j$ for every pair a_i, a_j in the terminal vocabulary. Such a grammar is thus considered to generate a string infinite to the left and to the right.

The input vocabulary and the transition probabilities for the Markov-source in Figure 6.1 are given in the following transition matrix, which gives a complete characterization of the source:

P	v	c
v	$p_{v,v}$	$p_{v,c}$
c	$p_{c,v}$	$p_{c,c}$

This source is a linguistic example *par excellence*. It is the model which Markov (1913) constructed for the description of the sequence of vowels (v) and consonants (c) in Pushkin's *Eugene Onegin*, and the origin of the Markov theory. It is a clear example of the problem of inference: given a corpus (Pushkin's text) and a grammar (the finite automaton in Figure 6.1), can the transition probabilities be estimated? Markov found estimates by determining, for 20,000 pairs of consecutive letters (*digrams*), to which of the four categories, *vv*, *vc*, *cv*, *cc*, they belonged. He found the frequencies given in Table 6.1, with the corresponding transition

TABLE 6.1. Digram frequencies and transition matrix for *Eugene Onegin*.

	vv	*vc*	*cv*	*cc*	*total*
Digram frequencies	1104	7534	7534	3827	19999

Transition probabilities:

P	v	c
v	0.128	0.872
c	0.663	0.337

matrix. (The number of digrams is, of course, one less than the number of letters.) The value $p_{v,v} = 0.128$ means that of the 1000 vowels, an average of 128 were followed by another vowel, etc. There appears to be a preference for the alternation of vowels and consonants, since the chance for two consecutive consonants

or two consecutive vowels is relatively small. How good is this model? Does it, for example, give correct predictions on the chances of *trigrams* such as *vvv*, *vcv*, etc.? If not, is there a better source for the description of vowel/consonant sequences? Before going into these questions, we give a somewhat more ambitious example of a Markov-source in linguistics, not for letter orders, but for word sequences.

Suppose that English has 100,000 words. We can imagine a Markov-source with 100,000 states, corresponding to the 100,000 words. The source can again be characterized completely by its transition matrix P. The matrix element $p_{i,j}$ stands for the chance that word i is followed by word j. The matrix will thus contain $100{,}000^2 = 10^{10}$ probabilities. Since the source is normalized, the rows of the matrix add up to 1. In each row, therefore, there are $100{,}000 - 1$ *independent* p-values, and the model contains a total of $10^{10} - 10^5$ independent parameters. It holds in general that a Markov-source with n elements in the input vocabulary has $n^2 - n$ independent parameters. Obviously no one has undertaken the impossible task of determining these parameters for English, and it seems excluded that we might make a judgment on the quality of the English generated by this Markov-source. However, means have been found for arriving at some impression of that which is generated by the source. One way is to present speaker A with a word, for example *the*, and to ask him to compose a sentence in which that word occurs. Let us suppose that A forms the sentence *that is the head*. We then go on to the word which follows *the*, namely *head*, and ask speaker B to compose a sentence in which that word occurs. If B in turn produces the sentences *head and feet are parts of the body*, we take the word following *head*, namely *and*, and go on to speaker C, and so forth. The sequence of words obtained in this way, *the*, *head*, *and*, etc., may be considered to be generated by the Markov-source. We call such a sequence a SECOND ORDER APPROXIMATION of English. A FIRST ORDER APPROXIMATION can be imagined by analogy; it is based on the probability of occurrence of the various individual words (and not pairs) in English. It could be composed, for example, by

taking the twenty-fifth word of every column in a newspaper, and forming a list of them in sequence. The more probable a word is in English, the greater the chance of meeting it in the local newspaper. For this we would imagine a probabilistic automaton with only one state, where the input of any word will bring the automaton to that state. The chance that a given loop be chosen is equal to the probability of the word in question in the language. This is shown in Figure 6.2.

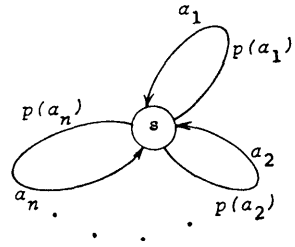


Fig. 6.2. A probabilistic automaton for first and zero order approximations.

A ZERO ORDER APPROXIMATION is a string of words chosen at random and without regard for their frequency of occurrence in the language. Such an approximation could be made, for example, by taking every thirteenth word which occurs on a page, chosen at random, of an English dictionary. If all the loops in the automaton in Figure 6.2 have equal probabilities, the automaton generates a zero order approximation.

The following are examples of zero, first and second order approximations (the first and second order approximations are taken from Miller and Chomsky (1963)); only the second order approximation can be taken as generated by a Markov-source.

ZERO ORDER: *splinter shadow dilapidate turtle pass stress grouse appropriate radio whereof also appropriate gourd keeper clarion wealth possession press blunt canter chancy vindicable corpus*

FIRST ORDER: *representing and speedily is an good apt or came can different natural here he the a in came the to of to expert gray come to furnish the line message had be these*

SECOND ORDER: *the head and in frontal attack on an English writer that the character of this point is therefore another method for the letter that the tired of who even told the problem for an unexpected*

Although the Markov-source clearly produces "better English" than the zero and first order approximations, the result is rather disappointing if we consider any other characteristic than immediate succession of words. Every sequence of five or more words looks strange. This is disappointing because the very limited result is attained by a model with an astronomically high number of parameters. It is more difficult to evaluate sequences of three or four words, and there is no method to determine how good the source is in this respect. This brings us back to Markov. For his vowel/consonant model it is possible to determine how well trigrams (and longer strings) are predicted, for we know the precise values of the transition probabilities in the model. The chance for a trigram *ccc* is equal to the chance for a digram *cc*, $p(cc)$, multiplied by the chance that the second *c* be followed by another *c*, $p_{c,c}$. The best estimate of $p(cc)$ is the relative digram frequency (cf. Table 6.1), $\frac{3827}{19999} = 0.191$.[1] The expected relative frequency for the trigram *ccc* is thus $0.191 \times 0.337 = 0.065$. Table 6.2 shows the

TABLE 6.2. Expected and Observed Relative Frequencies of Trigrams in *Eugene Onegin*.

	vvv	*vvc*	*vcv*	*vcc*	*cvv*	*cvc*	*ccv*	*ccc*
Expected	0.007	0.048	0.250	0.127	0.048	0.329	0.126	0.065
Observed	0.006	0.049	0.211	0.166	0.049	0.327	0.166	0.025

[1] One might wonder if $p(cc)$, which is determined on the basis of the digram frequency, is indeed predicted by the model, i.e. on the basis of the transition matrix. It may be argued as follows that this is in fact the case. For an ergodic Markov process, the probabilities of the various elements ($p(v)$ and $p(c)$ in the example) are given in the stochastic eigenvector of P, i.e. the vector α for which $\alpha P = \alpha$. In the example, $\alpha = (p(v), p(c))$, and the vector is stochastic because $p(v) + p(c) = 1$. The value of α can be found here by solving the equation $p(v)p_{v,v} + p(c)p_{c,v} = p(v)$. Substitution of $p_{v,v} = 0.128$ and $p_{c,v} = 0.663$ (cf. Table 6.1.) yields $p(v) = 0.432$ and $p(c) = 0.568$. The chance for the digram *cc* is then $p(c).p_{c,c} = 0.568 \times 0.337 = 0.191$, which corresponds to the actual value.

expected and observed frequencies for the eight possible trigrams. The Markov model is evidently quite accurate in this respect, but this is further left to the judgment of the reader.

If the predictions for trigrams (or n-grams of a higher order) are not considered satisfactory, a more complicated model can be chosen. One could select a model based on the probability of transition, not from letter to letter (or from word to word), but from *pair* of letters to letter; thus, for example, the probability of v after the sequence vc is $p_{vc,v}$. A finite automaton can also be constructed for this end. Such an automaton, unlike the Markov-source, will have a state for every *pair* of letters; for n vocabulary elements, then, there will be n^2 states in the automaton. Therefore for the vowel/consonant example, four states will be needed, s_{cv}, s_{cc}, s_{vv}, s_{vc}. The automaton is shown in Figure 6.3, and is called a PROJECTED MARKOV-SOURCE. It is a 2-limited automaton,

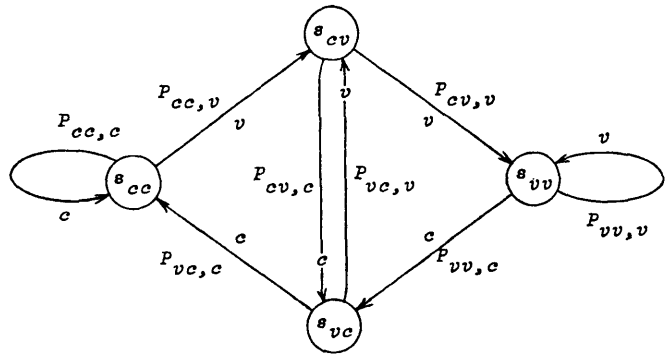


Fig. 6.3. A Projected Markov-source.

because every sequence of two input elements unambiguously determines the state of the automaton. Each state is labelled according to a sequence of two input elements which necessarily lead to it, and each state has as many inputs and outputs as there are vocabulary elements. In Figure 6.3 there are two vocabulary elements, and eight transition probabilities of the form $p_{ij,k}$, which logically find their places in the model. In the projected

Markov-source, the transition probabilities are also normalized according to state. Just as the model in Figure 6.1 represents the exact digram frequencies, the projected Markov-source represents the exact trigram frequencies. The characteristics of the automaton can once again be summarized completely in a transition matrix; this, however, will not be square (2×2), but rather rectangular ($2^2 \times 2$), with the rows labelled according to the pairs, and the columns according to the vocabulary elements.

P	v	c
vv	$p_{vv,v}$	$p_{vv,c}$
vc	$p_{vc,v}$	$p_{vc,c}$
cv	$p_{cv,v}$	$p_{cv,c}$
cc	$p_{cc,v}$	$p_{cc,c}$

Like the ordinary Markov-source, the projected Markov-source can be represented as a grammar. If the terminal vocabulary contains n elements, $\{a_1, \ldots, a_n\}$ the nonterminal vocabulary will contain n^2 elements $\{A_{11}, A_{12}, \ldots A_{21}, A_{22}, \ldots A_{nn}\}$ and n^3 productions of the form $A_{ij} \xrightarrow{p_{ij,k}} a_k A_{jk}$.

We can now attempt to generate English by means of the 2-limited projected Markov-source. This will give English to a *third order approximation*. With $n = 100{,}000$ words, there are $n^2 = 10^{10}$ states, and $n^3 = 10^{15}$ transition probabilities $p_{ij,k}$. Of these, there are $n^3 - n^2$ independent parameters (by normalization, one column of the transition matrix becomes redundant). Calculation of all these parameters is excluded, but to have an impression of how well the projected Markov-source can generate English, we can once again play the above game with speakers. We present speaker A with a *pair* of words (chosen at random from a newspaper or from a sentence composed by another speaker), for example, *family was*, and ask him to form a sentence in which the pair occurs. Suppose that the sentence which he produces is *the family was large*. We then present *was large* to speaker B, and request that he in turn form a sentence in which this pair occurs. If his sentence is *the forest was large, dark and dangerous* we present *large*

dark to speaker *C*, and so forth. The following string (Miller and Chomsky 1963) was obtained in this way.

THIRD ORDER: *family was large dark animal came roaring down the middle of my friends love books passionately every kiss is fine.*

Obviously we can go on to construct still higher order projected Markov-sources. A 3-limited source, the vocabulary of which has n elements, will have n^3 states, and each state will have n inputs and outputs. Every output of a state has a probability, and the transition probabilities are normalized for each state. An example of an approximation of English, generated by a 3-limited source, is the following (Miller and Selfridge 1950);

FOURTH ORDER: *went to movies with a man I used to go toward Harvard Square in Cambridge is mad fun for*

In general, a k-limited projected Markov-source has n^k states, and therefore a $n^k \times n$ matrix of transition probabilities. The number of independent parameters in such a model is thus $n^{k+1} - n^k$. The following is an example of a fifth order approximation of English (Miller and Chomsky 1963).

FIFTH ORDER: *road in the country was insane especially in dreary rooms where they have some books to buy for studying Greek*

All Markov-sources are k-limited, but as we have seen (in Volume I, Figure 4.5), not all finite automata are k-limited. Consequently it is not the case that all regular languages can be generated by Markov-sources.

The five approximations of English given in the course of this paragraph were progressively "better". The higher the order, the more predictable the text, and therefore the less "informative" (according to the definition given in communication theory). A zero order approximation is a string in which all elements have an equal chance to occur. Suppose we take a random segment of m elements from the infinite string produced by a zero order automaton. How great is the chance that a second random segment of m elements will contain the same elements as the first segment, and in the same order? The probability that the second segment

begins with the same element as the first is $\frac{1}{n}$, if the vocabulary contains n elements. The chance that the second element is the same is also $\frac{1}{n}$, and so forth. The chance that the entire segment is the same is therefore $p = \left(\frac{1}{n}\right)^m$. Suppose that the vocabulary contains only one element, $n = 1$; in that case any two segments of m elements will be identical, for $p = \left(\frac{1}{1}\right)^m = 1$. Predictability is then complete, the message does not reduce uncertainty, and there is no information. The uncertainty of a message is defined as the logarithm (base 2) of the probability p of that message. In the example, $p = 1$, and the uncertainty is therefore $\log p = 0$. Uncertainty increases with the number of vocabulary elements. For this source, the uncertainty relative to a segment of m elements is $\log\left(\frac{1}{n}\right)^m = m \log\frac{1}{n}$. The information H, the amount of uncertainty reduced, is defined as the complement of the uncertainty. With a zero order approximation, we therefore have $H(0) = -m \log\frac{1}{n}$ for a string of m elements.

For a first order approximation, the probabilities p_i of the various vocabulary elements a_i are not necessarily equal. How great is the information $H(1)$ of a random segment with m elements? If m is large, a string of m elements should contain the word a_1 about mp_1 times, the word a_2 about mp_2 times, and in general, the word a_i about mp_i times. The chance for the entire string is once again the product of the probabilities of the individual elements. Since the element a_i occurs approximately mp_i times, and a_i occurs with probability p_i, the probability of this string of m elements is $p = p_1^{mp_1} . p_2^{mp_2} . \ldots . p_2^{mp_n}$, and $H(1)$ is therefore approximately $-\log p = -(mp_1 \log p_1 + mp_2 \log p_2 + \ldots + mp_n \log p_n) = -m \sum_i p_i \log p_i$. If all p_i are equal, thus $p_i = \frac{1}{n}$, then the

information will, of course, be equal to that of the zero order approximation, $-m \log \frac{1}{n}$. If the probabilities are not equal, H is smaller. Therefore $H(0) \geqslant H(1)$.

One could go on to prove that $H(0) \geqslant H(1) \geqslant H(2),\ldots$, and in general that $H(i) \geqslant H(i+1)$. The information will be equal only when probabilities or transition probabilities are equal. For English, these are obviously unequal, and it holds, therefore, that $H(i) > H(i+1)$: the higher the order, the less informative (or more redundant) the text. In Volume III, Chapter 2, paragraph 2 we shall examine psychological applications of this. General introductions to communication theory may be found in other literature; a few sources are mentioned in the bibliographic survey at the end of this volume.

6.2. A PROBABILISTIC GRAMMAR FOR A CHILD'S LANGUAGE

The simplest case of statistical inference occurs when grammar and corpus are given, and production probabilities must be deduced. The procedure necessary for this is treated in detail in Voume I, Chapter 8, paragraph 5; the grammar in question was context-free.

An interesting linguistic example of this method is Suppes' analysis of the language of Adam, one of Brown's young subjects. The corpus analyzed by Suppes was recorded when Adam was two years and two months old, and consists of eight hours of tape recordings. After the elimination of immediate repetitions, the corpus contains 6109 words, over a vocabulary of 673 different words. It was segmented into 3497 utterances ("sentences"). Suppes analyzed this material in various ways. He attempted to give a complete grammar for it (which we shall discuss later), and he made an analysis of only the noun phrases in the material.

There is a certain amount of freedom for the definition of noun phrase, but as soon as a grammar is written, the sequences of categories which may be called "*NP*" are clearly determined.

Uncertainties concerning the frequencies in the corpus can only come about then through uncertainty in the categorization of individual words. Is *fly*, for example, a noun or a verb? This is the sort of problem of interpretation which typically occurs in applied linguistics. Suppes (1970, 1971) gives the following context-free *NP* grammar:

1. $NP \xrightarrow{a_1} N$
2. $NP \xrightarrow{a_2} AP$
3. $NP \xrightarrow{a_3} AP + N$
4. $NP \xrightarrow{a_4} P$
5. $NP \xrightarrow{a_5} NP + NP$
6. $AP \xrightarrow{b_1} AP + A$
7. $AP \xrightarrow{b_2} A$

The symbol P stands for *pronoun*, and AP for *adjective phrase*. The production probabilities are denoted by $a_1, \ldots a_5, b_1, b_2$. The grammar is normalized, and therefore $\Sigma a_i = 1$ and $\Sigma b_i = 1$. Consequently there are five independent parameters in the model.

A typical sentence in the corpus is *take off Adam paper*, with the *NP Adam paper*. The noun phrase is of the form $N + N$; a leftmost derivation of it is $NP \overset{a_5}{\Rightarrow} NP+NP \overset{a_1}{\Rightarrow} N+NP \overset{a_1}{\Rightarrow} N+N$. If it is supposed that the productions are applied independently, then $p(NN) = a_5.a_1.a_1$. The chances for all other observed *NP* forms are also determined in this way on the basis of the grammar. This led, according to the procedure presented in Volume I, Chapter 3, paragraph 5, to estimates of the seven production probabilities (cf. Table 6.2), and the calculation of expected frequencies for the various types of *NP*. The latter are given, together with the observed frequencies in Table 6.3. The difference in total between observed

TABLE 6.2. Estimated Production Probabilities for the *NP*-Grammar for Adam's Language

a_1	a_2	a_3	a_4	a_5	b_1	b_2
0.6391	0.0529	0.0497	0.1439	0.1144	0.0581	0.9419

TABLE 6.3. Observed and Expected Frequencies of Various Noun Phrase Types in Adam's Language

NP-Type	*Observed*	*Expected*	*NP-Type*	*Observed*	*Expected*
N	1445	1555.6	*PPN*	6	0.4
P	388	350.1	*ANN*	5	8.3
NN	231	113.7	*AAN*	4	6.6
AN	135	114.0	*PA*	4	2.0
A	114	121.3	*ANA*	3	0.7
PN	31	25.6	*APN*	3	0.1
NA	19	8.9	*AAA*	2	0.4
NNN	12	8.3	*APA*	2	0.0
AA	10	7.1	*NPP*	2	0.4
NAN	8	8.3	*PAA*	2	0.1
AP	6	2.0	*PAN*	2	1.9
			Total	2434	2335.8

and expected frequencies (98.2) is due to the fact that the grammar generates other (longer) noun phrases which do not occur in this corpus. There are also very noticeable differences in detail among the various types of noun phrase. Thus the actual frequency of the sequence $N+N$ is considerably underestimated in the theoretical expectations. Many *NN* sequences prove to be possessive from the context, such as *Adam bike* and *Daddy suitcase*. Others, however, are "is a" relations, like *toy train* and *lady Ursula*, and still other N/N relations have been distinguished in the material. One might consider introducing a separate possessive production rule to obtain better theoretical predictions. Or one could introduce statistical dependencies between productions. But linguists will be more inclined to treat these differences transformationally. Although there is no theoretical difficulty in writing a probabilistic transformational grammar, important practical problems are involved. It would be necessary (a) to assign production probabilities to the base grammar and (b) to assign probabilities to optional transformations. But then special provisions would have to be made for ambiguities, and, in grammars of some complication, for the treatment of transformational filtering. Suppes attempted to refine the grammar with regard to the *NN* sequences by means of a semantic analysis, which we shall not discuss further here.

The point here is to show the strength of such a probabilistic analysis. Direct information is obtained on which production rules do the actual work in the grammar, and which are used only occasionally. But above all there is a direct feed-back on which rules fail, and thus on the direction in which further improvement of the grammar must be sought. We shall return to this subject at the end of this paragraph.

Suppes attempted to write a complete grammar for the corpus; the form he chose was that of a probabilistic categorial grammar. The very limited number of rules in a categorial grammar (cf. Chapter 4, paragraph 2) restricts the freedom of movement to such a small number of parameters, that the undertaking — however interesting — is bound to fail, as was indeed the case for Suppes' grammar. Success would have meant a deep insight into the structure of the child's language; it would have meant that the child's syntax develops exclusively by the differentiation of categories, and changes in rule parameters. The number of parameters (and rules) would, however, be small and constant throughout the development. For details on this, we refer to Suppes (1970).

A probabilistic grammar also gives various additional information which can be of great use in applied linguistics. On the basis of such a grammar characteristics of the corpus can be treated, which might lie beyond the range of theoretical linguistics, but which are sometimes the *pièce de résistance* in practical applications. Thus *sentence length* is an essential variable in the analysis of style, in the analysis of children's languages, in the investigation of speech intelligibility, etc. An accurate probabilistic grammar also provides a description of sentence length in the corpus, as well as the distributions of length of other constituents.

An example can again be taken in Suppes' analysis of Adam's noun phrases. With the given grammar, a noun phrase of length 1 can be derived in three ways: (i) by applying production 1: NP will then be rewritten as N, with probability a_1; (ii) by first applying production 2, then production 7: $NP \Rightarrow AP \Rightarrow A$ with $p(A) = a_2.b_2$; (iii) by applying production 4: NP will be rewritten as P, with probability $p(P) = a_4$. The total probability of a noun

phrase of length 1 is thus $p(1) = a_1 + a_2.b_2 + a_4$. For the production probabilities in Table 6.2, this is $p(1) = 0.8329$. Of the 2434 noun phrases in the corpus, there should therefore be $2434 \times 0.8329 = 2027$ of length 1. The observed value is 1947. The expected value for length 2 can be calculated in the same way; for Adam's noun phrases the expected value is 314, and the observed value is 463. Likewise for length 3, the expected value is 67, the observed, 51, and 26 noun phrases are expected of length greater than 3, but none occur in Adam's speech.

One of the most noteworthy advances in the modern investigation of children's languages is that which one could call the *linguistic* method. In the 1960's explicit grammars were written for the first time for the languages of two and three-year-olds. Language development was studied for the first time from the point of view of grammar, and such matters as the differentiation of categories and rewrite rules and the growth of transformational skills such as in negation and question were investigated. In the meantime this research has begun to be integrated into a much wider framework, that of the cognitive-conceptual development of the child; we shall return to this subject in Volume III, Chapter 4. But in the beginning, the opinion of the transformational linguists of the time was the touchstone for this renewal, and consciously or unconsciously many accidental attitudes were taken over from them into the practice of research. One of these attitudes was an aversion to statistical concepts. In 1969 Chomsky wrote "It must be recognized that the notion 'probability of a sentence' is an entirely useless one, under any known interpretation of this term". Traditionally however, research on children's languages was very much interested in the development of the statistical aspects of the language, the development of sentence length, frequencies of the various types of sentences and classes of words, etc. There can be no doubt but that a complete theory of the development of children's languages must also be able to explain those phenomena. A(non-probabilistic) grammar is perhaps half the work in this, but it is still no more than a good beginning. Probabilistic grammars, however, make it possible to establish the relations between modern

structural linguistic insights and the abundance of traditional statistic data on the development of children's languages. The reason for such an approach is not simply the need to reconcile (apparent) contradictions, but rather the desire to find a structural explanation for the *patterns* which appear in those statistical phenomena. The change of one parameter in a probabilistic grammar can lead to statistical changes in very divergent aspects of the corpus generated, for example, simultaneous changes in the frequency of words of a certain class and in the distribution of sentence lengths. If the relationship were known, it would be possible to find an economical explanation for the development of phenomena which appear on the surface to be independent. This is precisely what is needed, but the traditional approach did not provide the means to accomplish this. Every statistical phenomenon was given a separate psychological "explanation": sentence length was said to grow with memory, verb/noun ratios with "functioning pleasure" *(Funktionslust)*, etc. Probabilistic grammars, applied with insight, can show how such apparently independent phenomena are in fact based on the same structural variable. Developmental language theory should therefore be oriented in this direction. Such an approach would not only be useful for developmental psychology, but also would help to attain explanatory adequacy in linguistic theory (cf. Chapter 1, paragraph 2). The question as to the cause of a universal systematics in natural languages should be traced back partially to the fundamental characteristics of human cognitive structure, and their development in the child. A probabilistic grammar is one of the means by which such fundamental characteristics can be localized, on the basis of the speech of the growing child. Thus statistical methods must not be excluded from theoretical linguistics. Theory and interpretation are interdependent, and interpretation often demands the use of statistical inference.

But what is the basis of the former aversion to statistics in linguistics? Chomsky, repeatedly and with great eloquence, emphasized the unpredictability of human language. Speech is creative, for every utterance is new (with the exception of a few

clichés); it shows no simple dependence on the situation in which it is generated. This virtual unlimitedness and freedom of human language is rightly used as an argument against over-simplistic theories of verbal conditioning, such as that of Skinner. But no argument against the statistical investigation of language can be based on these uncontrovertible facts. However this is precisely what Chomsky does. The newness of nearly every linguistic utterance means in statistical terms that every sentence has a probability of occurrence which is indistinguishable from zero. It is on this ground that Chomsky, and with him many other linguists, bans the concept of "probability" from linguistics. It is Suppes' merit to have refuted this argument. He points out that construction of statistical theory is necessary in science precisely where deterministic models are excluded in principle or in fact, and mentions quantum mechanics as the classic example of the impossibility of using a deterministic model. A sentence is precisely as unpredictable as the trajectory of an electron: in both cases the phenomena have a probability which is practically equal to zero. This is the situation in which statistics is applicable *par excellence*. A model is then tested by investigating various statistical parameters in their mutual relations. This holds as much for quantum mechanics as for linguistics. The fact that a sentence has a probability of zero does not mean that the sentence length involved does not occur in the corpus, nor does it mean that words or categories of words in the sentence have a probability of zero. It is on the basis of such data that a model can in fact be tested.

It should be pointed out that the situation here is essentially different from the usual empirical situation in linguistics, which involves the testing of linguistic intuitions. The linguist can question informants at will on their intuitions regarding a linguistic object, and if the phenomenon under study is of any importance, the answers will agree in that regard. But it is not possible, except in trivial circumstances, to make informants spontaneously produce a particular sentence. A sentence cannot be "repeated" like an intuition. It is this circumstance which makes the analysis of a corpus more difficult than work with informants.

As we have seen in Chapter 1, paragraph 2, the analysis of a corpus is one of the forms in which the problem of interpretation occurs in linguistics. Linguists who are concerned with such questions of interpretation must use other methods and types of analysis than those used by theoreticians. But theoretical linguistics is pointless, and ultimately impossible, without interpretation; both aspects of linguistics must develop in interaction. Methodological absolutism in linguistics would be entirely out of place.

HISTORICAL AND BIBLIOGRAPHICAL REMARKS

The distinction between theory and interpretation mentioned in Chapter 1 goes back directly to the work of Bar-Hillel, and indirectly to that of Carnap (cf. Bar-Hillel 1970, 364ff.). The notions of *language* and *observable linguistic phenomena* may be found in de Saussure (1916) as *langue* and *parole*, in Chomsky (in many places, especially Chomsky (1965)) as *competence* and *performance*. These distinctions, however, do not coincide precisely; the distinction between competence and performance in particular has not only the theoretical function emphasized in this volume, but also a psychological function which will be analyzed in Volume III. Literature on the metalinguistic character of linguistic data may be found in Bever (1970 a, b), Levelt and Schils (1971) Levelt (1972), and Watts (1970). The various forms of grammatical adequacy are treated extensively in Chomsky (1965) and in other places by the same author. A detailed treatment of concepts such as "utterance", "word", and "morpheme" may be found in Lyons (1968), to which we refer for further literature on the subject.

Nearly all the essential questions touched upon in Chapter 2 were dealt with by Chomsky before the publication of *Syntactic structures* (1957), in particular in *The Logical Structure of Linguistic Theory* (mimeo, 1955) and in *Three Models for the Description of Language* (1956). The last publications by Chomsky on this subject are those in the *Handbook of Mathematical Psychology* (1963). Our section on context-free grammars borrows some material from Postal (1964b). That article contains some errors, as well as a one-sided treatment of the work of a number of

linguists such as Harris and Halliday. Among others, the criticisms by Thorne (1965) and Robinson (1970) are interesting in this connection. Interest in finite automata has received a new impetus in the theory of formal languages, in two forms, (a) natural language parsing programs, based on *augmented transition networks* which are "expanded" finite automata, to be discussed further in Volume III, Chapter 3, paragraph 6.4 (cf. Woods 1970, and Kaplan 1972), and (b) in *tree automata*, which have tree-diagrams for their input and output, instead of terminal strings. These are finite automata, which can nevertheless recognize context-free languages (cf. Thatcher 1967, and Levy and Joshi 1971). There is an interesting future for language parsing programs in both.

The sources for Chapter 3 are Chomsky's *Aspects of the Theory of Syntax* (1965), and a few articles by Peters and Ritchie (1969 a, b, 1971, 1972). *Aspects* gives two different formulations for lexical insertion rules, and we follow the second. The general definition of transformations in Chapter 3, paragraph 2.2 follows Brainerd (1971), who also treats other grammatical systems formally. Chapter 3, paragraph 2.4 follows Peters and Ritchie (1973), a fundamental but extremely laborious formulation. We have tried to extend its readability by introducing the concept of "elementary factorization", and by omitting a few technical details of secondary importance, in particular with regard to definitions of transformational cycle and derivation. There is still no other summary of Peters and Ritchie's formalization of the *Aspects* theory. Later developments (Chapter 3, paragraph 3) originated in work by McCawley (1968 a, b) and by G. Lakoff (1970). The most important sources for the work of interpretative semanticists are Chomsky (1970a, 1971), Jackendoff (1969, 1971). A theoretical survey of generative semantics may be found in Lakoff (1971). This point of view may also be found in Postal (1970, 1971), articles in Bach and Harms (1968), Jacobs and Rosenbaum (1970), Steinberg and Jakobovits (1971), and others. A third trend originating in the *Aspects* theory is the work of Montague (1970, to be published), which was not discussed here. Before his sudden death, Montague had elaborated the formal

aspects of his theory in detail. Chapter 3 was written from a formal point of view. There are many introductions to transformational grammar which place more emphasis on content, such as Bach (1964), Lyons (1968), Liles (1971). Two articles by Hall-Partee (1971 a, b) give a good survey of later developments.

The four grammars treated in Chapter 4 come from the following literature. Categorial grammars are found in Leśniewski (1929) and Ajdukiewicz (1935), and related formal systems are treated by Curry (1961) and Lambek (1961). The work of Bar-Hillel, recapitulated in Bar-Hillel (1964), contains the principal background of Chapter 4, paragraph 2; it gave explicit linguistic motivation to the use of categorial grammars. A categorial variant of the base rules in *Aspects*, not discussed here, may be found in Miller (1968). Lewis (1970) treats the semantic component of a categorial grammar. The literature concerning operator grammars is sufficiently indicated in Chapter 4, paragraph 3. The most important source for Harris' work in the field of adjunct grammars, is Harris (1968), where an automaton is also developed to accept such *string languages*. The formal development of transformational adjunct grammars is the result of work by Joshi (1972) and Joshi, Kosaraju and Yamada (1972 a, b). Dependency grammars may be found in Tesnière (1959), Hays (1964), Robinson (1970), Anderson (1971). Articles by the last two authors as well as other important texts on case grammars are found in Abraham (1971). Gaifman (1965) provides a mathematical foundation for dependency grammars. Some material for Chapter 5 was also borrowed from an unpublished survey by Hirschman (1971).

The main point of Chapter 5, the undecidability of an *Aspects* type transformational grammar, was proved at almost the same time by Kimball (1967), Ginsburg and Hall (1969), Salomaa (1971) and Peters and Ritchie (1973). The dates here are misleading. The present writer remembers following a lecture by Ritchie at Harvard University in 1966; notes taken at that lecture show that proof was already given for transformational grammars with a context-sensitive base. Could not more rapid publication of that proof have been of great service to transformational linguistics?

Kimball was decidedly the first to give the proof for transformational grammars with a regular base.

Chapter 6, paragraph 1 is not intended as an introduction to information or communication theory. The most important mathematical source for this is Shannon and Weaver (1949). An excellent introduction is Cherry (1957). Miller (1951) gives more exclusively psycholinguistic applications. Miller and Chomsky (1963) place information theory in the framework of formal languages; the work offers the derivation of the information value of the various approximations of natural language. Adam's language (Chapter 6, paragraph 2) is described in Brown, Cazden and Bellugi (1968); other analyses of Adam's language can be found in McNeill (1970). Suppes' analysis is the only probabilistic approach to the grammar of children's languages available at the moment.

BIBLIOGRAPHY

Abraham, W. (ed.)
1971 *Kasustheorie* (Frankfurt: Athenäum).

Ajdukiewics, K.
1935 "Die sytaktische Konnexität", *Studia Philosophica* 1, 1-27.

Anderson, J. M.
1971 *The Grammar of Case. Toward a Localistic Theory* (Cambridge: Cambridge University Press).

Bach, E.
1964 *An Introduction to Transformational Grammars* (New York: Holt, Rinehart and Winston).

Bach, E. and R. T. Harms (eds.)
1968 *Universals in Linguistic Theory* (New York: Holt, Rinehart and Winston).

Bar-Hillel, Y.
1964 *Language and Information. Selected Essays on Theory and Application* (Reading, Mass.: Addison-Wesley).
1970 *Aspects of Language* (Amsterdam: North Holland).

Bar-Hillel, Y., C. Gaifman, and E. Shamir
1960 "On Categorial and Phrase Structure Grammars", *Bull. Res. Council of Israel* 9F, 1-16.

Bever, T. G.
1970a "The Influence of Speech Performance on Linguistic Structure", in: *Advances in Psycholinguistics*, G. B. Flores d'Arcais and W. J. M. Levelt (eds.) (Amsterdam: North Holland).
1970b "The Cognitive Basis for Linguistic Structures", in: *Cognition and the Development of Language*, J. R. Hayes (ed.) (New York: Wiley).

Brainerd, B.
1971 *Introduction to the Mathematics of Language Study* (New York: Elsevier).

Brown, R., C. Cazden and U. Bellugi
1968 "The Child's Grammar from I to III", in: *Minnesota Symposium on Child Psychology*, J. P. Hill (ed.) (Minneapolis: University of Minnesota Press).

Cherry, C.
1957 *On Human Communication* (Cambridge, Mass.: MIT Press).

Chomsky, N.
1955 *The Logical Structure of Linguistic Theory*. Microfilm, MIT-Library.
1956 "Three Models for the Description of Language", in: *I.R.E. Transactions on Information Theory*, vol. IT-2, Proceedings of the Symposium on Information Theory, Sept., 113-23.
1957 *Syntactic Structures* (The Hague: Mouton).
1963 "Formal Properties of Grammar", in: *Handbook of Mathematical Psychology*, R. D. Luce, R. R. Bush and E. Galanter (eds.) (New York: Wiley).
1965 *Aspects of the Theory of Syntax* (Cambridge, Mass.: MIT Press).
1969 "Quine's Empirical Assumptions", in: *Words and Objections. Essays on the Work of W. V. Quine*, D. Davidson and J. Hintikka (eds.) (Dordrecht: Reidel).
1970a "Remarks on Nominalization", in: *Readings in Transformational Grammar*, R. A. Jacobs and P. S. Rosenbaum (eds.) (Waltham: Ginn).
1971 "Deep Structure, Surface Structure, and Semantic Interpretation", in: *Semantics. An Interdisciplinary Reader in Philosophy, Linguistics and Psychology*, D. D. Steinberg and L. A. Jakobovits (eds.) (Cambridge: Cambridge University Press).
1972 "Some Empirical Issues in the Theory of Transformational Grammar". In N. Chomsky, *Studies on Semantics in Generative Grammar*. The Hague: Mouton

Chomsky, N. and M. Halle
1968 *The Sound Pattern of English* (New York: Harper and Row).

Chomsky, N. and G. A. Miller
1963 "Introduction to the Formal Analysis of Natural Languages", in: *Handbook of Mathematical Psychology*, R. D. Luce, R. R. Bush and E. Galanter (eds.) (New York: Wiley).

Curry, H. B.
1961 "Some Logical Aspects of Grammatical Structure", in: *Structure of Language and its Mathematical Aspects. Proc. Twelfth Symp. Appl. Math.*, R. Jakobson (ed) (Providence, R. I.: American Mathematical Society).

Dik, S. C.
1968 *Coordination: Its Implications for the Theory of General Linguistics* (Amsterdam: North Holland).

Fillmore, C. J.
1968 "The Case for Case", in: *Universals in Linguistic Theory*, E. Bach and R. T. Harms (eds.) (New York: Holt, Rinehart and Winston).
1969 "Verbs of Judging: An Exercise in Semantic Description", *Papers in Linguistics* 1, 91-117.

Fodor, J. A.
1970 Three Reasons for Not Deriving "kill" from "cause to die", *Linguistic Inquiry* 1, 429-38.

Gaifman, C.
1965 "Dependency Systems and Phrase Structure Systems", *Information and Control* 8, 304-337.
Geach, P. T.
1970 "A Program for Syntax", *Synthese* 22, 3-18.
Ginsburg, S. and B. Hall-Partee
1969 "A Mathematical Model of Transformational Grammar", *Information and Control* 15, 297-334.
Ginsburg, S. and H. G. Rice
1962 "Two Families of Languages Related to ALGOL", *J. Assoc. Comp. Mach.* 10, 350-71.
Hall-Partee, B.
1971a "Linguistic Metatheory", in: *A Survey of Linguistic Science*, W. O. Dingwall (ed.) (College Park: Linguistics Program/University of Maryland).
1971b "On the Requirement that Transformations Preserve Meaning", in: *Studies in Linguistic Semantics*, C. J. Fillmore and D. T. Langendoen (eds.) (New York: Holt, Rinehart and Winston).
Harman, G.
1970 Deep Structure as Logical Form", *Synthese* 21, 275-97.
Harris, Z. S.
1968 *Mathematical Structures in Language* (New York: Wiley).
1970a "The Two Systems of Grammar: Report and Paraphrase", in *Papers in Structural and Transformational Linguistics*, Z. S. Harris (Dordrecht: Reidel).
1970b *Papers in Structural and Transformational Linguistics* (Dordrecht: Reidel).
Hays, D. G.
1964 "Dependency Theory: A Formalism and Some Observations", *Language* 33, 283-340.
Hirschman, L.
1971 "A Comparison of Formalisms for Transformational Grammar", *Transformations and Discourse Analysis Papers* 87 (University of Pennsylvania).
Hopcroft, J. E. and J. D. Ullman
1969 *Formal Languages and Their Relations to Automata* (Reading, Mass.: Addison-Wesley).
Huddleston, R. D.
1971 *The Sentence in Written English: A Syntactic Study Based on an Analysis of Scientific Texts* (Cambridge: Cambridge University Press).
Hudson, R. A.
1971 *English Complex Sentences: An Introduction to Systemic Grammar* (Amsterdam: North Holland).
Jackendoff, R.
1969 "Some Rules for English Semantic Interpretation", MIT Dissertation.
1971 "Modal Structure in Semantic Representation", *Linguistic Inquiry* 2, 479-514.

Jacobs, A. K. and P. S. Rosenbaum, (eds.)
1970 *Readings in Transformational Grammar* (Waltham: Ginn).
Joshi, A. K.
1971 "How Much Hierarchical Structure is Necessary for Sentence Description?" *Proc. Intern. Symp. Comp. Ling.* (Debrecen, Hungary).
1972 "A Class of Transformational Grammars", in: *Formal Language Analysis*, M. Gross, M. Halle and M. P. Schützenberger (eds.) (The Hague, Mouton).
Joshi, A. K., S. Kosaraju and H. M. Yamada
1972a "String Adjunct Grammars I: Local and Distributed adjunction", *Information and Control* 21, 93-116.
1972b "String Adjunct Grammars II: Equational Representation, Null Symbols, and Linguistic Relevance", *Information and Control* 21, 235-60.
Kaplan, R. M.
1972 "Augmented Transition Networks as Psychological Models of Sentence Comprehension", *Artif. Intell.* 3, 77-100.
Katz, J. J.
1971 "Generative Semantics Is Interpretative Semantics", *Linguistic Inquiry* 2, 313-32.
Katz, J. J. and P. Postal
1964 *An Integrated Theory of Linguistic Descriptions* (Cambridge, Mass.: MIT Press).
Kimball, J. P.
1967 "Predicates Definable over Transformational Derivations by Intersection with Regular Languages", *Information and Control* 11, 177-95.
Kraak, A. and W. G. Klooster
1968 *Syntaxis* (Culemborg: Stam-Kemperman).
Kuroda, S. Y.
1972 "Généralisation de la notion d'équivalence de grammaires — une méthode topologique", in: *Formal Language Analysis*, M. Gross, M. Halle and M. P. Schützenberger (eds.) (The Hague: Mouton).
Lakoff, G.
1970 *Irregularity in Syntax* (New York: Holt, Rinehart and Winston).
1971 "On generative Semantics", in *Semantics: An Interdisciplinary Reader in Philosophy, Linguistics and Psychology*, D. D. Steinberg and L. A. Jakobovits (eds.) (Cambridge: Cambridge University Press).
Lambek, J.
1961 "On the Calculus of Syntactic Types", in: *Structure of Language and Its Mathematical Aspects. Proc. Twelfth Symp. Appl. Math.*, R. Jakobson (ed.) (Providence R. I.: American Mathematical Society).
Leśniewski, S.
1929 "Grundzüge eines neuen Systems der Grundlagen der Mathematik", *Fundamenta Mathematicae* 14, 1-81.

Levelt, W. J. M.
1966 "Generative Grammatika en Psycholinguistiek I. Inleiding in de Generatieve Grammatika", *Nederlands Tijdschrift voor Psychologie* 21, 317-37.
1972 "Some Psychological Aspects of Linguistic Data", *Linguistische Berichte* 17, 18-30.

Levelt, W. J. M. and E. Schils
1971 "Relaties tussen psychologie en linguistiek", in: *De plaats van de psychologie temidden van de wetenschap*, Annalen van het Thijmgenootschap (Bussum, Holland: Paul Brand).

Levy, L. S. and A. K. Joshi
1971 "Some Results in Tree Automata", *Proc. ACM Symposium on Theory of Computing* (Cleveland). *Mathematical Systems Theory* 6 (1973), 334-42.

Lewis, D.
1970 "General Semantics", *Synthese* 22, 18-67.

Liles, B. L.
1971 *An Introductory Transformational Grammar* (Englewood Cliffs, N. J.: Prentice-Hall).

Lyons, J.
1968 *Introduction to Theoretical Linguistics* (Cambridge: Cambridge University Press).

Markov, A. A.
1913 *Essai d'une recherche statistique sur le texte du roman "Eugène Onégin"* (= *Bulletin de l'Académie Impériale des Sciences*, St. Petersbourg, 7).

McCawley, J. D.
1968a "The Role of Semantics in Grammar", in: *Universals in Linguistic Theory*, E. Bach and R. T. Harms (eds.) (New York: Holt, Rinehart and Winston).
1968b "Concerning the Base Component of a Transformational Grammar", *Foundations of Language* 4, 55-81.

McNeill, D.
1970 *The Acquisition of Language. The Study of Developmental Psycholinguistics* (New York: Harper and Row).

Miller, G. A.
1951 *Language and Communication* (New York: McGraw-Hill).
1968 "Algebraic Models in Psycholinguistics", in: *Algebraic Models in Psychology*, C. A. J. Vlek (ed.) (The Hague, NUFFIC).

Miller, G. A. and N. Chomsky
1963 "Finitary Models of Language Users", in *Handbook of Mathematical Psychology*, R. D. Luce, R. R. Bush and E. Galanter (eds.) (New York: Wiley).

Miller, G. A. and J. A. Selfridge
1950 "Verbal Context and the Recall of Meaningful Material", *American Journal of Psychology* 63, 176-85.

Montague, R.
1970 "English as a Formal Language", in: *Linguaggi nella società e nella tecnica* (Milan: Edizioni di Comunità), 189-224.
to be published.
"The Proper Treatment of Quantification in English", *Proceedings of the 1970 Stanford Workshop on Grammar and Semantics*, J. Moravscik and P. Suppes (eds.) (Dordrecht: Reidel).
Peters, P. S.
1966 "A Note on the Equivalence of Ordered and Unordered Grammars", *Harvard Computation Laboratory Report of NSF* 17.
Peters, P. S. and R. W. Ritchie
1969a "A Note on the Universal Base Hypothesis", *Journal of Linguistics* 5, 150-52.
1969b "Context-sensitive Immediate Constituent Analysis — Context-free Languages Revisited", *Proceedings of the Ninth ACM Symposium on Theory of Computing*, 1-8. Also in *Mathematical Systems Theory* 6, 1973, 324-333.
1971 "On Restricting the Base Component of Transformational Grammars", *Information and Control* 18, 483-501.
1973 "On the Generative Power of Transformational Grammars", *Information Sciences* 6, 49-83.
Postal, P.
1964a "Limitations of Phrase Structure Grammars", in: *The Structure of Language*, J. A. Fodor and J. J. Katz (eds.) (Englewood Cliffs, N. J.: Prentice-Hall).
1964b *Constituent Structure: A Study of Contemporary Models of Syntactic Description* (Bloomington, Ind.: Indiana University Press).
1970 "On the Surface Verb 'remind'", *Linguistic Inquiry* 1, 37-120.
1972 "A Global Constraint on Pronominalization", *Linguistic Inquiry* 3, 35-60.
Reich, P. A.
1969 "The Finiteness of Natural Language", *Language* 45, 831-43.
Robinson, J. J.
1969 "Case, Category, and Configuration", *Journal of Linguistics* 6, 57-80.
1970 "Dependency Structures and Transformational Rules", *Language* 46, 259-85.
Ross, J. R.
1967 "Constraints on Variables in Syntax" (Cambridge, Mass.: MIT doctoral dissertation).
Sager, N.
1967 "Syntactic Analysis of Natural Language", in: *Advances in Computers* 8, F. Alt and M. Rubinoff (eds.) (New York: Academic Press).
Salomaa, A.
1971 "The Generative Capacity of Transformational Grammars of Ginsburg and Partee", *Information and Control* 18, 227-32.
Šaumjan, S. K. and Soboleva, P. A.
1963 *Applikativnaja poroždajuščaja model' i isčislenie transformacij*

v russkom jazyke (Moscow, Izdatel'stvo Akademii Nauk SSSR).

Saussure, F. de

1916 *Cours de linguistique générale* (Paris).

Schultink, H.

1967 "Transformationeel-generatieve taalbeschrijving", *De Nieuwe Taalgids* 60, 238-57.

Seuren, P. A. M.

1969 *Operators and Nucleus: A Contribution to the Theory of Grammar* (Cambridge: Cambridge University Press).

Shannon, C. and W. Weaver

1949 *The Mathematical Theory of Communication* (Urbana, Ill.: University of Illinois Press).

Steinberg, D. D. and L. A. Jakobovits

1971 *Semantics. An Interdisciplinary Reader in Philosophy, Linguistics and Psychology* (Cambridge: Cambridge University Press).

Suppes, P.

1970 "Probabilistic Grammars for Natural Languages", *Synthese* 22, 95-116.

1971 "Semantics of Context-free Fragments of Natural Languages", *Technical Report* 171, *Inst. Math. Stud. Soc. Sc.* (Stanford University).

Tesnière, L.

1959 *Eléments de syntaxe structurale* (Paris: Klincksieck).

Thatcher, J. W.

1967 "Characterizing Derivation Trees of Context-free Grammars through a Generalization of Finite Automata Theory", *Journal of Computer and Systems Sciences*, 317-22.

Thorne, J. P.

Review of P. Postal's *Constituent Structure: A Study of Contemporary Models of Syntactic Description.*

Watts, W. C.

1970 "On Two Hypotheses Concerning Psycholinguistics", in *Cognition and the Development of Language*, J. R. Hayes (ed.) (New York: Wiley).

Woods, W. A.

1970 "Transition Network Grammars for Natural Language Analysis", *Communications ACM* 13, 591-606.

Wundt, W.

1900 "Volkerpsychologie I und II", *Die Sprache* (Leipzig).

AUTHOR INDEX

SUBJECT INDEX

(Italicized numbers refer to definitions)

FORMAL GRAMMARS IN LINGUISTICS AND PSYCHOLINGUISTICS

VOLUME III

Psycholinguistic Applications

by

W. J. M. LEVELT

1974

MOUTON

THE HAGUE · PARIS

PREFACE

The marriage of linguistics and the psychology of language, while more than a century old, is one of doubtful stability. From time to time the partners get involved in a serious struggle for power, with the outcome that either the psychology of language becomes dependent on linguistics (Steinthal), or linguistics becomes dependent on the psychology of language (Wundt). There are also long periods of coldness in which the two parties tend to ignore each other, as was the case in the first quarter of this century.

Fortunately, however, from time to time one can witness a refreshing and intense cooperation between linguists and psychologists. Such was the case in the 1960's when the new transformational linguistics gained great influence on the psychology of language, in particular through the work of George Miller and his collaborators. During that period various formal models were enthusiastically studied and examined for their "psychological reality". The studies were based on Chomsky's distinction between competence and performance, with which linguists had joyfully thrown themselves into the arms of psychologists ("linguistics is a chapter of psychology").

In the long run psycholinguists, however, could not live up to it, and there followed a period of reflection — but certainly not of cooling-off — during which the relations between the linguistic and psychological models were examined with more objectivity and a greater sense of reality.

This volume will be devoted to a discussion of the connection between formal grammars and psycholinguistic models. The

connection has been worked out in various ways over the three main fields of psycholinguistics: (1) the study of the psychological basis of linguistic intuition; (2) the study of the "primary" linguistic behavior of the speaker-hearer; and (3) the study of language acquisition.

Our aim in this volume will be expressly limited. We do not attempt to present an introduction to the modern psychology of language, but rather only to show how the theory of formal languages and its applications to linguistics have penetrated psycholinguistics. That influence can be seen above all in the syntactic applications, but we shall show that formal language theory is of increasing importance to the semantic and conceptual aspects of the psychology of language, and that those aspects will draw growing attention.

Although this volume is addressed primarily to psychologists, it treats a number of topics which are also of interest to linguists, such as the problem of competence and performance, the structure of linguistic intuitions, and the language of the small child.

The presentation supposes that the reader is familiar with the material given in Volumes I and II, to which reference is often made.

W. J. M. Levelt
Nijmegen

August, 1973

TABLE OF CONTENTS

1

GRAMMARS IN THE PSYCHOLOGY OF LANGUAGE: THREE PROBLEMS

1.1. LANGUAGE USE, LINGUISTIC INTUITIONS, AND THE ACQUISITION OF LANGUAGE

The empirical psychology of language came into being in the second half of the nineteenth century, and developed rapidly to a tentative culmination in the classical work of Wilhelm Wundt (1900). The discipline has always been occupied with the psychological investigation of the acquisition and use of language. Language acquisition involves, in the first place, the process by which the growing child learns to use his native language, but it also includes the learning of a foreign language in later development, and the learning of artificial languages, such as the sign language used by the deaf, and more abstract systems of symbols. It corresponds to the developmental aspect of the psychology of language. The use of language corresponds in turn to the functional aspect. It has to do with such matters as the way man uses his language in communication situations, the way he formulates what he means while speaking, and the way he deciphers what another means while listening. It also takes in the derived processes of reading and writing, the way in which man memorizes verbal material over a longer or shorter period of time, and the way he later reconstructs it, and finally, the relations between speech and other psychological functions such as perception, thinking, decision making, and so forth.

We have stated, for the sake of caution, that the aim of the psychology of language (psycholinguistics) is the *psychological*

investigation of language acquisition and speech. But in this respect, it is difficult to establish the frontiers between linguistic and psychological research. In Volume II, Chapter 1, we stated that the formulation of psychological theory is essentially part of linguistic interpretation. The present volume is almost exclusively concerned with these psychological aspects of linguistic interpretation, and it will be quite evident that it is often impossible to draw a sharp line of demarcation between the two disciplines. In Volume II, Chapter 1, we showed that certain phenomena are considered to fall into the domain of linguistics merely because of more or less arbitrary traditions. Some intuitions of the native speaker are considered to be linguistic, while others are not. The inacceptability of very long sentences, for example, is seen as an intuition irrelevant to linguistics. The history of the psychology of language shows clearly that such traditions are indeed arbitrary. The line drawn between psychology and linguistics, or more precisely, the relationship established between the two sciences, will largely be dependent on some a priori philosophy concerning the relationship between language and the human mind. In recent history opinions have diverged considerably on this point. In the remainder of this section we shall mention, without attempting an exhaustive discussion, two prominent points of view on the subject while making some critical remarks on a number of current trends.

Some authors, such as Steinthal in the nineteenth century and Whorf in the twentieth, claim that the structure of the most important human cognitive functions, such as perception and thought, is determined by language, and that psychology should therefore be based on linguistics. Others reverse the argument, and state that the structure of human language can be understood only on the basis of the structure of the human mind. This latter point of view was already to be found in the nineteenth century among the *Junggrammatiker* (Paul and others), and explicitly in the work of Wundt and many of his disciples. They tried to find psychological explanations for linguistic phenomena, above all for diachronic regularities which were the *pièce de résistance* of nineteenth century linguistic research. But even since de Saussure

revived interest in synchronic relations, many linguists have maintained that their science should, in the final analysis, be drawn back to psychology. Sometimes only lip service was paid to this point of view, without any practical consequence (de Saussure, Bloomfield), but in Chomsky's work the conception was given a new and detailed formulation. For Chomsky, "linguistics is a chapter of human psychology" (Chomsky 1968), and he attempts carefully to delimit that chapter. He calls the psychological object of linguistics LINGUISTICS COMPETENCE, the creative faculty which allows the language user to understand and form an unlimited number of sentences. According to Chomsky, it is a whole of more or less unconscious knowledge which is applied in every act of linguistic behavior, such as speaking and listening. Chomsky calls this actual usage of language PERFORMANCE, and relates it to competence in a way which may be seen in the following quotation:

> To study actual linguistic performance, we must consider the interaction of a variety of factors, of which the underlying competence of the speaker-hearer is only one (Chomsky 1965).

Other psychological variables are such matters as attention, memory span, etc. But this is also an a priori delimitation of the domain of linguistic research, and Chomsky is aware of its tentativeness:

> It must, incidentally, be borne in mind that the specific competence-performance delimitation provided by a grammar represents a hypothesis that might prove to be in error... When a theory of performance ultimately emerges, we may find that some of the facts we are attempting to explain do not really belong to grammar but instead fall under the theory of performance, and that certain facts that we neglect, believing them to be features of performance, should really have been incorporated in the system of grammatical rules (Chomsky and Halle 1968).

It is interesting to notice in this quotation that although Chomsky is aware of the arbitrariness of the delimitation of linguistics, he nevertheless considers it an empirical question. When a complete theory of human linguistic behavior is prepared, it will be evident which part of the theory is concerned with competence and is

therefore the linguistic part. It is clearly presupposed here that empirical evidence will not have to do with the distinction between competence and performance itself, but only with the precise delimitation of competence. This position obscures the empirical character of the very starting point, which can be summarized in the following two points:

(1) The language user disposes of a system of rules called linguistic competence, which is the basis of actual linguistic behavior. Actual linguistic behavior (performance) is the result of the interaction of competence and other psychological factors.

The empirical character of this first point is comparable to that of the psychological notion of intelligence. In psychology, intelligence was originally no more than a theoretical construct with little empirical basis, but a promising program of research was defined with it. For the first time careful distinctions were made between such matters as "dumb" (competence) and "lazy" (performance), and external factors which stimulate or hinder intelligent behavior were discerned. At the same time it gradually became more plausible that a common factor, called "general intelligence", lay at the base of all forms of what could be called "intelligent behavior". Empirical research showed that when that common factor is subtracted, a whole gamut of specific capacities, which play a role in some forms of intelligent behavior and not in others, comes to light. Linguistic competence is an empirical notion in quite the same way. It is originally only a theoretical construct by which a program of research is defined. Only growing empirical evidence can prove that linguistic competence, like intelligence, can be clearly distinguished from other psychological factors, and we might mention in passing that three quarters of a century of research on intelligence has not been sufficient to be really decisive in this regard (cf., for example, Layzer 1973). Also, it is an empirical issue whether linguistic competence is indeed a general factor in human linguistic behavior, or to what extent various specific competences play a role in particular forms of verbal behavior. These latter can nevertheless be specifically verbal,

that is, relatively autonomous with respect to external factors such as attention and perception. The distinction between competence and performance is not a platonic truth, but an empirical psychological question.

(2) A grammar is a description of linguistic competence.

Some explanation will be needed to show that this is also an empirical issue. To do so we shall suppose that proposition (1) is indeed valid, i.e. that linguistic competence, a relatively autonomous factor at the basis of all linguistic behavior, does indeed exist. The question at this point is whether a grammar is a description of that linguistic competence. According to Chomsky, this should in fact be the case:

> We use the term "grammar" with a systematic ambiguity. On the one hand the term refers to the explicit theory constructed by the linguist and proposed as a description of the speaker's competence. On the other hand, we use the term to refer to this competence itself (Chomsky and Halle 1968).

The first part of this quotation states that a grammar should be a theory of linguistic competence, but according to the second part we may as well call the linguistic competence of the language user his grammar. Linguistic competence and linguistic theory would thus completely coincide. The empirical question is whether a (complete) theory of linguistic intuitions is identical with a (complete) theory of human linguistic competence. In Volume II, Chapter 1, we saw that the empirical domain of a Chomsky grammar consists of linguistic intuitions. Chomsky, then, has no doubt as to this identity:

> the grammar is justified to the extent that it correctly describes its object, namely the linguistic intuitions — the tacit competence — of the native speaker (Chomsky 1965).

This, however, is even less a self-evident truth than point (1), for as opposed to the first proposition, the second is unlikely to be confirmed empirically. The theory of one kind of linguistic behavior, namely, metalinguistic judgment on such things as grammaticality

and paraphrase, would then as a whole be built into theories on other forms of linguistic behavior such as speaking and understanding. The theory of linguistic intuitions is competence theory, according to (2); this in turn, according to (1), is part of every performance theory. The priority, given in this way to the theory of linguistic intuitions, has no empirical basis whatsoever. On the contrary, if we wish to think in terms of primary and derived forms of verbal behavior, the speaking and the understanding of language fall precisely into the category of primary forms, while metalinguistic judgments will be considered highly derived, artificial forms of linguistic behavior, which, moreover, are acquired late in development. We mentioned in Volume II, Chapter 1, that nothing is known of the origin of linguistic intuitions; we do not know what role factors such as imagery and verbal and nonverbal context might play, nor do we know to what extent intuitions are learnable, or how they originate and develop in the child.[1] We simply do not know the psychological factors which determine the formation of such intuitions. It would be foolish to make linguistic virtue of psychological necessity by concluding that these factors are unimportant simply because they are unknown, but this is precisely what is done when linguistic intuitions are made the key to linguistic competence. Some introspection, moreover, will make it plausible that the imagination of primary linguistic behavior — speaking and listening in a given situation of communication — plays an important role in the formation of linguistic judgments. Judgments on paraphrase are often based, at least from a phenomenological point of view, on a search of an imagery situation in which both sentences may be said with the same intention; if such a situation can be imagined, the two sentences are considered paraphrases of each other. In a similar way, judgments on grammaticality are perhaps dependent on the possibility of imagining a situation in which a sentence may be said. It would be an illusion to suppose that in such a case external factors, such as memory, motivation, etc., suddenly no longer play a role in the formation of

[1] During the translation of this volume Gleitman, Gleitman, and Shipley (1973) published a highly interesting article on this subject.

the judgment. We unfortunately do not know precisely what that role is, but there is at least as much reason to take a theory of primary verbal behavior as the basis of a theory of linguistic intuition as the inverse. It seems safer, however, to avoid connecting such theories a priori in series.

These remarks on the relation between psychology and grammar allow us at this point further to define the problems which will be discussed in the present volume. As we have already mentioned, the psychology of language is traditionally concerned with the usage of language and with language acquisition. From the point of view of grammars, the empirical problem in the psychology of language is in turn divided in two, the investigation of psychological factors in PRIMARY LANGUAGE USAGE, and the psychological investigation of LINGUISTIC INTUITIONS. The investigation of LANGUAGE ACQUISITION adds still a third problem to this — genetic aspects of grammars. Volume III is subdivided according to these three subjects. As was the case in Volume II, the principal accent will be laid on the formal aspects of the problems concerned, and questions of substance will not be treated systematically.

It should be noted that this division into three empirical fields is above all pragmatic. In principle the various subjects overlap in important and interesting ways. The remainder of the present chapter will evoke a few examples of such relations.

1.2. PRIMARY USAGE OF LANGUAGE AND LINGUISTIC INTUITIONS

In the preceding paragraph linguistic judgments were opposed to primary usage of language in speech and understanding. In the present paragraph we would emphasize that there is an area of rather fluent transition between the two, a field of research to which neither psychologists nor linguists have given much attention.

We can call this area of overlap that of METALINGUISTIC USE OF LANGUAGE. It is obvious that intuitive linguistic judgments are metalinguistic judgments. The object of a judgment of grammati-

cality such as *the sentence "John lives in town" is good English* is a linguistic object (*the sentence S*), and a judgment is made on it as a linguistic object. But the judgment itself is also a sentence, and as such it is subject to paraphrase relations with other sentences. The above example is thus a paraphrase of "*John lives in town*" *is good English.* An interesting question is whether grammatical relations between metalinguistic judgments are determined by the same rules as relations between "ordinary" sentences. The question is difficult to answer in this general form, but a step in the direction of analysis might involve the following, more elementary question: do metalinguistic forms exist in ordinary speech, and if so, how can they be described?

It is in fact easy to find various forms of "ordinary" metalinguistic speech. A few examples are the following:

(1) *John strikes Mary and inversely*
(2) *John strikes Mary and the latter strikes the former*
(3) *John and Mary entered in that order*

In these sentences *inversely*, *latter*, *former*, and *in that order* refer to the order of constituents in the sentence; in that respect they are metalinguistic.

There are other metalinguistic sentences in which the vocabulary of the language is extended, as it were, to include a linguistic auxiliary vocabulary, that is, to include words like *sentence*, *word*, *manner*, *rhyme*.

(4) *the sentence "John lives in town" has four words*
(5) *the word "biscuits" comes from French*
(6) *they sang a song called "By the Old Mill Stream"*
(7) *he looked at her in a "what do I care" manner*
(8) *"house" rhymes with "mouse"*

The use of these and other metalinguistic constructions is bound by specific rules. Restrictions of the use of *latter*, *former*, and *in that order* may be seen in the following constructions which are rather difficult to accept.

(9) **after they had entered the house in that order, John struck Mary*

(10) **after the latter had struck the former, John and Mary entered*

Latter and *former* do not simply function as pronouns, for we can say the following instead of (10):

(11) *after he had struck her, John and Mary entered*

and we obviously cannot paraphrase (2) with:

(12) *Mary was struck by John and the latter struck the former*

This shows that sentence (2) must have a complicated syntactic description. To derive sentence (2) it is evidently first necessary to generate the first half in such a way that the surface word order has *John* and *Mary*, and the second half of the sentence must in some way be generated so as to make reference to this surface order. In the syntactic description of metalinguistic sentences, coordination will often necessitate the introduction of the surface form of one subsentence into the underlying structure of the other subsentence. None of the transformational grammars in Volume II offers the technical means for accomplishing this. Nevertheless we are dealing here with ordinary speech, and it is not audacious to assert that linguistic judgments are an extension of it. Moreover, such usage can be consciously learned. This is precisely what happens in secondary and higher education: the language student becomes familiar with a whole vocabulary whose function is to refer to the language; the vocabulary includes such words as *vowel*, *morpheme*, *coordination*, *constituent*, *transformation*, and so forth.

Metalinguistic speech is not limited to one level. Just as one can speak of the language, he can also speak of metalinguistic sentences, and there again certain restrictions will apply. It is not to be expected that a system of rules which is adequate for one level should also be adequate for the next higher level. In this way linguists create a language to which their grammars are not applicable.

Not only is there a linguistic vacuum around metalinguistic usage, but there is also little known of this reflective behavior from a psychological point of view. Experiments have never been carried out on the way the hearer assimilates such metalinguistic sentences as examples (1) to (8). Likewise, psycholinguistics has yet to begin the investigation of the closely related field of the psychological background of metaphor. (Gleitman and Gleitman [1970] have, however, developed an ingenious experiment on the origin of the paraphrase judgment.)

1.3. LINGUISTIC INTUITIONS AND LANGUAGE ACQUISITION

We know as little about the acquisition of linguistic intuitions among children as we do of their nature among adults (see, however, footnote on page 6). But it is generally accepted in the investigation of children's languages that the small child is hardly capable of making judgments on his own language. As we have seen in Volume II, Chapter 1, this leads to an essentially different linguistic analysis; the child's grammar is not tested on the basis of intuitions, but rather on a corpus of utterances, complemented by understanding games, imitation, and careful notation of the situational circumstances in which certain utterances occur. The relation between such a speaker/hearer grammar and an intuition grammar is unknown. We mentioned the analysis of children's languages as a characteristic example of a linguistic problem of interpretation, and stated that psychological theory is indispensable to its solution.

A modest step toward research on the growth of linguistic intuitions in the child would be the investigation of metalinguistic usage. When does the child produce and understand metasentences such as examples (1) to (8)? Here, too, little is known, but some aspects of the problem are striking. The child, for example, is quite rapid in acquiring the notion of what a word is; if a three year old is asked what a word is, he will respond with a series of nouns and proper names. At the same age he will understand

what rhyme is. The child, therefore, apparently disposes of metalinguistic notions of word unity and sound associations quite early. But it is only later that he comes to the idea of sentence; two and three year olds notice ungrammaticality in word order only by exception (de Villiers and de Villiers 1972). It is quite normal for a child of six or seven not to know what a sentence is, or to reserve the notion for declarative sentences while excluding interrogative and imperative sentences. An early form of metalinguistic behavior was described by Weir (1962). She showed how the child practices syntactic forms by successively fitting various words into a given sentence frame: *what color — what color blanket — what color mop — what color glass* — etc.; this was done by the child immediately before falling asleep. The activity was spontaneously performed without training. The influence which metalinguistic behavior has on language acquisition itself is unknown, but it is probably considerable. In Chapter 4 we shall discuss a model of language acquisition in which metalinguistic feedback plays an essential role. In its explicit form, that feedback consists of an announcement on the part of the educator that a given utterance is or is not good English; there are, of course, other more subtle forms such as the giving of an inadequate reaction to an utterance which has not been understood.

There is also an interesting inverse relation between language acquisition and linguistic intuition. McNeill (1966) shows that adults have rather accurate intuitions on the level of development at which different sentences are formed. In an experiment, he asked adults to rank several sentences according to degree of grammaticality. The sentences were spontaneous utterances of an English-speaking child at the ages of twenty-six, twenty-eight and a half, and thirty-one months. They were selected for the experiment in such a way that age and sentence length were not correlated. The result was that the ranking of grammaticality as judged by the adults corresponded well with the stages of development at which the sentences were produced. Nearly the same was true of the judgment of adults who had acquired English only as a second language. McNeill relates this to his theory on the differen-

tiation of word classes. For him, all children, regardless of linguistic environment, begin with the same fundamental word classes, and the first differentiations which are introduced into those word classes are also universal. The differentiation of classes is also reflected in the syntactic structure of sentences produced at various levels of development. The last stage of differentiation is the adult's system of categories. The system of syntactic categories used by the adult, however, is not only a refinement of the various systems used by the child: the actual sequence of the phases of development is reflected in the adult's hierarchical subcategorization. Thus, for example, not only does the primitive distinction between verb and noun remain, but verbs are further differentiated at a later stage of development into subcategories of transitive and intransitive verbs, nouns are further distinguished as concrete or abstract, or divided into other subcategories. On this point, McNeill relates his model to Chomsky's grammaticality model (Chomsky 1964), according to which sentences become ungrammatical through the violation of category and subcategory features. Ungrammaticality increases when a more fundamental category is violated, that is, a category which is localized higher in the hierarchy. The the string *the students elephant the car* (violation of the distinction between noun and verb) is more seriously ungrammatical than *the students laugh a car* (violation of the distinction between transitive and intransitive). Sentences produced at a developmental stage where a certain category or subcategory has not yet been acquired, therefore, will, on the average, be more seriously ungrammatical than sentences produced at a later stage, where the differentiation in question has already been introduced.

One can rightly pose objections to Chomsky's classification into strictly hierarchical subcategories. Cross-classification of subcategories also occurs, as Chomsky shows in *Aspects*. That cross-classification model is in fact adopted by McNeill (1971) in his theory of the development of word class differentiation. But the fact that judgments on grammaticality and the stages of language development show a strong relationship calls for further research. There is, moreover, a set of related psycholinguistic phenomena,

such as the apparent ease with which adults speak children's languages and the ease with which small children imitate the languages of even smaller children. We know of no research already done on the development of this "feeling for language".

1.4. LANGUAGE ACQUISITION AND PRIMARY USAGE OF LANGUAGE

Nearly all research on language acquisition has been based on the child's usage of language. Research on the interaction of these two psycholinguistic aspects is therefore not easily distinguishable from research on language acquisition as such. We refer, therefore, to Chapter 4 of the present volume, in which this subject will be further discussed.

2

GRAMMARS AND LINGUISTIC INTUITIONS

2.1. THE UNRELIABILITY OF LINGUISTIC INTUITIONS

The empirical touchstone in the tradition of transformational linguistics is the linguistic intuition, either of the linguist himself or of an informant. This is also the case in other linguistic traditions, but not in all. Some linguists write grammars for a given corpus, at times on principle, and at times because they are forced to do so for lack of informants. Without taking position on the problem of whether or not intuitions constitute a sufficient basis for a complete language theory, we can in any case propose that their importance in linguistics is essentially limited by the degree to which they are unreliable. It is a dangerous practice in linguistics to conclude from the lack of psychological information on the process of linguistic judgment that intuitions are indeed reliable. Although incidental words of caution may be found in linguistic literature, their effect is negligible. Chomsky warns his readers that he does not mean "that the speaker's statements about his intuitive knowledge are necessarily accurate" (Chomsky 1965), and further states that

> in short, we must be careful not to overlook the fact that surface similarities may hide underlying distinctions of a fundamental nature, and that it may be necessary to guide and draw out the speaker's intuition in perhaps fairly subtle ways before we can determine what is the actual character of his knowledge of his language or of anything else.

As we pointed out in Volume II, Chapter 1, Chomsky (1957)

emphasizes that, as far as possible, grammars should be constructed on the basis of clear cases with regard to grammaticality. If the grammar is adequate for those cases, the status of less clear cases can be deduced from the grammar itself, and the intuitive judgment is no longer necessary.

After the first phase of the development of transformational generative linguistics, little seems to remain of these two directives in linguistic practice. Instead of an increasing number of cases in which the theory decides on the grammatical status of half-acceptable sentences, we find an enormous increase of examples in which sentences of doubtful grammaticality are applied as tests of syntactic rules.

In order to show how serious this development is, we offer an elaborate example of it. Fourteen sentences taken from a reader on transformational linguistics (Jacobs and Rosenbaum 1970) follow. In that book, each of the sentences is marked by the author concerned[1] as grammatical or ungrammatical. We shall allow the reader himself, however, to decide which of the sentences were marked ungrammatical in the original text. The original judgments of the respective authors may be found in a note at the end of this chapter. In making his judgment, the reader should imagine that the sentence is presented to him in spoken form.

(1) *Your making of reference to the book displeased the author* (Fraser)
(2) *No American, who was wise, remained in the country* (Postal)
(3) *They never insulted the men, who were democrats* (Postal)
(4) *They never agreed with us planners* (Postal)
(5) *The talking about the problem saved her* (Fraser)
(6) *The machine's crushing of the rock was noisy* (Fraser)
(7) *The giving of the lecture by the man who arrived yesterday assisted us* (Fraser)
(8) *Your making of a reference to the book displeased the author* (Fraser)
(9) *Her slicing up of the cake was clever* (Fraser)

[1] The name of the author is given in parentheses after each sentence.

(10) *John's cutting up of four cords of wood yesterday and his doing so again today was a welcome gesture* (Fraser)
(11) *John's tendency to sleep along with Mary's tendency not to do so ruined the party* (Fraser)
(12) *I didn't believe it, although Sid asserted that Max left* (Lakoff)
(13) *I did't believe that John would leave until tomorrow* (Lakoff)
(14) *His criticism of the book before he read it* (given as a noun phrase) (Chomsky)

We used these fourteen sentences as a demonstration example for a group of twenty-four trained linguists, and asked them to judge which sentences were marked as ungrammatical in the original text. The results of this little experiment, also given in the note at the end of this chapter, showed that the sentences marked ungrammatical by the authors had half as much chance of being judged ungrammatical by the linguists as those marked grammatical by the authors. This is precisely the opposite of what might have been expected. Though this experiment (reported in further detail in Levelt 1972) was not watertight because none of the judges was a native English speaker (but all had had higher education in English and many were specialists in the study of the English language); however, the results are alarming enough to incite us to caution in the use of linguistic intuitions. These fourteen sentences can also act as the basis of a discussion of a number of factors which contribute to the unreliability of linguistic judgment but are systematically underestimated and often denied by linguists.

The context of linguistic presentation. The grammatical status of many examples among sentences (1) to (14) is well indicated in the original articles, but outside of that context, the same sentences become problematic. The development of the argument in a linguistic article influences the grammaticality judgment in a way which has not yet been investigated.

Comparison with other sentences. A sentence which appears to be grammatical in isolation can nevertheless become ungrammatical

when it is compared with other sentences. Sentence (1), for example, loses much of its grammaticality if sentence (8) is presented first. Another example is the doubtful grammaticality of the following sentence:

(15) *Tom was not present, and many of the girls believed that the paper had been written by Ann and him himself.*

Ross marks the sentence grammatical (in Jacobs and Rosenbaum 1970) in contrast to the following sentence:

(16) *Tom was not present, and many of the girls believed that the paper had been written by Ann and himself*

which, in his opinion, is ungrammatical.

Judging isolated sentences differs very much from judging contrasting sentences. Which of the two methods is to be preferred? It is the exception rather than the rule that a stable criterion can be maintained by a judge in an actual judgment situation. Psychology makes it quite clear that such a criterion will be sensitive to pay off, that is, independently of the possibility of distinguishing grammaticality from ungrammaticality, the percentage of judgments "grammatical" will increase when the judge feels that such a reaction is desired of him. This is anything but an imaginary factor in present linguistic practice. As we have already pointed out, a linguistic article in itself can often induce a certain expectation in the first place, and that expectation can influence the criterion in one direction or in the other. But when the linguist is his own informant, reward or pay off and the criterion can no longer be distinguished from each other. The linguist's theoretical expectation on a given sentence is also determinant for the position of the criterion of grammaticality, but its influence will not necessarily be in a direction advantageous to the theory. The critical (or hypercritical) linguist can also show the reverse tendency. The point is that it is an illusion to think that an objective absolute judgment of grammaticality is possible. Abstracting from the effects of pay off, absolute judgments usually also show the effects of a central tendency. If a sequence of certainly grammatical sen-

tences is given, followed by a somewhat less grammatical sentence, this latter has a good chance of being judged as ungrammatical. Beside the reward effect, there is a tendency toward a fifty-fifty criterion. It would be best advice for the linguist who wishes to show a sentence of doubtful grammaticality to be grammatical to place that sentence at the end of a series of strongly ungrammatical examples.

There is, therefore, good reason to mistrust absolute judgments of grammaticality. Judgment of contrasting examples, in which the position of the criterion no longer plays a role, seems to be a considerably safer procedure. This does, however, lead to a type of linguistic data which is related to the linguistic theory in a different way. We shall return to this problem of interpretation later.

The use of unnatural and misleading examples. This is a common practice everywhere, as we see in the following examples, taken from the same reader;

(17) *The number of dollars that a dozen eggs cost in China is greater than the number of degrees centigrade the temperature was in Chicago* (Hale)

(18) *That Tom's told everyone that he's staying proves that he's thinking that it would be a good idea for him to show that he likes it here* (Langendoen)

(19) *I dreamed that I was a proton and fell in love with a shapely green-and-orange striped electron* (McCawley)

(20) *Tom thinks that I tried to get Mary to make you say that the paper had been written by Ann and him himself* (Ross)

In all of these cases, misleading factors are expressly introduced, rather than being eliminated. This can only increase the unreliability of the judgments. The reader may give his judgment on sentences (17) to (20); the judgments of the original authors are given at the end of this chapter.

The linguist as his own informant. The transformational linguist usually bases his arguments on his own intuitive judgments.

We have already pointed out that certain theoretical expectations on his part can influence the position of the criterion in the judgment situation. But the combination of linguist and informant can also be the source of problems in other judgment situations, such as the judgment of grammaticality on the basis of contrasting sentences and paraphrase judgments. The question is how the nature of the criterion is related to the linguistic training of the investigator. This is a particular instance of an old psychological problem — the use of trained subjects. At the beginning of this century, the Würzburg studies on thinking (Ach, Bühler) were among the first in which trained subjects were used for systematic introspection. The practice was a source of much vexation to Wundt, who on more than one occasion (1907; 1908) rejected it as unscientific. Van de Geer (1957) gives a survey of the discussions on the matter, and wonders in which field the subjects were actually trained. He writes that in Wundt's day

> training was assumed to be an unlearning of bad perceiving-habits, not the learning of a specific technique of perceiving. Nowadays we are inclined to say that the subjects were trained in a specific technique, and we recognize that different training systems may lead to different results

and further,

> one serious objection can be maintained: the special training of the subjects and the impossibility to see in how far the Würzburg results are a consequence of this training. This objection is the more cogent as other studies produced results which were at variance with those of the Würzburg school.

These considerations are almost literally applicable to the present situation in linguistic practice. Chomsky's warning, quoted at the beginning of this paragraph, is based on the supposition that a careful elimination of external factors (such as surface similarities) in the judgment situation will lead to the discovery of the "real" underlying knowledge. Linguistic training is useful in that it makes the judge aware of such factors: thanks to his training, the linguist unlearns his bad perceiving habits. But, in linguistic literature,

we seldom find instances of awareness of the fact that linguistic training also determines the form of the criterion itself, which expresses itself in a certain judgment technique. To illustrate this, we need not even compare the different schools of linguistics. A linguist trained in the transformational grammar of the type presented in *Syntactic Structures* will judge the string *colorless green ideas sleep furiously* as grammatical (in the restricted syntactic sense of the word), although it is semantically abnormal. A linguist trained in the *Aspects* theory will find the same string ungrammatical, because (syntactic) lexical insertion rules have not been respected in its derivation. The linguist trained in generative semantics, on the other hand, will in turn judge the string as grammatical, because the selection restrictions which have been violated are purely semantic in nature. We see here that the same phenomenon is alternately called semantic and syntactic, independently of the form of the theory, and this in turn determines the nature of the criterion of judgment. In this regard, judgments can only confirm the theory. If we hold the convention that selection restrictions are semantic, it is the theory which decides that *colorless green ideas sleep furiously* is syntactically correct. The judgment of the linguist adds nothing to this. We do not wish to say by this that it is pointless to train informants. One can make the judge aware of a particular characteristic of sentences, of theoretical importance at that moment. The theory can be tested on such judgments. One would, however, prefer that linguists were completely explicit on the nature of the criterion which they use in their judgments. With sufficient precautions, the use of trained subjects can indeed be useful, as is apparent in the history of psychology. The entire field of human psychophysics is based on experiments in which trained subjects were used, and it does not appear that any great problem occurred.

Written language versus spoken language. Linguistic judgment is often clearly based on the written form of the sentence, and at times even on the punctuation. Sentence (2) would be a good example of this. It remains an open question as to whether or not

there is an acoustic equivalent to the commas when the sentence is spoken. If not, it is impossible to hear whether the relative clause is restrictive or descriptive. It is also unclear whether this distinction is indeed of syntactic nature, or whether we are dealing with a semantic or even pragmatic characteristic which has come to be expressed in writing in our culture in the form of punctuation.

2.2 FROM DATA TO MODEL

If we suppose that all the problems of reliability mentioned in the preceding paragraph have been solved, we must still ask what the linguist can do with his reliable data. Data would offer the linguist the opportunity to test his theory, but this does not work only in one direction. The theory (grammar) determines which data are relevant, or, in other words, which linguistic intuitions must be investigated in order to justify certain conclusions. In Volume II, Chapter 1, we stated that the formal relations between data and theory are elaborated in the theory of linguistic interpretation. This theory may be said to indicate how the data (intuitions) are represented in the model (the grammar). In this respect the theory of interpretation fills the same function in linguistics as measurement theory in the social sciences (cf. Krantz, et al. 1971). But unlike the measurement theory, the theory of interpretation is only at the first stage of its development. In Volume II we discussed two cases in which the absence of a theory of interpretation had serious consequences for the testing of a linguistic model. The first case was Chomsky's rejection of the regular model for natural languages (Volume II, Chapter 2, section 2.2.). We showed that the data used by him were insufficient to justify his conclusion. On the ground of a bit of interpretation theory developed there ad hoc, however, it was possible to refer to, and (to a certain extent) to find, data which could tentatively justify the conclusion. The second case was Postal's rejection of the context-free model for natural languages (Volume II, Chapter 2, section 2.3.). His "proof" did not relate his data to the model he tested.

Postal's argument nevertheless appeared quite clear on first examination, and many references are made to it in linguistic theory. Without interpretation theory, such snares remain for the linguist. The choice between absolute judgment of grammaticality and judgment by contrast, for example, has repercussions for the possiblity of testing the grammar. In the following paragraph we shall illustrate this to a certain degree, but without a theory of interpretation, such considerations remain ad hoc remarks.

The remainder of this chapter will deal with the formal aspects of interpretation of two types of linguistic intuitions, namely, those concerning grammaticality and those concerning syntactic cohesion. On the first point, hardly any experimental work has been done on the testing of formal linguistic theories. Our remarks will therefore be limited to a few fundamental aspects of the relationship between data and theory, in particular concerning judgments of grammaticality by contrast. The second topic is an exercise in linguistic interpretation. Without arriving at definitive conclusions, we will show how a formal interpretation theory can be constructed and tested experimentally.

2.3. THE JUDGMENT OF GRAMMATICALITY: ABSOLUTE JUDGMENT VERSUS JUDGMENT BY CONTRAST

It is only since the generative point of view became common in linguistics that the absolute judgment of grammaticality came to be of great importance to theory. In *Syntactic Structures* Chomsky wrote:

> The fundamental aim in the linguistic analysis of a language L is to separate the *grammatical* sequences which are the sentences of L from the *ungrammatical* sequences which are not sentences of L and to study the structure of the grammatical sequences. The grammar of L will thus be a device that generates all of the grammatical sequences of L and none of the ungrammatical ones.

Only absolute grammaticality defines the language, and the language is what the linguist describes. Relative grammaticality or gradation of grammaticality is an interesting but secondary prob-

lem in this point of view. The principal distinction is that between grammatical and ungrammatical, and an order of grammaticality can be determined only for the ungrammatical sentences. Chomsky (1964) and others (Katz 1964; Ziff 1964; Lakoff 1971) have developed theories on degrees of ungrammaticality. They are all based on the consideration that given a grammar, by the systematic violation of certain rules, ungrammaticality can be varied as a function of the seriousness and the number of those violations. None of these theories has ever been the object of direct experimental tests.[1] Experimental work on grammaticality, such as that of Maclay and Sleator (1960) and of Quirk and Svartvik (1965), has been concerned principally with the relations between that which subjects understand by "ungrammaticality" and other experimental variables, such as judgment of "meaningfulness" of a sentence, and behavioral tests concerning the sentence (accuracy in making a semi-grammatical sentence passive or interrogative, etc.). Systematic predictions which can be made on the basis of the formal theory have never been tested. Given the secondary importance of the phenomenon of ungrammaticality, such tests would moreover have been rather indirect for such formal theories. According to the point of view of *Syntactic Structures*, it is in the grammatical sentences, and therefore in absolute grammaticality, that the real interest lies.

It should be pointed out, however, that, in other schools of linguistics, *relative* grammaticality is at least as important a notion. In Harris' transformation theory (both his earlier theory and his present operator grammar), paraphrastic transformation is defined by an equivalence relation between two classes of sentences in the language (for example, active and passive), with the property that the order of acceptability within the two classes is equal. The supposition behind this argument is that sentences in the language do indeed vary in acceptability. It should be noticed that acceptability is not identical to grammaticality, but Harris does not make

[1] During the translation of this book an article by Moore (1972) appeared in which Chomsky's theory was investigated experimentally with a completely negative result.

the distinction, and the context makes it clear that for him acceptability is a theoretical linguistic concept. It may therefore be said that RELATIVE GRAMMATICALITY is the central notion here; it is a condition on paraphrastic transformations. It is in fact nothing other than the supposition that transformations are independent of each other in their effect on the acceptability of the sentence. Suppose that sentence x is more acceptable than sentence y; we may represent this as $x \succ y$. Harris' condition states that if x' and y' are transformations of x and y, respectively, by means of the application of the optional (paraphrastic) transformation t_1, then it must hold that $x' \succ y'$. Suppose that x'' and y'' are, in turn, transformations of x' and y' by application of transformation t_2; it must then hold that $x'' \succ y''$. In other words, the order of acceptability resulting from the application of t_1 cannot be reversed by the application of t_2. But notice that this condition of independence says nothing of the effect of the transformation itself on acceptability. The effect of t_1 can as easily be $x \succ x'$ as $x \prec x'$. We shall return to this subject later.

There is nothing in *Aspects* to render this assumption of independence of optional transformations improbable. However, such an assumption is quite irrelevant to the framework of the *Aspects* theory in the first place. All the deep structures generated by the base are grammatical; on that level there is no gradation in grammaticality, and consequently, the assumption of independence is trivial. But in the practice of later developments, the very definition of deep structure becomes problematic. In the framework of generative semantics it is no longer clear how the underlying structures are generated, and one cannot be certain in advance that a given underlying structure can be accounted for on the basis of the (unknown) base grammar. This need not be a great hindrance to the investigation of transformational relations among sentences, provided that attention is paid to the fact that sentences of doubtful grammaticality can again play a role in the testing of transformations, and, as we have seen in the preceding section, this is a real problem. Under these circumstances the principle of independence becomes interesting once again.

One of the tests for the correct formulation of a transformation will be to see if it has any influence on the order of acceptability of the sentences. The investigation of grammaticality by contrasting sentences will be sufficent to test this prediction. For a given transformation a number of characteristic sentences x_1, x_2, ..., is determined where it holds for every pair x_i, x_j, that if $x_i \succ x_j$, then $x'_i \succ x'_j$ (when x and x' differ only in that x' has had the transformation in its derivation and x has not). It is therefore of secondary interest to know whether the sentences x satisfy the highest norms of grammaticality. Uncertainty in that regard does not effect the validity of the test to the extent that the transformation must in any case be rejected if the order of grammaticality is reversed. It is not the case, however, that the test is also *sufficient* for the demonstration of the transformation. Imagine, for example, the following transformation: t permutates every article with the rest of the noun phrase. Thus, if $x =$ *the child eats an ice cream*, then $t(x) = x' =$ *child the eats ice cream an*. Suppose we have $y =$ *the ice cream eats a child*, and $x \succ y$, then it decidedly also holds for $t(y) = y'$ that $x' \succ y'$: *child the eats ice cream an* is more grammatical than *ice cream the eats child an*. But t is clearly not a transformation in English. A further requisite which must apparently be stated is that an optional transformation should not decrease grammaticality; that is, it may never happen that $x \succ x'$, where x' is the transformed equivalent of x. Indifference, indicated by $x \sim x'$, seems to be an acceptable situation, however. A second condition on optional transformations is therefore $x \precsim x'$: the transformation may not lead to a decrease in grammaticality. It is also the case for this second condition that it is of little importance that there be some doubt as to the grammaticality of x, thanks to the condition of independence. Judgment of grammaticality by contrast is also sufficient for testing $x \precsim x'$.

As long as no theoretical objections can be brought against the assumption of the independence of optional transformations, judgment of grammaticality by contrasting sentences can play an important role in the testing of such transformations.

What is the situation for obligatory transformations? The

consequence of omitting the application of an obligatory transformation is the derivation of an ungrammatical string. If x is such a string and x' is the corresponding grammatical sentence, then $x \prec x'$. If there is uncertainty as to the grammaticality of x', as occurs at times in present practice, this inequality can nevertheless be investigated by means of judgment by contrast. We must, however, be sure that subsequent transformations cannot reverse the order of grammaticality. For this reason we must once again call upon the assumption of independence. As long as there are no theoretical objections, obligatory transformations can also be investigated to a large extent by means of judgment by contrast. In that case the absolute grammaticality of the example sentences is of minor importance. To sum up, the following tests by contrast may be performed for transformations:

for optional transformations: $x \precsim x'$
for obligatory transformations: $x \prec x'$, for all pairs x, x'
for both: if $x \succ y$, then $x' \succ y'$, for all pairs x, y and x', y'

Finally, we would point out that ranking judgments can give the linguist some insight into the futility of his problem. In the opinion of some authors (e.g. Suppes 1970), linguists are fond of all sorts of marginal transformational phenomena, while the attention paid to the more essential linguistic rules is considerably less. Obligatory linguistic rules could be ranked according to *importance in the grammar*. Violation of the more important rules leads to serious failure of communication, while violation of marginal rules leads only to rather mild forms of ungrammaticality. Let c be a central rule and m a marginal one. Let $x(c)$ be the sentence in which only rule p, and not rule c, is violated; let $x(m)$ be the inverse situation. In that case, $x(c, m) \succ x(c) \succ x(m)$. These inequalities can in turn also be investigated by means of judgment by contrast, and, in principle, rules of the grammar can be ranked according to importance. Thus one could retain minor problems for freshman students, and all dissertations could, on the other hand, be relevant.

2.4. THE JUDGMENT OF SYNTACTIC RELATEDNESS: A FEW MODELS OF INTERPRETATION

The intuition of grammaticality is the very basis on which generative linguistics defines a natural language. By definition, the correct prediction of that intuition by a grammar signifies observational adequacy. But if a grammar G is observationally adequate for language L, all grammars equivalent to G are also observationally adequate for L. Judgment of absolute grammaticality is therefore neutral in regard to the descriptive adequacy of a grammar. Thus we have seen, for example, that a context-free grammar is weakly equivalent to a dependency grammar. Judgments of absolute grammaticality can never differentiate between these two types of grammars. Judgments of relative grammaticality can say something on descriptive adequacy, but only in an indirect way. Two weakly equivalent grammars will, in general, define two different rank orders of ungrammaticality over the complement of the language. The testing of such gradations can therefore give indirect evidence for the descriptive adequacy of the grammar. But this is an unsatisfactory method — much as if psychology could only work with rats, or theology alone with fellow-men.

For the direct investigation of the descriptive adequacy of a grammar, that is, for the investigation of the correctness of the structural descriptions, intuitive judgments of another nature are needed; we call them STRUCTURAL INTUITIONS. One of the most often used structural intuitions is expressed in the paraphrase judgment. Two sentences with the same underlying structure are, according to the *Aspects* model, paraphrases of each other. The paraphrase judgment offers the possibility of concluding that two sentences differ in deep structure. Without underestimating the linguistic importance of such judgments, we would point out that the psychological problems which occur in this connection are even more treacherous — if that be possible — than those which occur in the judgment of grammaticality. The principal problem is that two sentences are never complete paraphrases of each other. The manner and degree in which they are paraphrases is dependent

on various syntactic, semantic, and pragmatic factors. Compare, for example, the following sentences, all of which are paraphrases:

(1) *I disapprove of the fact that John catches butterflies.*
(2) *I disapprove of the fact that butterflies are caught by John*
(3) *I disapprove of John's catching butterflies*
(4) *I do not like the idea of John's catching butterflies*
(5) *I object to John's catching butterflies*
(6) *I do not consent to John's catching butterflies*

We shall not discuss this question here. For the only psychological investigations on paraphrase judgments known to us, we refer to Gleitman and Gleitman (1970) and Honeck (1971).

In this section we shall discuss a type of structural intuition which is sometimes used in linguistic practice and which can offer direct insight into the structure of the sentence: intuitions on syntactic cohesion. Cohesion intuitions are expressed in judgments on whether or not words or phrases belong together in a sentence. Chomsky (1965) uses cohesion intuitions for the study of relations between the main verb and prepositional phrases:

> It is well known that in Verb—Prepositional Phrase constructions one can distinguish various degrees of "cohesion" between the verb and the accompanying Prepositional Phrase.

He illustrates this with the sentence *He decided on the boat* which can be read in two ways. *On the boat* refers either to the place or to the object of the decision. This is clear when we compare it with the following nonambiguous sentence: *He decided on the boat on the train.* Chomsky writes that in the latter sentence "the first prepositional phrase ... is in close construction to the verb", and he modifies the base grammar to agree with this insight. Cohesion is a direct and potentially valuable structural intuition, but its use in linguistics demands a theory of interpretation which establishes the relation between syntactic structure and cohesion judgment. The absence of such a theory easily leads to quasi-arguments and to confusion. Thus Uhlenbeck (1964) correctly pointed out that a parsing such as *(the man saw) (the*

boy) would not be in conflict with cohesion intuitions, quite in agreement with experimental results of the type to be discussed presently. The fact that a parsing such as *(the man put) (it into the box)* does conflict with intuition (Chomsky 1965) calls for explanation, but it is not an argument against the first analysis without further theory on cohesion intuitions.

2.4.1. *Methods for the Measurement of Syntactic Relatedness*

Let us use the simple sentence *John breaks in* as an example. There is a gamut of methods for having subjects judge how strong the syntactic relations are among the three words of this sentence. The following three, however, are the most common in the literature (in all three the judge is exposed to the complete sentence and is instructed constantly to relate his judgments to it).

(i) *Rank ordering of Word Pairs*

The example sentence contains three word pairs — *(John, breaks)*, *(breaks, in)*, and *(John, in)*. The subject is asked to rank these word pairs according to relatedness. The most probable result is (from strong to weak): *(breaks, in)*, *(John, breaks)*, *(John, in)*. For longer sentences, where the number of pairs becomes quite large, the task can be facilitated in several ways. One of these is TRIADIC COMPARISONS, in which the subject must indicate for every triad of words from the sentence which pair has the strongest relation in the sentence, and which has the weakest. The triads may be presented, for example, as shown in Figure 2.1. The subject marks his judgment in every triangle by placing a plus sign (+) at the side of the triangle showing the strongest relation, and a minus sign (−) at the side showing the weakest relation. When every triad for the sentence has been judged, each word pair can be assigned a number which represents the relatedness judgment. This can also be done in various ways. One of these consists of counting the number of times a word pair is judged as stronger than other word pairs. Thus, in Figure 2.1., the pair *(breaks, in)*

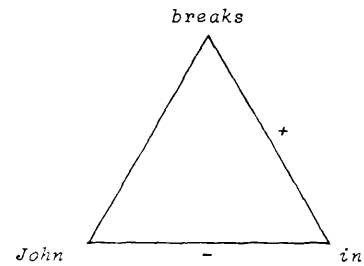


Fig. 2.1. An example of triadic comparison

is judged as more strongly related than either *(John, breaks)* or *(John, in)*; this gives a score of 2. The pair *(John, breaks)* has a score of 1, because it is more strongly related than only one other pair, *(John, in)*, which in turn has a score of 0. If there are more than three words in the sentence, the scores are added for all the triads in which the word pair occurs, yielding the final score for the pair. Other methods of determining the final score are also possible, but we need not describe them here.

(ii) *Assigning Scale Values to Word Pairs*

The subject may be asked to indicate the degree of relatedness of a word pair by means of a number. The most common method for this is the SEVEN-POINT SCALE. The subject indicates his judgment on each word pair by circling a scale value, as shown in Figure 2.2., where *(John, breaks)* has the value of 5, *(John, in)* has 2, and *(breaks, in)* has 6.

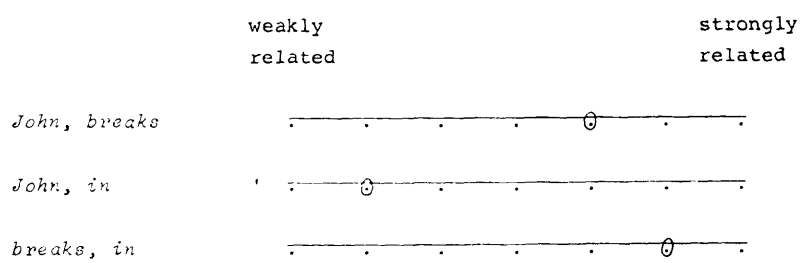


Fig. 2.2. An example of the seven-point scale method

Another common method is that of MAGNITUDE ESTIMATION. For a given word pair (the "standard"), a certain scale value is assigned beforehand, and it will be on the basis of this standard that the subject will make his judgments. Thus if the standard is *(John, breaks)* with an assigned standard value of 50, another pair, which intuitively has half as strong a relation, receives the judgment 25. This could for instance be the case for *(John, in)*. In the same way, a pair which is related twice as strongly will be assigned the value 100 by the subject. This could in the example be the case for the pair *(breaks, in)*. In general the subject will estimate the strength of the relations on the basis of the standard. There are many other methods by which the subject can assign numerical values to word pairs, but only those mentioned here are ordinarily used for this purpose.

(iii) *Word Sorting*

The subject can be presented with the entire sentence, as well as with the individual words, each of which is written on a separate card. He is then asked to sort out the cards into stacks according to the relatedness of the words in the sentence. Thus syntactically related words should be placed in the same stack, while words which have little to do with each other should, as far as possible, be placed in separate stacks. The informant may make as many stacks as he likes, and each of the stacks may contain as many cards as he likes.

In this way each word pair is given a score of either 1 or 0. When two cards are placed in the same stack, they have a relatedness score of 1, and when they are placed in separate stacks, they have a score of 0. Suppose that given the sentence *John breaks in*, a subject makes two stacks, {*John*} and {*breaks, in*}. This renders a score of 1 for *(breaks, in)*, and a score of 0 for the other two pairs. A dichotomy of this sort is not usually very informative. The sorting process would have to be repeated by other subjects (or possibly also the same subject) in order to obtain more gradation in the values. The scores for each word pair could then be added.

The maximum total score for a word pair would then be equal to the number of subjects (or trials); it would be obtained if two words were grouped together by all subjects, or by the same subject at all trials.

Each of these methods of judgment has its own advantages and disadvantages. We shall not discuss them here, but we shall mention them later when necessary.

It is obvious that all of these methods must be accompanied by careful instruction, in which it is explained to the subjects that the problem is one of sentence structure, and not of meaning relations which may by chance exist between the words of the sentence. This can be illustrated for the subjects by means of various examples. Experience has proven that subjects in general have little difficulty in understanding their task.

2.4.2. *A Constituent Model for Relatedness Judgments*

An interpretation theory is necessary in order to connect relatedness judgments to a linguistic theory. The purpose is, of course, to test the linguistic theory on the basis of as plausible an interpretation theory as possible. In this paragraph we shall elaborate a theory of interpretation, by way of example. In the following paragraph we shall mention another model in less detail, although that model is perhaps more promising.

As we have stated, a model for the judgment of syntactic relatedness has two components — a linguistic theory and an interpretation theory. For a simple linguistic theory we take a constituent structure grammar. We must first define the concept of COHESION within this theory. This means that the intuitive notion of cohesion as used by Chomsky and others must be formally related to the linguistic theory, and in particular, with the constituent structure of the sentence. To do this we return to Chomsky's example, the sentence *he decided on the boat on the train*, where *on the train* has less cohesion with *decided* than *decided* has with *on the boat*. In Chomsky's analysis, this is expressed in the structural description in Figure 2.3. We see in the figure that the difference in

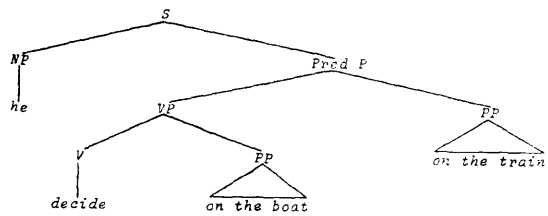


Fig. 2.3. Phrase marker for the sentence *he decided on the boat on the train* (abbreviated)

cohesion is tacitly attributed to a relation of inclusion of constituents in the phrase marker. The verb phrase *VP* is part of the predicate phrase *Pred P*, and this justifies the fact that the cohesion between the parts of the verb phrase *(decide, on the boat)* is greater than the cohesion between one of these and elements of the predicate phrase which lie outside the verb phrase *(on the train)*. A general formulation of this is as follows: the constituents of a sentence vary in cohesion, and the cohesion of a constituent is smaller than the cohesion of its parts. This is still nothing other than a faithful explicit representation of a more or less implicit linguistic notion. Without changing anything essential in the formulation, we can define the concept of cohesion mathematically as follows:

DEFINITION (Cohesion): A real-valued COHESION FUNCTION α is defined over the nodes of a phrase marker P, with the following property: if $A \rightsquigarrow B$, then $\alpha(A) < \alpha(B)$, for all nodes A, B in P, where $A \rightsquigarrow B$ means that there is a descending path in P from A to B. The COHESION of a constituent C, $\alpha(C)$, is defined as $\alpha(K)$, where K is the lowest node in P which dominates C and only C.

It follows from the definition that for every path from root to terminal element, the cohesion values of the nodes increase strictly. Consequently the cohesion of a constituent is necessarily smaller than that of its parts.

The following step is the formulation of the theory of interpretation. This theory must indicate how the strength of the

relation between two words, as judged by an informant, is connected with sentence structure. Let us imagine that we have performed such an experiment for a given sentence, and that the results of the experiment are summarized in a relatedness matrix R, in which the strength of the syntactic relation is indicated for every word pair in the sentence. Thus matrix element r_{ij} in R is the score for the degree of relatedness between words i and j. The score is obtained in one of the ways described in the preceding paragraph.[1] The interpretation theory must attempt plausibly to relate the observed r-values to the (theoretical) cohesion values α. An obvious place to begin would be to find the smallest constituent for every word pair (i, j) to which both words belong, and to compare their degree of relatedness with the cohesion value of the constituent. Let us call that constituent the SMALLEST COMMON CONSTITUENT, *SCC*, of the word pair. Each word pair in the sentence evidently has one *SCC* and only one. Thus in Figure 2.3. the smallest common constituent of the word pair *(decide, on)* is the verb phrase *decide on the boat*, with cohesion $\alpha(VP)$; the smallest common constituent of *(he, decide)* is the sentence *he decide on the boat on the train*, with cohesion $\alpha(S)$. Shall we make r_{ij} equal to the cohesion α of the *SCC*? That would not be wise, because the r-values are dependent on the experimental procedure followed. With the word sorting method, for example, the average r-value doubles when the number of subjects is doubled. Also one might expect that there is no linear relationship among the r-values given by the various methods. The word sorting method, for example, makes small differences between pairs with limited syntactic relation, while the seven-point scale yields considerable variations in r for the same pairs. The only thing which we can hope for and expect is that all methods yield the same rank order of r-values. The most careful approach, therefore, is to establish no direct relationship between r-values and α's, but only between the rank order of the r-values and the rank order of the α's. The following

[1] For the moment, we shall not discuss the effect of experimental noise. In fact the interpretation theory only regards real r-values, i.e. those corrected for errors in measurement.

interpretation axiom states that the rank order of the r-values must agree with the rank order of the α's of the smallest common constituents concerned.

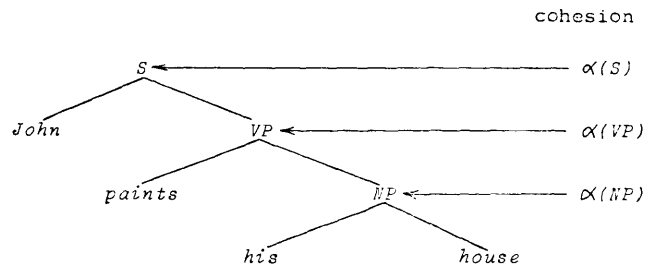
Interpretation axiom: For all words i, j, k, l in the sentence,

$$r_{ij} < r_{kl} \Leftrightarrow \alpha(SCC_{ij}) < \alpha(SCC_{kl}).$$

In this axiom, $\Leftrightarrow$ stands for "if and only if", and SCC_{ij} (SCC_{kl}) stand for the "smallest common constituent of words i and j (k and l)".

Although only inequalities are formulated in the axiom, it follows by exclusion that equal degrees of relatedness go together with equal cohesion values, and vice versa. It may be said that there is a strictly increasing relation between r and α. Equal r-values can, of course, occur by errors in measurement in the experiment, but in the theoretical error-free situation, it is a sufficient and necessary condition that the corresponding α's be equal. In particular it follows from the axiom that word pairs which have the same *SCC* will also have equal r-values.

EXAMPLE 2.1. In Figure 2.4. a theoretical phrase marker is given for the sentence *John paints his house*, together with the table of smallest



	John	*paints*	*his*	*house*
John	*John*	*S*	*S*	*S*
paints	*S*	*paints*	*VP*	*VP*
his	*S*	*VP*	*his*	*NP*
house	*S*	*VP*	*NP*	*house*

Fig. 2.4. Theoretical phrase marker for the sentence *John paints his house*, with cohesion values and table of smallest common constituents

common constituents for each word pair. On the basis of the definition of cohesion, the following inequality holds for this structure: $\alpha(S) < \alpha(VP) < (NP)$, and from this we can deduce the following equalities and inequalities by means of the interpretation axiom.

$$\begin{aligned} \textit{equalities:}\quad & r(\textit{John, paints}) = r(\textit{John, his}) = r(\textit{John, house}) \\ & r(\textit{paints, his}) = r(\textit{paints, house}) \end{aligned}$$

$$\begin{aligned} \textit{inequalities:}\quad & r(\textit{John, paints}) < \begin{cases} r(\textit{paints, his}) \\ r(\textit{paints, house}) \\ r(\textit{his, house}) \end{cases} \\ & r(\textit{John, his}) < \begin{cases} r(\textit{paints, his}) \\ r(\textit{paints, house}) \\ r(\textit{his, house}) \end{cases} \\ & r(\textit{John, house}) < \begin{cases} r(\textit{paints, his}) \\ r(\textit{paints, house}) \\ r(\textit{his, house}) \end{cases} \\ & r(\textit{paints, his}) < r(\textit{his, house}) \\ & r(\textit{paints, house}) < r(\textit{his, house}) \end{aligned}$$

These inequalities are not always independent of each other. Thus $r(\textit{John, house}) < r(\textit{his, house})$ follows from the combination of $r(\textit{John, house}) < r(\textit{paints, his})$ and $r(\textit{paints, his}) < r(\textit{his, house})$.

These predictions on equality and inequality can be tested by means of a judgment experiment. If the results of the experiment conflict with the predictions, the theoretical phrase marker or the interpretation axiom, or both, are incorrect. In judging deviations, account must be taken of errors of measurement. Absolute equalities in particular, will seldom occur. Strictly speaking, then, we must see if the observed relatedness values are equal within the tolerance of the measurement error. The measurement error can likewise change inequalities to equalities, or even to their opposites. However, we shall burden the further discussion in this section as little as possible with statistical considerations, and direct our attention to clear data from which it is possible to draw conclusions regarding our main problem, the relation between formal grammar and interpretation theory.

Given the interpretation axiom, we can study which phrase marker is most fitting for the observed relatedness values for a given sentence. If we have no particular theoretical expectation concerning the phrase marker, we can draw up a list of the predicted equalities and inequalities for every possible phrase marker in order to find the phrase marker which best agrees with the relatedness data. In doing so we should remember that different phrase markers for a single sentence do not always lead to the same number of equalities and inequalities. In general, however, we will certainly have particular theoretical expectations concerning syntactic structure, and it will be possible to limit the test to alternatives within that theoretical domain. The following is an experimental example of this.

For the sentence *the boy has lost a dollar*, only the phrase markers in Figure 2.5. are worth consideration. In an experiment described

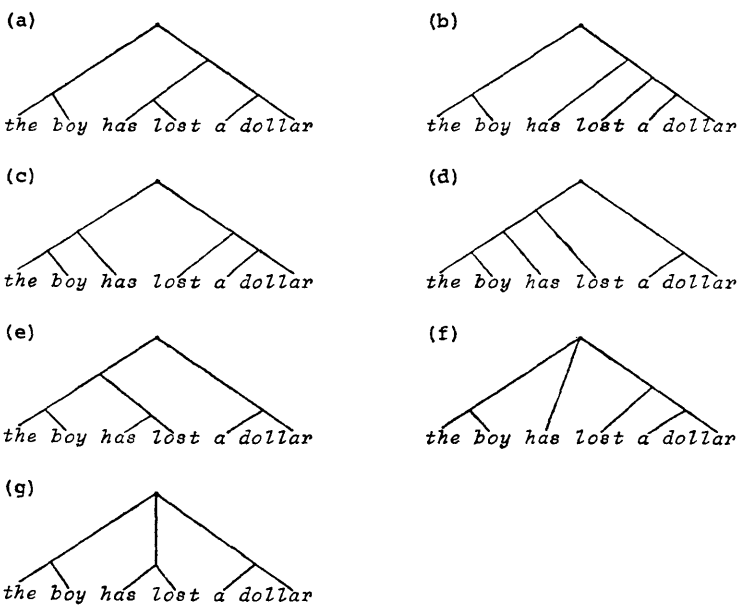


Fig. 2.5. Possible phrase markers for the sentence *the boy has lost a dollar* (node labels omitted)

elsewhere (Levelt 1967a), twenty-four native speakers of English judged this sentence by means of the method of triadic comparison. Table 2.1. shows the relatedness values obtained for the various word pairs. The value for a word pair was obtained by adding the scores for that pair in each triad and for each subject; it is expressed in a percentage.

TABLE 2.1. Relatedness Values for the Sentence *the boy has lost a dollar*

	the	*boy*	*has*	*lost*	*a*	*dollar*
the	—	99	43	29	19	16
boy		—	63	65	16	31
has			—	86	31	40
lost				—	42	70
a					—	94
dollar						—

Table 2.2. shows the number of inequalities predicted by means of the interpretation axiom for phrase markers (a) to (g), as well as the violations of these given Table 2.1. (also expressed in percentages in order to facilitate comparison of the models).

TABLE 2.2. Number of Predicted and Violated Inequalities for Phrase Markers (*a*) to (*g*) in Figure 2.5.

Phrase marker	(*a*)	(*b*)	(*c*)	(*d*)	(*e*)	(*f*)	(*g*)
Predicted Inequalities	64	67	58	67	64	46	36
Violations	9	11	7	12	8	5	0
Percentage of Violations	14	16	12	18	13	11	0

The predicted equalities are not taken into consideration here, but even without a statistical test it is quite clear that the results in this respect are in conflict with the expectations. It was predicted for all seven phrase markers that all relations between *the* or *boy* on the one hand, and *has*, *lost*, *a*, or *dollar* on the other, must be equal. Table 2.1., however, gives the very divergent values 43, 29, 19, 16, 63, 65, 16, and 31 for this. We shall return to this.

As for the inequalities, it is striking that in Table 2.2. only phrase marker (g) agrees perfectly with the data, and all other phrase markers show considerable percentages of deviation. Phrase marker (g) is the least hierarchical of the seven, and consequently the number of predictions (36) is the smallest. This is something which we meet quite regularly: the constituent model predicts less well as the hierarchy becomes more complicated. Theoretical refinements which accompany the complication of the hierarchy are not, in general, reflected in the relatedness judgments.

Sometimes the relatedness judgments for a sentence agree with no phrase marker whatsoever. This is a serious matter for, if we maintain the interpretation axiom, we must then reject the linguistic model as such, and not only some syntactic structure within the model. This gives a fundamental dimension to the investigation of cohesion. More important still than the question as to which phrase marker in a given case best agrees with the relatedness data is the logically preliminary question whether within the linguistic model (phrase structure grammar, dependency grammar, adjunct grammar, etc.) some structural description, in agreement with the relatedness data, can indeed be given. This can of course, only be investigated by seeing whether within the model at least one structural description can be given for each of various different sentences, in agreement with the relatedness data. But to answer the fundamental question, it is now less important to know precisely how that structural description looks, than to know if there is one at all. The problem is thus reduced to the following question: given a formal grammar, which properties must matrix R of relatedness values have in order to be able to find an accurate structural description within that grammatical model?

We shall at this point find that critical property for the constituent model. Let a, b, and c be three random (but different) elements (words) of a sentence s.[1] Let us imagine the three smallest common constituents for a and b, b and c, and a and c, respectively.

[1] The problem is trivial for sentences with fewer than three elements.

It is quite clear that for the three smallest common constituents, one and only one of the four hierarchical relations in Figure 2.5. must apply.

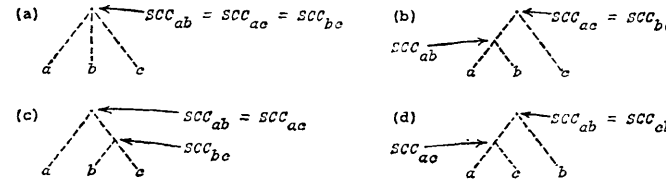


Fig. 2.5. The four possible hierarchies for three elements in a phrase marker (dotted lines indicate paths which can contain other nodes)

If (a) is the case for the phrase marker of *s*, we have the following definition of cohesion:

(1) $\alpha(SCC_{ab}) = \alpha(SCC_{ac}) = \alpha(SCC_{bc})$

If it is (b) we have the following relation:

(2) $\alpha(SCC_{ab}) > \alpha\,(SCC_{ac}) = \alpha(SCC_{bc})$

If the hierarchical relation is as in (c), we have:

(3) $\alpha(SCC_{ab}) = \alpha(SCC_{ac}) < \alpha(SCC_{bc})$

If (d) is the case, we have:

(4) $\alpha(SCC_{ab}) = \alpha(SCC_{cb}) < \alpha(SCC_{ac})$

By the interpretation axiom, it follows from (1) to (4) that one and only one of the relations (5) to (8) must hold for the observed degrees of relatedness of *a*, *b*, and *c*.

(5) $r_{ab} = r_{ac} = r_{bc}$
(6) $r_{ab} > r_{ac} = r_{bc}$
(7) $r_{ab} = r_{ac} < r_{bc}$
(8) $r_{ab} = r_{cb} < r_{ac}$

These relations (5) to (8) simply mean that r_{ab} must be equal to or greater than the smallest of the two other relations r_{ac} and r_{bc}. This

may be summarized as in (9):

(9) $r_{ab} \geqslant \min(r_{ac}, r_{bc})$

It follows from considerations of symmetry that the inequality also holds for every permutation of *a*, *b*, and *c*. (9) is called the ULTRAMETRIC INEQUALITY. In whichever way *a*, *b*, and *c* are chosen, the relatedness values in *R* must satisfy the condition of ultrametric inequality, if representation by phrase marker is to be possible. In a different context, S. C. Johnson (1967) showed that this is not only a necessary condition, but also a sufficient one: if the matrix is ultrametric, there is a tree diagram which agrees with that matrix.

To summarize, then, it holds that the formal constituent model can be tested by establishing whether relatedness matrices satisfy the condition of ultrametric inequality (9) for all triads. If this is not the case within the measurement error, when the interpretation axiom is maintained, the constituent model must be rejected as such.

Until a short time ago (cf. Loosen 1972), however, no algorithms, let alone computer programs, were available for testing the ultrametricity of a matrix. Instead, all sorts of ad hoc means were used which need not be mentioned here (cf. Levelt 1970a; 1970b). We would only point out that Johnson (1967) developed an algorithm which, given an ultrametric matrix *R*, reconstructs the corresponding tree diagram. If the matrix is not completely ultrametric, the algorithm yields two tree diagrams, both of which represent the relations as well as possible in certain (but different) respects. A rule of thumb is that the less matrix *R* satisfies the condition of ultrametric inequality, the more the two tree diagrams differ. Measures of agreement between two tree diagrams have been used as rough indications of the ultrametricity of the matrix *R* (cf. Levelt 1970b for a detailed description of the algorithm and measures of agreement). But it is not necessarily the case that one of the two tree diagrams is also the most fitting constituent structure. The solutions given by the Johnson algorithm are, in effect, solutions which, in terms of formal grammars, are as much as

possible in Chomsky normal-form (cf. Volume I, Chapter 2, section 2.3.1.), that is, they have a maximum of hierarchy. Less complicated hierarchies are avoided as much as possible by the program. This program for hierarchical clustering is therefore only of limited interest for our purposes.

We return to our original question: how adequate is the constituent model for the representation of judgments on the syntactic relations between words? In our opinion the answer is that it is inadequate. There are two reasons for this. In the first place, the condition of ultrametric inequality can be violated at will by the use of deep structure relations which cannot be expressed in the constituent structure. In the second place, independently of this, ultrametricity is systematically violated with regard to predicted equalities. We shall treat these two points successively.

The first reason can also be formulated affirmatively: that which is expressed in relatedness judgments is underlying relations among the words of a sentence. If this hypothesis is correct, we can create at will sentences whose relations no phrase marker can represent adequately. The best examples of such sentences are those from which certain words have been deleted transformationally. Take the following sentences for example: (a) *John eats apples and Peter eats pears*, which is related through the deletion of *eats* to (b) *John eats apples and Peter pears*. We should expect the model to fail in dealing with the second sentence, as we see in the following experiment (Levelt 1969; 1970c). Using the seven-point scale method, eight subjects judged all the word pair relations in sentence (a), and eight others, those of sentence (b). (In the original experiment the following equivalent Dutch sentences were used: (a) *Jan eet appels en Piet eet peren*, (b) *Jan eet appels en Piet peren*. In this case, as may be seen, the word order is the same in Dutch as it is in English.) Table 2.3. shows the total scores for the two sentences and the eight subjects (the minimum score is 8, the maximum 56).

The theoretical phrase marker in Figure 2.6.(a) gives ninety-five predictions of inequality for the relations in (a). All of them are confirmed. This constituent representation is also fitting from

TABLE 2.3. Relatedness Values for the Sentences (*a*) *John eats apples and Peter eats pears* (*Jan eet appels en Piet eet peren*) and (*b*) *John eats apples and Peter pears* (*Jan eet appels en Piet peren*).

(*a*)	*John* (*Jan*)	*eats*$_1$ (*eet*)	*apples* (*appels*)	*and* (*en*)	*Peter* (*Piet*)	*eats*$_2$ (*eet*)	*pears* (*peren*)
John (*Jan*)	—	55	48	24	36	9	10
eats$_1$ (*eet*)		—	50	16	10	32	10
apples (*appels*)			—	17	10	10	31
and (*en*)				—	19	15	16
Peter (*Piet*)					—	56	46
eats$_2$ (*eet*)						—	45
pears (*peren*)							—

(*b*)	*John* (*Jan*)	*eats* (*eet*)	*apples* (*appels*)	*and* (*en*)	*Peter* (*Piet*)	*pears* (*peren*)
John (*Jan*)	—	52	42	11	35	15
eats (*eet*)		—	49	15	45	47
apples (*appels*)			—	11	10	33
and (*en*)				—	33	24
Peter (*Piet*)					—	44
pears (*peren*)						—

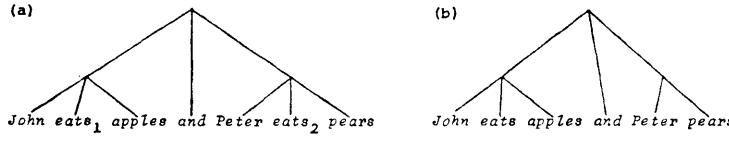


Fig. 2.6. Theoretical phrase marker for (a) *John eats apples and Peter eats pears* and (b) *John eats apples and Peter pears* (omitting category labels)

the point of view of predicted equalities,[1] but inequalities are sufficient for our argument. The corresponding constituent diagram for sentence (b), however, is considerably less in agreement with the relatedness values (Table 2.3.(b)). In fewer than half as many predicted inequalities (44), we find 4 violations, all of which are the result of the high values of *r(eats, Peter)* and *r(eats, pears)*, respectively 45 and 47. These are precisely the two values for which a deviation would be expected if the subjects do in fact judge the underlying relations, for *Peter* and *pears* are the only surface elements which are directly related to the deleted element *eats*. There is no other structure for (b) which better predicts the inequalities (except, of course, the trivial coordination of all elements, on the basis of which no prediction of inequality whatsoever is possible. That structure, however, fails largely in the case of predictions of equality.) As was our plan, we have thus found a sentence whose relatedness matrix is not ultrametric for predictable reasons. This experiment was repeated with seven other deletion sentences, and the results were similar (Clarisse, unpublished undergraduate thesis, 1969). These findings are in themselves sufficient to justify the rejection of the constituent model (if the interpretation axiom is to be maintained). Before going on to discuss difficulties with equalities, the second argument for the rejection of the model, we might wonder how the model could be modified in order to meet the objections of this first type. We know how the ultrametricity of a matrix can be violated systematically, and we should be able to try to avoid this in the model without introducing important changes in the definition of cohesion or the interpretation axiom.

We therefore take a transformational grammar with a phrase structure base, as in *Aspects*, and we define the cohesion function, not for the surface structure of a sentence, but rather for its deep structure. In all other respects the definition remains the same

[1] The only great deviations from the predictions of equality result from the values for *r(John, Peter)*, $r(eat_1, eat_2)$, and *r(apples, pears)* which are too large. It is rather certain that this is due to a nonsyntactic factor.

as in the original constituent model. The interpretation axiom also remains unchanged.

Let us see how well such a transformational model predicts the relations for our experimental sentence *John eats apples and Peter pears*. We take the diagram in Figure 2.6.(a) as the deep structure of the sentence. We are then faced with the problem of deciding on which deep pair a surface pair must be mapped. This decision must follow from the transformation rules of the grammar. Without defining these explicitly for this sentence, we can consider the relation between *John* and *eats* as a relation between *John* and $eats_1$ in the deep structure. Likewise *r(eats, apples)* can be considered as the expression of the relation between $eats_1$ and *apples*. As for the relations between *Peter*, *pears*, and *eats*, we take *eats* as a reflection of $eats_2$ in the deep structure. For *r(and, eats)* it is irrelevant whether we consider *eats* as $eats_1$ or $eats_2$, while the relation between $eats_1$ and $eats_2$ is not expressed in the judgments of the informant. Some interpretation of the data is therefore necessary before the model can be tested. Explicit rules can be established for that purpose. One of these might be that for a given surface pair the deep pair must be selected in such a way that the smallest common constituent in the deep structure is as small as possible. That is in fact the rule which we have followed here. However we shall not discuss this point further.

With the transformational model, fifty-four inequalities are predicted for sentence (b), and all of them are verified. The transformational model also gives good predictions on inequality and degree of relatedness for other sentences in which deletion transformations have been applied (Clarisse, unpublished undergraduate thesis, 1969).

How adequate is the model for sentences in the derivation of which transformations other than deletions have been applied? In another experiment (Arnolli, unpublished undergraduate thesis, 1969) we have shown that, in agreement with the model, the degrees of relatedness for a simple declarative sentence do not differ statistically from those of its passive and interrogative equivalents. In the following four Dutch sentences, the

respective word pairs did not differ significantly in degree of relatedness.

de man betaalde het geld aan een agent
'the man paid the money to a policeman'

het geld werd door de man aan een agent betaald
'the money was paid to a policeman by the man'

betaalde de man het geld aan een agent?
'did the man pay the money to a policeman?'

werd het geld door de man aan een agent betaald?
'was the money paid to a policeman by the man?'

As far as the corresponding words are concerned, these sentences do indeed have the same deep structure in the *Aspects* model. We may conclude from these and other experiments that the transformational formulation of the cohesion model agrees much better with the relatedness data, with regard to inequalities, than with the original constituent model. If such results are maintained, the model can be used for practical purposes, namely, for finding the most appropriate deep structure. Relatedness judgments allow one in a sense to bypass the transformational superstructure of a sentence and to arrive directly at the underlying relations. It is important to notice, however, that the ultrametric inequality is neither necessary nor sufficient for testing the transformational model as such. We have already seen that the matrix for *John eats apples and Peter pears* is not ultrametric, while the transformational model is in complete agreement with the relatedness data as far as inequalities are concerned. The important point is the transformation relation between the surface and deep structures. Ultrametric inequality retains its critical value only for sentences with the same deep structure as surface structure (abstraction made of nonessential changes). Such sentences are called KERNEL SENTENCES. Examples are *John has lost a dollar* and *the man paid the money to a policeman.* Whether or not a sentence is a kernel sentence will, of course, depend closely on the transformational component. If, however,

the relatedness matrix for kernel sentences in a given model is not ultrametric within the measurement error, the transformational model must be rejected. Such negative information does not exist at the moment, at least as far as ultrametric inequality predictions are concerned. But the case is different for equality predictions. This brings us to the second argument against the constituent model; the argument will also hold for the transformational extension of the constituent model.

Ultrametricity can be systematically violated in matter of predicted equalities. The introduction of an endocentric construction into the sentence will be sufficient to cause this. We shall limit the discussion to constructions of the type article+noun (*the child, a policeman*, etc.). Whether we test the parsing of the surface structure or that of the deep structure, article and noun in the cohesion determinant phrase marker will always be connected at a relatively low level in the hierarchy. Only at a higher level does the noun phrase as a whole come to be related to the other elements of the sentence. But this means that for every third element x in the sentence the smallest common constituent of article and x is the same as that of noun and x. It follows from the interpretation axiom that with the same degree of cohesion the same relatedness value should be expected for these pairs. For the sentence *the child cried for help*, for example, the theory predicts the following equalities:

$$r(\textit{the}, \textit{cried}) = r(\textit{child}, \textit{cried})$$
$$r(\textit{the}, \textit{for}) = r(\textit{child}, \textit{for})$$
$$r(\textit{the}, \textit{help}) = r(\textit{child}, \textit{help})$$

This holds, no matter what the sentence structure is, provided that the smallest common constituent of *the* and *child* includes no other smallest common constituent. Any theory which allows the contrary is a priori in disagreement with current relatedness data, for the relation between the article and its corresponding noun is always stronger than any other relation in an experimental matrix. But the reader can clearly see that the predicted equalities conflict with intuition; one feels that the relations with the article are

systematically weaker than those with the noun, and this is indeed what is regularly found in judgment experiments. For the dozens of sentences with article/noun pairs which we have investigated, we have always found, without exception, that the average strength of the relation between the noun and the other words of the sentence is considerably greater than that between the article and the other words. An example of this is the following. The Dutch sentence *Meester geeft de doos aan Jetty of aan Thea* ('Teacher gives the box to Jetty or to Thea') was presented to eight subjects, who judged the word pair relations on a seven-point scale. The relatedness values (total scores) for *de* 'the' and *doos* 'box' are given in Table 2.4.

TABLE 2.4. Experimental relatedness values for the relations between *de* 'the' and *doos* 'box' on the one hand, and on the other, the remaining words in the sentence *Meester geeft de doos aan Jetty of aan Thea* ('Teacher gives the box to Jetty or to Thea')

	Meester 'Teacher'	*geeft* 'gives'	*aan*$_1$ 'to'	*Jetty* 'Jetty'	*of* 'or'	*aan*$_2$ 'to'	*Thea* 'Thea'
de 'the'	10	11	9	9	9	10	9
doos 'box'	38	45	20	38	9	22	35

r (*de*, *doos*) = 55

The relations with *doos* 'box' are systematically stronger than those with *de* 'the'. Only the minimal relation with *of* 'or' shows the predicted equality. This result is also characteristic for the strength of the effect: the relations with the article are always close to the absolute minimum score (the minimum score is 8 for eight subjects), while those with the noun tend to cluster around the middle of the scale. It is possible to produce systematic deviations from ultrametricity by introducing article/noun constructions into the test sentence. In general, relations with the head of an endocentric construction are systematically stronger than those with the modifiers.

We must add, however, that not all judgment techniques are equally suited for demonstrating this systematic deviation from ultrametric predictions of equality. Martin (1970) investigated (to use our own terminology) the Chomsky normal-form of the pure surface constituent model. Using the Johnson algorithm, he found good hierarchical solutions for two types of sentences, examples of which are *parents were assisting the advanced teenage pupils* and *children who attend regularly appreciate lessons greatly.* The dominant solution for the first sentence type is (in labeled bracketing notation):

$((N_1(Aux\ V))(D(Adj_1(Adj_2\ N_2))))$,

or, for example,

((parents (where assisting)) (the (advanced (teenage pupils)))).

The solution found for the second sentence type is:

$((N_1\ ((Wh\ V_1)\ Adv_1))((V_2\ N_2)\ Adv_2))$,

or, for example,

((children ((who attend) regularly)) ((appreciate lessons) greatly)).

In the first sentence type the main verb is drawn to the subject, as is the case in Uhlenbeck's analysis and in many of our own experiments. This is not the case for the second sentence type. Later we shall see that the degree of relatedness between the main verb and the subject or the object decreases with the length of the subject or the object, and the highest degree of relatedness is obtained by the pronominalization of that subject or object. But we mention Martin's findings here particularly because of the fact that he did not obtain the same great differences between the degrees of relatedness of articles and those of nouns as we regularly found, and which are so much in conflict with the constituent model. The reason for this, however, is quite clear. With minor modifications, Martin used the word-sorting method in his experiments. That method is not suited for demonstrating these systematic inequali-

ties, and that for an obvious reason. As we have seen, the relation between article and noun is among the strongest in the sentence. This is also the case in Martin's experiments: in four out of five cases in the experiment, the subject grouped the article and its corresponding noun together. If there is a third element X in weak relation to the article and in strong relation to the noun, it is impossible to distinguish the relation of the article to X from that of the noun to X when the informant groups the article and the noun together. If X is placed in the same stack as the article and the noun, the scores of both D, X and N, X (article to X, and noun to X respectively) are equal to 1, and if X is placed in a different stack, both scores are equal to 0. With the word sorting method, then, strongly related words cannot be distinguished with respect to a third element; both relations are always equal. This contributes considerably to the satisfaction of the condition of ultrametric inequality, according to which the two weakest relations among three elements must be equal. In our own experiments with the word sorting method, we also found a high frequency of ultrametricity in the experimental data. The word sorting method camouflages the characteristic deviation peculiar to the constituent model, and therefore should not be used for testing that model. Moreover, Martin does not mention the other source of systematic deviations, the deep structure relations.

We may then conclude that the transformational extension of the constituent model must also be rejected when the interpretation axiom is maintained. The model is not capable of accounting for either the strong relation between the article and the corresponding noun, or the weak relation between the same article and the other words in the sentence. Yet this result is not surprising to the intuition. It shows that the relation between article and noun is asymmetric; the article is dependent on the noun, and the noun is the head of the noun phrase. A phrase structure grammar or constituent model is not suited for the representation of such dependencies (cf. Volume II, section 4.1.). An obvious alternative is to use a dependency grammar as a linguistic theory, and to adapt the formulation of the interpretation axiom accordingly.

2.4.3. *A Dependency Model for Relatedness Judgments*[1]

In the preceding paragraph we found that relatedness judgments are more a reflection of the relations in the deep structure than of those in the surface structure. We suppose in the present paragraph that the dependency model must be a transformational model. Here, too, the theory has two aspects: a linguistic definition and an interpretation axiom. In a dependency grammar the equivalent of cohesion consists of the two notions of dependency and connectedness. We define a dependency function over the nodes of a dependency diagram (for the formal structure of a dependency grammar, see Volume II, Chapter 4, section 4.5.).

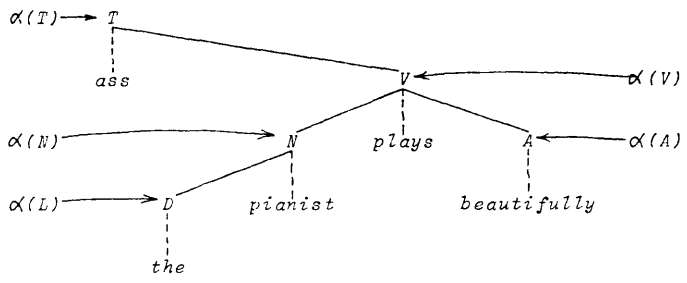
DEFINITION (Dependency). A real-valued DEPENDENCY function α is defined over the nodes of a dependency diagram D, with the property that if $A \rightsquigarrow B$, then $\alpha(A) < \alpha(B)$ for all nodes A, B in D, where $A \rightsquigarrow B$ means that B is directly dependent on A.

The nodes of a dependency diagram thus have values expressed as real numbers; these values increase in all descending paths of the diagram. The head (the start symbol of the grammar) has the smallest degree of dependency.

If we suppose, by convention, that every element in a dependency diagram is dependent on itself, then for every pair of elements there is at least one element on which both are dependent. The FIRST COMMON HEAD *FCH* of two elements in a dependency diagram is the element with the highest dependency value α, on which both elements are dependent. This may be illustrated by the following example.

EXAMPLE 2.2. Figure 2.7. gives a dependency diagram for the underlying structure of the sentence *the pianist plays beautifully*, and an *FCH* table for all pairs of elements in the diagram. *N* and *A*, for example, are both dependent on *V*, but also indirectly on *T*. The first common head of *N* and *A* is the element with the highest

[1] The suggestion of a dependency model as well as other considerations in this paragraph originated in the work of Mr. E. Schils, which will be reported in a separate publication.



FCH	T	D	N	V	A
T	T	T	T	T	T
D	T	L	N	V	V
N	T	N	N	V	V
V	T	V	V	V	V
A	T	V	V	V	A

Fig. 2.7. Hypothetical dependency diagram for the sentence *the pianist plays beautifully*, with degrees of dependency and *FCH* table

dependency value. It follows from the definition of the dependency function that V has a higher dependency value than T, and V is therefore the first common head of N and A. Or consider nodes D and N. They are both dependent on V, but also on N and T. Because $\alpha(N) > \alpha(V) > \alpha(T)$, $FCH_{DN} = N$, as may be seen in the *FCH* table.

We now define the notion of connectedness negatively as follows:

DEFINITION (Disconnectedness). The DEGREE OF DISCONNECTEDNESS of two elements A and B, $\delta(A, B)$ in a dependency diagram is defined as follows: $\delta(A, B) = [\alpha(A) - \alpha(FCH_{AB})] + [\alpha(B) - \alpha(FCH_{AB})] = \alpha(A) + \alpha(B) - 2\alpha(FCH_{AB})$.

Two situations can occur here. The first is that in which FCH_{AB} is different from A and B themselves. In Figure 2.7., that is the case for D and A: $FCH_{DA} = V$ and $\delta(D, A) = \alpha(L) - \alpha(V) + \alpha(A) - \alpha(V)$. This is the sum of the two reductions in dependency which

occur when we pass from the two elements to V. The other case is that in which one of the elements is the *FCH* of both. This holds, for example, for D and V in Figure 2.7., where V is the first common head of D and V. The disconnectedness is thus $\delta(D, V) = [\alpha(D) - \alpha(V)] + [\alpha(V) - \alpha(V)] = \alpha(D) - \alpha(V)$, which is the difference in the degree of dependency of D and V. In both cases δ is a non-negative real number.[1]

We must now give the interpretation theory which relates experimentally measured degrees of relatedness to this linguistic theory of dependency and connectedness.

Interpretation Axiom. $r_{ij} < r_{kl} \Leftrightarrow \delta_{ij} > \delta_{kl}$, for all words i, j, k, l, in a sentence.

It should be noted that the degree of connectedness of two words is considered to be equal to that of the syntactic category which dominates them directly (it will be remembered that lexical insertion does not take place by dependency rules in a dependency grammar; cf. Volume II, Chapter 4, section 4.5.).

The degree of relatedness of two words is therefore greater to the extent that their connectedness in the dependency diagram is stronger, and vice versa.

Given the interpretation axiom, it is not difficult to reconstruct the dependency diagram which corresponds to a given error-free relatedness matrix R. More precisely, one can reconstruct a connectedness diagram (or GRAPH) which corresponds to the dependency diagram. This calls for some explanation. The interpretation axiom is relatively weak; it does not relate the degrees of relatedness directly to the degrees of dependency in the diagram, but only indirectly, by way of the connectedness of the elements. The axiom says nothing about the direction of the dependency which is at the basis of a given connectedness. Thus we cannot tell from $r(D, N)$ whether $\alpha(D) > \alpha(N)$ or $\alpha(N) > \alpha(D)$, nor do

[1] It is not difficult to show that the degree of disconnectedness δ is a distance metric: (1) $\delta(A, B) = \delta(B, A)$ for every A and B; (2) $\delta(A, A) = 0$ for every A; and (3) $\delta(A, C) \leqslant \delta(A, B) + \delta(B, C)$ for every A, B and C in the dependency diagram (triangular inequality).

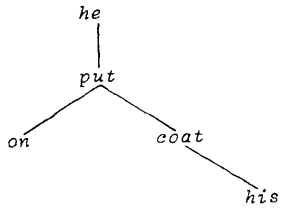
we know that *D* and *N* are directly connected. We can take either *D* or *N* as the head of the dependency relation. Only linguistic considerations can be decisive here, and not the relatedness data. The latter only tell us how the dependency diagram is connected, that is, they show the elements between which relations of direct dependency exist. If we abstract from the transformational character of the model, or limit ourselves to kernel sentences, it holds that for a sentence with *n* words there are *n* and only *n* dependency diagrams which are isomorphous with it as far as connectedness is concerned. Each of the *n* words can, in effect, be taken as the start symbol in the dependency diagram, and the other dependencies will follow naturally from this. This is illustrated in Figure 2.8.; in it the hypothetical connectedness graph for the sentence *he put his coat on* is given, together with the five dependency diagrams which correspond to it.[1] On close inspection, we see that the interpretation axiom allows us only to find the most accurate connectedness graph for a given relatedness matrix; it does not decide among the various dependency diagrams. But the axiom is nevertheless strong enough, in our opinion: given the root (or start symbol), it determines the form of the dependency diagram. Every linguistic dependency theory will indicate the element which is to be taken as the start symbol. With the interpretation axiom, we suppose that that choice cannot be justified empirically, and this is decidedly a realistic point of view.

In respresenting experimental data, we will take the main verb as the root of dependency diagrams. The graph of Figure 2.8. will thus have the form of diagram (b).

How do we find the dependency diagram for an error-free relatedness matrix *R*? Just as was the case for the Johnson algorithm for hierarchical clustering in the preceding paragraph, we shall refrain in this section from deducing and justifying the dependency algorithm in any detail. We shall illustrate the procedure with an example. Figure 2.9. gives a hypothetical dependency

[1] For the generation of diagrams with split constituent, such as *(c)* and *(d)*, more complete dependency rules are necessary than those given in Volume II, Chapter 4, section 4.5.

connectedness graph



dependency diagrams

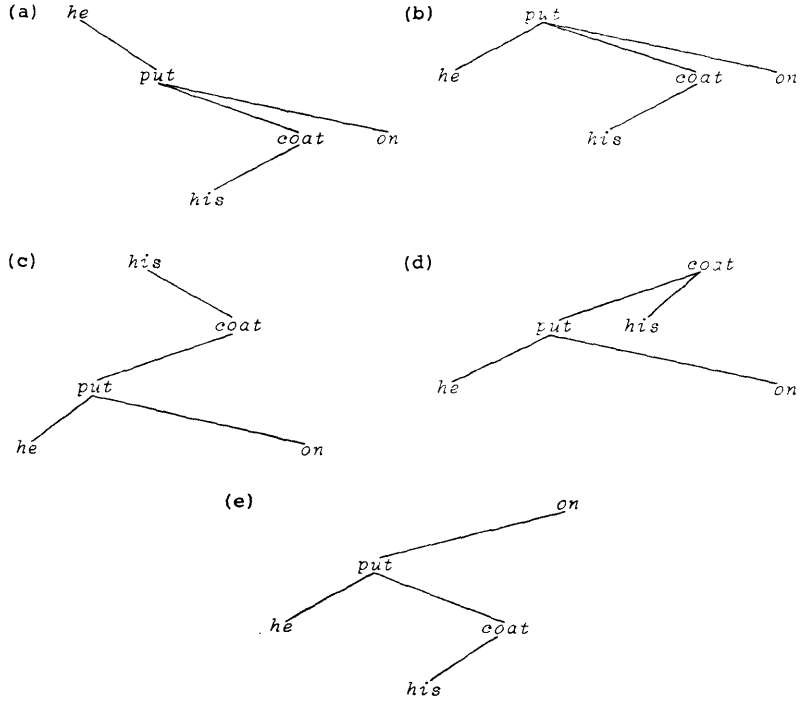
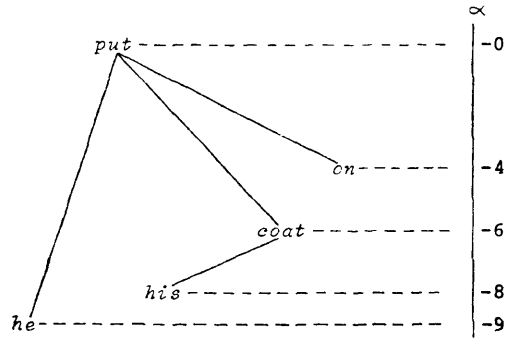


Fig. 2.8. Connectedness graph and corresponding dependency diagrams for the sentence *he put his coat on*

diagram for the sentence *he put his coat on* with the corresponding α-values (*a*) and a δ table with the degrees of disconnectedness for all the word pairs in the sentence (*b*). These degrees of

disconnectedness, as the reader can verify, can easily be deduced from the diagram, thanks to the definition of disconnectedness given above. The interpretation axiom states that the rank order of the degrees of relatedness must be the opposite of that of the δ's. The rank numbers for r, from weak (1) to strong (10), are given in a separate table (c) in the figure. Measured r-values will have the

(a) Theoretical diagram with dependency values



(b) δ-matrix

δ	*he*	*put*	*his*	*coat*	*on*
he	—	9	17	15	13
put		—	8	6	4
his			—	2	12
coat				—	10
on					—

(c) Relatedness ranks from weak (1) to strong (10)

r	*he*	*put*	*his*	*coat*	*on*
he	—	6	1	2	3
put		—	7	8	9
his			—	10	4
coat				—	5
on					—

(*d*) Steps in the construction of the graph

rank number	word pair	result
10	*his, coat*	*his—coat*
9	*put, on*	*his—coat* *put—on*
8	*put, coat*	*his—coat* / *put—on*
7	*put, his* (the path is already present)	*his—coat* / *put—on*
6	*he, put*	*his—coat* / *put—on* \ *he*

(*e*) Dependency diagram

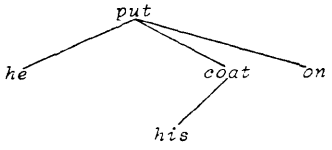


Fig. 2.9. Algorithm for the reconstruction of a dependency diagram from relatedness matrix.

same rank order if the model (disregarding error of measurement) is correct. The problem is now to reconstruct the form of the graph from the experimentally observed order of *r*-values; thence, when the root is selected, we must reconstruct the dependency diagram. The method is as follows:

(a) select the word pair with the highest ranking degree of relatedness;
(b) connect the words, if there is not yet a path between them;
(c) lower the rank by 1, and select the corresponding word pair;
(d) repeat (b) and (c) until the graph is connected, that is, until there is a path between each element and every other element.[1]

[1] It can occur that two or more word pairs are tied for the same rank. In that case, any of the tied pairs may be selected in (a); when (b) has been per-

These steps are shown in Figure 2.9.(*d*), and the resulting dependency diagram, with the main verb as the root, is given in Figure 2.9.(*e*).

Although the construction of the graph in Figure 2.9. is based only on the five strongest relations, the graph is in agreement with the entire matrix. We shall now check this for a few properties of the matrix.

It is not difficult to see that, on the basis of the definitions of dependency and connectedness, the following should be the case: If two elements B and C lie in the path between two other elements A and D, then the connectedness between B and C is greater than that between A and D. By the interpretation axiom, it follows from this that $r(B, C) > r(A, D)$. This holds likewise when the two pairs have one element in common: with a path $A - B - C$ we find $\delta(A, B) < \delta(A, C)$, and therefore $r(A, B) > r(A, C)$. Thus if Figure 2.9.(*c*) correctly represents the relatedness matrix (*c*), then according to the rule mentioned above, the following inequalities hold:

$$r(he, put) > \begin{cases} r(he, his) \\ r(he, coat) \\ r(he, on) \end{cases} \qquad r(put, his) > \begin{cases} r(he, his) \\ r(his, on) \end{cases}$$

$$r(put, coat) > \begin{cases} r(he, coat) \\ r(put, his) \\ r(he, his) \\ r(his, on) \\ r(on, coat) \end{cases} \qquad r(his, coat) > \begin{cases} r(his, put) \\ r(his, he) \\ r(his, on) \end{cases}$$

$$r(put, on) > \begin{cases} r(he, on) \\ r(his, on) \\ r(coat, on) \end{cases}$$

$$r(he, coat) > r(he, his)$$

$$r(coat, on) > r(he, his)$$

Not all of these inequalities are independent; the right hand member of one can be the left hand member of another. In

---

formed for that word pair, we return to (a) and continue the operation until all the equal ranking pairs are connected. Only when this has been done can we proceed to (c).

fact, only nine of the inequalities in the list are independent; the other nine can be deduced from these. One can easily verify that the matrix in Figure 2.9.(*c*) is correct for all eighteen inequalities predicted in the dependency diagram (*e*).

There are probably more conditions which a relatedness matrix must fulfill in order to correspond to at least one graph or dependency diagram. The fundamental question here, as was the case for the constituent model, concerns the characteristics of the matrix which are necessary and sufficient for the matrix to correspond to at least one graph. But, unlike the case for the hierarchical model, no solution for this has yet been found, as far as we know, though attention has been paid to this question within topology (cf. Goodman 1966).

This problem remains unsolved, and the experimental research on the dependency model is yet in its first stages. Therefore we shall only mention a few preliminary findings which appear to be promising with regard to the model, and a number of points on which problems might be expected.

Within the context of the investigation of another problem, we examined the way in which degrees of relatedness behave under pronominalization (cf. Visser-Bijkerk, unpublished undergraduate thesis, 1969). Every reasonable linguistic theory recognizes that *the boy gave the ice cream to a child* and *he gave the ice cream to a child* have the same structure, with the exception of the substitution of *he* for *the boy*. Likewise, the substitution of *it* for *the ice cream*, or of *him* for *a child*, will also leave the structure unchanged. Three noun phrases can thus be pronominalized in this sentence. Alternate pronominalization of one, two, or all three of those noun phrases will produce seven new sentences, beside the original complete sentence. The eight sentences (including the original) will all have the same structure, with the exception of the pronominalizations. We examined this in the context of the constituent model as well as within that of the dependency model. In the experiment this sentence (in Dutch) was used together with seven others, all with corresponding syntactic structure. The eight sentences were the following:

de jongen gaf het ijsje aan een kind
'the boy gave the ice cream to a child'

de man betaalt het geld aan een agent
'the man pays the money to a policeman'

de miljonair schonk het schilderij aan een pastoor
'the millionaire presented the painting to a priest'

de directeur stuurde het honorarium aan een advocaat
'the director sent the fee to a lawyer'

de meester leende het boek aan een leerling
'the teacher lent the book to a pupil'

de slager overhandigde het vlees aan een klant
'the butcher handed the meat to a customer'

de eigenaar vermaakte het huis aan een invalide
'the owner bequeathed the house to an invalid'

de grossier leverde het hout aan een timmerman
'the wholesaler delivered the wood to a carpenter'

With all the pronominalizations, this gave sixty-four experimental sentences. Each subject was presented with all the forms of pronominalization, and asked to judge them on seven-point scales. Each form was derived from a different sentence content, and the sixty-four sentences were distributed in such a way to eight subjects that each sentence was judged only once. We shall limit our discussion to the results of each form of pronominalization, that is, the totals for the various forms over subject and sentence content; therefore we shall indicate the various words with their category symbols. The sentences on which no pronominalization has been carried out have the form $D_1N_1VD_2N_2$ *to* D_3N_3; those in which the first noun phrase has been pronominalized have the form *he* VD_2N_2 *to* D_3N_3, and so forth. Note that the three articles are all different in Dutch *(de, het, een)*, and thus no confusion was possible.

Analysis showed that the data obtained seriously conflicted with the constituent model. The principal deviation had to do with the

predicted equalities for the relations with article and noun. With one exception, the relations with the noun are stronger than those with the corresponding article, quite in agreement with that which was discussed in the preceding paragraph. The single exception is the relation between two articles, which is stronger than that between an article and a noun which does not correspond to it. We shall return to this later. There were also great deviations from the constituent model concerning inequalities. The ultrametricity of the matrices was limited, and alternative phrase markers were always found for the various forms of pronominalization. Only one general tendency could be found in this: when a noun phrase is pronominalized, cohesion with the verb increases. The average degree of relatedness, for example, between N_1 and V (e.g. *boy, gave*) is 5.6 (on a scale running from 1 to 7); for *he* and V, it increases to 6.3. For V and N_2 (e.g. *gave, ice cream*) we find an average scale value of 4.8, but for V and *it*, the value increases to 5.3. The effect is stronger still for N_3; $r(V, N_3)$ has an average of 3.2, while $r(V, him)$ has an average scale value of 4.8. It is not surprising, therefore, that in the best fitting phrase marker the main verb is joined to either the subject, the object, or the indirect object, depending on the pronominalization. In other words, one can, at will, elicit the Uhlenbeck or the Chomsky structure (cf. section 2.4.2. of this chapter).

We performed a graph analysis on the same data by means of the algorithm given in Figure 2.9. As the data were not error-free, we were not certain of finding the most suitable dependency diagram. Therefore, for every reconstructed diagram, we examined the degree to which the relatedness data agreed with the inequalities which could be predicted on the basis of that diagram. By making minor variations in the diagrams, we attempted to find better agreement with the data. With a single exception, however, this never led to improved fit. Figure 2.10. shows the most satisfactory solutions, obtained in this way, for the eight forms of pronominalization; the main verbs were used as the roots. Each diagram is accompanied by the percentage of deviations from the inequalities predicted by it.

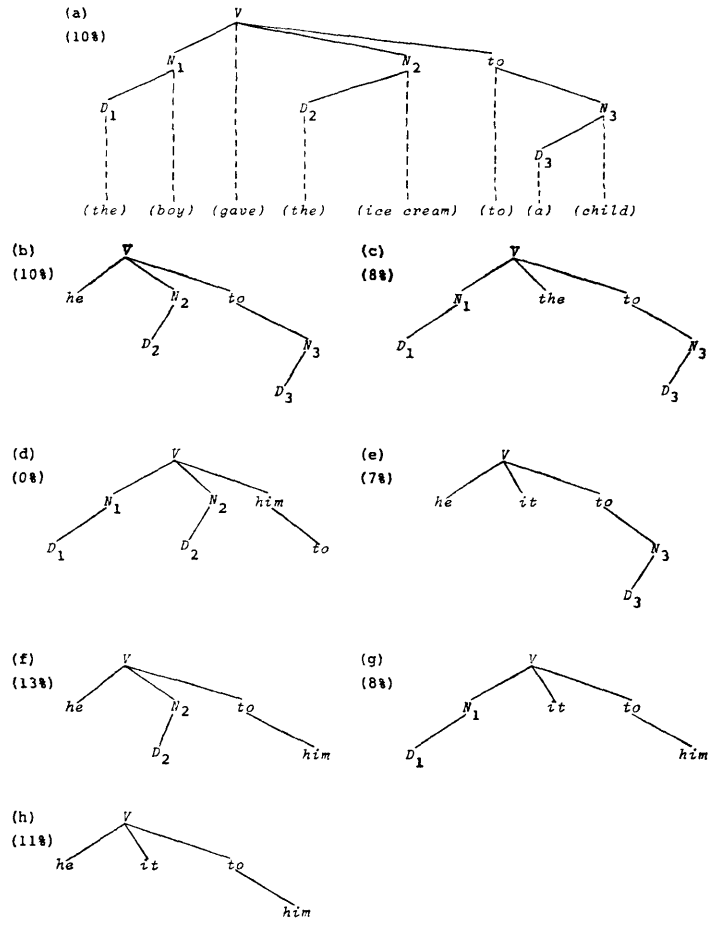


Fig. 2.10. Dependency diagrams for eight forms of pronominalization (percentages of violations are given between parentheses)

The figure shows that the syntactic structure remains constant throughout the eight forms. There is one small exception, however. In (d) *him* and *to* exchange places; otherwise the violations would increase to 9 percent. With this exception, the structures are those which would be expected on linguistic grounds (cf. Volume II, Chapter 4, section 4.5.). The structure, consisting of the verb and

its three cases, agentive, objective, and dative, remains constant under pronominalization. If this finding is confirmed by further research, it will be possible to conclude that syntactic relatedness is a concept which can be described in terms of case dependency structure.

The violations reported in Figure 2.10. vary from 0 to 13 percent; these percentages are high enough to consider whether any system can be found in them. It appears that most violations can be explained by a word class effect: degrees of relatedness between words of the same class show a certain inflation. Too strong a relation between artilces, for example, accounts for 25 percent of the violations, and 37 percent of the violations are due to the same effect with nouns or pronouns. Both percentages are too high to be written off to chance. This effect is not syntactic in nature, and no place has yet been made for it in the model.

The experiment reported here by way of example is no proof of the correctness of the dependency model. Further experimentation will certainly lead to modifications and additions. The purpose of this section was to show that to an explicitly formulated grammar an equally explicitly formulated interpretation theory could be added, making it possible to investigate the descriptive adequacy of the linguistic theory. We found that a transformational grammar with a phrase structure grammar as its base is not descriptively adequate in a number of regards, and that a dependency grammar as base avoids many of the difficulties. In both cases, the linguist can set these findings aside by rejecting the interpretation theory. To do so, however, will oblige him to find a better interpretation theory, and it is by no means excluded that this is possible. In that case, the linguist will finally have to attend to a matter which he usually neglects, namely, the theory of the relationship between formal linguistic model and concrete linguistic data.

2.5. CONCEPTUAL FACTORS IN THE JUDGMENT PROCESS

An axiomatic model, however adequate it may be, tells us nothing about the process of judgment itself. It deals only with the output

of that process, and not with the way in which that output is produced. This holds for the processes both of judgments on grammaticality and of judgments on syntactic relatedness. It is on introspective grounds alone that we presume conceptual factors of various sorts to play a role in such judgments. The mental representations which we have while the linguistic judgment is formed and the deductions which we make from them are not negligible epiphenomena; they are part of the essence itself of such behavior. Hill (1961) mentions that three of his subjects found the following sentence ungrammatical: *I never heard a green horse smoke a dozen oranges.* But "two changed their votes when it was pointed out that the sentence was strictly true". Conceptual factors are investigated in this framework only in order that they might be eliminated as troublesome variables. No work has yet been done on the problem of how semantic and other conceptual processes in fact determine intuitive linguistic judgments.

NOTE TO SECTION 2.1.

The following table shows the authors' original grammaticality judgments on sentences (1) to (14) and the number of the twenty-four linguists who judged the sentences as ungrammatical.

Sentence	*Author's Judgment*	*Number of Linguists Judging Sentence as Ungrammatical*
(1)	Ungrammatical	9
(2)	Ungrammatical	0
(3)	Ungrammatical	0
(4)	Grammatical	4
(5)	Ungrammatical	7
(6)	Ungrammatical	3
(7)	Grammatical	16
(8)	Grammatical	11
(9)	Ungrammatical	5
(10)	Grammatical	0

(11)	Ungrammatical	1
(12)	Ungrammatical	8
(13)	Grammatical	12
(14)	Ungrammatical	5

The average number of "ungrammatical" judgments was thus 4.2 for the sentences marked ungrammatical by the authors, and 8.6 for those marked grammatical by the authors.

Sentences (17) to (20) were all marked grammatical.

3

GRAMMARS IN MODELS OF THE LANGUAGE USER

The central problem in psycholinguistics is the explanation of primary language usage, that is, speaking and understanding. Language is the means *par excellence* of human communication. By the spoken word (and in a derived form, by the written word), we express our intentions, thoughts, feelings, questions; by it also we are able to decipher the intentions of others. Verbal communication is possible on every subject imaginable. It is a general and considerably flexible medium, supple enough to fit not only the direct topic of the conversation, but also the suppositions shared by speaker and hearer, the special and social relations which exist between them, the perceptions which they share during the conversation, verbal communication which has preceded the conversation, the supposed intentions of the other, and much more still. One can only imagine the number of variables which would have to be taken into account for a complete analysis of the following conversation:

She: Shall I simply go alone to the PTA meeting?
He: You are only reminding me that I dread it.

Contemplation of such examples is enough to drive a psycholinguist to distraction. The flexibility of the use of language forces us to study so many variables that every concrete investigation already suffers from the odium of futility before it even begins. On the other hand, given the importance of the growing insight into human verbal communication, from the points of view of both science and application, the psycholinguist has little alternative

but to assume that odium and try little by little to acquire insight into this complex problem.

The only reasonable way of doing this is to isolate a number of variables for close investigation, and to keep all other variables constant. Although reasonable, this approach is not without repercussions for the examination of so complicated a phenomenon. There is the danger that the investigator will lose sight of the whole and will systematically undererstimate or even deny the importance of other variables. The research situation is almost inevitably accompanied by a number of presuppositions which tend to blind one to the limitations of an experiment. In the history of psycholinguistics, unfortunately, this has been more the rule than the exception.

For this reason, even an incomplete general model can act as a corrective to more specific investigation. Not every model, of course, is suited to this purpose. An important requirement for such a model is that it be formulated in the same language as the more specific models on which concrete research is based. Communication must remain possible between general insights and experimental findings. The only general model which could fulfill this requirement at the moment is a computer simulation model which takes the human being as an information-processing system. The main reason for this is the following. In experimental research an effort is made to investigate each procedure in such detail that every step in it can be described explicitly. If this is successful, a Turing machine can be found which can simulate the process (cf. Volume I, Chapter 7); in other words, a computer can imitate the procedure in principle. To know this, one need not go so far as to program a computer for the task, and in fact this is done only rarely. The point is, however, that on the basis of this notion of simulation, and with the theory of Turing machines in the background, the language of artificial intelligence is used more and more for the description of psychological processes. At the moment it is quite common to see cognitive processes represented in the form of flow diagrams, that is, *as if* they were to be programmed. This way of modelling has a respectable theoretical basis: each

process which can be formulated explicitly can be represented in this way. The same theoretical basis provides guarantees for the possibility of simulating verbal communication as a whole. Finally, there are universal Turing machines on which every imaginable procedure can be performed. Anything which can be formulated explicitly can therefore be incorporated in principle into such a general model. Here too, it is not necessary to program a computer for such a model, although it would be informative to attempt to do so, as we shall see in the course of this chapter. One can be satisfied to begin with general flow diagrams in which important aspects of the language usage model are represented as empty black boxes; but even in that case the general theory is put in the same language as that used by the experimenter who deals with such matters.

The subject of this chapter is the place of formal grammars in models of the language user. No effort will be made to present a complete survey of the experimental work already done in this field, nor will any effort be made to treat information-producing models exhaustively. We shall only consider the question of how the theory of formal grammars has contributed to both. The nature of that contribution is determined principally by the extent to which formal grammar itself is used as a general model for speaking and understanding. To that extent, psycholinguistic models can be subdivided into ISOMORPHISTIC, SEMI-ISOMORPHISTIC, and NON-ISOMORPHISTIC models.

3.1. ISOMORPHISTIC, SEMI-ISOMORPHISTIC, AND NON-ISOMORPHISTIC MODELS

Opinions differ on the degree of directness with which the grammar is used in speaking and understanding. Some researchers claim that the grammar has a central place in the model of hearer and speaker, while others give it only a peripheral function. The former implicitly or explicitly suppose that the hearer, in understanding a sentence, either literally runs through the list of rules of the grammar by which the sentence is generated, or performs a

series of operations, each of which corresponds to a rule in a one-to-one fashion. They thus presuppose an ISOMORPHISM between linguistic rules and psychological operations. An implication of this point of view is that a given partitioning in the linguistic grammar must correspond to a parallel partitioning in the psychological process. As the input and output of every linguistic rule is copied psychologically, this must also hold for groups of rules. If, for example, the formal model is a transformational grammar, the distinction between the base grammar and the transformational component will be reflected in a parallel segmentation of psychological processes; the deep structures would be the output of one process, for example, and the input of another.

Other investigators reject rule-for-rule isomorphism (MICRO-ISOMORPHISM), but maintain the general agreement between the partitioning of the grammar and that of the psychological mechanism. For them, components of the grammar correspond to relatively independent processes in the language user (MACRO-ISOMORPHISM). In their details, these show little structural agreement with the rules of the grammar, but input and output remain linguistically defined entities, such as surface structure and deep structure. This school thus omits isomorphism on the microlevel, but retains it for the major steps. We shall refer to this kind of model with the term SEMI-ISOMORPHISTIC.

Finally, this whole approach may be dropped, and one can attempt to construct a model of the language user which is not patterned after the rules or components of the grammar. We can call such models non-isomorphistic; in them psychological theory is not patterned after the linguistic grammar. The role of grammars in such models is restricted to that of a minor subcomponent, or to the formal representation of nonlinguistic aspects in the model, or to both. The only non-isomorphistic models at present in which formal grammars play such a role are those which were developed within the framework of research on artificial intelligence; they are known as SEMANTIC MODELS. These are usually relatively general models of the sort to which we referred earlier. The term "semantic" might be somewhat confusing here, since to at least one

point of view semantics is a subdivision of linguistics, while most of these models are characterized by assumptions which are conceptual rather than linguistic in nature. Apart from their intrinsic significance, these non-isomorphistic theories, by their general character, have cast new light on the relations among linguistics, psycholinguistics, and the theory of formal languages. It is primarily for that reason that we have chosen to discuss them as well in this chapter. They descend, however, from a completely different tradition of research than isomorphistic and semi-isomorphistic models.

Let us first return to isomorphistic and semi-isomorphistic theories. The distinction between these two approaches is a historical one, and it appears in various forms. Isomorphistic theories enjoyed priority in history, almost by necessity. The notion of "a psychology of grammar" came into being precisely when psychologists were once again becoming aware of the importance of linguistic variables in verbal communication. It was the work done in collaboration between Chomsky and Miller around 1960 which caused the rebirth of this interest. Obviously, the first thing to be shown was that linguistic entities such as constituents, transformations, deep structures, etc., play a demonstrable role in the processes of speaking and understanding. This was done by means of what was called the study of the PSYCHOLOGICAL REALITY OF LINGUISTIC CONCEPTS, an understandable, but somewhat misleading term. In effect, it is decidedly not so that without such study the linguistic concepts in question would have no claim to psychological reality. In Volume II, Chapter 1, we showed that the empirical domain of (transformational) linguistics consists precisely of linguistic intuitions. A linguistic concept is psychologically real to the extent that it contributes to the explanation of behavior relative to linguistic judgments, and nothing more is necessary for this. Although the term is misleading, it does indeed have content in that it refers to the question as to whether constructions which are suited to the description of one form of verbal behavior (intuitive judgments) are equally suited to the description of other verbal processes (the comprehension and retention of sentences,

etc.). But for this the term STUDIES OF PSYCHOLOGICAL VALIDITY might be more fitting. The fact that originally an affirmative answer was expected to the above question is largely the result of that which we described as the identification of grammar and linguistic competence in Chapter 1 of this volume. Linguistic competence is at the basis of all verbal behavior, and grammar, therefore, is expressed in all verbal processes. Consequently, it should take little effort to prove the psychological reality of grammar on a large scale. Grammatical considerations were in fact so exclusive in this kind of experimental research that in most cases no effort whatsoever was made to produce a general model in which the relationship between grammar and psychological processes is outlined.

Retrospectively, however, we notice that all the models had an implicitly isomorphistic character; the more rules there were in the grammar, the more complicated were the psychological processes. In Matthews' analysis-by-synthesis model (1962), the assumption of isomorphism is made explicitly rather than implicitly, but the model characterizes the period well. It is a model of the hearer in which the grammar is a source from which structural descriptions are generated. The sentence introduced as input is temporarily stored in a memory, and the generator recursively enumerates structural descriptions, the terminal strings of which are compared with the input stored in the memory. When that generated terminal string coincides with the input string stored in the memory (this is established by a "comparator"), the process stops, and the structural description is accepted as the analysis of the sentence introduced as input.

Naturally, the more rules employed in the linguistic description of the sentence, the more time the synthesis process will take. There is thus a close relationship between the grammar and the process of comprehension. From the beginning it was also quite clear that this way of proceeding cannot be done in real time. It was calculated that if the generation of a twenty-word sentence took one second, it would take 10^{42} seconds to find the correct structural description of such a twenty-word sentence introduced

as input, a time longer then the history of the earth. Therefore Matthews and others made a number of additions to the model. One of these was the so-called "preliminary analysis"; in it a preprocessor directly produces a structure which corresponds to the sentence in a number of respects. The comparator then shows the differences which remain, and the generator attempts to find a structure more accurate with regard to those aspects of the sentence. Alternation between comparator and generator reduces the differences until they are considered sufficiently small according to some criterion. Also, the preprocessor and the generator are restricted to the generation of sentences which are not greater in length than the input. Unfortunately, however, these and other modifications have not been able to save the model; the only way to do this would be to extend the function of the preprocessor to such a degree that the lion's part of the work would be accomplished by it. In that case, however, we would be dealing with a semi-isomorphistic model in which only the output, and not the mechanism, is linguistically defined. The question as to how the preprocessor really works would then come to the foreground; its answer has never been quite clear in the analysis-by-synthesis model.

As we have stated, also other (implicitly) isomorphistic models have been in the background of studies on the psychological reality of linguistic concepts; in the following we shall mention these briefly where necessary.

It was only after a veritable rage of "reality studies" that psychologists came to realize that there are systematic exceptions to this isomorphism, and that those exceptions indicate that grammar plays a less direct role than was originally thought to be the case. One started studying psychological processes of understanding in their own right, without much recourse to grammar. Only one supposition of the isomorphistic point of view was retained, namely that the output of the hearer model or the input of the speaker model is a linguistically defined object, the deep structure of the sentence. The necessity a priori of building a complete grammar into a model of the language user came to be considered less urgent, and it was possible to make the matter the subject of renewed discussion.

Non-isomorphistic conceptual models lack that supposition as well. Investigators in that camp have initially been indifferent or even hostile to the idea of linguistic parallels.

In the present chapter we shall treat the isomorphistic models first. Because they are directly inspired on formal linguistic theory, they can best be subdivided according to the formal structure of that theory, thus, according to types 3, 2, 1, and 0. Regular models go back, for the most part, to communication theory. Other phrase structure models (types 2 and 1) stem from Chomsky and Miller's formulations (section 3.3.), as do the transformational models (type-0, to be discussed in section 3.4.). The object of semi-ismorphistic models, almost without exception, has been the hearer. Their most characteristic representation is found in the theory of processing strategies and lexical complexity, as developed by Fodor, Garret, and Bever (section 3.5.). Non-isomorphistic conceptual models will be treated in section 3.6., and some general conclusions will be drawn in section 3.7.

3.2. THE LANGUAGE USER AS A FINITE AUTOMATON

It is commonplace that human information processing capacities are limited. If we consider man to be an automaton, he must decidedly be a finite automaton. In theory, then, he should only be able to deal with regular languages (cf. Volume I, Chapters 4 and 5). But we have seen that natural languages are almost certainly of a more complicated sort (cf. Volume II, Chapter 2), and consequently we should conclude that man must necessarily err in the use of his own language. This matter has been sufficiently explained to show us that it is no mysterious paradox. The real questions have to do with the way in which man errs as language user, and whether we can learn something about the nature of his limitations. In terms of automata, we must find out whether we can consider man as a finite automaton limited as such, as a push-down automaton with a limited push-down store, or as a linear-bounded automaton with an upper limit to the input tape, etc. All of these constructions are equivalent to finite autom-

ata, but there are great differences in the types of failure which they will show. A limitation on the push-down store is expressed in an upper limit to the number of self-embeddings which can be accepted by the automaton; a limitation of the linear-bounded store places a direct restriction on the size of the sentence. In this paragraph we shall discuss the way in which a finite automaton will err with respect to natural languages, and we shall consider the extent to which the characteristics of that model correspond to those of human linguistic behavior.

The only finite automata which have been examined as models of linguistic behavior are k-limited probabilistic automata or higher order Markov sources (cf. Volume II, Chapter 6, section 6.1.). Attempts were made to prove the psychological reality of this model by varying the order (k) of the text; the aim of this was to show that as k increases (or uncertainty decreases), the text can be more easily processed and memorized. The classical experiment for this was that of Miller and Selfridge (1950). They used zero to seventh order approximations of English, as well as an ordinary text. Some of their examples were given in Volume II, Chapter 6, section 6.1. They calculated the average percentage of words which subjects could reproduce after seeing a series of words once. The series varied in length from ten to fifty words. The results are given in Figure 3.1.; they show that the probability of recall increases to the fourth or fifth order. But from that point no further improvement took place; the level of the ordinary text had been reached. This indicates that a role is played by something other than decreasing uncertainty, and what that precisely is became apparent from later work by Miller (1962). He showed that verbal memory is particularly sensitive to the grouping of words into units of a certain size, and suggested that in the perception of speech that size roughly corresponds to the phrase, i.e. nominal, verbal, and other linguistically defined phrases containing two to six words. If the material cannot be grouped into such linguistic units, it is relatively difficult to memorize; this is the case for the lower order approximations. In higher order approximations, such groups can be formed (cf. Tulving and Patkau 1962), and the capacity of

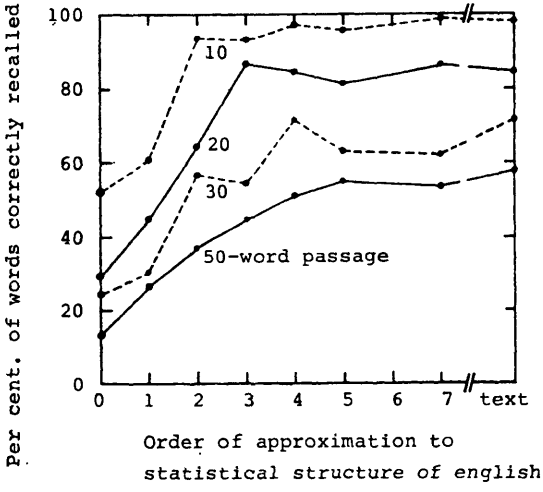


Fig. 3.1. Recall of word strings as a function of order of approximation to English and string length (after Miller & Selfridge).

direct memory is used to a maximum. Meaning-bearing relations between phrases, as in an actual text, do not add much to this. The important variable here is therefore probably the possibility of phrase formation rather than uncertainty. In the following section we shall return to the subject of the phrase as a processing unit.

In any case, the results indicate that man, considered as a Markov source, is no more than 4-limited; limitations of a higher order are not reflected in his performance. For other reasons, however, the 4-limited automaton is an absurd model for the human language user, given considerations of the following kind. Normal speakers and listeners have no difficulty in understanding the sentence *the person, whom you invited recently to come and give a lecture later in the year, seems to be out of town.* If the word *person* in this sentence is changed to *persons*, anyone will immediately realize that *seems* should be changed to *seem*. In general the language user will not err on this dependency, although it spans fifteen words. That is far more than four!

A Markov source with $k = 15$ will contain an enormous number of parameters. Let us calculate that number. We suppose that the source generates word types rather than words (if words themselves are generated, the argument becomes much stronger still), and that there are no more than four word types (an extremely low estimate). The vocabulary thus has four elements, $n = 4$. With the formulas given in Volume II, Chapter 6, section 6.1., we find that the model contains $4^{15} \times (4 - 1)$ parameters, which is more than three billion. A child who wishes to assimilate the language of his environment would thus be obliged to estimate about thirty parameters per second throughout his entire childhood. This is completely unrealistic, especially if we consider that for the estimation of every transition probability at least a few presentations of the word sequence in question are required. A Markov model of the human language user, therefore, does not err in the correct way. With a reasonably limited number of parameters, the model decidedly cannot recognize grammatical dependencies over long sequences, while a human being can do this without difficulty.

Other finite automata have never seriously been studied as models of the language user, but we can expect that examples will always be found in which the model either fails where man does not, or it contains an impossibly high number of parameters.

3.3. NON-REGULAR PHRASE STRUCTURE MODELS

The property of self-embedding (Theorem 2.8. in Volume I) places all non-regular languages outside the reach of finite automata. Only with the addition of an unlimited store can an automaton accept such languages. The reader may also remember that push-down and linear-bounded automata may have a structure analogous to the corresponding grammars; in the proofs of equivalence, each of the transition rules reflects a production of the grammar (cf. the constructions in Volume I, Chapter 5, section 5.2., and Chapter 6, section 6.2.). As a consequence of this, the sentence is accepted in

steps which correspond to the structural descriptions in question. In this respect, these are optimal recognition automata, for the history of acceptance precisely reflects the structural description of the sentence.

If these models are selected as models of the human language user with his limited capacity, it will be necessary to limit the store. It follows directly from this that an upper limit will have to be set to the number of times self-embedding can occur in a sentence which is to be processed by the model. If we wish also to maintain the feature according to which the model produces the correct structural description as long as that limit is not attained, the same upper limit will have to be established for all nestings of elements, and not only for self-embedding. A push-down automaton, in particular, retains a nonterminal symbol in the push-down store until all nonterminal symbols in which it was nested have been removed from the store (see Volume I, Chapter 5, section 5.2. for an example of this). A limited push-down automaton, therefore, has an upper limit to the number of nestings, and that limit is the same for self-embedding and other nestings. This also holds, *mutatis mutandis,* for linear-bounded automata.

The question is whether man also has such an upper limit to nesting, and whether that limit is the same for self-embedding and other forms of nesting. The first part of the question can clearly be answered in the affirmative, the second probably in the negative. A whole series of studies (Blumenthal 1966; Fodor and Garrett 1967; Foss and Lynch 1969; Freedle and Craun 1969; Perchonock-Schaefer 1971; Phillips and Miller 1966; Stolz 1967) shows that multiple nesting as well as self-embedding renders a sentence incomprehensible. Chomsky and Miller (1963) give the following example of five-fold nesting:

Anyone who feels that if so many more students whom we haven't actually admitted are sitting in on the course than ones we have that the room had to be changed, then probably auditors will have to be excluded, is likely to agree that the curriculum needs revision.

The sentence is quite incomprehensible on first reading. But it shows no self-embedding. In order to compare self-embedding

and other forms of nesting, let us compare the following sentences. Sentences (1) and (2) contain one and two self-embeddings, respectively, and sentences (3) and (4) contain one and two non-self-embedding nestings, respectively.

(1) *if if John comes Peter comes Charles comes*
(2) *the dog that the cat that the mouse killed scratched is large*
(3) *John, who has seen everything, will tell you about it*
(4) *John, who has seen everything you mentioned, will tell you about it*

The last two sentences are strikingly easier to understand than the first two. The human observer can manage two or three nestings, provided that they are not cases of self-embedding. Chomsky and Miller (1963) suggest that the observer not only has a limited memory, but also is subject to the condition that a perceptual operation may not interrupt *itself* more than once (cf. Bever 1970a for further possibilities of explanation).

Whatever the origin of the problem of self-embedding may be, the fact that self-embedding is much more difficult than other forms of nesting indicates that the push-down automaton does not fail in the same way as man.

Yet the push-down automaton has sometimes proved to be an apt model for the analysis of some general aspects of speech. An example of this concerns a general characterization of the language of schizophrenics. There is a good deal of literature on the formal aspects of the language of schizophrenics (see, for example, Border 1940 and Ellsworth 1951). In comparison with normal people, schizophrenics use (i) more objects per subject, (ii) fewer qualifications per verb, (iii) fewer different words, (iv) fewer adjectives, (v) fewer adjectives per verb, (vi) shorter sentences, and (vii) more incomplete sentences. These findings are a gold mine for psychiatric interpretation. Each of these various phenomena can be considered as an indicator of certain changes in the emotional and thought structure.

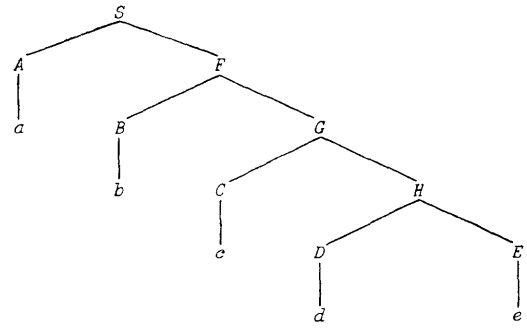
Masters (1970) simulated a push-down automaton on a computer; the automaton was based on a context-free grammar of

English. He had it generate sentences by random procedure, under various limits on the push-down store. With an unlimited store, each output would, of course, be grammatical according to the context-free grammar, but by limiting the store, the automaton produces not only grammatical sentences, but also various partial constructions which cannot be completed because of overflow in the store. Masters varied the number of push-down symbols from two to ten. He showed that above a limit of six, the percentage of incomplete constructions came close to zero (0.8 percent with seven symbols). Apparently then, no more than seven elements need be held in the memory in order to process syntactically nearly all the English sentences produced by this grammar. It is encouraging to note the correspondence of this number with the capacity of man's immediate memory (cf. Miller 1956), though it is difficult to come to judge the quality of Master's grammar. The interesting point is that further reduction of the capacity of the store brings about all the characteristics of the language of schizophrenics mentioned in points (i) to (vii). Although at first sight these characteristics seem very different from each other, apparently they can all be drawn back to one underlying factor, a store of limited capacity. Psychiatrists would do well to investigate whether the immediate memory of schizophrenics is indeed of relatively limited capacity, and if so, what the cause of this can be. It would therefore be pointless to give separate explanations for each of the phenomena mentioned.

Push-down automata and context-free grammars are also the heart of Yngve's model of the speaker (Yngve 1961). The model simply states that the speaker gives a binary leftmost generation of the sentence (cf. Volume I, Chapter 2, section 2.3.4.). Starting with the start symbol *S*, he successively rewrites each leftmost nonterminal symbol in the string. In this way, the various words are derived in the correct syntactic order. Yngve then defines the concept of DEPTH. When a word is derived, it is assigned a number which indicates how many nonterminal elements the string still contains at that moment. The number is equal to that of the push-down symbols (excluding the start symbol) in the store of the

corresponding push-down automaton at the moment the word in question is generated. The depth is the number of push-down or nonterminal symbols which remain at that moment. A measure of complexity could be the maximum depth (storage capacity) required by the sentence. Sentence structures which are left-branching have greater depth than those which branch to the right. With binary right-branching there are never more than two nonterminal symbols in the store, while with binary left-branching the maximum is $n - 1$ for a sentence with n words. Compare, for example, the right- and left- branching structures for the sentence *abcde* at the bottom of this page and the top of the next. According to Yngve, right branching structures are less burdensome for the speaker than left branching structures, and consequently the former are predominant in most languages (this also holds for the languages which are usually said to be left-branching languages, such as Turkish; these are really right-branching, but to a lesser degree than other languages). This is called Yngve's DEPTH HYPOTHESIS.

right-branching



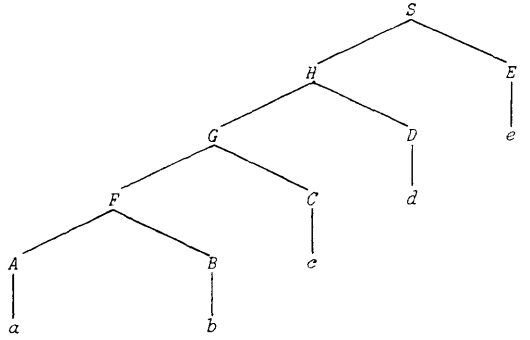
Leftmost derivation:

$S \Rightarrow AF \Rightarrow aF \Rightarrow aBG \Rightarrow abG \Rightarrow abCH \Rightarrow$
Depth 1 2 1 2

$abcH \Rightarrow abcDE \Rightarrow abcDE \Rightarrow abcde$
1 2 1 0

Maximum Depth: 2

left-branching



Leftmost derivation:

$S \Rightarrow HE \Rightarrow GDE \Rightarrow FCDE \Rightarrow ABCDE \Rightarrow$

$aBCDE \Rightarrow abCDE \Rightarrow abcDE \Rightarrow abcdE \Rightarrow abcde$

Depth: 4 3 2 1 0

Maximum Depth: 4

Miller and Chomsky, on the other hand, state that the memory of the hearer is better served by left-branching constructions (Miller and Chomsky 1963). The hearer must try to process incoming words as rapidly as possible. In other words, he must remove them from his immediate memory by replacing groups of words with a code, by replacing the constituent with the nonterminal symbol. With constructions which are right-branching, this process can begin only when all the words have been received; with left-branching constructions, it can begin at the first word. They conclude that it is unclear why a language should especially serve the interests of the speaker, as Yngve's hypothesis presupposes. However, there is a rather good reason for this: as we shall explain later, the hearer often need not perform any syntactic analysis in order to understand the sentence. The speaker, on the other hand, must cast the sentence in the proper syntactc form. Yngve's hypothesis certainly contains a source of truth.

Beside Yngve's measure of depth, other measures of complexity have been derived from the phrase structure model. The reason for

this is that in using an isomorphistic model, one can expect the linguistic complexity of the structure to be expressed in some way in verbal behavior. Rejecting Yngve's measure of depth, Miller and Chomsky (1963) in turn propose a measure which is not dependent on the right-left progression of the structure, but only on the quantity of hierarchy in that structure. The maximum of hierarchy is found in a completely binary tree diagram in which each nonterminal node has two outgoing branches, as in a grammar in Chomsky normal-form (cf. Volume I, Chapter 2, section 2.3.1.). A minimum of hierarchy is found when all terminal elements with multiple branching proceed from a single node. A measure of hierarchy is therefore the node-to-terminal-node ratio or NTN RATIO; this is the number of nodes (including the terminal elements) divided by the number of terminal elements. This ratio would roughly indicate the amount of processing per word necessary to both speaker and hearer. Unlike Yngve's measure of depth, the psychological attractiveness of this ratio has never been proven.

But both these measures have the disadvantage of ignoring completely the lesson of information theory. Given the grammar, which alternative structures are possible and what are their probabilities? A measure of complexity for a phrase structure grammar might more effectively be based on the number of alternative rewrites per nonterminal symbol in the grammar. In the simplest case, one could consider the probabilities of all these alternatives to be equal, and calculate $p(s)$, as given in Volume I, Chapter 3, section 3.4. The measure of complexity would then be the uncertainty, $H = -\log p(s)$. This is in fact the measure of complexity which we used in comparing grammars generated by a grammar-grammar in Volume I, Chapter 8. Further refinements, taking, among other things, conditional probabilities into consideration, then become obvious.

To this point we have discussed explicit type-2 models of the language user (we refer the reader also to Osgood 1963 for a kind of probabilistic context-free model). As far as we know, the value of the linear-bounded automaton as a (type 1-) model has not yet been studied, although it has some rather attractive features,

such as a store which, up to a certain limit, adapts to the size of the input sentence.

In the remainder of this section we shall mention work to which the title of "studies on the psychological reality of linguistic concepts" applies more directly. There is a vast literature in which the constituent is taken as a unit of processing in perception, memory, and reproduction of the sentence. We shall not attempt a complete survey of the material here (for further information, see Neisser 1966; Levelt 1966; Fillenbaum 1971; Mehler and de Boysson-Bardies 1971; Loosen 1972). Our aim here is only to offer an idea of the kind of evidence which was sought for the testing of isomorphistic phrase-structure models of the language user.

The most striking feature of all the experiments in question is a lack of solicitude for the psychological details of the processing model. Authors tended to be satisfied with proving the "psychological reality" of a constituent structure, but took little trouble to find the psychological mechanisms which were responsible for this. This is a characteristic of the implicit isomorphistic approach. In Miller and Chomsky (1963) we find at least a rough indication of such a processing model. They consider the psychological reality of constituents to be the consequence of the features of a PREPROCESSING MECHANISM. The hearer, in processing a sentence, first tentatively parses the string in smaller and larger phrases on the basis of various indications, such as intonation, function words, articles, and so forth. Thus segmented, the signal is available in immediate memory and will serve in turn as the input of the MAIN PROCESSING which derives the syntactic and semantic relations among the various parts of the sentence. Aside from the fact that the details of the preprocessor are not discussed at all, this distinction between preprocessing and main processing finds little experimental support. Indeed, the whole idea that the psychological reality of linguistic entities such as phrases is caused by features of perceptual processing is used more as a presupposition than as a proposition to be tested experimentally. In many of these "reality studies", the subject is presented a sentence and asked to reproduce it either immediately or after a certain lapse of time,

and it thus becomes impossible to distinguish perception from reproduction. Loosen (1972) correctly points out that a so obtained constituent-wise parsing structure could as easily reflect a characteristic of the retrieval process as one of the perceptual mechanism. When the two phases can be distinguished, for example in reaction time experiments in which subjects are not asked to reproduce the sentence, no evidence can be found for a strict temporal separation between preprocessing and main processing. This can be seen, for instance, in the so-called garden path phenomenon, demonstrated by the following example:

The cherry blossoms during summer into full bloom.

We reach a dead end halfway through the sentence by interpreting *blossoms* as a noun, and only at the end can we introduce a correction to this. If the entire sentence were first recorded in the memory and segmented before further interpretation is begun, the correct interpretation would have been given from the very beginning. Such effects can easily be demonstrated experimentally (see the chapter on ambiguity in Flores d'Arcais and Levelt 1970). If there is indeed a preprocessing phase, it will have to do with smaller word groups than the sentence as a whole. The only reasonable interpretation of such a model would then be that although preprocessing and main processing alternate while the sentence is being listened to, they should still be distinguished since syntactic and semantic decisions made in the main processing only concern complete word groups which should therefore have been recognized as such in an earlier preprocessing phase. But on introsprective grounds this is not plausible. It seems for instance possible first to process the relations among the endocenters of phrases, and only later to deal with their internal structure. It is not very likely that the first three words of a sentence such as *Mary watched Trudy who played in the garden* are processed differently than the sentence *Mary watched Trudy*. The verb-to-object relation between *watched* and *Trudy* can be grasped before the relative clause is processed.

We shall now mention two kinds of experiments which are characteristic of the studies of the psychological reality of linguistic

concepts; these are the sentence reproduction paradigm and the click experiment. Dozens of other approaches can also be found in the literature on the subject.

We shall begin with an example of the sentence reproduction paradigm. Levelt (1970a) had 120 subjects listen to sentences whose intelligibility had been decreased by the addition of white noise. After each sentence the subject had to write down what he had understood. The following calculation was performed for each sentence. Let i and j stand for two words in a sentence. Of the subjects who had correctly reproduced i we determined the percentage who had also understood j. This percentage was taken as an estimate of the conditional probability that word j can be reproduced if i can be reproduced; the notation for this is $p(j|i)$. If the sentence undergoes a hierarchical analysis, like that of a phrase marker, during the transmission from perception to reproduction, then we can expect the following. Let i, j, and k be three words in the sentence; j and k belong to a phrase to which i does not belong. If the phrase functions as a whole during the transmission (i.e. if it is "psychologically real"), we should expect that $p(j|i) = p(k|i)$. We can develop this reasoning in a way analogous to that in Chapter 2, section 2.4.2., of the present volume, and deduce that a hierarchical organization of the sentence must result in an ULTRAMETRIC matrix of forward[1] transition probabilities. The results of the experiment showed that this ultrametric feature was satisfied strikingly well. Figure 3.2. shows the best fitting binary tree diagrams for two syntactically related sentences in the experiment. They are nearly perfect representations of the observed transition probabilities. The experiment not only showed that a hierarchical analysis takes place, but also that the larger psychological entities generally correspond to linguistic constituents such as *het water onder de brug* 'the water under the bridge', *onder de brug* 'under the bridge', *draait in kolken* 'whirls in eddies', and that syntactically equal sentences elicit the same analysis. The smaller units, however,

[1] Backward conditional probabilities of the type $p(i|j)$ can also be studied. An interesting analysis of this for the present experiment can be found in Loosen (1972).

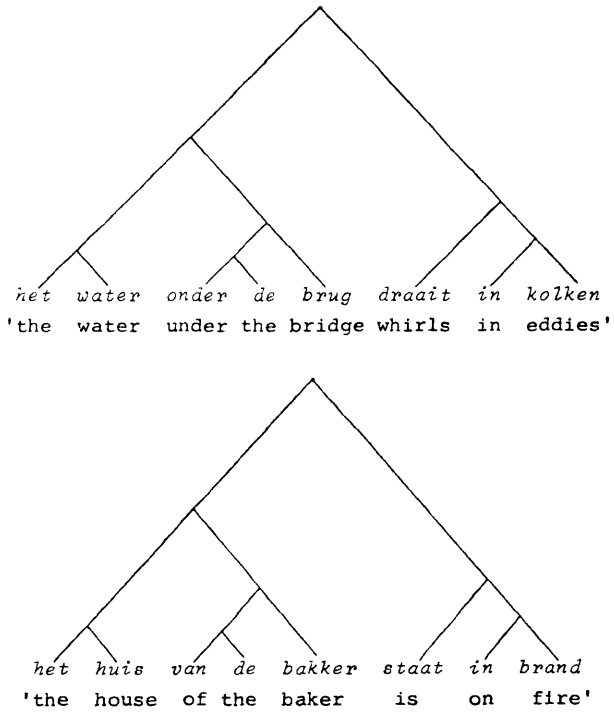


Fig. 3.2. Observed hierarchical analysis for two experimental sentences (after Levelt 1970a)

such as *onder de* 'under the' and *van de* 'of the', do not always agree with the linguistic structure.

It is by no means clear how the hierarchical parsing occurs. We supposed (Levelt 1970a) that it is a feature of the perceptual mechanism. Already in the perceptual phase, the hearer begins to group words into more and more extensive larger phrases. If he is able to understand a given word, the chance increases that he will also understand a later phrase *as a whole.* But nothing in our data would prevent an interpretation in terms of retrieval. The subject was aware that he would be asked to reproduce the sentence. Suppose that he understood a few words (e.g. *is, fire*), but was

unsure of a few other words (e.g. *on*). If he constructs a more or less plausible sentence around the words which he has understood, there is a certain chance that he will correctly guess some of the words which he did not understand. The choice is rather simple for a string such as *is? fire.* Loosen (1972) repeated the experiment, but he presented the words in random, rather than syntactic, order. He instructed the subjects before the presentation of the strings of words, to reproduce what they could understand, as far as possible in the form of a sentence. The results of his experiment basically agreed with those of our own. This uncertainty as to the cause of such psychological constituent structures holds for all experiments on sentence reproduction, and in particular for Johnson's (1965) experiment, one of the first on the subject.

The situation is no different for the click experiments. It was at first thought that a method had been found to prove that syntactic parsing of a sentence is operative already during perception. The click procedure is the following. The subject wears earphones, and hears the test sentence in one ear, while at a given moment during the presentation of the sentence, he hears a brief click in the other ear. The subject must later say at which point in the sentence he heard the click. This method was developed for nonsyntactic material by Ladefoged and Broadbent (1960), and was first used by Garrett for the analysis of sentence perception. The characteristic result of such experiments is that the click is not located at the correct place, but rather shifts to the nearest large constituent boundary. The first article in which the click experiment was used as a test of "psychological reality" was titled "The Psychological Reality of Linguistic Segments" (Fodor and Bever 1965). The following example is taken from Garrett, Bever, and Fodor (1966). When the sentence *your hope of marrying Anna was surely impractical* was presented with the click in the middle of the word *Anna*, the subjects described the click as occurring between the words *Anna* and *was*, thus at the major constituent boundary in the sentence. This result was not due to intonation or pause patterns in the sentence. When the word *your* was cut from the sound tape and replaced by

in her, yielding the sentence *in her hope of marrying Anna was surely impractical*, the subjects located the click between the words *marrying* and *Anna*, once again in agreement with the linguistic segmentation. In the article reporting this experiment (entitled "The Active Use of Grammar in Speech Perception") the authors presupposed that, even without special acoustic indications, the hearer, by making active use of his syntactic knowledge, perceives the sentence in a particular way. Other experiments (Bever, Lackner, and Kirk 1969) and various unpublished material (Fodor, Bever, and Garrett i.p.) show not only that the boundaries between the large constituents attract the click, but also that the underlying structure of the sentence is a determinant for the localization of the click. Thus, for the sentence *they watched the light green car*, subjects showed no tendency to dislocate the click to the boundary between the words *watched* and *the* (a minor constituent boundary), while such a tendency was indeed in evidence for the sentence *they watched the light turn green*, the constituent structure of which is the same at the critical place. The difference is that *the light turn green* is itself an embedded sentence, whereas *the light green car* is not. This seems to indicate that the click experiments confirm the psychological reality of the major and minor constituents, provided that these latter reflect subsentences in the deep structure. Other experiments (Feldmar 1969, Ladefoged 1967, Reber and Anderson 1970, Reber 1973) call the perceptual linguistic interpretation of these findings seriously into question. Although the origin of the click-shift phenomenon has not yet been explained, there is solid experimental ground for the following propositions: (1) if a syntactic factor is operative, it is so by response bias, and not by perceptual mechanisms; (2) the most important perceptual determinant of the click shift is the intonation pattern of the sentence, and not the constituent boundary; (3) nonlinguistic factors play a dominant role in the clickshift. We shall briefly discuss each of these points.

(1) In the original experiments the subjects were asked first to write down the sentence which they heard, and then to indicate the place at which they heard the click. The writing down of the

sentence could give rise to response bias. Feldmar performed the experiment as follows. He presented the sentence twice, each time with the click located at a different place. Two conditions were introduced. In the first of these, the subject was asked to say in which of the two presentations the click occurred earlier; only afterward did he write the sentence down and indicate the places at which he heard the click. In the second condition the subject wrote down the sentence immediately and then marked the positions of the click. The result was that the shift toward major constituent boundaries occurred in the latter case, but not in the former. Obviously, then, the reporting of the sentence is itself a factor which influences subjects in judging the place at which they heard the click. If the subject must first reflect on the position of the click, the syntactic effect is eliminated, or at least decreased.

A more direct and convincing study of response bias was presented by Reber and Anderson; it also led to conclusions, (2) and (3). Their experimental material consisted of strings of six two-syllable words. Strings which were sentences were of the following type: *open roadside markets display tasty products.* The string was presented on one loudspeaker, and the click on another. In principle the click could occur within any syllable but the first and last of the strings, or between any two words. The major constituent boundary lay between the third and fourth words (position 0). After each presentation, subjects were given a printed copy of the sentence and asked to mark the place at which the click occurred. The original click-shift effect was reproduced without difficulty. The condition which was interesting with regard to point (1), however, was the following. A group of subjects was presented with the sentence, but not the click. They were told that there would be a click which they might not notice, but that they would be likely to locate it with more than random probability, as "was the case in other experiments". The subjects showed a strong tendency to locate the fictitious click at position 0, confirming the standard results of experiments of this type. At least in this case, response bias, and not perception, is a sufficient explanation. Response bias could also explain the results

obtained with the *Anna* sentence mentioned above, but syntactic factors might have had a perceptual effect as well.

The experiment performed by Reber and Anderson also contained a condition according to which the words of the sentence were not presented in syntactic order, but rather randomly. In that case the subjects made more errors in locating the click, whereas the very factor supposed to be determinant for the clickshift had been eliminated. Yet the tendency to locate the click in the middle of the string (i.e. at position 0) was still observed, although that tendency was somewhat weaker than in the case of the syntactically normal sentences. Although this latter, limited effect might be attributed to a perceptual factor, it would be more natural to consider it as a response bias. Faced with uncertainty, the subject will show a stronger tendency to locate the click at position 0 when given a real sentence than when given a random string of six words.

(2) Another condition of the experiment concerned the comparison of natural and monotonous intonation of the sentence. Of all the stimulus variables employed, this was the most important to the click-shift effect. Not only did the number of errors in the location of the click greatly decrease with monotonous intonation, but the direction bias (i.e. the tendency to move the click to position 0) was also largely eliminated. The characteristic response bias was likewise considerably stronger for sentences presented with normal intonation than for sentences read in monotone in the experiment in which the click was not given, and consequently where only response bias was in question. The most important factor for response bias, thus, is intonation.

(3) The authors repeated the experiment with nonlinguistic stimulus material. The words were replaced by stretches of white noise, represented on the answer sheet as a sequence of six rectangles. Once again the characteristic click-shift toward position 0 was observed. No linguistic factor whatsoever can have come into play here. It was also found in this, as in other experiments, that errors in the location of the click were greater when the click occurred at the beginning of the string than when it occurred at

the end. Two non-linguistic factors may have played a role in this experiment, and consequently in the others. The first of these is immediate memory. As the lapse of time between click and answer decreases, the location becomes more precise. The second factor is less obvious; it has to do with the reason why clicks which occur early in the string are shifted "to the right", and those which occur late in the sentence are shifted "to the left". The authors sought a perceptual explanation. On the basis of Broadbent's single channal hypothesis, they supposed that a subject, once trained, concentrates his attention at first on the string, whether it be a sentence, a series of random words, or of noises. He must then turn his attention to the click when it occurs, and will lose time in doing so, thus slowing down the perception of the click. The situation is reversed when the click occurs near the end of the string. By that time the string has become redundant — certainly if it is well intonated and syntactically correct, but also if it consists of series of noises — and the expectation of the click, which has not yet occurred, increases considerably. Consequently, attention is concentrated on the click-channel, and the perception of the click is somewhat anticipated. It should be noticed, however, that this point can also be explained on the basis of response bias.[1]

The classical click-experiments therefore prove no perceptual constituent effect, and they become considerably less attractive from a psychological point of view when we realize that we are dealing only with response bias. Therefore one could as easily ask the subjects immediately to analyze the sentences by marking off the constituents. This may be seen in experiments in which the hearer reacts during perception. Foss and Lynch (1969), for example, asked subjects to push a reaction key as soon as they heard the letter *b* in the sentence. The result was that reaction times were particularly long near major constituent boundaries (which is quite contrary, moreover, to the idea behind the original click-experiments, according to which such a boundary is a suitable place to change channels and to allow the click to pass). At the

[1] As this translation went to press, a sequel to this experiment was published (Reber 1973), in which the results mentioned above were largely confirmed.

end of a major constituent, there is obviously a greater mental load. Relatively slow reactions are likewise observed after an ambiguous segment, because it is rather difficult at that point to free attention for the extra task. Mental load is also expressed in autonomic reactions. Bever, Lackner, and Kirk (1969) showed that the effect of slight electric shocks on the psychogalvanic reflex varies in function to the moment in the sentence at which such shocks are administered. The effect was more limited when the shock was given near a major constituent boundary. This confirms the impression that mental load at such a moment is relatively great.

In summary we can state that if constituents are indeed a psychological reality, the origin of that reality is unknown. Certainty may be had only as to the effects of intonation pattern and response bias.

3.4. TRANSFORMATIONAL COMPLEXITY AND THE CODING HYPOTHESIS

Thanks to transformational grammar, a principle used in the psychology of language (by Wundt and others) around the turn of the century was reintroduced. The principle is that the superficial word order reflects a more abstract structure of relations and functions: *aussere Sprachform* versus *innere Sprachform*. In the early 1960's the distinction between surface structure and deep structure was the subject of feverish activity in psychological research. Only in retrospect can we see that that work was based on two working hypotheses, which we shall discuss here. These hypotheses are not entirely independent of each other, and both are forms — each in a different sense — of the isomorphism discussed in the first section of this chapter.

(1a) *The Coding Hypothesis.* The coding hypothesis describes the results of the processing of a sentence by the hearer, or the code in which the sentence is put into memory. This hypothesis has one of two forms, depending on the precise form of the transformational

theory used. If the transformations are considered to be paraphrastic, transformations such as the passive or the interrogative must already be marked in the deep structure; the base rules generate such transformation labels as *pass* and *Q*. In this case, the coding hypothesis is as follows: the memory code of a sentence is isomorphous with the deep structure of that sentence. Two further hypotheses are usually directly added to this:

(1b) The transformation labels are retained in the memory without interaction either with the rest of the deep structure (the base relations of subject, object, etc.) or with each other.

(1c) The code for the base relations has priority over transformation labels. The latter are learned with greater difficulty, and are forgotten more easily.

When we refer to the coding hypothesis without further qualification, we mean hypotheses (1a), (1b), and (1c) taken together. If, together with the interpretative semanticists and lexicalists (cf. Volume II, Chapter 3, section 3.4.), or as in the period before the publication of Katz and Postal (1964), we do allow transformations which change meaning, then hypothesis (1a) should be read as follows: the memory code for a sentence is isomorphous with the deep structure plus the list of transformations which are applied in the generation of a sentence. In hypotheses (1b) and (1c), "transformation labels" will then be changed to "transformation-operations". The hypothesis has never been reformulated on the basis of the generative semantic point of view, where the very notion of deep structure is considered to be on rather slippery ground. But the formulations of the coding hypothesis are as numerous as the articles which deal with it. No attempt has ever been made to unify these formulations and make them more precise.

(2) *The Derivational Theory of Complexity (DTC)*. According to this hypothesis, the processing of the sentence simulates the transformational derivation of the sentence. The hypothesis is known in several different forms, one of which is the analysis-

by-synthesis model, and another, a model which we shall call the "onion model". In the analysis-by-synthesis model, the hearer tries to approach the sentence iteratively by attempting various transformational derivations (the generation of the deep structure itself is not considered here) until the difference between the sentence heard and the sentence generated has been eliminated. In the onion model, for every transformation there is a psychological de-transformation. The sentence, as it were, is peeled, transformation by transformation, until the deep structure is accessible. Notice that this theory supposes that the preprocessor has already made the surface structure accessible (we have already expressed our objections to this idea), and that it is possible to reverse the transformational derivation (which, as we have seen in Volume II, Chapter 5, proved not to hold). A point which the two theories have in common is that for every transformation there is a corresponding operation; thus the more complex the transformational structure of the sentence is, the more difficult is the processing. Here we see the isomorphism at its purest, but applied to the microstructure of the transformational derivation rather than to the derivation of constituents, as was the case in the preceding paragraph.

As opposed to the derivational theory of complexity, the coding hypothesis is what we have called isomorphism on the macroscale (in the first paragraph of this chapter). The supposition is that the process by which a string of words is understood runs parallel to the linguistic levels of surface structure interpretation, deep structure interpretation, and semantic interpretation.

Nearly overwhelming evidence was originally presented for both of these hypotheses. But as many of the experiments used have since been completely by-passed, we shall not discuss this evidence in detail here. It seems more efficient to chose a number of characteristic examples, as was done in the preceding paragraph from the best of those studies, and to refer the reader to the bibliography at the end of this volume for further information. It will thus likewise be possible to pay more attention to the arguments which have sounded the death knell for this model.

The coding hypothesis was first studied by Mehler (1963) within the framework of a research program, under the direction of George Miller, in which the use of transformational grammar in psycholinguistics was first introduced. Almost all the studies in the program were characterized by experimental material in which interrogatory, passive, and negative transformations were the principal variables. Thus in Mehler's experiment, sentences of the following eight transformational forms were used:

K (kernel sentence)	*the secretary has typed the paper*
P (passive)	*the paper has been typed by the secretary*
N (negative)	*the secretary has not typed the paper*
Q (question)	*has the secretary typed the paper*?
NQ	*hasn't the secretary typed the paper*?
PN	*the paper hasn't been typed by the secretary*
PQ	*has the paper been typed by the secretary*?
PNQ	*hasn't the paper been typed by the secretary*?

The deep structure of these sentences was considered to consist of the deep structure of the kernel sentence and the additional *P*, *N*, and *Q* labels. Mehler asked his subjects to learn the sentences, and to reproduce them when a prompt word, such as *paper*, was given. It was expected that the sentences would become more difficult to memorize as the number of labels increased. The results confirmed the expectation. Kernel sentences were correctly reproduced in an average of 75 percent of the cases; *N*, *Q*, and *P* sentences in an average of 57 percent of the cases; *NQ*, *NP*, and *PQ* sentences in 47.2 percent of the cases; and *PNQ* sentences in an average of 46 percent of the cases. The most frequent reproduction errors concerned the change of one label (e.g. *K* instead of *P*, or *PN* instead of *PQ*), changes of two or three labels seldom occurred. The distribution of errors corresponded in general to the trinomial distribution expected on the basis of hypothesis (1b). Deviation from the expectation was observed only with regard to *NQ* sentences, which proved to be approximately as easy to learn as *Q* sentences.

Hypothesis (1c) was the object of a study performed by Mehler and Miller (1964). Subjects were given a list of eight sentences to learn. As in the experiment which we have just mentioned, the sentences differed in content, but each of the eight syntactic forms occurred once in the list. When the sentences were learned, the subjects were given a second list of eight sentences to learn in order to cause interference with the first eight sentences. The interference list (*IL*) could differ from the original list (*OL*) in a number of ways. In order to test syntactic interference, the interference list was composed in such a way that the sentences differed in 0, 1, 2, or 3 transformations[1] interference from the original list, although the content was the same. Semantic interference was caused by introducing an interference list of eight sentences of totally different content than the original list, but in which each of the forms, *K*, ..., *PQN*, occurred. As a control, a third group of subjects was given an addition task as interference. All three groups were asked to reproduce the original list. Subjects proved to be very resistant to semantic interference. Syntactic interference, however, was considerable, and the group with the syntactic interference list frequently reproduced sentences in incorrect syntactic forms. It appeared to be possible independently to interfere with a syntactic label in the deep structure. Moreover the experiment gave reason to suppose that the syntactic label was learned only after the rest of the deep structure had been established (hypothesis (1c)).

An experiment dealing with hypothesis (1b) which drew a great deal of attention at the time, was performed by Savin and Perchonock (1965). The writers attempted to measure the memory space taken up by the deep structure. Their method was as follows. The sentence was first presented acoustically, and after five seconds of silence, a string of eight disconnected words was presented. The subjects were asked to reproduce the sentence and the disconnected words as well as they could. The sentence material in the experiment consisted of forty-five kernel sentences, their variants

[1] One of them was a negation transformation, which in those days was treated as a purely syntactic matter.

in the *Q*, ..., *PQN* forms, as well as a *Wh* form *(who has typed the paper?)*, and *E* (emphatic) form *(the secretary did type the paper)*, and an *EP* form for each of the kernel sentences. When the number of words retained from a passive sentence is subtracted from the number of words retained from the corresponding kernel sentence, the difference, $K - P$, can be considered as the memory space taken up by the transformation label *P*. This is an operationalization of hypothesis (1b). If those labels are indeed coded independently of each other, the memory load for *P* can also be estimated on the basis of $E - EP$, $Q - QP$, and $QN - PQN$. The experiment showed that, within the measurement error, the differences were indeed equal. Similar results were obtained for the interrogative, the negative, and the emphatic labels.

Indications of the correctness of hypothesis (1a) may be found in particular in Blumenthal (1967) and Blumenthal and Boakes (1967). They asked subjects to learn a sequence of sentences and then to reproduce them one by one, when a prompt word from the sentence was given (prompted recall). Blumenthal's idea was that under hypothesis (1a) the deep structure of the sentence, not its surface structure, should determine which word would be an effective prompt word. The following two sentences, for example, have the same surface form, but they differ in underlying structure:

(a) *the meat was sold by the pound*
(b) *the meat was sold by the poor*

In sentence (a), *by the pound* modifies *sold*, which is realized in the deep structure as an embedded sentence, approximately as follows: $(_S somebody\ sold\ (_S\ it\ was\ by\ the\ pound)_S\ the\ meat)_S$. Sentence (b), however, is the passive form of the simple sentence *the poor sold the meat*. Let us examine *pound* and *poor* as prompt words. We can expect that *poor* would be a better instrument for the recall of the sentence in which it occurs than *pound* would be for its respective sentence. *Poor* is an element of the main clause (its subject), while *pound* figures only in the subordinate clause (notice that an additional hypothesis is tacitly given here on the relative importance of the main and subordinate clauses). This expectation

was confirmed by the experiment when care was taken that the subjects really understood the sentences (for this, Blumenthal included a paraphrasing task for the subjects). Similar results were obtained with pairs of sentences such as *John is eager to please* and *John is easy to please*. In the first sentence *John* is the subject of the main clause, while in the second, *John* is the object of the subordinate clause in the underlying structure. Indeed, *John* is a more effective prompt word in the first sentence.

The purest test of the derivational theory of complexity may be found in McMahon's dissertation (1963). He asked subjects to push a reaction button as soon as they had judged the correctness of sentences such as the following:

(a) seven (thirteen) precedes thirteen (seven) - *K*
(b) thirteen (seven) is preceded by seven (thirteen) - *P*
(c) thirteen (seven) does not precede seven (thirteen) - *N*
(d) seven (thirteen) is not preceded by thirteen (seven) - *PN*

It appeared that the reaction times increased with the complexity of the transformational structure. Moreover, the reaction time for (d) could be predicted from the reaction times for (a), (b), and (c) from an additive model: $RT(PN) = RT(P) + RT(N) - RT(K)$.

Miller and McKean (1964) reported another much quoted, but at first sight less successful, experiment dealing with hypothesis (2). We refer the reader to Levelt (1966) for a critical analysis of that experiment.

Criticism of this isomorphistic transformational model came slowly but surely in the second half of the sixties. Experimental shortcomings were shown in nearly all the experiments in the series. Foa and Schlesinger (1964) pointed out a number of possible alternative explanations for the results obtained by Mehler. Thus the number of morphemic differences between sentences could determine the nature and number of errors of reproduction. They also indicated methodological errors, such as insufficient control of variables like sentence length and word frequency. For another critique of the experiment, see Howe (1970). Savin's experiments were repeated, but never systematically confirmed.

For critical replications, see Matthews (1968) and Wright (1968). It was Glucksberg and Danks (1969) who proved that, in reality, there was another variable working in this experiment — the time lapse between the presentation and the reproduction of the disconnected words. Reproduction, in effect, begins after the sentence is correctly reported. Depending on the syntax and the length of the sentence, that reporting will take more or less time; the number of words retained is a neat decreasing function of that time lapse. It could be argued that the transformational complexity is nevertheless indirectly expressed by the possibility of reproduction of the sentence. This would be seen in the latency time, that is, the time between the end of the presentation of the sentence and the beginning of the reporting. But these latency times show no simple relation to the transformational complexity of the sentence.

Blumenthal's experiments have likewise often been repeated, with various degrees of success. A problem with all prompted recall experiments is that the two experimental sentences compared are always different. The difference can always be interpreted in a different way than as a difference in deep structure. *Sold* and *poor*, for example, might have a higher degree of association than *sold* and *pound*. Most often totally different sentences are used for the two syntactic conditions, for example, with *children are anxious to play* in the one condition and *Rome is fun to visit* in the other, with *children* and *Rome* as the respective prompt words. The greater effectivity of *children* could then be attributed to a strong association between *children* and *play*. Although nothing in Blumenthal's experiments especially indicates that the main factor is the degree of association, there is nothing to exclude the possibility.

Levelt and Bonarius (1968) eliminated this factor by working with ambiguous Dutch sentences of the type *de studenten zijn te jong om te ontgroenen* which means both 'the students are too young to be initiated' and 'the students are too young to initiate (somebody)'. The two different deep structures were established by presenting the sentences to two different groups of subjects in different unambiguous contexts, so that one group

took *the students* as the subject of *initiate*, the other as its object. When this procedure was followed, no difference was found in the effectivity of *the students* as prompt word for the reproduction of the sentence.[1]

McMahon's results were never seriously challenged, but this whole "verification research" has since developed into a separate branch of cognitive psychology (cf. Clark i.p., Wason and Johnson-Laird 1972; Trabasso, Rollins, and Shaughnessy 1971); in it there seems to be little need for an isomorphistic transformational model. Discussion of this field, however, would take us too far from the principal subject of this book.

More interesting than critical refutations are studies in which systematic deviations from hypotheses (1) and (2) are shown. An early example of such a study is by Fillenbaum (1966). He showed that in deviations from the coding hypothesis (1b) there certainly is interaction between the content of the sentence and the transformational labels. He used groups of four sentences, such as the following:

(a) *the fireman is dead*
(b) *the fireman is alive*
(c) *the fireman is not dead*
(d) *the fireman is not alive*

He asked each subject to memorize one of these sentences, and a number of other sentences. Later the subject was asked to recognize which of the four sentences, (a), (b), (c), or (d), he had memorized. The coding hypothesis predicts confusion between (a) and (c), and between (b) and (d). In each of those pairs, both sentences have the same basic relations, and differ from each other only with regard to the transformation label, and according to hypothesis (1c) this is most easily forgotten. Fillenbaum, however, found that such confusion seldom took place, while the sentences with the same meaning — (a) and (d), and (b) and (c) — were often confused. He concluded that what the subject retains is not the deep structure,

[1] Notice, however, that *the students* is in both cases the subject of the underlying main clause *the students are too young*.

but the gist of the sentence. It should be pointed out that Fillenbaum explicitly instructed his subjects to "try to get the gist or sense of each sentence". When the subjects are instructed to memorize and to reproduce the sentences verbatim, the coding hypothesis is strongly confirmed. Thus Clark and Card (1969) found that with sentences of the type *A is (not)* $\left\{\begin{matrix}better\\worse\end{matrix}\right\}$ *than B*, the confusion between *not better* and *better* is greater than that between *not better* and *worse*. Obviously, the response pattern is strongly dependent on the instructions given to the subjects. The coding hypothesis is only confirmed when a particular method of memorization is explicitly or implicitly offered. That this hypothesis has decidedly no general validity is clear from the following experiment, performed by Bransford, Barclay, and Franks (1972), in which the strategy of memorization is oriented toward the construction of a visual representation. Under such circumstances, obviously, the coding hypothesis is not confirmed. The two following sentences were offered to subjects:

(a) *three turtles rested on the floating log and a fish swam beneath it*
(b) *three turtles rested on the floating log and a fish swam beneath them*

The subjects could not tell afterwards which of the sentences they had heard, although the deep structures of the sentences are quite different. The following sentences were also given:

(c) *three turtles rested beside the floating log and a fish swam beneath it*
(d) *three turtles rested beside the floating log and a fish swam beneath them*

In this case memory was perfect. The representation in the memory is obviously not that of the deep structure, but some nonlinguistic representation of the situation described. Only in the case of the second pair of sentences are there two different imaginary situations; with the first pair, the situation is the same for both sentences. Paivio (1971a; 1971b) pointed out that sentences are retained

in the form of concrete visuo-spatial images and that this imagination factor is controlled in hardly any of the classical experiments. See Kintsch (1972), however, for a refutation of a too facile spatial imagery interpretation of the verbal memory.

Yet a general procedure is given with such experiments for the refutation of the coding hypothesis. In other words, hypothesis (1) can best explain a syntactic effect in situations in which the gist, the visual imagery, etc., is held constant. The coding hypothesis is therefore condemned to be of relatively minor importance in the theory of language perception where the transmission of the gist or reference of the message is precisely the essential factor.

Hypothesis (2) could be correct, even if the coding hypothesis is incorrect. A sentence can be more difficult to understand if its transformational structure is more complicated, regardless of whether or not the hearer reconstructs the deep structure. A series of experiments by Fodor, Garrett, and Bever (1967; 1968) was directed to the elicitation of systematic deviations from the derivational theory of complexity. The subject heard or read a sentence and was asked to repeat it as soon as possible in his own words. The latency time, the time between presentation and reaction, was recorded and used as a measure of the difficulty of the processing. (Notice that in such tasks, a reproduction factor also plays a role, but that role was not investigated.) Thought experiments alone are enough to show that the derivational theory of complexity must err here, as in fact it did in these experiments. Compare the following sentences:

(a) *the red house is on fire*
(b) *the house which is red, is on fire*

Sentence (b) ought to be easier to process because it has a simpler transformational derivation. In the *Aspects* model, prefixed adjectives like those in (a) are derived from relative clauses like those in (b). Sentence (a) is therefore more complex, contrary to a common sense expectation. The simple addition of an adjective to a noun should lead to important increases in the complexity of the sentence: the *Aspects* model gives no less than three transfor-

mations for the generation of a prefixed adjective. In one of their experiments, however, the authors show that the addition of an adjective does not increase complexity. Examples of deletion are still stronger. In the following sentences, transformational complexity increases from sentence (c) to sentence (d), and from sentence (d) to sentence (e).

(c) *John swims faster than Bob swims*
(d) *John swims faster than Bob* (deletion of *swim*)
(e) *John swims faster than Bob does* (insertion of *do*)

But it is clear that sentence (c) is not the easiest to understand; it is, on the contrary, the most difficult. This had already been shown by Fodor, Jenkins, and Saporta (1965). Deletion transformations obviously do serve a purpose.

These and similar examples make isomorphistic transformational theories improbable. The experiments by Fodor et al. were in fact intended to show that although there is some connection between the difficulty of processing and syntactic structure, it is of a completely different nature than the isomorphistic model would suggest. We shall presently go more deeply into the ideas behind these experiments and their conclusions, but we would first point out that these studies are in essence still based on the coding hypothesis. They suppose that at a certain stage the output of the processing of the sentence is its deep structure. Although the two articles of 1967 and 1968 have effectively refuted isomorphism on the microscale, they left isomorphism untouched as far as the major steps in the process of comprehension are concerned. This is what we called semi-isomorphism in the first section of this chapter.

3.5. PERCEPTUAL STRATEGIES

The following quotation from Fodor and Garrett (1967) shows how the derivational theory of complexity was rejected while maintaining the coding hypothesis.

The most profound problem in psycholinguistics is perhaps to specify the nature of the relation between the grammar and the recognition

routine. We have seen that the only a priori requirement upon that relation is simply that the recognition routine must recover the structural descriptions output by the grammar.

It is clear from the context that the authors mean "deep structure" by the term "structural description". But no derivational theory of complexity is needed for this. The idea of the authors is that direct conclusions regarding the deep structure configuration can be drawn from certain properties of the surface structure without need of referring to de-transformations or anything else. Such (hypothetical) processes are called FUNCTIONAL RELATION STRATEGIES. We shall mention a few of these later, but it should first be noted that this point of view supposes that those strategies concern a sentence which has already been segmented to a certain degree by a preprocessor, and that that segmentation is more or less in agreement with the surface structure. Fodor, Garrett, and Bever (1968)

have presupposed as input to the sentence recognition process a representation of the sentence which makes at least a crude segmentation, including the identification of the main verb.

In section 3.3. of this chapter, we have seen that there is not much ground for this supposition, and that segmentation can as well be the output of the processing as it can be its input. Nevertheless it is probable that certain elements in the sentence, and above all its prosody, can give strong indications that some words belong together and that others do not; at various stages in the processing, this can be important information. On the whole, however, this does not maintain that segmentation completely precedes further processing. One could easily imagine that some decisions on segmentation are made only after a schema of functional relations has been composed. Be this as it may, decisions on the segmentation of the sentence are made at some time during the processing. The term SEGMENTATION STRATEGIES refers to the way in which the hearer does this, and to the information on the basis of which he does it.

We shall first mention a number of segmentation strategies

which have been proposed. We shall limit the discussion, however, to a few comments. A systematic treatment of this field is in an advanced stage of preparation (Fodor, Garrett, and Bever i.p.). But at present only a few suggestive references are available in the literature; their significance cannot be judged without a coherent theory (cf. Bever 1970a; 1970b).

SEGMENTATION STRATEGIES. It is supposed that segmentations are preferably of a form which is tuned to the deep structure; in particular, the sentence would be examined on word groups which correspond to the subsentences in the deep structure. Such strategies may be called MAIN AND SUBORDINATE CLAUSE STRATEGIES. Fodor and Garrett (1967) showed that if the relative pronoun is present, it is an important cue for such strategies. In a series of experiments, they proved that the omission of this pronoun — which is often possible in English — makes the processing of a sentence more difficult. Of the following two sentences, sentence (1) takes more time to paraphrase than sentence (2).

(1) *the man the dog bit died*
(2) *the man whom the dog bit died*

The experiments were set up in such a way that the greater transformational complexity of sentence (1) could not be considered as the cause of the difference. The prosody of the sentence can also be an indication of the place at which a subordinate clause interrupts the main clause. Not only Fodor and Garret have found that the processing of sentences like sentence (1) was considerably facilitated when it was spoken with expressive intonation; others also have proven the role of prosody in the identification of phrases. Levelt, Zwanenburg, and Ouweneel (1970), for example, found that ambiguous French sentences such as *on a tourné ce film intéressant pour les étudiants* (which can mean either 'they showed this film, which is of interest to the students' or 'they showed this interesting film to the students') were understood correctly only when spoken with expressive intonation (as opposed to natural intonation). The speaker, therefore, can if necessary

provide the information needed for distinguishing the constituent *intéressant pour les étudiants* as a subordinate clause, and the hearer evidently makes use of that information. Finally, there are various conjunctions which could index syntactic clauses: *but or*, *because*, etc. Little research has yet been done on these.

Smaller phrases should also be recognized as such; this holds in particular for noun phrases. There are indications that noun phrase strategies (NP STRATEGIES) exist. One such strategy for Dutch might function as follows: (i) interpret each occurrence of the following words in the first place as an article (*D*), *de*, *het*, and *een* (these are the three Dutch articles); (ii) check whether *D* is followed by a word which is of category *N*; (iii) interpret the sequence *D* + *N* as a noun phrase *NP*. In experiments using sentences like the Dutch sentences (1) and (2) below, we found that, with visual presentation, sentences of type (1) were always rapidly and correctly paraphrased by subjects, while sentences of type (2) yielded longer latency times and more errors (cf. Keers, unpublished undergraduate thesis 1968, and Stehouwer, unpublished undergraduate thesis, 1969).

(1) *het jongetje merkte dat het vlees lekker smaakte*
'the little boy noticed that the meat tasted delicious'

(2) *het jongetje merkte dat het vlees lekker vond*
'the little boy noticed that he found the meat delicious'

This was possible in the experiment since in Dutch both the singular definite neuter article and the singular neuter pronoun are the word *het*. Introspection as well as the errors made showed that the subjects held strongly to the interpretation of *het vlees* as a noun phrase in sentence (2), instead of interpreting *het* as a pronoun, referring to *het jongetje*, followed by a noun. Let us mention in passing that the detection of noun phrases poses a major problem in the artificial (computer) processing of language; we shall return to this subject in the following paragraph. In this connection, however, we would point out that Brandt Corstius (1970) has developed a program which isolates noun phrases in Dutch

texts. The program is based on a context-free grammar and nearly infallibly marks every noun phrase which occurs in good Dutch. Although the aim of the program is not to simulate human linguistic perception, some of the errors which it makes are typically human errors. Thus, like our subjects, the program misinterprets sentence (2), and obstinately considers *het vlees* to be a noun phrase. It would not be surprising if other noun phrases not recognized by the program would also lead to difficulties for human beings.

FUNCTIONAL RELATION STRATEGIES. Supposing that the hearer is able to distinguish main clauses from subordinate clauses, as well as main verbs, noun phrases, and other phrases, how does he proceed to determine the semantic relations, also called "functional relations" between words? Many possibilities can be imagined, but only little experimental work has been done on the subject. One of the earliest propositions (Fodor and Garrett 1967; Levelt 1967b) was that the hearer can derive parts of the deep structure configuration from the lexical structure of the verb. Such a strategy is called a LEXICAL STRATEGY. We quote from Levelt (1967b):

> There appear to be considerable restrictions on the use of certain words. If one such word occurs, the listener knows at once that the syntactic restrictions in question are realized. If, for example, the word *convince* occurs in a sentence, we know immediately that there must be a *somebody* and a *something* such that *somebody* is convinced of *something*. It is possible that both be explicitly mentioned in the sentence. This, for example, is the case for *John convinces Peter of his error.* The word *convinces* can at once elicit deep structure relations in the hearer, deep structures in which *Peter* and *his error* fit like the keys of a lock. However, it is not necessary that both *somebody* and *something* be explicitly mentioned in the sentence. This is the case for the sentence *convincing is a difficult matter;* yet the word *convincing* here indicates that there is somebody who is to be convinced of *something*. The transformational grammar also indicates these elements in the description of the deep structure of such a sentence. The hearer can interpret the *somebody* and the *something* only on the basis of the context in which the sentence is spoken. The following might be said, for example: *John cannot convince Peter that he is wrong. Convincing is a difficult matter.* Further interpretation of *somebody* and *something* then becomes an easy matter.

Sometimes information also lies in the non-linguistic context. The point here is that certain words directly indicate the existence of certain grammatical relations. The occurrence of such a word in a sentence can be the means for the hearer to decide directly on a particular deep structure.

The earliest experiments on lexical strategies may be found in Fodor, Garrett, and Bever (1968). We shall mention the most characteristic results here. The authors made the non-trivial prediction that of the following two sentences, sentence (1) is more difficult to understand than sentence (2).

(1) *the box the man the child saw carried was empty*
(2) *the box the man the child hit carried was empty*

The sentences differ only in the main verb, *saw* as opposed to *hit*. The prediction is based on the different lexical structure of those verbs. In the lexicon, they have the following subcategorizations:

$$hit\ [+\ -\ NP] \text{ and } see \begin{cases} [+\ -\ NP] \\ [+\ -\ S] \end{cases}$$

Beside its normal noun phrase object, *see* can also have a complement *(I see John walking)*, by which the object of the main clause is itself a clause. *See* can thus occur in more deep structure contexts than *hit*, and it is therefore less informative. When subjects were asked to paraphrase the sentences, the result was that significantly more errors were made with sentence (1) than with sentence (2).[1]

In Volume II, Chapter 4, section 4.5., we gave a dependency representation of case relations. That is also a fitting formalization for the description of lexical strategies. The lexical information for the verb contains the cases with which that verb may be connected. The verb *give* induces the schema (A^*OD) in the hearer, or in other words, the procedure looks in the sentence for an agentive,

[1] Objection could be raised against the use of the word *carried* in the sentences of the experiment. It could too easily be interpreted in the construction with *saw* as a past participle. The same objection could be raised for a verb such as *take*. We would point out, however, that the authors also used sentences other than (1) and (2), and came to the same results.

an objective, and a dative. Parts of this procedure can be performed by testing the sentence for "case-related features". For the agentive, for example, we look for a word with the characteristic [+ animate], etc. We saw in the same paragraph that cases are sometimes marked by prepositions (*by, with,* etc.), or by suffixes. Therefore an efficient strategy would be first to test the sentence for such characteristics, and only later to test them for case-related lexical features. No serious experimental work has yet been done, however, on the perceptual importance of prepositions. Inflected languages often carry case information in the suffix structure of nouns. This holds in particular for Finnish. Levelt and Bonarius (1968) studied the effects of that information in an experiment in which sentences were to be reproduced on the basis of case-marked prompt words.

But words other than verbs and prepositions can also carry direct information relative to the underlying relations. Fodor and Garrett showed that the relative pronoun has more than just the segmentation function which we have seen. For example, expressive ("segmenting") prosody was never as effective for the comprehensibility of constructions such as *the man the dog bit died* as the insertion of the relative pronoun: *the man whom the dog bit died.* The relative pronoun contains specific information on the syntactic relations between main and subordinate clauses which could facilitate the processing of the sentence. The sequence NP_1 + *Rel* + NP_2 + V_t can occur in an English surface structure only if NP_2 (in the example, *the dog*) is the subject of the simple transitive verb V_t *(bit)*, the object of which is NP_1 *(the man)*. This information is lost in sentences where *Rel (whom)* is deleted.

If lexical strategies can provide the hearer with hypotheses on the deep structure configuration of the sentence, the task of filling in the various open spaces in that configuration remains. We have already seen that morphemically realized case characteristics (such as prepositions) can play a role in this, but no experimental work has yet been done on the subject. We also pointed out that case-related features, such as [+ animate] for the agentive function, can be used. In a somewhat broader connection the term SEMANTIC STRATEGIES is used to refer to the

latter because decisions are made on grounds of the semantic characteristics of words. Finally, functions can be assigned to phrases on grounds of their order in the sentence. This is called a WORD-ORDER STRATEGY. We shall first discuss a number of word-order strategies, and then turn our attention to semantic strategies.

A word-order strategy for which some evidence exists involves interpreting every sequence $NP + (Aux +) V + NP$ as agentive — action — object. This, of course, will often be successful, and it seems that we have a natural tendency to do this, as may be seen in the meaningless string *the dur sefted the dat*. The articles are sufficient to make us presume that we are dealing with noun phrases, and the past tense morpheme *ed* leads the observer to the conclusion that he is dealing with a verb. The critical sequence $NP + V + NP$ is thus present. The interpretation is clear, and without any problem we paraphrase the sentence with *the dat is sefted by the dur*. It is even more interesting to examine situations in which the strategy is not applicable or where it would produce an awkward effect. The strategy is not applicable in passive sentences in which the agentive and the objective have exchanged places and the word *by* has been added. The often proven difference in comprehensibility between active and passive sentences could be ascribed to the fact that this strategy cannot be applied to the latter. But it can also account for other phenomena. Of the following two sentences, for example, sentence (1) is easier to understand than sentence (2), as shown in an experiment performed by Mehler and Carey (1968).

(1) *they are fixing benches* (progressive form)
(2) *they are performing monkeys* (participial form)

The strategy works only for the progressive form; in participial constructions it leads to the incorrect conclusion that NP_2 *(monkeys)* is the object of *V (performing)*. The strategy also leads to errors of interpretation with nested constructions. Thus Bever (personal communication) showed that subjects were extremely difficult to convince that the doubly nested construction *the*

editor authors the newspaper hired liked laughed should not be read as an ungrammatical form beginning with the *NP + V + NP* construction *the editor authors the newspaper* (for further examples, see Bever 1970a; 1970b).

The input of semantic strategies consists of case-related, and, in general, semantic features. As we have mentioned, active sentences are generally easier to understand than passive sentences. Of the following, sentence (1) is easier than sentence (2).

(1) *the cow followed the horse*
(2) *the horse was followed by the cow*

Slobin (1966) showed that this characteristic difference does not occur in the following sentences (3) and (4).

(3) *the dog ate the cookie*
(4) *the cookie was eaten by the dog*

The explanation for this in terms of case is that for sentences (3) and (4) the agentive related feature [+ animate] is found only in *dog* and not in *cookie*. The agent is thus determined unequivocally. This is not the case for sentences (1) and (2), where both *horse* and *cow* have the feature [+ animate]. There it will be necessary to use a different strategy, such as the *NP + V + NP* strategy. This fails, however, with sentence (2). Children at an early age give the correct interpretation of sentence (4), while at the same time they do not do so for sentence (2), where either *horse* or *cow* is chosen at random as the agentive (Turner and Rommetveit 1968). Semantic strategies are apparently available earlier than syntactic and word-order strategies.

Semantic factors also play a role in the results obtained by Schlesinger (1966). He proved that of the following two sentences, the doubly nested construction (1) is easier to understand than sentence (2).

(1) *the question the girl the lion bit answered was complex*
(2) *the lion the dog the monkey chased bit died*

The reason for this is that in sentence (1) there are semantic limitations on the case roles of noun phrases. Situations in which girls who bite lions offer answers to questions are rather unlikely, so that the interpretation of the sentence should be straightforward on semantic grounds only. But also we see from this example that there is only a vague distinction between linguistic selection restrictions and that which may be called "knowledge of the world" In fact, that distinction might well be superfluous for psycholinguistic purposes. It is only a small step from the above example to the following two sentences, taken from Garrett (1970).

(3) *the boy chased the dog with a bone*
(4) *the boy chased the dog with a car*

Both sentences are ambiguous, but it is on semantic or, rather, on conceptual grounds that the hearer decides that *bone* in the first sentence belongs with *dog*, and that *car* in the second sentence belongs with *boy*. This touches on the essential question of which knowledge is linguistic and which is not. We shall return to this question in the following section. It should suffice here to point out that ambiguous sentences are particularly appropriate material for the study of the hierarchy of strategies. To this end, sentences could be constructed which have one interpretation in one strategy, and another in another strategy. It would thus be possible, for example, to weigh a semantic strategy against the $NP + V + NP$ strategy on the basis of ambiguous sentences such as *they are lecturing professors*, in which the semantic strategy will yield the participial interpretation *(lecturing* will be taken to modify *professors)*, and the word-order strategy will interpret *are lecturing* as the progressive form. For a rather complete survey of the literature and problems involved in the experimental study of ambiguous sentences, we refer the reader to Flores d'Arcais and Levelt (1970). This concludes our remarks on processing strategies.

Semi-isomorphistic models were given a new source of inspiration in modern work in the field of artificial parsing. We do not refer so much to special purpose programs, such as Brandt Corstius' program for Dutch mentioned above, as to a new style

of general parsing programs which have been so successful that it can be said without exaggeration that the problem of automatic syntactic analysis has been solved in principle. The idea behind these programs lies in the work done by Thorne and his collaborators at Edinburgh. We shall briefly return to this in section 3.6.4. of this chapter. The point here is the establishment of the fact that these programs differ from the original programs for transformational analysis (e.g. Zwicky, et al. 1965) precisely in the same way as semi-isomorphistic psycholinguistic models differ from isomorphistic ones. At first attempts were made to reverse the grammar in the computer; transformational derivations were undone step by step, as in the onion model of the hearer mentioned above. Thorne, however, paid less attention to the rules of the grammar than to the structural descriptions generated by them. His aim was to produce both the surface structure and the deep structure (*Aspects* model), by having the program process the sentence in one run from left to right by analogy with the human observer. The result was what is now known as an AUGMENTED TRANSITION NETWORK, a sort of extended finite automaton (cf. section 3.6.4. of this chapter). Although it was constructed exclusively for linguistic purposes, the strong analogy with the hearer which was one of the aims of the development of the program, resulted in its exhibiting a number of human traits which call for some attention in psycholinguistic investigation. The program, for example, makes only a limited use of memory. In particular, for most words in the lexicon, it contains no information on syntactic categories. Only for articles, prepositions, and other grammatical formatives is the category stored. The others are deduced from suffixes, affixes and word order, just as man does with strings like *the dur sefted the dat*. Moreover, surface and deep structures are derived simultaneously. The fact that this is evidently possible refutes the obstinate supposition in psychology that a syntactic preprocessing, the output of which is a rough parsing, is necessary for an adequate perception model. While this program is an inspiration for psycholinguists and much discussion can be heard on the subject, no publications have appeared in which it is investigated for its

simulation value, and the same holds for the second generation of programs of this type (cf. section 3.6.4. of this chapter).[1]

3.6. CONCEPTUAL MODELS

In this section we shall consider a number of general models of the language user in which no attempt is made to establish direct relations with the grammar. In other words, the models which we are to discuss have never been based on syntactic suppositions because most non-isomorphistic models proceed from the tradition of artificial intelligence — the study of information processing systems. Let us consider the relation between the theory of artificial intelligence and psychological models of the language user. We are dealing here with a completely different approach, which, on the whole, is not an extension of the models in the preceding paragraphs. The theory of artificial intelligence is a general one. Human in-

[1] As this book was going to press, an article by Kaplan (1972) appeared which contains a study of the psychological importance of augmented transition networks. Kaplan makes it plausible that networks can be made which are equivalent to an (*Aspects* type) transformational grammar, and which in their parsing follow strategies such as those described above. The article gives, in particular, a number of functional relation strategies elaborated in detail. The perceptual complexity of a sentence is determined in the model by the number of transitions which the automaton must make in order to accept the sentence.

To the extent that such a transition network works, we return to a strict isomorphistic model; each psychological understanding operation corresponds to a linguistic transition rule. But the situation is quite different from that given in section 3.4. of this chapter. It is no longer the psychological theory which is adapted to the grammar, but rather the grammar which is written for the representation of psychological processing operations. If such a network at the same time provides all input sentences with their correct grammatical parsing, this new isomorphism is of a more acceptable kind than the naïve isomorphism discussed in section 3.4. of this chapter. Quite correctly, Kaplan is careful concerning the possibility of generalizing this approach.

During the translation of this book another article appeared in which augmented transition networks are used in a strongly psychological way. Simmons and Slocum (1972) present a sentence-generating system in which the nodes represent word meanings and paths represent mainly case relations. The model can certainly be considered as a speaker model.

formation processing is, within the theory, a special case, just as processing by computer or other systems. Language usage can be studied from this general point of view; one can develop language processing systems without posing the question as to the extent to which such systems are adequate for human behavior. It is even argued that this method is the most fruitful: if the general problem is solved, it should not be difficult to decipher the organization of a concrete language processing system such as man. We would like to make three remarks on this.

(1) No convincing demonstration of the fruitfulness of this approach has yet been given. Until now we have only seen that no clear boundary can be drawn between general language processing systems and rough computer models of human linguistic behavior. The latter type of investigation comes under the category of "computer simulation research". It aims at having a computer imitate certain forms of human behavior. The development of a general theory of language processing appears to be so basically dependent on what we know about human linguistic behavior, that, for the present, that theory will not go beyond the study of computer simulation, let alone fruitfully produce feedback for it.

(2) The empirical basis of language processing models, even in their reduced form, is quite limited. No more than incidental verification is available for any of the models which will be mentioned in this paragraph, and even this is sometimes lacking. Likewise, it is mostly unclear what kind of empirical results could verify a given model, and authors are only rarely explicit on this point. In the rare cases of experimental testing, additional assumptions, not essential to the model, are nearly always made, and consequently we never know whether it is the model or the assumptions which are being tested.

(3) The fact that man can be described as an information processing system is less enlightening than it seems to be at first sight, despite the commonness of the idea.

In the introduction to this chapter we showed that a general model which is formulated in the language of information processing

systems has the advantage of being able to communicate with concrete experimentation, because anything which can be described explicitly can be described as a computer procedure. Everything which can ever be formulated explicitly concerning human linguistic behavior can thus in principle be put into such a model. While we have also pointed out that this proposition has a solid basis in computing theory, we cannot say that it gives reason for optimism. The situation is no different in our opinion from that which inspired mechanismic philosophers at the beginning of the eighteenth century to their irresponsible optimism. They expected that the future of the universe, including the future of the human mind, would, with the development of natural science, swiftly become predictable. That expectation was based, among other things, on Laplace's idea that the future of the universe lay completely contained in the position, velocity, and direction of movement of all parts of the universe at a given moment. But even if that idea would prove to be correct, no such expectation could be based on it, given the impossibility of determining the position, velocity, and direction of movement of all parts of the universe. The idea of man as an information processing system is no different. It would be just as difficult to program the universal Turing machine which simulates human linguistic behavior as it is to write up all *the details* of raising a child to be a language-using adult. Moreover, such simulation could never be the goal of an empirical theory. A theory should not attempt to imitate reality in all its details; it should rather strive for the strongest possible generalizations on reality in the most economical possible way. It is not at all clear, however, how the metaphor of man as an information processing system can lead to strong generalizations concerning human verbal behavior.

Although computers have played an important role in the computer simulation of human linguistic behavior, it is incorrect to consider computer programs as psychological theories, as was often done in the study of simulation. Theoretical principles can also be formulated independently, and much information processing theory which is explicit and can be tested has never been put

on the computer. The computer program, moreover, is dependent in many details on the computer employed, the compiler, etc. A psychological theory is at least one step of abstraction away from this. In the necessarily concise discussion in this section, we shall avoid computer jargon as much as possible, and limit ourselves to the treatment of a number of general theoretical principles. By doing so, we hope to put the role of grammar in a model of the language user into a larger context, and thus to relativize it.

3.6.1. *General Organization of the Models*

The models which may be found in literature diverge considerably in organization. Some of them have been developed only for certain aspects of language usage, such as the organization of word memory, while others are more general theories which are less elaborate with regard to various details. These more general theories differ enormously in their internal organization. We shall first mention a number of components which occur in some or in all of the theories, and give a rough indication of their function. Later we shall treat some of those components in greater detail.

A trait common to all models of this sort is that their basis is a system of concepts, a CONCEPTUAL BASIS. In the ideal case, this basis includes a representation of the inside and outside world, and that representation is intelligent to the extent that it makes various inferences possible concerning that inside and outside world. In its most general form (and, for the moment, its ideal form), the basis should contain knowledge on the effect of own actions, knowledge on the temporal, spatial, and causal characteristics of the physical environment, as well as a model of the conversation partner — assumptions on his knowledge and intentions. In concrete models, no more than minor and relatively arbitrary portions of this have been elaborated. However this may be, the conceptual basis distinguishes these models from all the more or less isomorphistic models treated earlier. The conceptual basis characteristically is not linguistic in nature.

The other components connect the basis to the linguistic input

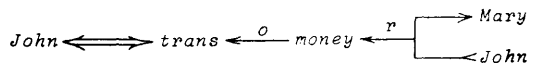
and output. For the analysis of the input, most systems include a smaller or a larger SYNTACTIC ANALYZER. The connection between this analyzer and the conceptual basis generally goes through a SEMANTIC SYSTEM, which at least includes a dictionary in which words are related to their conceptual meanings. The input can also contain nonlinguistic information — perceptions of the internal or external surroundings. This is processed by what we will call the EYE. Finally, the system must be able to respond, either in linguistic or in nonlinguistic form. We shall call these the TEXT-GENERATOR and the HAND, respectively.

3.6.2. *The Conceptual Basis*

The conceptual basis contains the knowledge conveyed by the messages in the language. It is a representation of the internal and the external world. The representation is constructed in such a way that it is possible to draw conclusions concerning this information, to add new information, to answer questions on the information, and so forth. Generally speaking, it is common to all models that at least three units of information can be re-represented in the basis: *objects*, *relations*, and *properties*. The definitions of these three units of information differ somewhat from one model to another, but for the present we can think of objects (ideas) as everything to which a noun phrase in the language can refer. Relations and properties say something about the objects; they are predicates. Relations are predicates over one or more objects (arguments), and properties are predicates which can have only one argument (one-place relations). Properties are usually expressed in the language as modifiers.

Nearly all the theories allow relations to function as the arguments of other relations; relations (including properties), therefore, can themselves be treated as objects. But it is at this point that the correspondence among the various theories comes to an end. There are great differences in the way in which relations and objects are represented in the various theories, but this is not the place to go into this in detail. For an excellent survey of the subject,

see Frijda (1972). In order to show how different these various conceptions can be, we shall only show how the information that John gives money to Mary is represented in the respective theories of Winograd (1972) and Schank (1972). For Winograd, *give* (not the word, but the concept) is considered as a relation (indicated by the symbol #) which can have three arguments (each of which is indicated by the symbol :). The situation is thus represented as (*# give :John :Mary :money*). This is analogous to the way in which an operator grammar (cf. Volume II, Chapter 4, section 4.3.) writes the base forms of sentences. While Winograd considers the concept *give* as a three-place relation, Schank regards it as an object of a particular kind, an action concept. The action to which the word *give* refers is a concept which Schank indicates with *trans*. *John*, *Mary*, and *money* are also objects, and are called nominal concepts. Schank connects the four objects *(trans, John, Mary* and *money)* by means of three relations which he calls "dependency relations". The first relation connects *John* and *trans*; it is an abstract actor-relation which is indicated by ⇔. The second connects *trans* and *money*; Schank calls this the objective case, *o*, indicated by $\overset{o}{\leftarrow}$. The third relation is the recipient case, *r*, which shows from whom to whom the action *(trans)* is directed (from *John* to *Mary* in the case of our example); it is indicated by the fork shown below, in which we give all the conceptual information according to Schank's theory.



Whereas in Winograd's and other systems everything which is expressed as a verb in the language can have the form of a relation (and consequently the number of relations is in principle great), for Schank the number of possible relations is limited to four abstract case relations (objective, instrumental, recipient, and directive), or five, if the actor relation ⇔ is also counted.[1] In

[1] For some reason which is not clear, Schank does not call this a case relation. The reader should, by the way, bear in mind that these are conceptual, not syntactic, case relations.

some systems, however, the number of possible relations is even further reduced; Kempen (1970) allows only one fundamental relation, the inclusion.

Beside the number of relations and the representations of actions (object versus relation representation), the systems differ in the number of arguments which they allow for the predicate. Winograd allows in principle that any number of objects figure in a relation, while Simmons (1970) and Frijda (1972) do not allow more than two objects for each relation (these objects, however, can in turn also be relations).

The kind of information which we have discussed until now can be called SIMPLE INFORMATION. Another, more complicated kind of information, however, should also be specified in the basis — information with quantifiers like "every" or "some", with the logical connectives such as "implies", "and", "or", "if...then", and with negations. It is impossible to identify such COMPLEX INFORMATION in the simple form of relations between objects. Compare the information which is contained in the following sentence: *If some tables or chairs are outside, not all the furniture is inside.* Representation of this demands a formal language such as predicate logic. It is striking to see how some models completely deny information of this kind, or at least neglect it (cf., for example, Schank 1972).

The representation of knowledge in the basis must be intelligent. New information must be easy to add, questions must be easy to translate into the format of that information, and it must be possible to perform various deductive and, if possible, also inductive procedures. In this regard, Winograd (1972) classifies the models according to five categories:

(1) Specific Systems. These are developed only in order to react intelligently to questions on a specific area of information. Many of the older programs were of this kind. The STUDENT program by Bobrow (1964), for example, was good for the solution of what is called "algebraic word problems". These were analyzed and represented as linear equations. The solution of the problem thus consisted of the solution of a system of linear equations.

Another example is Brandt Corstius' program (1970) which includes quadratic equations and does a similar but better job, particularly in the translation of ordinary Dutch into the representation of the base, the system of linear equations. Such programs, however, from the point of view of simulation, are too one-sided.

(2) Systems with a Text Basis. To a large extent the objects and relations in these systems are words, sometimes with an additional formalization. The information is stored, in slightly edited input and output format, and the intelligence of the system resides in an association network of references. A good example of this is Quillian's (1967) program in which the basis consists of a system of dictionary definitions. Each of the words has a list in which is indicated which other words occur in its definition, with mention of the type of relation. The other words, in turn, are themselves heads of definition lists, and so forth, so that a whole network of definitions is present. One can ask the computer what the relation is between two words, and the answer will be constructed by finding the shortest path through the network from the one word to the other and producing the chain of the words concerned in the output. PROTOSYNTEX I by Simmons, et al. (1962) also falls into this category, as does Kempen's (1970) model. The answers given by such systems are in fact mere portions of the network, and not so much intelligent conclusions drawn from the information stored. This is, of course, not a necessary consequence of the fact of having a text basis. But no attempt has ever been made to develop a strong deductive system for a text basis. In other words, it is not known to what extent knowledge can be stored efficiently (from the point of view of storage, deduction, and retrieval) in terms of words taken from the natural language.

(3) Limited Logic Systems. Relations and objects in these systems are stored in a formal language, and not in the form of an English text, but the representation is limited to what we have above called simple information. This can be called a network of elementary predicates. Although the formal languages used for the various models differ somewhat, they all approach that which

is called FUNCTIONAL NOTATION, i.e. the logical notation for propositional functions with zero or more variables containing no quantifiers. Such systems can only process input of the same sort. Their intelligence resides in the subprogram which mediates between the English language input (and output) and the basis, that is, the semantic component. Some of these programs were effective in the translation of a complex English sentence into a series of predicates in functional notation, which in turn could be compared in that form with the available data in the basis (cf., for example, Simmons 1970).

(4) General Deductive Systems. These were developed for the intelligent storage of complex information. The functional notation is extended to a complete predicate logic, and the information in the basis consists of a set of complex predicates. This way of representing knowledge has great advantages. If a question is asked of the system and the semantic component is able to translate the question into predicate logic notation (i.e. as a theorem to be proven), then the answering of the question consists of finding the proof. The axioms for this are the predicates which are stored in the basis. The attractive point here is that there are uniform proof procedures for first-order predicate logic.[1] This means that there are procedures which guarantee that if a proof of the proposition is at all possible, that proof will be found, regardless of the subject to which the proposition refers (Robinson 1965). Such conceptual bases thus have a considerable degree of generalness. Question-answer procedures are completely independent of the content of the question, and are steered exclusively by the logical notation; information on any arbitrary subject can be stored and used, and all the information relevant to the answering of the question will be found. See, for example, Green and Raphael (1968) for applications.

Though the generality of such a basis is a great improvement over earlier systems, it is also true that, with it, the system easily

[1] First order predicate logic with quantifiers is not decidable as such. However, by the introduction of certain limitations to the order of quantifiers in the formulas, decidability can be attained (cf. Kleene 1967).

becomes impractical and "unpsychological". As the conceptual basis grows, a uniform proof procedure becomes terribly time consuming because there is no means whatsoever to limit the search of the basis in a reasonable way. Human beings, however, are apt to search in the right place. The subject of the question usually limits the searching process much more than is logically justifiable, with the consequence that the answer is given in a relatively short time, and with only little chance that relevant information has been overlooked.

(5) Systems with Procedural Deduction. These are not only suited to storing complex information in logical notation; they also give a considerable degree of structure to the information stored. The structure is established in the form of procedures. Every unit of information is represented in logical notation or an equivalent form, just as in a general deductive system, but at the same time it contains directions on how that information can be used. Suppose, for example, that we ask the system whether the Pope is fallible. The question is first translated by the semantic system into logical notation as a proposition to be proven: *fallible (Pope)*. Let us suppose that the system is not well educated, and that neither this proposition nor its negation are available in its conceptual basis. The program must therefore deduce the proposition from the axioms which are present in the basis. Suppose further that the system does know that everything human is fallible. In the procedural system, such information is established as a procedure which says "*for fallible (X), test human (X)*". PLANNER, a programming language for such procedural propositions, was developed by Hewitt (1969). The system deduces *fallible (Pope)* by giving itself a new goal, namely, to find *human (Pope)*. It searches for this in the set of axioms which are characterized by *Y (Pope)*, and let us suppose that it finds the information *human (Pope)*. It immediately comes to the conclusion *fallible (Pope)*. The form of the procedure for information can be much more complicated than in this example. In particular, a whole hierarchy of tests may be needed. If *human (X)* yields nothing, reference can be made to a secondary goal in the proce-

dure, for example, *living (X)*, etc. In the least favorable of cases, a program like PLANNER can still function as a uniform proof procedure, and ultimately miss no relevant axiom. Winograd's (1972) model has such a procedural conceptual basis. It serves for the representation of a world of blocks, boxes, and other physical objects, which can be observed by an eye and manipulated by a hand.[1] The system understands and produces a written natural language. It is possible to hold a conversation on the block world, and the system can execute all sorts of commands. The following is a specimen of conversation:

Q. Had you touched any pyramid before you put the green one on the little cube?
A. Yes, the green one
Q. When did you pick it up?
A. While I was stacking up the red cube, a large red block and a large green cube
Q. Why?
A. To get rid of it
Q. Why did you do that?
A. To clear off the red cube
Q. Why did you clear off that cube?
A. To put it on a large green cube
Q. Why did you do that?
A. To stack up the red cube, a large red block and a large green cube
Q. Why did you do that?
A. Because you asked me to
etc.

In this model the procedural basis proves to be an extraordinarily fruitful approach. The representation of knowledge is extremely

[1] This hand is not in fact realized, but the output of the program includes the proper instructions to the imaginary hand. The eye is nothing other than the input of the initial coordinates and the names of the objects. A logbook of self-initiated changes is kept. The perceptual system is thus limited, or in fact absent.

flexible for the purposes. It does not follow, however, that if the base is expanded considerably, the same degree of flexibility and speed will be maintained. Nevertheless this format of representation is decidedly promising. It should be added that objections can be made to the way in which Winograd represents the block world. From a psychological point of view it is naïve to establish the size and place of objects in the form of Euclidian coordinates, as Winograd does. This unpsychological representation of knowledge makes judgment on the simulation capacity of the model rather difficult (as the study was an artificial intelligence study, simulation in the strict sense was not the aim, but on the other hand it is significant that the article was published in a psychological journal).

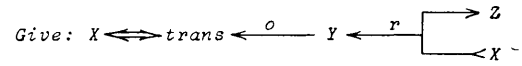
In more general terms, we should state that at the moment there is no domain of knowledge for which a psychologically justfiable formal representation has been developed. There is great need of a "psychological physics" which would define the naïve knowledge of man in his physical surroundings. Take, for example, the sentence *because it was very slanted, the cat slid off the roof.* There is no linguistic reason whatsoever to take the pronoun *it* by preference as referring to the roof (in fact, there are reasons for the contrary). Nevertheless we understand that it was the roof which was slanted and not the cat, thanks to our naïve physical knowledge. The experiment by Bransford, et al., mentioned earlier (cf. section 3.4.) should also be interpreted in terms of such a representation. More particularly, such a naïve physics should describe our knowledge concerning the location of objects (inside, outside, in front of, behind, on top of, etc.), of causality, substance, permanence, etc., and of functions of objects (to put something on, to be sat upon, to cut with, etc.). The foundations of such a study of naïve physics have been laid in the schools of Piaget and of Michotte, but until now such theories of knowledge have neither been formalized nor incorporated into models of the language user.

Still more deplorable is the lack of naïve theory of motivation. How does man represent his own and others' motivations, inten-

tions, actions? At present there is no psychological predicate logic of motivation (see, however, Nowakowska 1973). As long as these fields have not been developed, models of the language user will continue to show important hiatuses.

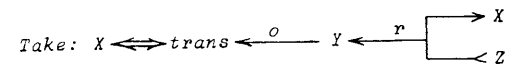
3.6.3. *The Semantic System*

The semantic system mediates between the syntactic analyzer and the conceptual basis. Its purpose is to connect the words in the text correctly to the concepts in the basis. In the first place, a dictionary, or internal lexicon, will be necessary. The dictionary shows how words and concepts are related. The way in which this is done is quite different in each of the models. Schank, for example, defines verbs in terms of the conceptual configurations to which they can refer (possibly more than one for a given verb). Thus we find the following dictionary entry for the verb *give*:



where *X* is *human*
Y is *physical obj.*
Z is *human*

The verb *take* has almost the same dictionary definition; the conceptual action is the same, but the variables have been exchanged:



This shows that it is the conceptual configuration which defines the verb. Schank gives fewer details on the dictionary definitions of nouns, but in any case these definitions contain such characteristics as *human, physical obj.*, etc. In Schank's model, part of the process of understanding is a lexical strategy in the sense used by Fodor, et al. (i.p.). A relational structure is derived from the verb, and further information is fitted into that structure. Although we once again have a system of case and case related features,

it should be pointed out that these are now conceptual and not linguistic entities. Schank takes care to distinguish them from linguistic cases.

For Winograd, on the other hand, words are defined by procedures. Some words, especially function words, are defined by syntactic procedures. Other words, especially content words, are defined by conceptual procedures. A word such as *powerful* is given in the lexicon as "having the property *power*", where power is the concept in the information basis. Semantic relations with other words pass through the procedures which surround this concept in the base. In Winograd's system, thus, the lexicon carries little information in itself. This contrasts with systems with a text basis, such as Quillian's (1967) model. In these there is no distinction between the conceptual and the semantic systems; instead, there is an associative network of word definitions. A real, although unnecessary disadvantage of such representations is their limited deductive capacity. From a purely semantic point of view, on the other hand, one of their advantages is that they often allow an easy simulation of associative relations between words. The system searches for the shortest connection in the network of semantic definitions, and this is not necessarily the most logical relation between the corresponding concepts. Yet, in understanding language we often make use of such jumps in thought which are directed more by the words than by the concepts. In some models, finally, careful distinction is made between words and concepts, but no deep attention is given to the problem of their interaction. This may be seen in the model developed by Rumelhart, Lindsay, and Norman (1972) which was meant in the first place as a model of verbal memory. It inventively combines features of the Schank and Quillian models, but pays relatively little attention to the problems of input and output. Consequently the semantic component is not worked out in much detail.

Beside indicating the relations between simple words and concepts (or procedures), the semantic system must show how word meanings are connected, given a certain syntactic structure. For this it is necessary that the internal lexicon give the grammatical

categories of the words, or that these be derivable in some other way by the syntactic analyzer. On the basis of this, the syntactic analyzer can determine how the meanings of individual words should be connected, and it is this which the semantic system must perform. In Winograd's model, there are separate semantic subprograms for the various phrases: one each for noun phrases (which, in general, correspond to conceptual objects), for prepositional phrases, for adjective phrases, for verb phrases, and for clauses and sentences. The aim of each of these subprograms is the finding of a conceptual representation for the component concerned. For this, it has access both to the syntactic analyzer (if necessary, it can ask for more syntactic details) and to the procedures in the basis. Semantic, syntactic, and conceptual analyses constantly alternate, guided by the priorities in the various procedures. We cannot go more deeply into the details of these programs here.

In Schank's model, this "sentential semantics", as he calls this part of the semantic system, has only a limited role. It combines the words in such a way that corresponding semantic features fit into the spaces of the case network of the verb in question. In doing so, it can use redundancies in the system of semantic features (thus everything with the feature *human* also has the feature *animate*, and so forth). Such redundancies are also used in Winograd's system. When this work of fitting into the conceptual network meets no further conceptual problems (it could, for example, conflict with the foregoing context), the semantic work is completed as far as the sentence structure is concerned.

A final task of the semantic system is interpretation in the light of earlier text or conversation. This has to do, among other things, with the treatment of pronouns and of other words and phrases which can only be interpreted by reference to preceding sentences. Winograd's system keeps a sort of logbook in which the order of conceptual representations is recorded, as obtained in the course of an input text. The order determines the hierarchy of references in the semantic procedures. Many models lack this option, or at least a detailed implementation of it in a computer program.

Before we go on to the discussion of the syntactic analyzer, we must, in this connection, consider the question which was touched upon in the treatment of semantic strategies in paragraph 5 of this chapter: to what extent are selection restrictions linguistic in nature? Are they systematic properties of the combinability of words, or is that possibility of combining words above all determined by nonlinguistic conceptual factors? In conceptual models the latter alternative is always taken. If a conceptual basis is used, the role of semantic and syntactic features appears to be very limited. They serve in the testing of a given conceptualization, but if it does not work, they are ignored. For every selection restriction, one can find exceptions by finding the correct conceptual surroundings. We need no computer simulation to see this. Linguists tell us that the object of *drink* must have the characteristic [+*liquid*]. But a sentence such as *we drank a delicious cup of molten iron* is quite as ungrammatical as *we ate a whole plate of tea.* The fact that these linguistic selection restrictions are satisfied does not guarantee grammaticality any more than ungrammaticality is guaranteed by the fact that they are violated. The ungrammatical sentence *he drank the whole dish of ice cream* (violation of the liquidity restriction) is completely acceptable if it is clear from the context that the ice cream had melted. Should *ice cream* therefore be given two alternative characteristics in the lexicon [+*solid*] and [+*liquid*]? That would mean removing the content of the concept of selection restrictions. The point is, of course, that we are dealing with a conceptualization of *drinkability,* usually absent in molten iron, and sometimes present in ice cream. In neither of the cases is it a systematic characteristic of the word; rather, it is only a semantic characteristic of a conceptual situation. It is better, therefore, to consider selection restrictions as useful linguistic summaries of nonlinguistic conditions. Likewise it is often better to consider subcategorization as a rough conceptual categorization than as a strict linguistic characteristic of words. The distinction, for example, between count and mass nouns *(bicycle* as opposed to *water)* is in fact a conceptual distinction within "naïve physics". It is indeed the case that

syntactic characteristics are connected to enumerability, but enumerability itself is not bound to the words themselves, but rather only to the concepts to which those words refer in a given context. If there is great regularity in the reference, one can, as it were, short circuit the theory by treating the subcategorization as a grammatical property. But in this case, a principle called the *principle of diminishing returns* by Lyons (1968) comes into play: each grammatical refinement concerns a smaller group of sentences, and at the limit each new sentence which is not completely justified by the grammar will demand the introduction of an idiosyncratic rule. The reason for this is that selection restrictions and some or all of the forms of subcategorization ultimately go back to properties of concepts of which grammatical categories are only rough summaries. Language is not a closed system.

3.6.4. *The Syntactic Analyzer*

The emphasis on syntactic analysis varies considerably from model to model. Quillian (1967) gives little room to syntax in his model, and hardly any syntactic analysis is performed in the understanding of a sentence. The input words lead directly to the activation of nodes in the basis, and the model finds the shortest path between the words. This is usually sufficient for the construction of an answer: "a generative grammar need not be in any sense a "component" of the underlying language processor" writes Quillian (1968). Only for the formulation of the answer does he consider some grammar necessary. For this, in his opinion, a kind of transformational component must be built in, but there is little or no need of a base grammar. For a model with a basis such as Quillian's (1967), this is a trivial conclusion because the sentences are nearly ready-made available in the basis, and syntactic rules are necessary only for the production of transformational variants of those sentences. Problems do arise, however, in more realistic models with more abstract bases.

Some of these models are almost exclusively oriented to the understanding of sentences. Their production capacity is limited, e.g.

to the giving of yes/no answers, numbers, or simple cliché sentences. This holds for Raphael's (1968) SIR, Bobrow's (1964) STUDENT, Weizenbaum's (1966) ELIZA, and other early programs. All of these models practically did without syntactic analysis. In the circles dealing with simulation at that time, the impression reigned that a syntactic component was more or less superfluous. Since 1969, however, a clear reaction to this can be observed. As the conceptual basis of the models moves farther away from text and less specifically concerns a particular subject (such as linear equations) and at the same time more attention is paid to flexible text production, there appears to be a growing need for a rather extensive syntactic analysis. It is on this point that the new developments in the field of automatic syntactic analysis, which we have already mentioned, is beginning to bear fruit.

Thorne's analysis program has since been followed in the work of Bobrow and Fraser (1969), Sager (1967; 1971 — this is probably a completely independent development), Winograd (1972), and Woods (1970). Woods calls the analysis model an augmented transition network. This is essentially a finite automaton which is augmented in the following sense. State transitions are caused by the input not only of terminal elements, but also of nonterminal elements. To determine whether a given nonterminal symbol is present at a given transition, a subroutine is used, which, again as a finite automaton, determines whether the following portion of text belongs to the category in question. When this subautomaton reaches a final state, it is reported that the original transition can be made. In this form, such an automaton has the capacity of a push-down automaton. All the programs mentioned, however, call for further conditions on the execution of a transition, by which, certainly at least for Woods' and Winograd's programs, the capacity of a Turing machine is obtained. The quasi-finite automaton at the heart of these programs guarantees that the sentence is scanned once from left to right: at each new word there is a limited number of syntactically possible continuations, and the subroutines test which of these continuations is in fact present.

The augmented transition network approach is neutral with respect to the grammar used. Every explicit grammar can be implemented on it. Thorne's original work was based on an *Aspects* grammar. Sager's program, which is the most detailed, proceeded from an adjunct grammar, while Winograd's syntactic analyzer is based on Halliday's systemic grammar. The differences among grammars become considerably less conspicuous when the grammars are written in the form of analysis programs. The origin of Woods' syntax, for example, is difficult to determine. When such a syntactic analysis is integrated into a conceptual model, new problems, of course, occur. It would not be wise, for example, to have the syntactic analysis strictly precede the semantic and conceptual analyses. This integration is executed in detail only in Winograd's model. In that program the syntactic analyzer can call upon semantic and conceptual subroutines at any time in order to solve syntactic problems, or to test analyses on their conceptual tenability. According to Winograd, it is better to consider an augmented transition network more generally as a program which must perform particular subroutines for each transition. The subroutines may be of a syntactic kind, or they may lead to calling upon semantic or conceptual procedures.

3.6.5. *The Text-Generator*

A syntactic component is indispensable to a system which must produce a text. This, however, is the Achilles' heel of most conceptual models, for the text generated (in all cases it is in the form of computer print) is always simple or trite. Of course, fixed expressions and the filling in of standard schemas should play a certain role in every simulation program. *I don't know*, *what do you mean*?, *which X*? are patterns which the human language user also has ready at hand. But more than this is needed. Winograd's program can form not only such fixed expressions, but also new answers to *why*, *when*, and *how* questions concerning the block world and its manipulation. We have already given an example of a conversation thus produced. A *why* question is

answered by calling upon the consecutive aims which have led to the answer or the action to which the question refers. We have seen how the program would answer the question *Is the Pope fallible?* If after receiving the affirmative response, we ask *why?*, the intermediate aim *human (Pope)* is called upon, and the answer becomes *because the Pope is human.* For this, of course, the program must be able to translate conceptual information into natural language. In fact Winograd's program can only speak about the block world. It contains more or less ad hoc schemas for nominalization and the formation of sentences in dependency on the question asked. Some style is added to this by the introduction of pronouns where a given object has already been mentioned in the sentence, or by using a number and a plural form when more than one object of the same kind are referred to *(three cubes*, instead of *a cube and a cube and a cube)*. However there is still no general system according to which information in functional logical notation, or an equivalent form of it, can be translated into natural language.

3.6.6. *The "Hand"*

The hand need only be discussed briefly. Motor programs are lacking in all models except that of Winograd, but, as we have seen, even there that component is based on a psychologically naïve representation of a block world. The motor program translates instructions such as *grasp (X)* or *get rid of (X)* into three-dimensional movement instructions. It is only worth mentioning here that also in this case the program can generate subtasks if a particular action cannot be performed. If the instructions are *grasp (X)* while *X* is under *Y*, the subtask *get rid of (Y)* is first generated (compare with the conversation given in section 3.6.4.). Unfortunately studies of the relationships between verbal instructions and motor actions are not only lacking in artificial intelligence work, but also in psycholinguistics this is still a virginal subject.

3.5.7. *The "Eye" and the Theory of Pattern Grammars*

None of the models contains a nonlinguistic perceptual component. As we have already seen, in Winograd's program this is only mentioned *pro forma*. If someone were to manipulate the block world, it would not be noticed. The only input is the original position coordinates of the various objects, and changes executed by the model itself are recorded in a logbook. It is a bit like the eye of a blind man who walks on familiar ground. Naturally it is necessary that nonlinguistic visual input should lead to modifications in the representation of the surroundings in the conceptual basis. The problem of how such information can be deduced has not yet been studied in natural language programs.

It may be expected, however, that in the near future attempts will be made to integrate the rapidly developing theory of pattern grammars, and more generally, work in the field of scene analysis, into models of the language user. What is a pattern grammar? Just as the linguistic grammars treated in this book generate sentences, pattern grammars generate n-dimensional patterns. The simplest case is a one-dimensional pattern with black and white squares as its elements. The hierarchy of formal grammars for such patterns is precisely the Chomsky hierarchy discussed in Volume I. The terminal vocabulary consists of two elements, a and b, where a is interpreted as "black square", and b as "white square". If other shades or colors are admitted, these will correspond to further elements in the terminal vocabulary. The grammars describe how the black and white (or other) squares can be placed in sequence. Wagenaar (1972) discusses regular patterns of this kind, and a number of their psychological properties. Just as with sentences generated by a linguistic grammar, the pattern grammar assigns a structural description to each pattern.

The pattern can also be expanded to two dimensions, that is, we can give rules for horizontal and vertical arrangement of the elements. The grammar will then generate two-dimensional patterns.

Seen from the point of view of patterns, pattern grammars can

best be subdivided according to the following three characteristics: (1) the definition of the terminal elements, (2) the number of dimensions, and (3) the place in the Chomsky hierarchy (if there is one for the given class of patterns).

In the just given example, the terminal elements were squares. Grammars with such elements are called ARRAY GRAMMARS or MATRIX GRAMMARS. The production rules are rewrites of arrays, configurations of black, white, or other colored areas. There is a Chomsky hierarchy (regular/context-free/context-sensitive) for two-dimensional matrix grammars, cf. Chang 1971).

It is also possible to take any other form, in n dimensions, as the terminal elements; these can be lines, ovals, angles, or anything else. In order to give the rules of arrangement for such elements, one can define points of connection in each element. There may be two points (Shaw 1969), or an indefinite number of points (Feder 1971). Such grammars are called PLEX GRAMMARS, these also define a Chomsky hierarchy of patterns.

The notion of "point of connection" can also be generalized. Each terminal element can be provided with a system of coordinates. In the rewrite rules, one can indicate at which points a terminal element is connected to other elements. Such grammars are called COORDINATE GRAMMARS (Anderson 1968). As a consequence of the properties of continuity of these grammars, they do not define a Chomsky hierarchy.

Finally, there are pattern grammars in which the terminal elements are arbitrary labeled graphs. These are networks of nodes and the connections among the nodes. Nodes and connections can differ from each other in various respects. The nature of each is indicated by a label. Such a graph may be called a "web", and the corresponding grammars may be called WEB GRAMMARS (Rosenfield and Pfaltz 1969). Plex grammars are particular cases of web grammars. There is a Chomsky hierarchy for two-dimensional web grammars.

If a Chomsky hierarchy exists, one can also expect that recognition automata can be constructed. In other words, it should be possible to construct systems which can accept given classes of

patterns, and to reconstruct the corresponding structural descriptions while doing so.

How can pattern grammars be used in a model of the language user? This question takes us to the more general problem of scene analysis. Scene analysis deals with the inferences which can be made from a two-dimensional picture as to how a three-dimensional layout or scene may be. In general it is not possible to map pictures and scenes in a one-to-one fashion. On the one hand, not all pictures can be mapped on a scene (under reasonable restrictions). Examples are the Devil's fork, Penrose's figure, and some of the famous Escher graphics. On the other hand, most pictures correspond to an infinity of possible solid objects. Numerous examples of this have been given by Ames. If the domain of three-dimensional scenes is well-defined and, by experience or otherwise, sufficiently restricted for the language user, he might be able to make intelligent and quick inference from the pictures to plausible scenes. Picture grammars may enter the analysis of such processes by specifying the well-formedness conditions on pictures, i.e. by generating the language of possible two-dimensional pictures, while providing them with structural descriptions which are three-dimensional representations. As in natural language understanding programs, such grammars might be helpful in formulating the parsing heuristics, although, once again, one should not expect such parsing programs to be grammars-in-reverse.

In fact, as in Winograd's language parsing program, one should expect intelligent parsing of pictures to be strongly semantic. Possible three-dimensional representations should agree with available conceptual information. If the subject expects that the object is a table, he might try a representation where there is a flat horizontal surface. If this is successful, the conceptual base might then decide that the object is fit to support something, to work at, etc., because such functional properties are part of the definition of a table. For an excellent review of intelligent picture processing, see Sutherland (1973).

Dynamic structural descriptions likewise must undergo semantic

interpretation. The work performed by Michotte (1964) in this field has been exemplary. It is based on certain sensory movement relations between objects and shows how these are interpreted perceptually. Under certain circumstances a characteristic structural description comes into being; Michotte calls this an AMPLIATION. This is then conceptually represented as a form of causality. Minor sensory variations can influence the perceptual structural description considerably. Thus we obtain the class of perceptions which we indicate with descriptions such as "object *A* pushes object *B*", "*A* launches *B*", "*A* releases *B*", "*A* obstructs *B*", and so forth. All of these words express conceptualizations, in which the property "causal" is an integral part of the relation between *A* and *B*. They differ in additional characteristics.

Of course the "eye" is intended to do much more than interpret visual patterns. We have used the word for all forms of sensory input and introspection. In particular, the perceptions of the intentions of oneself and of another fall into this category. This field of knowledge, however, is a blank page from a formal point of view, and even a vague prediction of future developments would lack any foundation in reality.

3.7. GRAMMARS AND MODELS OF THE LANGUAGE USER

Let us return to the principal theme of this chapter, the question as to the relation between grammars and models of the language user. This question has two sides. The first of these is the question concerning the place of a linguistic grammar in the theory of the language user. The second is the more general question of the applicability of the theory of formal grammars to models of the language user.

Let us first consider the linguistic aspect. In the beginning of the present volume, we discussed Chomsky's conception of the coincidence of linguistic grammar and the creative linguistic capacity of the language user. For those who hold that point of view, the linguistic grammar is necessarily an integral part of a

model of the language user. We showed that the empirical basis on which linguistics rests makes it highly improbable that this conception is correct.

The original form of the study of the psychological reality of grammars, as we have seen, was based on the assumption of isomorphism between linguistic rules and psychological operations. This isomorphism is not a logical conclusion from Chomsky's proposition, although Chomsky highly encouraged the development of such an isomorphistic psycholinguistics. This line of research clearly showed that if the grammar should be part of the model of the language user, this will certainly not take on the form of isomorphism. Isomorphistic models, regular as well as context-free ones, proved to be useful only for the analysis of rough statistical aspects of human linguistic behavior. Moreover cases constantly occurred in which the process of understanding was obviously directed by nonlinguistic factors, such as visual representations and knowledge of the situation.

This brought us into contact with a completely different tradition in which the basic assumption had always been conceptual: the study of artificial intelligence and computer simulation. In spite of the hardly surprising fact that at the present stage of development no model in that tradition simulates the human language user very well, the models cast a new light on the question as to the place of the grammar in a model of the language user. In fact the answer to the question is rather independent of the actual success of the simulation. That success is namely highly dependent on the degree to which the various components of the model are elaborated in detail, while the question is mainly concerned with the nature of the relation between those components. It was by way of the general approach from the point of view of information processing systems that the linguistically inspired approach was brought back into perspective. Models in this line are all characterized by their conceptual intelligent basis. The messages in the language concern a nonlinguistic system of concepts. As soon as one introduces such a conceptual system, however (and no reasonable psychological objection can be brought against doing

so), the role of the grammar, semantics included, becomes less extensive than was suggested in the linguistic tradition. In the first place the creative linguistic capacity of the language user is shifted in an obvious way to the conceptual basis. It is there that the information is combined in order to make various inferences possible, and it is there that new concepts come into being. Only in the second place can this newly formed information be expressed linguistically. The creative aspect of human language is largely accounted for in these models by a nonlinguistic subsystem.

To what extent, then, is the grammar still a necessary component? There are as many opinions on this point as there are models. At first there was a tendency to minimalize the role of grammar, but we could ascribe that either to the text basis which was sometimes chosen, or to the limited conceptual domain of some models. In the newer models, the need for syntactic and semantic analyses increases with the level of abstraction and the size of the conceptual systems. When the basis becomes less linguistic, the road from linguistic input to conceptual representation grows longer. Winograd's model includes extensive syntactic and semantic analysis programs, and it appears that these will have to be expanded even further if the conceptual basis is to be given a more realistic capacity. Moreover it is not known how much more syntax will be needed in order to provide the output of the model with any more than extremely meager possibilities. One can therefore expect that grammars will indeed go on being an important part of such models. Four qualifications should, however, be added to this.

(1) Grammars in such models are not isomorphistic with linguistic grammars. No one-to-one relation with the rules of the grammar may be seen in the parts of either the analysis program or the synthesis program.

(2) The model grammars are also not equivalent to linguistic grammars. The set of sentences accepted includes the grammatical sentences, but it also contains much more. In other words, the model grammar is less detailed than the linguistic grammar. The reason for this is clear. The criterion for acceptability of the sentence no longer resides in the possiblity of syntactic-semantic

analysis, but rather in the possibility of finding a conceptual representation for it. Thus we have seen that the whole mechanism of selection restrictions, or at least an important part of it, need not be defined in the linguistic components of the model. One can, of course, by the same right, argue that such things are equally out of place in linguistic grammars in the first place (this is in fact what Harris does: cf. Volume II, Chapter 4, section 4.3.), but this idea is not widely held.

(3) Even when such an internal grammar is available, only a minimal use is made of it in the comprehension of many, if not most, sentences. As Schank emphasizes in his model, it will often be possible to go directly from individual words to conceptual representation. Only when ambiguities occur will further syntactic analysis be needed. The idea is that, in general, the internal lexicon, together with the deductive capacity of the conceptual basis, will be sufficient for the comprehension of sentences. The use of grammar will naturally be more intensive in the production of speech or text. But here, too, it will generally not be necessary that the grammar exercise control over selection restrictions, and so forth. The conceptual input in the text-generator will tend to be such that that kind of restriction will be satisfied automatically. And if this is not the case, then there is apparently good reason for the violation of such restrictions.

(4) As we have already pointed out, it is not the internal grammar which accounts for the creative aspect of human language, but rather the intelligent conceptual basis.

These four qualifications are not only statements of fact concerning computer simulation models, but, given the development in psycholinguistic research in recent years, they can also be considered as a realistic evaluation of the relation between linguistic grammar and human language usage, and consequently, as the most important conclusions of this chapter.

Regarding the role of formal grammars in models of the language user, we can conclude that a considerable nonlinguistic expansion may be expected in the future. We have seen that the input and output of each component of a simulation program are

formal expressions, and even without computer simulation, a theory of the language user will have to give explicit formal representations for the coding of the information in the various components. Such coding systems are formal languages, and their analysis is a part of the theory of formal grammars. But there is still a great distance between the available formal languages whose mathematical structure is known, and the psychologically attractive coding systems, whose formal structure is not well known. The mathematical structure of predicate logic, for example, is well known, but it is hardly clear how suited that formal language is for the representation of conceptual information. On the other hand, the mathematical properties of perceptual coding systems which were developed from empirical work in psychology (see, for example, Leeuwenberg 1971) are not sufficiently known, and no parsing programs have yet been developed for them. Consequently, it is not possible at present to connect them to a linguistic or a conceptual subsystem in the theory.

In summary, then, we see that on the one hand the role of linguistic grammars in models of the language user is diminishing in a way, although it is decidedly not about to disappear, and on the other hand, the theory of formal languages and grammars appears to be of increasing importance for the nonlinguistic aspects of such models.

4

GRAMMARS AND LANGUAGE ACQUISITION

4.1. ASPECTS OF LANGUAGE ACQUISITION

The extent of the theme "grammar and language acquisition" depends on what is understood by the term grammar. If grammar is taken in the limited syntactic sense of Chomsky's *Syntactic Structures*, the theme concerns one interesting aspect of language development — the growth of sentence structure. As it is quite impossible to study as a whole the development of language in the child, it would by all means be acceptable to limit one's research to one important aspect, such as syntax. But this may be done, as we have seen in the preceding chapter, only on condition that one does not lose sight of the whole, for the development of syntax is not a closed system, but part of the total cognitive growth of the child. The expansion which the notion of grammar has undergone in the direction of semantics since about 1965 has recently begun to bear fruit in the study of language development. The stiff interaction between syntactic theory and developmental psychology which could be observed for a few years, is now giving way, thanks to a growing semantic interest, to a more integrated psycholinguistic approach to language development.

In the study of language development, however, the period from 1963 to 1968 was characterized by a strong accent on syntax. Syntactic grammars were written for the various stages of language development in the child between the ages of one and a half and three years, the period in which sentence structure develops most strikingly. Research was much inspired by the theory of

formal grammars, both directly and through its applications in linguistics. As this is the subject of this book, we shall spend the first half of this chapter on a discussion of that interaction. In section 4.2. we first treat Chomsky's HYPOTHESIS-TESTING MODEL which stems from the theory of formal languages; some of its consequences can be evaluated in the light of later developments. In section 4.3. we shall discuss the concretization which the model underwent in the light of developing transformational grammar. This can be called a RATIONALISTIC acquisition theory, as opposed to an EMPIRICIST theory. Although most researchers pointed out at some time that syntactic development is only a part of general cognitive growth, this seldom amounted to more than lip service. There was in fact a tacit assumption in studies inspired on transformational grammar to the effect that grammatical development can be described as a rather closed system. Around 1968 this attitude began to change gradually. The question as to the origin of universal grammatical forms and their systematic development in children's languages could no longer be put off by the simplistic label "inborn" or derived forms of it. It became clear that at least two other points of view were necessary for an explanatory theory (in the sense of Volume II, Chapter 1, section 1.2.). The first of these concerns the general possibilities and limitations of the child's information processing (perception, memory, etc.). We shall call these PROCESS FACTORS. In section 4.4. we shall give a few incidental examples of these. The second point of view is that of the intention of the sentence; here one often speaks of SEMANTIC FACTORS. In the final paragraph of this chapter, section 4.5., we shall discuss a number of aspects of this, under the more general heading of CONCEPTUAL FACTORS. Semantic analysis of a child's language is impossible without a study of how the system of concepts — called the "conceptual basis" in the preceding chapter — develops in the child. In this discussion there will be no pretention to completeness, and the choice of topics for discussion will once again be determined by the formal language theme of this book.

4.2. LAD, A GENERAL INFERENCE SCHEMA

This schema was first described by Chomsky (1962). After mentioning that the child acquires the language of his environment in a strikingly short time and without special training, Chomsky proposes that linguists and psycholinguists consider the construction of a formal system, which, given the same input as the child (a set of linguistic utterances), will produce the same output as the child, a grammar of the language. He calls this a language learning device, and in later publications renames it a LANGUAGE ACQUISITION DEVICE, or simply LAD. In Chomsky's opinion, if such a language acquisition device is successfully constructed, its structure would be an hypothesis on the innate intellectual capacities which the child uses in the learning of a language.

For the construction of LAD, we must have at our disposal a linguistic theory which defines the class of possible grammars from which the child, and therefore also LAD, can choose. Such a theory must be a general characterization of natural languages, or a theory of linguistic universals. In terms used in Volume I, Chapter 8, this is the HYPOTHESIS SPACE of LAD.

One then must consider the input of LAD. Chomsky names a number of possibilities: LAD might only be given positive instances (called TEXT PRESENTATION in Volume I, Chapter 8), or both positive and negative instances (INFORMANT PRESENTATION), or even pairs of positive and negative instances (which might be called CORRECTIONS). Other input might also be required, such as meaning, but Chomsky (1962) rejects this as being probably irrelevant:

> For example, it might be maintained, not without plausibility, that semantic information of some sort is essential even if the formalized grammar that is the output of the device does not contain statements of direct semantic nature. Here, care is necessary. It may well be that a child given only the input of (2) [LAD] as nonsense elements would not come to learn the principles of sentence formation. This is not necessarily a relevant observation, however, even if true. It may only indicate that meaningfulness and semantic function provide the motivation for language learning, while playing no necessary part in its mechanism, which is what concerns us here.

And Chomsky repeats essentially the same argument in *Aspects* (1965, p. 33). Syntactic input of a certain composition is for LAD what was called OBSERVATION SPACE in Volume I, Chapter 8.

LAD must also dispose of an EVALUATION PROCEDURE. This is needed in order to be able to choose from among those grammars in the hypothesis space which are capable of accounting for all observations. The evaluation procedure must lead to the choice of an optimal grammar. What the criterion for this must be is not defined by Chomsky, but he suggests that it will have to do with the intuitive adequacy of the structural descriptions which a grammar assigns to sentences. A condition for evaluation by LAD is that a mechanism be included by which, given a grammar G and a sentence s in $L(G)$, the structural description of s can be derived in terms of G.

Chomsky and Miller (1963: 276-277) add to this that LAD must also be provided with HEURISTIC PROCEDURES, by which a number of promising grammars in the hypothesis space can be evaluated rapidly and in greater detail. Such procedures should comprise a set of intelligent induction methods which can accelerate the learning process. But Chomsky and Miller also point out that an a priori limitation of the hypothesis space is much more favorable to a rapid learning process than a large battery of heuristic methods:

> The proper division of labor between heuristic methods and specification of form remains to be decided, of course, but too much faith should not be put in the powers of induction, even when aided by intelligent heuristics, to discover the right grammar. After all, stupid people learn to talk, but even the brightest apes do not.

And in *Aspects*, where the theory is once again summarized, we read,

> This requires a precise and narrow delimitation of the notion "generative grammar" — a restrictive and rich hypothesis concerning the universal properties that determine the form of the language.

Basically LAD is nothing other than a schema for the analysis of the language acquisition situation. It makes explicit the relations

among a priori hypothesis space, the nature and extent of observations, and the deductive capacity of the child, which together lead to the discovery of the grammar. Chomsky correctly indicates the need to choose between an hypothesis space a priori limited on the one hand, and a strong combination of observations and intelligent heuristics on the other. In *Aspects*, he relates that choice to two characteristic conceptions which can be found in literature on language learning; he calls these "empiricist" and "rationalistic". The empiricist point of view, in terms of LAD, is that there is hardly any limitation on the a priori hypothesis space, and that LAD includes strong heuristic principles by which, with any input, it can ultimately deduce the grammar. Within this framework much remains to be said on the nature of these heuristic principles. One might, for example, imagine general association and generalization principles, and so forth, but that is beside the point here. For the empiricist point of view, the point is that there is no, or very little, limitation on the hypothesis space with which the child begins his language acquisition. In other words, LAD would then contain no linguistic theory worth mentioning. The rationalistic conception, on the other hand, sees LAD's power in a restrictive a priori hypothesis space; here the heuristic principles become relatively unimportant. Here no inductive effort, no difficult learning process, is needed. According to Chomsky (1965:51), already Humboldt held the view that

> one cannot really teach language, but can only present the conditions under which it will develop spontaneously in the mind in its own way.

Chomsky then states that the essential question in comparing these two points of view is the question, mentioned above, concerning the possibility of constructing LAD, or, as he calls it, the ADEQUACY IN PRINCIPLE of the theory of language acquisition. Only when that question is answered, either in an empiricist or in a rationalistic sense, can we think of how the acquisition algorithm of LAD should be composed in order to have properties comparable to those of the child's language acquisition, especially as far as the REAL TIME ASPECTS of the model are concerned. He

then informs us in *Aspects*:

> In fact, the second question [as to the *real time* algorithm] has rarely been raised in any serious way in connection with empiricist views... since study of the first question [the adequacy in principle] has been sufficient to rule out whatever explicit proposals of an essentially empiricist character have emerged in modern discussions of language acquisition.

This, however, is a highly one-sided statement. In 1965 the question of the feasibility of language acquisition algorithms had not yet been solved, and it became evident from the later work by Gold and others that it was not only impossible to construct an empiricist LAD, but also a rationalistic one.

Before we go into this, however, we should point out that Chomsky's schema of analysis, as well as his own rationalistic position within it, were highly stimulating to the study of language acquisition in the 1960's. On the one hand, linguists once again realized how important it is to develop a general linguistic theory in close interaction with the study of language acquistion. On the other hand, developmental psychologists began consciously to search — at first, not without success — for the early universals in the language of the child. In these the hypothesis space of LAD should be reflected in a direct way. In the following paragraph we shall discuss the way in which general linguistic theory and language acquisition theory are connected *in concreto*. For the present we shall only say that, thanks to the LAD model, renewed attention was paid to the study of the relations between linguistic universals and early language acquisition. One could also observe a growing interest in the nature of the input, the observation space of LAD and therefore also of the child. Interesting new attempts were made to find out what the linguistic environment of the child really consists of, and to discover the basis on which the child modifies his language. We shall also return to this later. Finally, research employing various experimental means was performed on the structural descriptions which children implicitly assign to sentences. Several new techniques, such as the use of play-acting and imitation, were developed for this.

This research did much to awaken interest, but, rather re-

markably, no one attempted to answer Chomsky's basic question as to whether LAD could be constructed in the first place. That possibility was simply taken for granted. Braine (1971) was the first psycholinguist to examine the LAD schema as such, and to declare it intrinsically impossible as a model of language acquisition. This whole development took place entirely without contact with the parallel research which was being done on grammatical inference, which we have discussed in Volume I, Chapter 8 (to which should be added that those who performed that research were in their turn rather aware of the psycholinguistic problems). Braine's treatment and rejection of the LAD model could, in fact, have been more complete on the basis of the results obtained in that field. In fact, in his discussion Braine confused the LAD schema itself with its rationalistic pole, as we will show presently.

Let us first have a closer look at the question of the possibility, in principle, of constructing LAD. LAD must do two things: in the hypothesis space it must find a subset of weakly equivalent grammars for L, and on the basis of an evaluation procedure it must decide which of those grammars is the best.

Gold (1967) — cf. Volume I, Chapter 8 — deals only with the first problem, whether an observationally adequate grammar for L can be found in the limit. As we have seen, he tried to determine whether it was possible to construct a Turing machine for various input conditions and language classes, by which the language is learnable in this weak sense of the word. This is precisely Chomsky's basic question, although the application of Gold's conclusions is too optimistic in a number of respects. In the first place, if LAD is to be a theory of the language-acquiring child, it can never be a Turing machine. We could temporarily get around this objection by supposing that the finite properties of the child do indeed lead to characteristic human errors in the learning process. In the second place, Gold had to suppose that the presentations are complete, that is, that every positive and negative instance (every string over V_T^+ for informant presentation, and every sentence in L for text presentation) appears within a finite amount of time in the information sequence. This assumption will become more

important when the secondary question of the real time properties of the system will be treated. However this may be, Gold's assumptions are more tolerant than can be allowed for LAD, and conclusions concerning LAD must therefore be at least as strong as Gold's results.

Table 8.2. in Volume I shows that only finite languages are learnable by means of text presentation (with only positive instances). And for finite languages, the only possible algorithm is based on the idea that the whole language is observed in finite time. If one wished to maintain that for the learning situation natural languages can be considered as finite sets, then there could be no doubt as to the extremely large size of the language, and LAD would consequently have impossible real time properties. So, for text presentation, LAD cannot be constructed by any schema whatsoever, be it empiricist or rationalistic.

Informant presentation is much more favorable. There is a learning algorithm in principle even for the class of primitive recursive languages. Is this encouraging for the possibility of constructing LAD? There are two reasons for answering this question in the negative. The first of these reasons is of limited significance, but as a programmatic question, it is worth considering. In Volume II, Chapter 5, we showed that the *Aspects* grammar and its later developments precisely define the class of type-0 languages. Table 8.2. in Volume I shows that in principle such languages are unlearnable, even with informant presentation. This is an extreme form of the empiricist learning situation: the hypothesis space is unlimited. There is thus no heuristic possibility whatsoever for the discovery of the grammar. It is therefore necessary, even from the empiricist point of view, and obviously also from the rationalistic point of view, to limit the hypothesis space to the level of primitive recursive. The second reason for doubt as to the possibility of constructing LAD — and this from any point of view — is Braine's argument that the child, at best, is in a situation of text presentation, not of informant presentation. There are indeed strong arguments to support this, and we shall mention a number of them. (1) By social and cultural circumstances,

the speech of many children is never corrected; that is, a string of words is never marked as incorrect. Nevertheless such children learn the language. (2) When strings are indeed marked as incorrect, this has strikingly little effect. Experiences of Braine (1971) and McNeil (1966) and experiments by Brown (1970) show that directed corrections, even in the form of expansions (for example, Child: *Eve lunch*; Mother: *Eve is having lunch*), have little or no effect on language development. Such negative information cannot be considered as input for LAD if the child never reacts to it. (3) Learnability-in-principle demands a complete presentation. This implies that it should be possible for any negative instance to occur in a finite amount of time. A study by Ervin-Tripp (1971) showed, however, that the language spoken to children is strikingly grammatical, while utterances not addressed to the child are generally ignored by him, and therefore cannot be considered as real input. Ungrammatical sentences, thus, are not only not marked as such, but also occur only seldom in the input. However, for every realistic real time mechanism, positive and negative instances should occur with about the same frequency, and this is decidedly not the case for the language-acquiring child. (4) One might argue that the lack of reaction to the ungrammatical utterance of the child (ungrammatical, that is, with respect to the adult language) is an indirect indication of ungrammaticality. It is indeed the case that ungrammatical sentences stand a greater chance of being unintelligible than grammatical sentences, and consequently the adult or the other child has also a greater chance of not giving an adequate reaction. This argument, however, is a two-edged sword. At first nearly all utterances made by the child are ungrammatical, but every utterance which can be interpreted can nevertheless be followed by an adequate reaction. It is impossible that grammaticality be a condition for reaction, and the child who supposes that it is, will hopelessly confuse the positive instances with the negative.

With a text presentation, as we have said, it will be impossible, in principle, to construct LAD, and any linguistic or psychological program of research within that framework is doomed to failure.

It must be emphasized that this holds not only for a rationalistic LAD, but also for an empiricist one. Braine (1971), who rejects the possibility of constructing LAD on grounds of the relative absence of negative information for the child, goes on to replace LAD with what he calls a DISCOVERY PROCEDURES acquisition model, a model with strong inductive possibilities. In doing so, however, he was not aware that he did not depart from the LAD schema at all, but only from its rationalistic pole. His model has all the LAD characteristics: on the basis of a text presentation as the minimum input it generates a grammar as the output, by means of a very wide hypothesis space and a number of strong heuristic principles (to which we shall return in section 4.3. of this chapter). But this is precisely the empiricist version of LAD, and we can conclude from Braine's own arguments that this discovery procedures theory is quite as impossible to realize as the model he rejects. Only if essentially different aspects are added, such as meaning input, can one avoid the above mentioned difficulties. It should be said that Braine does indeed do this (cf. section 4.3.), but his argument for the discovery procedures model is not based on those additions.

This of course does not mean that the heuristic principles proposed by Braine are a priori without empirical foundation. Quite the contrary is true. But the point here is that the basic question as to whether LAD can be constructed must be answered in the negative, from either a rationalistic or an empiricist point of view, and that we must therefore suppose that the language acquiring child must receive information other than text presentation, if he is to learn his language. One might well think of semantic information in this connection. Although we shall see later that a particular form of text presentation does offer certain possibilities, it remains difficult to imagine that anything but semantic information will prove to be effective. (Of course one could introduce the grammar itself into the model, but that would not only be trivial, it would also be quite unrealistic for a model of language acquisition.) We can therefore conclude that the role of semantic input is not simply incidental, as Chomsky suggests, but that it is entirely essential. Without something like semantic input, the language

will not be learnable. The minimum formal requirements on the form of the semantic input are still a completely open question, however. One could consider a whole gamut of information, from simple paraphrases of sentences (as parents sometimes make) to a complete visual, acoustic, and motor acting out of the intention of the sentences. This is an interesting and as yet untouched field of formal research.[1]

In the rest of this section we shall concentrate our attention on the two remaining problems, the evaluation procedure and the real time characteristics of the model of language acquisition. The discussion of these leads to the consideration of presentation procedures other than Gold's text presentation, and, consequently, to a number of qualifications of the results just mentioned.

In Volume I, Chapter 8, we treated the Bayes-type evaluation procedure for context-free grammars which was given by Horning. For the development of that theory it was necessary to define the notion of learnability in a weaker form, as follows: there is a procedure such that every nonoptimal grammar is rejected within a finite amount of time. With this definition and a stochastic text presentation, the class of nonambiguous context-free grammars is learnable, provided that an a priori probability distribution or a complexity distribution is given for that class. If we are prepared to accept such a weak form of learnability in a model of language acquisition, we might wonder once again how far we can come without semantic information. It should first be noted that stochastic text presentation seems to be an acceptable model for the input of the child. On the one hand Horning shows that considerable deviations from stochastic text do not noticeably influence the results of his analysis, and, on the other hand, that natural language is a sufficient approximation of stochastic text. Moreover, a real probability distribution over the hypothesis space is not essential to Horning's argument, provided that in one way

[1] Semantic relations define equivalence classes over the sentences of a language. Thus the class of paraphrases is defined for every sentence (relation: equality of interpretation). Is a language learnable if a set of semantic relations is offered in the form of equivalence classes?

or another a complexity distribution is given. The probabilistic aspect is therefore less essential than it appears. Still a probabilistic model of language acquisition has many advantages which are lacking in a deterministic model such as LAD or as Gold's version of LAD. We mean the advantages regarding the basic question of the feasibility- in-principle of LAD. It is the case, for example, that a stochastic model is noise resistant, a realistic characteristic. The child might indeed occasionally hear utterances which are ungrammatical, but are not marked as such. With Gold's procedure, the child would then be permanently confused. Horning shows that his procedure is highly resistent to such noise. An experiment by Braine (1971) also shows that human beings have the same characteristic. He presented adult subjects with sentences from an artificial language with the following grammar:

$$S \rightarrow B'\ q\ A'\ r \qquad B' \rightarrow B\ f$$
$$S \rightarrow p\ A'\ B' \qquad A \rightarrow \{a_1, a_2, \ldots, a_6\}$$
$$A' \rightarrow A\ f \qquad B \rightarrow \{b_1, b_2, \ldots, b_6\}$$

The lower case letters belong to the terminal vocabulary, but in fact in this experiment they formed meaningless but pronounceable syllables. Together with grammatical sentences, the subjects were also presented with ungrammatical strings; these latter made up 7 percent of the total number of strings presented. The ungrammatical strings were first and second order approximations of the language (the language is finite, and consequently, with the assumption of a uniform probability distribution over L, all conditional probabilities are known precisely). The ungrammatical strings were three to eleven words long. The degree of learning was tested by means of as yet unpresented sentences of the languages, either in the form of a completion test (filling in omitted words), or in the form of a recognition text. A control group was presented only with grammatical sentences. Both groups showed a high degree of learning; many of the subjects attained full mastery of the language, although none could formulate the rules. The most striking point was that the addition of ungrammatical

sentences ("noise") had no negative effect. Most subjects realized slowly but surely that there were a few strange sentences among the others, but they simply ignored them. The noise resistance of Horning's model is therefore a psychologically attractive characteristic.

But, on the basis of Horning's results, it is not possible to reject in principle the possibility of constructing a stochastic model of language acquisition. All that can be said at present is that nothing is known on that possibility for anything higher than the nonambiguous context-free languages. As for the learning of natural languages, the possibility of constructing such a stochastic model of language acquisition is still an undecided question.

But however the solution of this may be, it seems that such a model is bound to fail with respect to the real time aspects. Horning (1969), who investigated various heuristic procedures with the intention of accelerating the learning process, writes:

> It is clear, however, that grammars as large as the ALGOL-60 grammar will not be attainable simply by improving the deductive processing,

and further:

> But adequate grammars for natural languages are certainly more complex than the ALGOL-60 grammar, and the range of observed natural languages is sufficiently large to require a rather rich hypothesis space — probably much richer than anything we have considered in this study.

On the basis of real time considerations, therefore, this model is also out of the question. It holds here, too, that more input of a different nature than (stochastic) text will be needed. In this connection Horning remarks that the child might receive a completely different kind of presentation of his language:

> he is confronted with a very limited subset, both in syntax and vocabulary, which is gradually (albeit haphazardly) expanded as his competence grows.

This is an important suggestion. The hypothesis space of the child would then be limited at first, and the language addressed to him

would fall more or less into the same class; it would thus be possible to develop an optimal subgrammar. This would be incorporated at a later stage into a new hypothesis space of larger grammars, and the observation space (the language of adults) would in turn adapt to this, and so forth. This would amount to an "intelligent" presentation of the language. It is interesting in this connection to refer to a number of studies done at Berkeley (cf. Ervin-Tripp 1971) on the subject of the language which adults address to children (for further references see Historical and Bibliographical Remarks). That language proved to be simple. It consisted of short sentences, included few conjunctions, passive forms, and subordinate clauses; on the other hand, it contained many imitations of the child's language. Paraphrases and expansions of the child's utterances were to be found, but the general syntactic form of the child's language was nevertheless maintained. Moreover, it was shown that mothers allow the complexity of their sentences to grow with the age of the child, so that, as Ervin-Tripp points out, "the input maintains a consistent relation to the child's interpretative skill". One might say that the child learns a miniature language, and that the parents adapt themselves to the limited hypothesis space of the child.

Although Horning's proposition agrees well with the facts, we are once again in uncertainty. The boundaries of learnability with such an "intelligent" presentation are not known. For natural languages, ambiguity will continue to constitute a serious problem, but for the present we can only state that, with this type of presentation, learnability without additional semantic input cannot yet be excluded a priori.

But even in that case it is not clear why semantic input in the model of language acquisition must necessarily occupy a secondary place. The acquisition mechanism might be able to work much more efficiently if it can dispose of such information at the same time. Finally, it is less a question of what is possible than of what is in fact the case.

It is tragic to cut off from the domain of research the large field of cognitive relations which are found in early sentences (...) by assuming

a priori that there are no interesting problems in their acquisition. Dogmatism without evidence is to say the least presumptuous (Ervin-Tripp 1971).

4.3. UNIVERSALS OF ACQUISITION FROM THE RATIONALISTIC AND THE EMPIRICIST POINTS OF VIEW

Although every theory of language development which has a syntactic input and a grammar as output can be brought under the LAD schema, it is an historical fact that only theories of a rationalistic strain were explicitly based on that schema. As we have already stated above, rationalistic here means the point of view according to which LAD disposes of a very limited hypothesis space. Thus the language-acquiring child would have a strong starting position: he has implicit knowledge of the universal characteristics of human language, and for a large part those universals determine the characteristics of the specific language which he is to learn. Inference is needed only for the nonuniversal, specific characteristics. McNeill (1966; 1971) is the most extreme exponent of this point of view. The following quotation shows how he relates language development to general linguistics.

> However, by accepting linguistic theory as a description of a child's capacity for language, one is led to a very natural explanation of the development of abstract linguistic features. Apparently many (but not all) features in the deep structure of sentences correspond to linguistic universals (Chomsky, 1965). The general form of human language is largely defined in its underlying structure. There is, in addition, the general idea of a grammatical transformation, which likewise contributes to the general form of human language as a formal universal. However, particular transformations (...) are unique, and so must be acquired from a corpus of speech. Accordingly we may think of LAD (or children) *making* such universal features as the basic grammatical relations abstract by acquiring the particular transformations of the language to which they are exposed. We may even say that languages possess such features as the basic grammatical relations as abstractions because they first correspond to children's basic capacity for language, but are subsequently buried under a mass of particular transformations. Accordingly we should expect to find that the earliest grammatical production of children will contain the abstract features of the deep

structure but few of the locally appropriate transformations. Young children should "talk" deep structures directly. And that is precisely what an examination of children's early speech shows (Miller and McNeill 1968).

The child thus begins by nature with a grammar which is an important part of the base component of a transformational grammar. He talks deep structures, or better, in "deep sentences". He later learns the transformations specific to the language with the help of a kind of foreknowledge concerning the structure of transformations. We see here, albeit in another form, the isomorphism of the preceding chapter. Just as the grammar generates a sentence from the deep structure to the surface structure, the language usage of the child develops from deep sentences to surface sentences.

But the empiricist extreme of LAD, on the other hand, does not lead to the conclusion that the child begins with talking deep sentences. In that point of view, the child has in fact little a priori insight into the language of his environment, and can learn only because of strong inductive principles. Inference of the grammar is principally based on regularities noticed in the observed language. Such regularities will have to be visible in surface phenomena. By induction, the child can then make connections between the surface regularities he notices, a new level of abstraction is then reached, and so forth. This is precisely the model which Braine (1971) proposes. As we have mentioned in the preceding paragraph, Braine incorrectly opposed his model to LAD, while in fact it is opposed only to the rationalistic version of LAD which historically became confused with LAD itself. Braine correctly calls his version a DISCOVERY PROCEDURES model. In its most simple form, it has a READER and a MEMORY. The reader scans the input (linguistic utterances) and determines what the characteristics of that input are. For the input *John walks*, for example, it would, among other things, register "two words", "*John* + word", "word + *walks*", and so forth. These characteristics are stored in the memory. In the simplest form of the model, the memory consists of a number of layers. The characteristics of the input are first

put into the uppermost layer. If a characteristic is again observed in later input, it is moved to the next layer, etc. Frequently occurring characteristics are ultimately placed in the lowest layer, the permanent memory. The reader can in turn use the characteristics stored in the permanent memory for a more efficient analysis of new input. It might find that the input has certain combinations of characteristics, and register this fact as a feature in the uppermost layer of the memory. The model is further refined in a number of ways. Features in the intermediate layers are supposed to disappear slowly, thus protecting the model against noise. It is also supposed that the permanent memory at first contains little information, only short input sentences are analyzed, and only some of their features are registered. Other properties have also been added to this model.

This theory thus predicts that the child will begin by registering the frequent and regular characteristics of short sentences. In principle these can as easily be specific to the language as universal. If the first speech of the child is guided by knowledge thus obtained, it is at least not necessary that the child should speak in deep sentences. In that respect, the rationalistic model thus makes stronger predictions. The empiricist model, on the other hand, is very sensitive to frequency. Frequently occurring characteristics (of short sentences) are registered more rapidly in the permanent memory than infrequently occurring ones. We might thus expect a certain frequency matching between the language of the child and the language of his environment. This is not predicted by the rationalistic model.

As was shown in the preceding paragraph, both theories are on rather weak ground as long as no additional information is introduced into the model. Braine (1971) does, however, introduce additional information by pointing out the importance of the semantic input (more will be said on this in section 4.5.). In the present section, however, we shall only discuss the simple syntactic versions of the two models, and in doing so will not do complete justice to Braine's ideas. The theory of McNeill, on the other hand, still fits completely into the LAD schema.

Later we shall mention a few of the many studies which have been done in the field of tension between the rationalistic and the empiricist theories. Before doing so, however, we must clarify a number of notions which have often and unnecessarily troubled the discussion, and still lead to all sorts of confusion.

(1) We have already seen that LAD as an inference schema is incorrectly confused with its rationalistic pole. Rationalistic in this connection means nothing other than that the model disposes of a limited a priori hypothesis space.

(2) Although foreknowledge necessarily results in universals, that is, generally or universally occurring characteristics of human language (because each acquired grammar ultimately satisfies all the characteristic properties of the a priori hypothesis space), it does not hold that all universals must lie in the hypothesis space. Both general characteristics of the observation space (regularities in the observed language) and general features of the inference mechanism (such as the perceptual and cognitive faculties of the child) can lead to language universals. It is thus not the case that only rationalistic models are capable of explaining the existence of universals, although this is often argued. Characteristics proceeding from the limitations of the hypothesis space are called STRONG UNIVERSALS by McNeill (1971). He calls universals proceeding from the nature of the inference mechanism WEAK UNIVERSALS. He does not mention the possibility that the observation space might yield additional universal characteristics.

(3) Within the rationalist camp it is often held that universal characteristics are necessarily innate and not learned. But this is the result of two errors in thinking. In the first place, the possibility that universals might proceed from characteristics in the observation space is overlooked, as McNeill does. In the second place it is tacitly supposed that that which is present in the organism at the beginning of language learning is innate. This can be interpreted in two ways, to one point of view the proposition is incorrect, and to the other it is meaningless, If "innate" is taken to mean

"given at birth" in the traditional sense, it implies that the child learns nothing between birth and the beginning of language learning, but this is absurd. The proposition becomes meaningless when later learning is nevertheless excluded and innate is taken to mean "genetically given". That which is given in the genes can develop only in interaction with its surroundings (the intracellular and extracellular prenatal environment at first, and the internal and external surroundings of the organism later). It is quite arbitrary to call some of these interactions learning and others not, and the question becomes semantic, rather than empirical. It is wiser to avoid words like innate, as ethologists have since long taught us (cf. Hinde 1966). This, however, is not to deny the high degree of specificity of languages. The genetic equipment of the human is in some way particularly suited for the development of language. See Lenneberg (1967) for a thorough treatment of this subject.[1] This does not mean that other animals are incapable of acquiring a productive language form. Work done by the Gardners (cf. Brown 1970) casts a whole new light on this. The Gardners taught sign language to their pet chimpanzee, Washoe. Washoe developed a good vocabulary, and made understandable sentences up to four words long. The experiment ended when Washoe reached maturity. Of course Washoe's language, although much more complex than any other attained by an animal, remained different from children's languages. Washoe used a free word order, for example. But these and other findings by the Gardners (1969) show that chimpanzees possess considerable linguistic potentialities.

(4) Finally, confusion is frequently caused by the identification of the empiricist version of LAD with behavioristic learning theory. An empiricist theory only states that the organism disposes of little or no foreknowledge of the grammar, but that it can deduce the grammar by means of strong heuristic procedures. These procedures are part of the a priori equipment of the child. Just as

[1] We do not share Lenneberg's conception that language is so much biological that it has the usual critical maturation period, in any case not as far as syntactic and semantic structures are concerned. In our opinion, these can as easily be learned at any later age.

in the rationalistic model the nature of the foreknowledge in question must still be defined more completely, empiricists still face the task of analyzing the structure of the inference mechanism of which they speak. Some (together with Staats 1971) will take first recourse to stimulus response mechanisms, such as association, generalization, and discrimination, but that is by no means intrinsic to the empiricist model of language acquisition. At present, however, the only thing which is known with precision on this point is that all regular grammars, as well as all the more complex grammars which fall into Miller and Chomsky's (1963) *tote schema*, are learnable by means of a certain *S-R* mechanism in which correct responses are confirmed and incorrect responses are not confirmed (Suppes 1969). Suppes rightly points out, however, that such a model is quite far away from the observations. It is as if a computer program were written in the form of a network of flip-flops. It works well in the end, but it does not increase our insight into what actually happens during learning. An empiricist theory should be explicit on the components of the inference mechanism, or, in other words, on the subroutines. The way in which they could be realized by means of the elementary *S-R* mechanisms — or any other units for that matter — is only of secondary importance to an empiricist theory; it is not an essential part of the theory any more than the physiological basis of the foreknowledge is an essential part of a rationalistic theory.

To summarize, we can ask, from rationalistic and empiricist points of view, what may be expected concerning the development of syntax in the child.

The *rationalistic model* states that the linguistic development enjoys a high degree of independence from the observations. All children will at first use the same syntactic structures and talk in "deep sentences", regardless of their social, cultural, or linguistic circumstances. Training will do little to change this. The later transformational development, on the other hand, is specific to the language and can be ascribed either to the a priori hypothesis space (strong universals) or to the a priori characteristics of the acquisition mechanism (weak universals).

The *empiricist model* states that the language of the child will at first reflect the superficial characteristics of the corpus presented to him (or at least, those of that which he actually observes). This corpus, and consequently the earliest linguistic behavior, will vary with the social, cultural, and linguistic circumstances of the child. Therefore it is not expected that children will at first use the same syntax, nor that they will talk in "deep sentences". They will, on the other hand, be sensitive to training (systematic influence on the input). Where universal phenomena occur in children's languages, it should be possible to ascribe them to universal characteristics of the corpora observed, or to universal characteristics of the information processing mechanism.

What is the evidence? In the 1960's much activity took place in the field of syntactic language development. It would be impossible to attempt to give a complete account of that work within the scope of this book. We refer the reader to the short survey given by Slobin (1971a), or to the detailed and complete discussion by Braine (1972). We shall limit the discussion here to a few data concerning the first two-word sentences of the child.

Most of the studies were characterized by the analysis of extensive corpora of speech, collected within short periods of time. Each study dealt with one or only a few children, but was worked out in great depth. Most often the speech of the child (including dialogues with others) was recorded on tape and processed later. This information was usually completed by the addition of data on the situational context, and sometimes of experimental data, such as the results of imitation and comprehension tasks. By the examination of the same child at various stages of development, longitudinal data were also collected on a number of them.

The first three studies in the period in question were those of Braine (1963), Brown and Fraser (1963), and Miller and Ervin (1964). They all contained analyses of the brief period of two-word sentences (the child is about a year and a half old), and it was felt that striking agreement was found in syntactic structure. That agreement can be expressed in the following PIVOT GRAMMAR:

$S \rightarrow P_1 + O,\ S \rightarrow O + P_2,\ S \rightarrow O$

Braine called P_1 and P_2 PIVOT categories. They contain a small number of frequently used words, which can only occur exclusively in the first position (P_1) or exclusively in the second position (P_2). All other words fall into the open class O. A typical P_1 word is *allgone*, and a typical P_2 word is *on*. Characteristic sentences are *allgone shoe, allgone bandage, shoe on, bandage on.* The two-word sentence period is not completely free of still another construction, $O + O$, but it is argued that that form is infrequent at first.

Rationalist authors considered this a first universal phenomenon,[1] the expression of an innate grammatical relation which in the language of adults could be defined only over the deep structure. The pivot-open construction, in particular, was taken as a relation of modification (McNeill 1966), and P and O were considered to stand for the modifier and the head, respectively. The class of modifiers is at first extensive, containing, for example, demonstrative pronouns *(this, that)*, articles *(a, the)*, adjectives *(big)*, and possessive pronouns *(my, your)*. Later, according to McNeill, these categories are further differentiated, as may be seen in the distributional properties of the first three-word sentences (*that my car* occurs, while *my that car* does not). Details of this analysis are not important for the present purposes. What is important is that on the basis of these distributional characteristics it was decided that there are fundamental grammatical relations which are apparently innate. In the *Aspects* model these could only be defined over the deep structure, but in the language of the child, they simply appear as surface constructions. The rationalist argument thus rests on a rather high degree of interpretation, that is, (a) the assignment of categories to words on grounds of distributional analysis, and (b) the assignment of universal relation names to the (hierarchic) order relations between the categories. Concerning the pivot grammar, both of these points appear to be quite arbitrary. As for (a), it can be pointed out that

[1] Looking back at the literature, it is striking to notice how so few English-speaking children have been made responsible for so many universals.

the pivot grammar is by no means an exhaustive description of the distributional relations. Even in the original material on which McNeill based his universality argument, deviant constructions appear. Thus we find $P + P$ constructions in Brown and Fraser's material, and in all the material some pivot words can occur in either the first or the second position. Such words obviously belong to two grammatical classes (P_1 and P_2), but this is not further analyzed. On (b) we can say that the assignment of fundamental grammatical relations to such pivot-open constructions as well as to open-open constructions is an arbitrary matter from a distributional point of view. It is quite clear that one must often be able to assign more than one grammatical interpretation to a given order of categories. This is most evident for ambiguous two-word sentences. As an example of this, Bloom (1970) mentions the sequence *Mommy sock*, which in one situation can clearly mean "Mommy's sock",a nd in another, "Mommy is putting on my sock". McNeill also freely gives multiple interpretations to certain category sequences. However this means that children at first do not simply speak in "deep sentences" in which the relations should be immediately evident, but that they express different functional relations sometimes in the same word order. But this means that if functional relations, such as those of subject, object, and modification, are to be defined configurationally, as in the *Aspects* model, a transformational instead of a purely phrase-structure grammar will be needed for the level of the two-word sentences. This conclusion is also drawn by Bloom, as we shall see in the following section.

There is still another problem with the rationalistic interpretation of the pivot grammar, even if we suppose that it is observationally adequate. This problem has to do with the distributional differentiation of categories in later development. The supposed universality of syntactic categories in the base grammars of various languages forces McNeill to find a rationalistic genetic explanation. In his original theory (1966) he stated that those categories grow by a gradual differentiation on the basis of the primitive categories P and O. This differentiation is first expressed in increasing distributional refinement, an example of which has just been given (cf.

Chapter 1, section 1.3. for further details). But if this is so, it is difficult to understand why some words which will later become adjectives are at first found in the *P* class while others are found in class *O*. How, in the course of language development, they manage to arrive in the same category is a riddle. Therefore McNeill had to change his position (1971). He states that for every word not only a category feature, but also one or more "contextual features" are learned. In the lexicon of a child at a given stage of development, for example the word *big* might have the category feature *Adj* as well as the following two contextual features: [+ — *NP*] and [+*NP* —]. The child can thus make constructions such as *big house* and *house big*, both of which, despite the difference in word order, express a universal modification relation, namely the adjunction of an adjective to a nominal element. The word order of the first two-word sentences will depend upon which of the two contextual features is acquired first by the child. It may be one feature for one adjective, and the other feature for another. In both cases the feature lacking will be added at a later stage of development when this is required by the language in question. Consequently the final category structures will converge, or better, in spite of the distributional differences, there was never any difference in category structure. Notice that McNeill uses the same notation for these contextual features as is used in *Aspects* for subcategory features (cf. Volume II, Chapter 3, section 3.1.1.). But McNeill expands the notion considerably. Thus a contextual feature can show that a word can be the subject of a sentence [+ — *Pred P*], which is not the case in *Aspects*. All McNeill is actually doing here is replacing the context-free base grammar with what is essentially a categorial grammar (cf. Volume II, Chapter 4, section 4.2.). The word order is determined by a set of contextual features; in terms of a categorial grammar this means that one or more (complex) categories are assigned to each word. The child then need only learn the set of categories for the various words, and the word order in the sentence constructions will then directly follow from the mechanism of the categorial grammar. McNeill tries to save the pivot grammar in this way, while

maintaining the rationalistic point of view. But when we formulate his theory of contextual features as a categorial grammar, where words have multiple categories acquired in an indefinite order, we see that McNeill abandons precisely the most characteristic property of the pivot grammar, namely, fixed word order. With the contextual features, every pair of words can occur in any order, provided that there is sufficient choice among the complex categories for the words. Or, in McNeill's formulation, each unexplained word order can be accounted for by the addition of a new contextual feature without the necessity of readjusting the category. These categories are universal and given from the beginning. In this way McNeill saves the rationalistic point of view by making it untestable. But at the same time the pivot grammar is implicitly rejected.

More recently, however, the pivot grammar has been rejected much more explicitly. Bowerman (1971) studied a Finnish child who showed no evidence of a pivot grammar. Two of the three children studied by Bloom (1970) likewise exhibited no pivot grammar. Schaerlaekens (1973) in a strikingly complete study of the two-word phase of six Dutch-speaking children (two sets of triplets) also found no evidence of a pivot grammar. Moreover, using explicit criteria, she made a more detailed study of the corpus of one of the children with the purpose of trying to find a pivot grammar for it, but this did not prove possible.

Concerning the first two-word sentences of the child, therefore, we can safely conclude that the rationalistic expectation of those sentences having the same syntactic structure for all children cannot be confirmed. There are remarkable differences in word-order even at this first stage of development (notice that we did not as yet formulate anything concerning semantic universals, but are still dealing with purely syntactic statements).

What about the predictions made on the basis of the empiricist model? The empiricist version of LAD predicts feature matching: the word order in the earliest sentences should reflect the dominant word order in the language of the environment, or at least that of the language addressed to the child. But empiricist researchers

have never seriously tried to prove this. The interlinguistic research on language acquisition done at Berkeley (cf. Slobin 1970 among others) showed that many, if not all, children in various linguistic environments begin with a few pivot-like constructions, and that they most often use the dominant word order of their surroundings. In cases where the word order is very free (as, for example, in Russian) and there is therefore no basis for adaptation to surroundings as far as word order is concerned, children nevertheless appear to choose a fixed word order and to maintain it. More precise data may be found in Bowerman (1971), where the author treats naturalistic observations of verbal communication between a Finnish child and its mother. When the average length of the child's sentences was 1.42 morphemes, the language of mother and child showed the following frequencies in the word orders of subject (*S*), verb (*V*), and object (*O*) (notice that these categories were not defined distributionally, but semantically, in Bowerman's study. It is generally not difficult to decide which word is the subject and which is the object if one knows what the discussion is about.)

	SV	*VS*	*VO*	*OV*	*SVO*	*OSV*	*OVS*	*VSO*	*SOV*
Mother	47	5	16	3	32	0	1	1	1
Child	44	4	4	1	7	1	0	0	1

The child has obviously taken over the dominant word order in Finnish, and in particular the most frequent *SV* and *SVO* patterns. In Schaerlaekens' work, although the imitation of word order was not the object of separate study, rather strong arguments can be found in support of the word order imitation theory in two-word sentences. The genitive relation, for example, which Schaerlaekens calls the "relation of fixed allocation" (and which once again is defined semantically and not distributionally) for each of the six children, is reflected in the fixed word order POSSESSION-POSSESSOR: *auto Piet* 'car Piet', *boek madame* 'book lady', *boot Diederik* 'boat Diederik', *koek Gijs* 'cookie Gijs', *poep kindje* 'bottom child', *koets Karel* 'wheelbarrow Karel'. The genitive relation in English usually shows precisely the opposite order. Bloom reports

that her subjects used the word order POSSESSOR-POSSESSION: *Kathryn sock, Wendy hair, baby raisin,* etc. It is quite possible that in Dutch the genitive relation is predominantly expressed by way of constructions with *van* 'of' and *heeft* 'has', while in English the *NP*'s+*NP* construction is more frequent. However, precise data on this are lacking.

Slobin (1970) is quite aware of the systematic differences in word order, but he apparently is not inclined to draw the empiricist conclusion from this. He writes,

> If you ignore word order, and read through transcriptions of two-word utterances in the various languages we have studied, the utterances read like direct translations of one another.

They agree with respect to the various stages of development (children never begin with sentences more than two words long, although they produce long sequences of babbling noises, and can link a few sentences together which have to do with a single thematic situation). The sentences are telegraphic, that is, they consist principally of content words rather than function words. They are always without inflection, articles, or copula. This is enough to make Slobin remark "what is remarkable at first glance is the uniformity in rate and pattern of development".

Concerning the possibility of training, another prediction made on the basis of empiricist suppositions, there are several incidental examples which indicate that in the first phase of language development the child is insensitive to explicit attempts at correction of his language (McNeill 1966; Braine 1971). Training programs appear to be effective at later stages, when the child is three or four years old. But at that point it is mainly a question of the transformational structure, and both rationalists and empiricists recognize the necessity of learning this. The only nontransformational study of receptivity to training is only loosely related to children's language acquisition; it dealt with adults' acquisition of simple artificial languages. We refer the reader to Miller (1967) and to the excellent survey by Smith and Braine (1972). The conclusion drawn by Miller is that it is nearly impossible to learn

artificial languages without access to semantic information. Using a different method of experimentation, however, Smith and Braine come to the opposite conclusion. They showed which aspects of the meaningless language could be learned and which could not. An interesting result was that adults could learn an artificial pivot grammar of the form $S \rightarrow P_1 + O$, $S \rightarrow O + P_2$, $S \rightarrow O$ only to a certain extent. In fact, all they could learn was that P_1 elements always occur in the first position, P_2 elements always occur in the second position, and some words could also occur alone. Expressed in rules, this gives: $S \rightarrow P_1 + W$, $S \rightarrow W + P_2$, $S \rightarrow W$, where $W = P_1$, P_2, or O. The learning of positions is seen here to the extent that the first and the last positions are learned, but this does not lead to a complete differentiation of the three categories. If this is a general characteristic of the human inference mechanism, the two-word situation can be interpreted in the following way. Sentences spoken to small children contain frequently occurring words, some of which usually appear in the first position *(hi, look)*, while others usually appear in the last position *(on, off)*. The child will begin to suppose the grammar just mentioned, and not the pivot grammar. The fact that the child is then additionally able to differentiate class O, while the adult is not, might be the consequence of an a priori hypothesis, or of semantic knowledge: the child would simply know that *hi on* has no possible interpretation in his world.

The early two-word stage, to which we have limited the discussion in this paragraph, can, of course, give no decisive answer on the empiricist and rationalistic predictions concerning the transformational development. Nevertheless, before drawing up a balance sheet, we would first make a number of remarks on that subject. In itself, transformational development is a highly interesting field of research. Thanks to transformational grammar, aspect of language development which had never been studied before have become accessible to research. We refer the reader, for example, to the splendid work of Bellugi (1967) on the acquisition of the negative in English, and to publications by Menyuk (1969) and C. Chomsky (1969).

According to the rationalists, the universal base matures more or less without help. The acquisition of transformations, however, is highly dependent on the nature of the input, and because that which is learned comes later than that which is innate, it is predicted that the development of transformations will occur relatively late. This prediction agrees with the facts. But this is a spurious argument in favor of the rationalist position. By definition, a transformation is a relation between constituent structures. Their acquisition, therefore, supposes that those structures are present to start with, and thus stated, there is no empirical argument. But if we ask whether the result of the transformations is reflected in the first simple sentences of the child as the empiricists predict, the answer will simply be affirmative. If the child says *plat eendje* 'flat duckling' (Schaerlaekens), he is not speaking a "deep sentence", because adjectives are transformationally derived from propositions like *eendje is plat* 'duckling is flat'. The traces of transformations which can be found in the language of the adult can also be found in the first utterances of the child, completely in agreement with the empiricist point of view.

If we now consider the whole state of affairs, we must conclude that some aspects of the observation space do indeed have a specific influence on early syntax. A purely rationalistic point of view must be rejected. On the other hand, even from a purely syntactic point of view, there is remarkable agreement in the language development of children in different cultures and linguistic surroundings. This calls for some explanation. We have seen that the reasons for this may be sought in (i) the hypothesis space, (ii) the deductive information processing procedures, and in (iii) the observation space. Rationalists accentuate the first two, empiricists the latter two. Let us examine these possibilities.

The investigation of the hypothesis space, essential for the rationalist school, is seriously handicapped by the lack of independent insight into the structure of the base grammar. As we have seen in Volume II, Chapter 5, as long as an *Aspects*-type formalization is used, nearly every base will be universal for purely

logical reasons. It is impossible to choose among such bases on empirical grounds. But until that universal base is well defined, it is useless to establish on it the explanation for the evidence of universals in children's languages. On the other hand, it is quite obvious that under these circumstances no empirical problem will be solved simply by defining the universal base in terms of the universals which are to be found in children's languages. In our opinion, therefore, in the near future rationalists cannot be expected to gain much credit on this point, although it is the only ground on which the nationalistic point of view could be verified and the empiricist rejected. This type of confrontation is not to be expected soon. However, it is not at all excluded that the empiricist stand will be verified and the rationalistic rejected as we shall see presently.

For both rationalists and empiricists the heuristic procedures would be an acceptable source of universals. They are McNeill's "weak universals". The future here is rather promising. Not only will it be possible for both camps to cooperate in research, but that research will be empirical in nature. It is quite possible, in effect, to collect data on the information processing capacities of the small child, and to do so by nonlinguistic means. One could then examine the way those processing procedures operate on linguistic input. A number of recent publications (Bever 1970a; 1970b; Slobin 1971b) deal with such process factors in language development. We shall return to this in the following section.

Research on the observation space, finally, is essential for the exclusive proof of the empiricist version of LAD as opposed to the rationalist one. The study reported by Ervin-Tripp (1971) on the language which mothers speak to their children is a start in this direction. Not only is such research possible; it is also of great interest to theory. The investigation of receptivity to training also falls into this category (on this point, see a number of studies by Brown, et al., in Brown 1970). It may seem to be unsatisfactory, however, to attempt to explain the universals in children's languages on the basis of the universals in the languages presented to them, for the question as to how those universals come into

being in the first place remains open. Yet we cannot exclude that possibility of explanation. The problem is basically one of "the chicken and the egg", and there is no reason to seek the solution only in the egg.

4.4. PROCESS FACTORS IN LANGUAGE ACQUISITION

The argument that universal characteristics in the language development of the child proceed largely or completely from general characteristics of cognitive development is not new. It has always been an axiom of Piaget's school at Geneva; see, for example, the work of Sinclair-de Zwart (1967). But it is not easy to discover why certain phenomena which occur in language development must necessarily be the consequence of circumstances in the field of concept formation and the information processing capacities of the growing child. Such research concerning syntactic phenomena in the child's language is just beginning. Bever (1970a; 1970b) shows that some linguistic rules might be the natural consequence of general perceptual characteristics of the child. With evidence which is largely incidental, he shows that some perceptual strategies for language processing (cf. Chapter 3, section 3.5.) are essentially not specifically linguistic, but are also operative in the recording of other kinds of information. This latter involves an important obligation in this kind of research, for there is real danger that universal linguistic phenomena may simply be translated into general cognitive principles, without independent — that is, on a nonlinguistic basis — proof of the existence of such principles. This may be seen in much of the literature of Wundt's school of the psychology of language, above all in the work of van Ginneken (1904) who explained the word order used by children on the basis of the temporal order of images. This amounts to nothing other than terminological magic if this order is not determined independently. No proposition in the literature on the subject is completely immune to this objection, and the real research has still to begin.

Only for the sake of orientation, we shall now mention a number of attempts to account for linguistic universals by reference to

general information processing capacities. Most of the examples have been borrowed from Slobin (1971b).

One of the first information processing capacities of the child is attention to the order of elements. A universal linguistic consequence of this would be that the earliest sentences would reflect the dominant word order in the language of the surroundings. In the preceding paragraph we showed that there is some evidence for this. But two points should not escape our attention. In the first place, the early capacity for the processing of order must still be proven on the basis of nonverbal tasks. Slobin simply supposes that this is the case. In the second place, the word "universal" refers to the child's way of reacting linguistically (for example, in maintaining word order), and not to the final grammatical form. This latter will in fact be *different* in various linguistic environments, in consequence of this principle. This comes close to the proposition that humans are universally different.

A better example is: the end of an information sequence receives primary attention.[1] This proposition must also be proven independently, and the notion of information sequence calls for further definition. The experiment by Smith and Braine in which a pivot-like artificial language was used (cf. section 4.3.) could be taken in this sense. They found that subjects rapidly discovered which elements could occur in the last position (but the first position was also marked). On the basis of this principle, Slobin predicts that suffixes and postpositions will universally be acquired earlier than prefixes and prepositions. There is some evidence for this, for example, the fact that French-speaking children, in learning the negation *ne...pas*, acquire *pas* earlier than *ne*. Slobin also gives further instances of this kind.

Other examples have to do with the tendency to avoid interruptions in information processing (thus, permutations which occur with some transformations are acquired at a late stage of development), with the tendency to avoid exceptions (so that after a short period weak forms are preferred to strong: *breaked, mans*), and

[1] Slobin's formulation of this is linguistic: "pay attention to the end of words".

with semantic processing principles, on which we shall have more to say in the following paragraph.

4.5. CONCEPTUAL FACTORS IN LANGUAGE ACQUISITION

With this subject, we definitively leave the domain of the LAD schema. Reasoning on the basis of the schema itself, we showed, in paragraph 2 of this chapter, that syntactic input alone is not sufficient for a realistic model of language acquisition. One can, of course, still try to stay as close to the schema as possible while adding a handful of semantic aspects to LAD's input, but such an approach would be harmful to the actual development of the study of language acquisition. Around 1970 a basic reorientation began to be observed in this field of research. A semantic point of view came to replace the common syntactic approach to the study of early language.

One of the first studies to appear from this view point was that of Bloom (1970). In the preceding paragraph we showed that McNeill's syntactic analysis of the two-word stage is essentially distributional, and we mentioned that Bloom proved that a given word order can go together with different semantic or functional relations. In that connection we cited her example of the ambiguous string *Mommy sock*. If the grammatical functions of subject, object, etc., are still to be defined configurationally as in *Aspects*, it will be necessary to develop a transformational grammar for the two-word stage, in which an ambiguous sentence such as *Mommy sock* can have two deep structures, both of which result transformationally in the same surface structure. This is essentially the method followed by Bloom. On the ground of semantic considerations, she categorizes the functional relations which are expressed in the two-word sentences. To account for this she writes a context-free base grammar in which those functions are defined configurationally. Terminal strings generated by the base grammar can contain more than two elements. A system of transformational rules, however, reduces the length to two words at most (for the two-word stage), and this results in the observed am-

biguities. This is fine, but it remains a somewhat hesitant beginning for the semantic approach. What is new to Bloom's approach is her nondistributional semantic point of departure, which, if care is taken, can yield far better insight into the first sentences of children. But there is no need whatsoever to use a system of description for the functional relations thus found which was developed for the language of adults (and even then without being all too convincing); we are referring to the definition of functional relations in terms of the hierarchic relations among syntactic categories. For purely formal reasons, we would then have to suppose that at an early stage the child disposes of a differentiated system of grammatical categories (for Kathryn, one of Bloom's subjects, the two-word stage demanded ten such categories, as well as four nonterminal categories), and we would also have to assume that the sentences have abstract parsings, not expressed in the surface form. No independent empirical evidence is available for either of these suppositions; they are only artifacts of the grammar's formalization.

Schaerlaekens (1973) avoids these complications. She starts with the same semantic point of view, and tries, like Bloom, to discover which functional relations the child expresses in his two-word sentences. But she does not impose a transformational model which would raise more problems than it would solve. She only shows that many of those functional relations are accompanied by a characteristic word order, but that word order is defined in terms of the functional role which the two words play in the relation, and not in terms of abstract grammatical categories. If the functional roles were reversed, the words would change places correspondingly. Thus the bothersome problem of the multiple categorization of words is avoided. The most important relations which Schaerlaekens found in the two-word sentences of the six children she studied are the following:

(1) The RELATION OF FIXED ALLOCATION *(car Piet, book Arnold)*; in general this is a genitive relation *(the car of Piet, the book of Arnold)* with a fixed word order: possession-possessor.

(2) The RELATION OF COINCIDENCE, which is usually a locative relation *(Joost bed, mister bicycle)* has a fixed word order: object-place *(Joost is in bed, mister is on the bicycle)*.

(3) The SUBJECT-VERB RELATION is an actor-action or agentive relation *(Karel cries, father shaves)*; there is no fixed word order, but the *S-V* order appears to be dominant.

(4) The OBJECT-VERB RELATION is an objective relation *(airplane take, watch trashcan)*; there is a fixed word order for some children, *O-V*, but not for all.

(5) QUALIFICATION RELATIONS, such as DEIXIS *(look car, hears bim-bam)*, NEGATION *(no bed, no milk)*, PLACE RELATION *(there tower, here bus)*. Not all of these relations were observed for all the subjects, nor were they always clearly differentiated. Word order aspects are likewise precarious and complex.

Other infrequent and idiosyncratic relations were also observed, but we shall not discuss them here.

The point is that in all cases we are dealing with functional relations among concepts which may be expressed in a characteristic word order, but even this is not necessarily the case. This situation is analogous to that of the conceptual models of language users. In those models attempts were made to understand linguistic behavior as the communication of essentially nonlinguistic conceptual relations. In the present case, language development is to be explained in terms of the development of the conceptual basis. Slobin (1971b) writes:

> Is it possible, then, to trace out a universal course of linguistic development on the basis of what we know about the universal course of cognitive development? (Can one take Piaget as a handbook of psycholinguistic development?)

The analysis of the first words and sentences of the child must therefore be primarily oriented to the question of the intentions which the child wishes to express. The first words stand for objects and

actions in the child's world, but there is no reason to suppose that this first subjective lexicon also contains information on categories. Grammatical categories will rather develop slowly but surely from the discovery that certain conceptual relations ordinarily accompany certain relative word positions in the language of adults.[1] The child will try to imitate this, and will eventually have the actor precede the action *(Karel cries)* regardless of the grammatical categories of the words in the language of adults (the same relation is expressed in a sentence such as *airplane by*, where *by* indicates the action, although it is not a verb). Only later will the child learn that not all words which can stand for actions may be placed in the second position in this relation. The words for which this does hold according to the language of adults will be specially marked as the "regular" ones. This is only an early stage of syntax acquisition. But it has already been preceded by rather extensive conceptual and verbal experience, according to this conceptual point of view. Language acquisition would then follow the sequence conceptual (concepts and relations) — semantic (words and their conceptual references) — syntactic (syntactic categories of words and rules of syntax).

As we wish also in this last paragraph to concentrate our attention on relations with formal grammars, we shall not discuss the actual content of this conceptual approach (we refer the reader especially to Campbell and Wales 1970; Donaldson 1970; and Luria 1961). In closing, we shall treat the only formalization known to us of the cognitive-semantic approach, to be found in Schlesinger (1971). According to Schlesinger, also for the child, the input of the speech mechanism is the intention. He calls its conceptual representation the *I*-MARKER. The output of the system is the sentence — at any rate the author limits himself to that. The *I*-marker is transformed into a sentence by means of REALIZATION RULES.

The *I*-marker is a relational network of objects and relations. Just as in Schank's model (cf. Chapter 3, section 3.6.2.), the number

[1] It is quite possible that at this first stage also inflection and accent will be used to mark conceptual relations.

of conceptual relations is limited; they principally include the conceptual case relations of agent (*Ag*), object (*Obj*), locative (*Loc*). There are also attributive relations (*Att*), determiner relations (*Det*), and operations such as negation and ostension.

Suppose that the speaker wishes to say that John catches the red ball. For the formal treatment we shall write concepts in upper case letters and words in lower case letters. The following relations exist among JOHN, CATCHES, THE, RED, and BALL (abstraction made of tense and number relations):

(i) *Att* (RED, BALL)
(ii) *Det* (THE, RED BALL)
(iii) *Obj* (THE, RED BALL CATCHES)
(iv) *Ag* (JOHN, CATCHES THE RED BALL)

According to Schlesinger, the formal structure is then

Ag(JOHN, [*Obj* ([*Det* (THE, [*Att* (RED, BALL)])], CATCHES)])

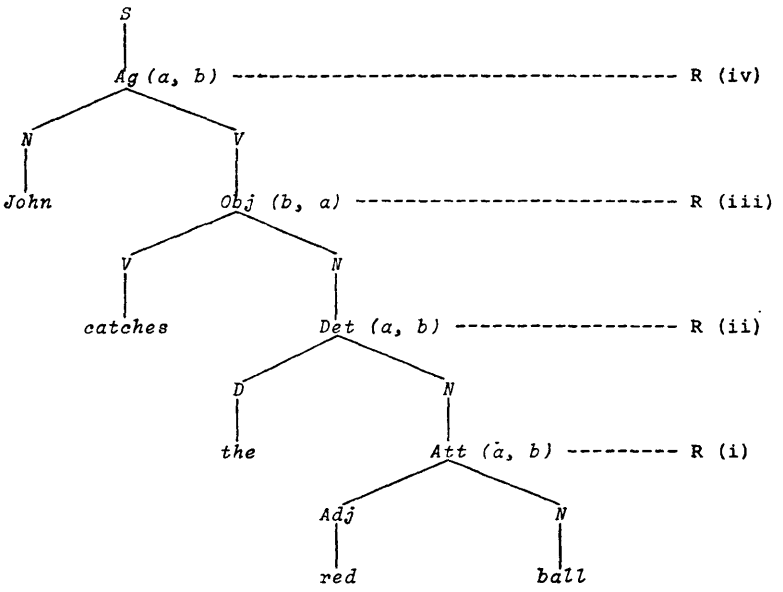
where the elements are not ordered from left to right, but only have a hierarchic organization.

The realization rules assign a place and a syntactic category to each element in the *I*-marker. This means that the *I*-marker is realized verbally by finding words or phrases of the category indicated and putting them into the correct order. As an example, Schlesinger gives the following realization rules for the relations shown above in (i) to (iv):

R(i) $Att(a, b) \rightarrow N(Adj_a + N_b)$
R(ii) $Det(a, b) \rightarrow N(D_a + N_b)$
R(iii) $Obj(a, b) \rightarrow V(V_b + N_a)$
R(iv) $Ag(a, b) \rightarrow S(N_a + V_b)$

R(i) means that the attributive relation between concepts *a* and *b* is expressed syntactically by finding an adjective word for *a*, followed by a substantive word for *b*. In other possible realization rules this sequence will then behave as a substantive or noun phrase. Suppose that the speech mechanism finds the word *red* for Adj

RED, the word *ball* for *N* BALL, and so forth. The resulting linguistic structure will then be:



The conceptual relations are thus marked in the surface (more or less as in Halliday's systemic grammar). Some of the realization rules bring about that which in *Aspects* is done by transformations, for example, the positioning of the adjective. But additional position rules will be needed for the realization of paraphrastic transformations. Schlesinger, however, does not work this out, because he wishes by way of the schema to focus attention on language development. Before going into that, we will make two remarks on the formalization up to this point. The *I*-markers show that which Chomsky calls a SET SYSTEM (cf. Volume II, Chapter 4, section 4.2.). They define the hierarchic relations among elements, but not their arrangement. Their formal structure was studied by Curry (1961), among others. Chomsky's objection to this is that such a system must be complemented by a set of ordering rules (Schlesinger's realization rules) and, in his opinion, one could

quite as well begin directly with an ordered structure. But in Schlesinger's case the matter is obviously somewhat different. The *I*-marker is not a phrase marker, but a network of conceptual relations which is also based on arguments which are not linguistic in character. The same networks must serve as the input for non-linguistic motor actions (the "hand" in Winograd's model), and they are the output of nonverbal input (the "eye" of that model). It would not be easy to find strong arguments for an ordering of concepts. It is, moreover, an old, but completely unproven proposition that word order reflects the sequence of thoughts (cf. Levelt 1967b). A set system can have its attractive sides as a conceptual representation, but the problem here is finally no different from the one discussed in Chapter 3, section 3.7., namely that of finding a suitable formal system for the representation of knowledge. In that regard, Schlesinger's proposals are significantly more limited than the procedural approach presented by Winograd. Schlesinger is also more limited insofar as he considers the sentence to be the maximum framework of expression for the *I*-marker; for Winograd, an intention can be expressed in a whole paragraph.

Our second remark concerns the aspects of the intention which are expressed linguistically. Schlesinger defines the *I*-marker as "the formalized representation of those of the speaker's intentions which are expressed in the linguistic output". This not only disregards everything which precedes the immediate input in the realization rules, especially the whole intelligent processing in the conceptual base, but it also excludes from further investigation all factors, such as limitations of memory or vocabulary, which cause some intentions which do constitute input for the speech mechanism not to result in linguistic output. Such factors, however, are important for the study of language acquisition.

Concerning the development of language in the child, Schlesinger states that the earliest realization rules which appear in the child's language are POSITION RULES, that is, rules which project these conceptual relations on a particular word order, just as was seen in Schaerlaekens' study. Syntactic categories play no role at this stage. Schlesinger mentions the following position rules among

others. (The two concepts are given in the order to be realized. The examples are taken from naturalistic observations in which the context unambiguously revealed the intention.)

(1) actor + action	ex. *Bambi go, airplane by*
(2) action + object	ex. *pick glove*
(3) actor + object	ex. *Eve lunch*
(4) modifier + head	ex. *big boat*
(5) negation + *X*	ex. *no wash*
(6) ostension + *X*	ex. *here bed, it ball, that blue*
(7) *X* + locative	ex. *sat wall, baby room*

Although these rules are based on the linguistic corpora of English-speaking children, the same or similar position rules hold for the Dutch corpora studied by Schaerlaekens; differences in terminology should not lead us astray. It is indeed the case that, according to research on the two-word stage, not all position rules are functional. Even in later stages, some children exhibit a free word order. Gruber (1965) mentions such a case, and it is worth mentioning that Washoe, the Gardners' chimpanzee, never felt bound by position rules, although she seems to have expressed the same kind of relations as those involved in position rules (1) to (7). The child learns to express conceptual relations not only by means of position rules, but also by the use of special morphemes, such as inflections and prepositions (take, for example, the plural, possessive, and tense morphemes which appear quite early).

According to Schlesinger, only after having learned various position rules does the child learn the syntactic categories, the category rules. How does this happen? Little is known about this, but two principles follow directly from the conceptual theory of language acquisition:

(1) If a syntactic category has a conceptual corollary, that category is not acquired before the child is able to make the conceptual distinction. The category of possessive pronouns is not learned before the child can understand the genitive relation "belongs to". Tense categories appear after the development of the capacity

to distinguish time, and so forth. Although this all seems to be quite obvious, the principle is not a trivial one. In the first place, it does not follow from the LAD schema, according to which syntactic categories can first develop and only later be used for semantic expression. In the second place, it is not easy to prove in a nonlinguistic way that a given conceptual relation lies within the reach of the small child. How, for example, could one test the distinction between "count" and "mass" concepts? In the third place, it is seldom clear which conceptual distinctions are in question. Thus it is sometimes argued that substantives and verbs are distinguished conceptually as things and actions. Braine (1972) proves convincingly that this is a misinterpretation.

(2) A category is more rapidly acquired the better it is marked. Slobin (1971b) gives many interesting examples of this principle. One of them concerns a child who was raised bilingually to speak Hungarian and Serbian. In Hungarian, locatives are expressed in an orderly way, with a suffix added to the noun. In Serbian, on the other hand, locatives are expressed in the form of a relation between preposition and an inflection of the noun, and that inflection depends on the gender of the noun. The child learned the Hungarian locative categories long before he used the Serbian correctly. More generally, we can expect that the simpler the syntactic means is, the earlier the children will begin to express relations and concepts in correct syntax. The rather late acquisition of the future tense in Dutch and English can be better explained on the basis of the relatively complicated syntactic form than on the proposition that the notion of the future is acquired late in development. Before the child begins to speak, he already follows with his eyes the disappearance of an object behind a screen, and directs his attention to the point at which the object is expected to reappear; a primitive notion of the future is thus present already.

How far this conceptual point of view will lead remains an open question. For the present, we can expect important findings in research on early language. The more the language and the thought of the child become entwined, however, the more feedback will

occur from syntax to conceptualization. Later the child will also become able to draw conclusions from the syntactic category of a word on the concept to which the word refers. Brown (1957) gives experimental examples of this. Apparently, the acquisition of concepts can in its turn be the consequence of syntactic categorization.

However this may be, the study of language acquisition, like that of the language user, will certainly be served by the development of suitable formal systems for the representation of knowledge. Here, too, the theory of formal languages is becoming increasingly important to the nonlinguistic aspects of theory.

HISTORICAL AND BIBLIOGRAPHICAL REMARKS

For a historical survey of psycholinguistics, see Blumenthal (1970). Some of Whorf's articles have been collected in Whorf (1956). Steinthal's linguistic psychology is published in his book *Grammatik, Logik, und Psychologie* (1855). Paul's principal work appeared in 1886. Wundt's linguistic theory may be found in the two volumes of his *Völkerpsychologie* (1900).

Intuitive linguistic judgment has been neglected as an object of research by both linguists and psychologists. Beside the literature mentioned in Chapter 2, the following sources on ungrammaticality should be noted: Coleman (1965), Marks (1965), and Seuren (1969). Bever (1970a; 1970b) is the only linguist to give thorough attention to the empirical aspects of the problem of intuition.

The psychology of grammar discussed in Chapter 3 is largely the work of George Miller and his collaborators. Both his fundamental contributions, such as the articles written in collaboration with Chomsky and published in the *Handbook of Mathematical Psychology* (1963), and his excellent popular writings made this type of psycholinguistics dominant in the 1960's. A complete survey of the developments in this field until 1966 may be found in Levelt (1966); the developments from 1966 to 1971 are covered in Fillenbaum (1971). It would be incorrect to continue to identify Miller with this tradition; since 1965 he has been almost exclusively concerned with semantic problems. Much information on the early conceptual or semantic models can be found in Minsky (1968). For our chapter we have borrowed much from Frijda (1972), Schank (1972), and Winograd (1972). We refer the reader

also to Tulving and Donaldson (1972). More information on pattern grammars can be found in Rosenfeld (1969), Grasselli (1969), Lipkin and Rosenfeld (1970), and the two special editions of *Pattern Recognition* — 3: 4 (1971); and 4: 1 (1972).

It is no accident that the LAD schema corresponds well to the theory of grammatical inference. Both originated in Chomsky's work. But it is surprising that the two lines of reasearch lost contact with each other at such an early stage of development. This was neither to the advantage of the study of language acquisition, where mathematical precision tended to be replaced by dogmatism, nor to the advantage of the study of artificial intelligence, where the possibility of application of the findings to psychology was considerably limited. The terms "empiricist" and "rationalistic" have a much wider meaning than the technically limited use in Chapter 4. Chomsky's broader use of the term may be found, for example, in Chomsky (1968). The chapter is model oriented, and limited to the formal grammar problem. Consequently it should not be consulted as a general survey of the literature on the subject. A rather complete bibliography was composed by Appel (1971). Likewise Chapter 4 does not give a complete treatment of the learning of artificial languages. The first studies on the subject were those of Esper (1925). Miller's *Grammarama* project was a heroic but not very successful undertaking in this field (cf. Miller 1967). An evidently more fruitful approach is that of Smith and Braine; a good survey of literature may be found in Smith and Braine (1972).

Since the writing of this text much progress has been made in the study of mother/child conversation, cf. Broen (1972), Holzman (1972), Phillips (1973), Sachs et al (1972), Snow (1972). Van der Geest (1974), and other papers. They all give considerable support to the "intelligent text presentation" model, and add importantly to our knowledge about the role semantic input plays in the acquisition of language.

BIBLIOGRAPHY

Anderson, A.

1968 "Syntax-Directed Recognition of Hand-Printed Two-Dimensional Mathematics", Ph. D. dissertation, Applied Mathematics (Harvard University).

Appel, R.

1971 *Bibliography of Child Language* (=*Publications of the Institute for General Linguistics*) (University of Amsterdam).

Bellugi, U.

1967 "The Acquisition of Negation", unpublished Ph. D. thesis (Harvard University).

Bever, T. G.

1970a "The Influence of Speech Performance on Linguistic Structure", in G. B. Flores d'Arcais and W. J. M. Levelt (eds.) (Amsterdam: North Holland).

1970b "The Cognitive Basis for Linguistic Structures", in J. R. Hayes, (ed.), *Cognition and the Development of Language* (New York: Wiley).

Bever, T. G., J. R. Lackner, and R. Kirk

1969 "The Underlying Structures of Sentences Are the Primary Units of Immediate Speech Processing", *Perception and Psychophysics* 5, 225-234.

1969 "An Autonomic Reflexion of Syntactic Structure", *Neuro-psychologica* 7, 23-28.

Bloom, L.

1970 *Language Development. Form and Function in Emerging Grammars* (Cambridge, Mass.: MIT Press).

Blumenthal, A. L.

1966 "Observations with Self-Embedded Sentences", *Psychonomic Science* 6, 453-454.

1967 "Prompted Recall of Sentence", *J. Verb. Learn. Verb. Beh.* 6, 203-206.

1970 *Language and Psychology* (New York: Wiley).

Blumenthal, A. L., and R. Boakes

1967 "Prompted Recall of Sentences", *J. Verb. Learn. Verb. Beh.* 6, 674-676.

Bobrow, D. G.
1964 "A Question-Answering System for High School Algebra Word Problems", *AFIPS Conference Proceedings* 26, Fall Joint Conference (Baltimore: Spartan Books).

Bobrow, D. G., and J. B. Fraser
1969 "An Augmented State Transition Network Analysis Procedure", *Proceedings of the International Joint Conference on Artificial Intelligence* (Bedford, Mass.: Mitre Corporation).

Border, D. P.
1940 "Adjective-Verb Quotient; A Contribution to the Psychology of Language", *Psychol. Rev.* 3, 310-343.

Bowerman, M. F.
1971 "Acquisition of Finnish", unpublished Ph. D. dissertation (Harvard University).

Braine, M. D. S.
1963 "The Ontogeny of English Phrase Structure. The First Phase", *Language* 39, 1-13.
1971 "On Two Models of the Internalization of Grammars", in D. I. Slobin (ed.), *The Ontogenesis of Grammar. A Theoretical Symposium* (New York: Academic Press).
1972 "The Acquisition of language in Infant and Child", in: C. Reed, (ed.), *The Learning of Language* (New York: Appleton Century Crofts).

Brandt Corstius, H.
1970 *Exercises in Computational Linguistics* (Amsterdam: Mathematisch Centrum).

Bransford, J. D., J. R. Barclay, and J. J. Franks
1972 "Sentence Memory: A Constructive vs. Interpretive Approach", *Cognitive Psychology* 3, 193-209.

Broen, P.
1972 "The verbal environment of the language-learning child", *American Speech and Hearing Association Monograph* No 17.

Brown, R. W.
1957 "Linguistic Determinism and the Part of Speech", *J. Abnorm. Soc. Psychol.* 55, 1-5.
1970 *Psycholinguistics, Selected Papers* (New York: The Free Press).

Brown, R., and C. Fraser
1963 "The Acquisition of Syntax", in C. N. Cofer and B. S. Musgrave (eds.), *Verbal Behavior and Learning: Problems and Processes* (New York: McGraw-Hill).

Campbell, R. N., and R. J. Wales
1970 "The Study of Language Acquisition", in J. Lyons (ed.), *New Horizons in Linguistics* (Harmondsworth: Penguin Books).

Chang, S. K.
1971 "Picture Processing Grammars and Its Applications", *Information Sc.* 3, 121-148.

Chomsky, C.
1969 *The Acquisition of Syntax in Children from 5 to 10* (Cambridge, Mass.: MIT Press).

Chomsky, N.
1957 *Syntactic Structures* (The Hague: Mouton).
1962 "Explanatory Models in Linguistics", in: E. Nagel, P. Suppes, and A. Tarski, (eds.), *Logic, Methodology and Philosophy of Science: Proceedings of the* 1960 *International Congress* (Stanford: Stanford University Press).
1964 "Degrees of Grammaticalness", in J. A. Fodor and J. J. Katz, (eds.), *The Structure of Language. Readings in the Philosophy of Language* (Englewood Cliffs, N. J.: Prentice-Hall).
1965 *Aspects of the Theory of Syntax* (Cambridge, Mass.: MIT Press).
1968 *Language and Mind* (New York: Harcourt, Brace, and World).
Chomsky, N., and M. Halle
1968 *The Sound Pattern of English* (New York: Harper and Row).
Chomsky, N., and G. A. Miller
1963 "Introduction to the Formal Analysis of Natural Languages". in: R. D. Luce, R. R. Bush, and E. Galanter (eds.), *Handbook of Mathematical Psychology* (New York: Wiley).
Clark, H. H.
i.p. "Semantics and Comprehension", in *Current Trends in Linguistics*, 12, T. A. Sebeck, (ed.), (The Hague: Mouton).
Clark, H. H., and Card, S.
1969 "The Role of Semantics in Remembering Comparative Sentences", *J. exp. Psychol.* 82, 545-553.
Coleman, E. B.
1965 "Responses to a Scale of Grammaticalness", *J. Verb. Learn. Verb. Beh.* 4, 521-527.
Curry, H. B.
1961 "Some Logical Aspects of Grammatical Structure", in: R. Jakobson (ed.), *Structure of Language and Its Mathematical Aspects Proc. Twelfth Symposium of Applied Mathematics* (Providence, R. I.: American Mathematical Society).
Donaldson, M.
1970 "Developmental Aspects of Performance with Negatives", in: G. B. Flores d'Arcais and W. J. M. Levelt (eds.), *Advances in Psycholinguistics* (Amsterdam: North Holland).
Ellsworth, R. B.
1951 "The Regression of Schizophrenic Language", *J. Consult. Psychol.* 15, 387-391.
Ervin-Tripp, S.
1971 "An Overview of Theories of Grammatical Development", in D .I. Slobin (ed.), *The Ontogenesis of Grammar, A Theoretical Symposium* (New York: Academic Press).
Esper, E.A
1925 "A Technique for the Experimental Investigation of Associative Interference in Artificial Linguistic Material", *Language Monographs* 1, 1-47.
Feder, J.
1971 "Plex Languages", *Information Sc.* 3, 225-241.

Feldmar, A.
1969 "Syntactic Structure and Speech Decoding: the Judgment of Sequence in Auditory Events, unpublished Masters thesis (University of Western Ontario).
Fillenbaum, S.
1966 "Memory for Gist: Some Relevant Variables", *Language and Speech* 9, 217-227.
1971 "Psycholinguistics", *Annual Review Psychol.* 22, 251-308.
Flores d'Arcais, G. B., and W. J. M. Levelt (eds.)
1970 *Advances in Psycholinguistics* (Amsterdam: North Holland).
Foa, U., and I. Schlesinger
1964 "Syntactic Errors in Sentence Recall (An Analysis of Mehler's Data)", report (Hebrew University, Jerusalem).
Fodor, J. A., and T. G. Bever
1965 "The Psychological Reality of Linguistic Segments", *J. Verb. Learn. Verb. Beh.* 4, 414-420.
Fodor, J. A., and M. Garrett
1967 "Some Syntactic Determinants of Sentential Complexity", *Perception and Psychophysics* 2, 289-296.
Fodor, J. A., M. Garrett, and T. G. Bever
1968 "Some Syntactic Determinants of Sentence Complexity. II Verb Structure", *Perception and Psychophysics* 3, 453-461.
i.p. *The psychology of language* (New York: McGraw Hill).
Fodor, J. A., J. J. Jenkins, and S. Saporta
1965 unpublished report (Center for Advanced Studies, Palo Alto, Calif.).
Foss, D., and R. H. Lynch
1969 "Decision Processes during Sentence Comprehension: Effects of Surface Structure on Decision Times", *Perception and Psychophysics* 5, 145-148.
Freedle, R., and M. Craun
1969 "Observations with Self-Embedded Sentences", *Report* E.T.S. (Princeton, N. J.).
Frijda, N. H.
1972 "Simulation of Human Long-Term Memory", *Psychological Bulletin* 77, 1-31.
Garrett, M.
1970 "Does Ambiguity Complicate the Perception of Sentences?" in G. B. Flores d'Arcais and W. J. M. Levelt (eds.), *Advances in Psycholinguistics* (Amsterdam: North Holland).
Garrett, M., T. G. Bever, and J. A. Fodor
1966 "The Active Use of Grammar in Speech Perception", *Perception and Psychophysics* 1, 30-32.
Geer, J. P., van de
1957 *A Psychological Study of Problem Solving* (Haarlem: De Toorts).
Ginneken, J. van
1904 *Grondbeginselen der Psychologische Taalwetenschap* (Lier).
Gleitman, L. R., and H. Gleitman
1970 *Phrase and Paraphrase. Some Innovative Uses of Language* (New York: Norton).

Gleitman, L. R., H. Gleitman, and F. Shipley
1972 "The emergence of the child as grammarian", *Cognition* 1, 137-164.

Glucksberg, S., and J. H. Danks
1969 "Grammatical Structure and Recall: A Function of the Space in Immediate Memory or of Recall Delay?", *Perception and Psychophysics* 6, 113-117.

Gold, E. M.
1967 "Language Identification in the Limit", *Information and Control* 10, 447-474.

Goodman, N.
1966 *The Structure of Appearance* (Indianapolis: Bobbs-Merrill).

Grasselli, A. (ed.)
1969 *Automatic Interpretation and Classification of Images* (New York: Academic Press).

Green, C., and B. Raphael
1968 "The Use of Theorem Proving Techniques in Question-Answering Systems", *Proc. 23d National Conference of the A.C.M.* (Washington, D. C.: Thompson Book Co.).

Gruber, J. S.
1965 "Topicalization in Child Language", *Foundations of Language* 3, 37-65.

Hewitt, C.
1969 "PLANNER: A Language for Proving Theorems in Robots", *Proc. International Joint Conference on Artificial Intelligence* (Bedford, Mass.: Mitre Corp.).

Hill, A. A.
1961 "Grammaticality", *Word* 17, 1-10.

Hinde, R.
1966 *Animal Behavior* (New York: McGraw-Hill).

Holzman, M.
1972 "The use of interrogative forms in the verbal interaction of three mothers and their children", *Journal of Psycholinguistic Research* 1, 311-36.

Honeck, R. P.
1971 "A study of paraphrases", *J. Verb Learn Verb Beh.* 10, 367-381.

Horning, J. J.
1969 "A Study of Grammatical Inference", *Technical Report CS 139, Stanford Artificial Intelligence Project* (Stanford: Computer Science Department).

Howe, E. S.
1970 "Transformation, Associative Uncertainty, and Free Recall of Sentences", *J. Verb. Learn. Verb. Beh.* 9, 425-431.

Jacobs, R. A., and P. S. Rosenbaum (eds.),
1970 *Readings in Transformational Grammar* (Waltham: Ginn).

Johnson, N. F.
1965 "The Psychological Reality of Phrase-Structure Rules", *J. Verb. Learn. Verb. Beh.* 4, 469-475.

Johnson, S. C.
1967 "Hierarchical Clustering Schemes", *Psychometrica* 32, 241-254.
Kaplan, R. M.
1972 "Augmented Transition Networks as Psychological Models of Sentence Comprehension" *Artificial Intelligence* 3, 77-100.
Katz, J. J.
1964 "Semi-Sentences", in: J. A. Fodor and J. J. Katz (eds.), *The Structure of Language* (Englewood Cliffs, N. J.: Prentice-Hall).
Katz, J. J., and P. Postal
1964 *An Integrated Theory of Linguistic Descriptions* (Cambridge, Mass.: MIT Press).
Kempen, G.
1970 *Memory for Word and Sentence Meanings, A Set-Feature Model* (Nijmegen: Schippers).
Kintsch, W.
1972 "Abstract Nouns: Imagery versus Lexical Complexity", *J. Verb. Learn. Verb. Beh.* 11, 59-65.
Kleene, S. C.
1967 *Mathematical Logic* (New York: Wiley).
Krantz, D., R. D. Luce, P. Suppes, A. Tversky
1971 *Foundations of Measurement I* (New York: Academic Press).
Ladefoged, P.
1967 *Three Areas of Experimental Phonetics* (London: Oxford University Press).
Ladefoged, P., and D. E. Broadbent
1960 "Perception of Sequence in Auditory Events", *Quart. J. Exp. Psychol.* 12, 162-170.
Lakoff, G.
1971 "Presuppositions and Relative Well-Formedness", in D. D. Steinberg and L. A. Jakobovits (eds.), *Semantics* (Cambridge: Cambridge University Press).
Layzer, D.
1972 "Science or Superstition?", *Cognition* 1, 265-299.
Leeuwenberg, E. L. J.
1971 "A Perceptual Coding Language for Visual and Auditory Patterns", *American Journal of Psychology* 84, 307-349.
Lenneberg, E. H.
1967 *Biological Foundations of Language* (New York: Wiley).
Levelt, W. J. M.
1966 "Generatieve Grammatica en Psycholinguistiek II", *Nederlands Tijdschrift voor Psychologie* 21, 367-400.
1967a "Psychological Representations of Syntactic Structures", in T. G. Bever and W. Weksel (eds.), *The Structure and Psychology of Language* (in preparation). Available as *Heymans Bulletin* HB-69-36 Ex, Department of Psychology, Groningen University).
1967b *Over het Waarnemen van Zinnen* (Groningen: Wolters).
1969 "The Scaling of Syntactic Relatedness, A New Method in Psycholinguistics", *Psychon. Sc.* 17, 351-352.

1970a "Hierarchical Chunking in Sentence Processing", *Perception and Psychophysics* 8, 99-103.

1970b "Hierarchical Clustering Algorithms in the Psychology of Grammar", in G. B. Flores d'Arcais and W. J. M. Levelt (eds.), *Advances in Psycholinguistics* (Amsterdam: North Holland).

1970c "A Scaling Approach to the Study of Syntactic Relations", in G. B. Flores d'Arcais and W. J. M. Levelt (eds.), *Advances in Psycholinguistics* (Amsterdam: North Holland).

1972 "Some Psychological Aspects of Linguistic Data", *Linguistische Berichte* 17, 18-30.

Levelt, W. J. M., and M. Bonarius

1968 *Suffixes as Deep Structure Clues.* (= *Heymans Bulletin* HB-68-22 EX) (Groningen University).

Levelt, W. J. M., W. Zwanenburg, and G. R. E. Ouweneel

1970 "Ambiguous Surface Structure and Phonetic Form in French", *Foundations of Language* 6, 260-273.

Lipkin, F. S., and A. Rosenfeld (eds.)

1970 *Picture Processing and Psychopictorics* (New York: Academic Press).

Loosen, F.

1972 "Cognitieve Organisatie van Zinnen in het Geheugen", dissertation (University of Louvain).

Luria, A. R.

1961 *The Role of Speech in the Regulation of Normal and Abnormal Behaviour* (London: Pergamon).

Lyons, J.

1968 *Introduction to Theoretical Linguistics* (Cambridge: Cambridge University Press).

Maclay, H., and M. D. Sleator

1960 "Responses to Language: Judgments of Grammaticalness", *International Journal of Applied Linguistics* 26, 275-282.

Marks, L. E.

1965 "Psychological Investigations of Semi-Grammaticalness in English", Ph. D. thesis (Harvard).

Martin, E.

1970 "Toward an Analysis of Subjective Phrase Structure", *Psych. Bull.* 74, 153-166.

Masters, J. M.

1970 "Pushdown Automata and Schizophrenic Language", unpublished report (University of Sydney).

Matthews, G. H.

1962 "Analysis by Synthesis of Sentences of Natural Languages", in: *Proceedings International Congress on Machine Translation and Applied Language Analysis* (London: H. M. S. O.).

Matthews, W. A.

1968 "Transformational Complexity and Short Term Recall", *Language and Speech* 11, 120-128.

McMahon, L.
1963 "Grammatical Analysis as Part of Understanding a Sentence", unpublished Ph. D. thesis (Harvard University).
McNeill, D.
1966 "Developmental Psycholinguistics", in F. Smith and G.A. Miller (es.), *The Genesis of Language, A Psycholinguistic Approach* (Cambridge, Mass.: MIT Press).
1971 "The Capacity for the Ontogenesis of Grammar", in D. I. Slobin (eds.), *The Ontogenesis of Grammar, A Theoretical Symposium* (New York: Academic Press).
Mehler, J.
1963 "Some Effects of Grammatical Transformation on the Recall of English Sentences", *J. Verb. Learn. Verb. Beh.* 2, 346-351.
Mehler, J., and B. de Boysson-Bardies
1971 "Psycholinguistique. Messages et Codage Verbal. II Etudes sur le Rappel de Phrases", *Année Psychologique* 71, 547-581.
Mehler, J. and P. Carey
1968 "The Interaction of Veracity and Syntax in the Processing of Sentences", *Perception and Psychophysics* 3, 109-111.
Mehler, J., and G. A. Miller
1964 "Retroactive Interference in the Recall of Simple Sentences", *British Journal of Psychology* 55, 295-301.
Menyuk, P.
1969 *Sentences Children Use* (Cambridge, Mass.: MIT Press).
Michotte, A.
1964 *The Perception of Causality* (London: Methuen).
Miller, G. A.
1956 "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information", *Psychol. Rev.* 63, 81-97.
1962 "Decision Units in the Perception of Speech", *IRE Transactions on Information Theory, IT* 8 2, 81-83.
1967 "Project Grammarama", *The Psychology of Communication. Seven Essays* (New York: Basic Books — Penguin Books, Pelican Series, 1970.)
Miller, G. A., and N. Chomsky
1963 "Finitary Models of Language Users", in R.D. Luce, R.R. Bush and E. Galanter (eds.), *Handbook of Mathematical Psychology* (New York: Wiley).
Miller, G. A., and McKean
1964 "A Chronometric Study of Some Relations between Sentences", *Quart. J. Exp. Psychol.* 16, 297-308.
Miller, G. A., and D. McNeill
1968 "Psycholinguistics", in G. Lindzey and E. Aaronson (eds.), *Handbook of Social Psychology* 3 (Reading, Mass.: Addison Wesley).
Miller, G. A., and J. A. Selfridge
1950 "Verbal Context and the Recall of Meaningful Material", *American Journal of Psychology* 63, 176-185.

Miller, W. R., and S. M. Ervin
1964 "The Development of Grammar in Child Language", in U. Bellugi and R. Brown (eds.), *The Acquisition of Language* (= *Monographs of the Society for Research in Child Development*).
Minsky, M. (ed.)
1968 *Semantic Information Processing* (Cambridge, Mass.: MIT Press).
Moore, T. E.
1972 "Speeded Recognition of Ungrammaticality", *J. Verb. Learn. Verb. Beh.* 11, 550-560.
Neisser, U.
1966 *Cognitive Psychology* (New York: Appleton Century Crofts).
Nowakowska, M.
1973 *Language of Motivation and Language of Action* (The Hague: Mouton).
Osgood, C. E.
1963 "On Understanding and Creating Sentences", *Amer. Psychol.* 18, 735-751.
Paivio, A.
1971a "Imagery and Deep Structure in the Recall of English Nominalisations", *J. Verb. Learn. Verb. Beh.* 10, 1-12.
1971b *Imagery and Verbal Processes* (New York: Holt, Rinehart, and Winston).
Paul, H.
1886 *Prinzipen der Sprachgeschichte* (Halle: Niemeyer).
Perchonock-Schaefer, E.L.
1971 Comprehension of Self-Embedded Sentences as a Function of Context", unpublished paper.
Phillips, J.
1973 "Syntax and Vocabulary of Mother's Speech to Children: Age and Sex Comparisons", *Child Development* 44, 182-185.
Phillips, J. R., and G. A. Miller
1966 "An Experimental Method to Investigate Sentence Comprehension", unpublished paper (Center for Cognitive Studies, Harvard University).
Quillian, M. R.
1967 "Word Concepts: A Theory and Simulation of Some Basic Semantic Capabilities", *Behavioral Science* 12, 410-430.
1968 "Semantic Memory", in M. Minsky (ed.), *Semantic Information Processing* (Cambridge, Mass.: MIT Press).
Quirk, R., and J. Svartvik
1965 *Investigating Linguistic Acceptability* (The Hague: Mouton).
Raphael, B.
1968 "SIR: A Computer for Semantic Information Retrieval", in M. Minsky, (ed.), *Semantic Information Processing* (Cambridge, Mass.: MIT Press).
Reber, A. S.
1973 "Locating Clicks in Sentences: Left, Center and Right", *Perception and Psychophysics* 13, 133-138.

Reber, A. S., and J. R. Anderson
1970 "The Perception of Clicks in Linguistic and Non-Linguistic Messages", *Perception and Psychophysics* 8, 81-89.

Robinson, J. A.
1965 "A Machine-Oriented Logic Based on the Resolution Principle", *Journal of the ACM* 8, 536-541.

Rosenfeld, A.
1969 *Picture Processing by Computer* (New York: Academic Press).

Rosenfeld, A., and J. L. Pfaltz
1969 "Web Grammars", *Proc. Joint International Conference on Artificial Intelligence* (Washington, D. C.).

Rumelhart, D. E., P. H. Lindsay, and D. A. Norman
1972 "A Process Model for Long-Term Memory", in E. Tulving and W. Donaldson (eds.), *Organization of Memory* (New York: Academic Press).

Sachs, J., R. Brown, and R. Salerno
1972 "Adults' Speech to Children". Paper presented at the *International Symposium on First Language Acquisition*, Florence, Italy.

Sager, N.
1967 "Syntactic Analysis of Natural Language", in F. Alt and M. Rubinoff (eds.), *Advances in Computers* (New York: Academic Press).
1971 "The String Parser for Scientific Literature", *Natural Language Processing*, R. Rustin, (ed.), (Englewood Cliffs, Prentice Hall).

Saussure, F., de
1916 *Cours de Linguistique Générale* (Paris).

Savin, H. B., and E. Perchonock
1965 "Grammatical Structure and the Immediate Recall of English Sentences", *J. Verb. Learn. Verb. Beh.* 4, 348-353.

Schaerlaekens, A. M.
1973 *The Two-word Sentence in Child Language Development.* (The Hague: Mouton).

Schank, R. C.
1972 "Conceptual Dependency: A Theory of Natural Language Understanding", *Journal of Cognitive Psychology* 3, 532-631.

Schlesinger, I. M.
1966 "The Influence of Sentence Structure on the Reading Process", *Tech. Rep.* 24, *U. S. Office of Naval Research.*
1971 "Production of Utterances and Language Acquisition", in D.I. Slobin (ed.), *The Ontogenesis of Grammar. A Theoretical Symposium* (New York: Academic Press).

Seuren, P. A. M.
1969 *Operators and Nucleus: A Contribution to the Theory of Grammar* (Cambridge: Cambridge University Press).

Shaw, A. S.
1969 "A Formal Picture Description Scheme as a Basis for Picture Processing Systems", *Information and Control* 14, 9-52.

Simmons, R. F.
1966 "Storage and Retrieval of Aspects of Meaning in Directed Graph Structures", *Communications of the ACM* 9, 211-214.
1970 "Natural Language Question Answering Systems: 1969", *Communications of the ACM* 13, 15-29.
Simmons, R.F., S. Klein, and K.L. McConlogue
1962 "Toward the Synthesis of Human Language Behavior", *Behavioral Science* 7, 402-407.
Simmons, R. F., and J. Slocum
1972 "Generating English Disourse from Semantic Networks", *Communications of the ACM* 15, 891-905.
Sinclair-de Zwart, H.
1967 *Acquisition du Langage et Développement de la Pensée* (Paris: Dunod).
Slobin, D. I.
1966 "Grammatical Transformations and Sentence Comprehension in Childhood and Adulthood", *J. Verb. Learn. Verb. Beh.* 5, 219-227.
1970 "Universals of Grammatical Development in Children", in G. B. Flores d'Arcais and W.J.M. Levelt (eds.), *Advances in Psycholinguistics* (Amsterdam: North Holland).
1971a "Data for the Symposium", in D. I. Slobin (ed.), *The Ontogenesis of Grammar, A Theoretical Symposium* (New York: Academic Press).
1971b "Development Psycholinguistics", in W. O. Dingwall (ed.), *A Survey of Linguistic Science* (College Park: Linguistics Program, University of Maryland).
Smith, K. H., and M. D. S. Braine
1972 "Miniature Languages and the Problem of Language Acquisition, 1972", in T. G. Bever and W. Weksel (eds.), *The Structure and Psychology of Language.*
Snow, C. E.
1972 "Mother's Speech to Children Learning Language", *Child Development* 43, 549-565.
Staats, A. W.
1971 "Linguistic-Mentalistic Theory versus an Explanatory S-R Learning Theory of Language Development", in D. I. Slobin (ed.), *The Ontogenesis of Grammar. A Theoretical Symposium* (New York: Academic Press).
Steinthal, H.
1855 *Grammatik, Logik und Psychologie* (Berlin: Dümmler).
Stolz, W. S.
1967 "A Study of the Ability to Decode Grammatically Novel Sentences", *J. Verb. Learn. Verb. Beh.* 6, 867-873.
Sutherland, N. S.
1973 "Intelligent Picture Processing". *Conference on the Evolution of the Nervous System and Behavior* (Florida State University).
Suppes, P.
1969 "Stimulus-Response Theory of Finite Automata", *J. Mathem. Psychol.*, 327-355.

1970 "Probabilistic Grammars for Natural Languages", *Synthese* 22, 95-116.

Trabasso, T., H. Rollins, and E. Shaughnessy

1971 "Storage and Verification Stages in Processing Concepts", *Cognitive Psychology* 2, 239-289.

Tulving, E., and W. Donaldson (eds.)

1972 *Organization of Memory* (New York: Academic Press).

Tulving, E., and J. E. Patkau

1962 "Concurrent Effect of Contextual Constraint and Word Frequency on Immediate Recall and Learning of Verbal Material", *Canadian Journal of Psychology* 16, 83-95.

Turner, E. A. and R. Rommetveit

1968 "Focus of Attention in Recall of Active and Passive Sentences", *J. Verb. Learn Verb. Beh.* 7, 543-548.

Uhlenbeck, E. M.

1964 "Discussie", in H. Lunt (ed.), *Proceedings of the Ninth Congress of Linguists* (The Hague: Mouton).

Van der Geest, T., C. Snow, and A. Drewes-Drubbel

1974 "Developmental Aspects of Mother-Child Conversation" (= *Publications of the Institute of General Linguistics*) (University of Amsterdam).

Villiers, P. A., de, and J. G. de Villiers

1972 "Early Judgments of Semantic and Syntactic Acceptability by Children", *J. Psycholing. Res.* 1, 299-310.

Wagenaar, W. A.

1972 *Sequential Response Bias* (Rotterdam: Bronder).

Wason, P. C., and P. N. Johnson-Laird

1964 *The Psychology of Reasoning: Structure and Content* (London: Batsford).

Weir, R.

1962 *Language in the Crib* (The Hague: Mouton).

Weizenbaum, J.

1966 "ELIZA", *Communications of the ACM* 9, 36-45.

Whorf, B. L.

1956 *Language, Thought and Reality: Selected Papers*, J. B. Carroll (ed.), (New York: Wiley).

Winograd, T.

1972 "Understanding Natural Language", *Cognitive Psychology* 3, 1-191.

Woods, W. A.

1970 "Transition Network Grammars for Natural Language Analysis", *Communication of the ACM* 13, 591-606.

Wright, P.

1968 "Sentence Retention and Transformational Theory", *Quart.J.Exp. Psychol.* 20, 265-272.

Wundt, W.

1900 "Völkerpsychologie I and II", *Die Sprache* (Leipzig).

1907 "Über Ausfrage-Experimente", *Psychol. Studien* 3, 301-360.

1908 "Kritische Nachlese zur Ausfragemethode", *Arch. f. d. ges. Psychol.* 11, 445-459.

Yngve, V. H.
1961 "The Depth Hypothesis". in R. Jakobson (ed.), *Structure of Language and Its Mathematical Aspects* (Providence, R. I.: American Mathematical Society).

Ziff, P.
1964 "On Understanding 'Understanding Utterances'", in J.A. Fodor and J. J. Katz (eds.), *The Structure of Language* (Englewood Cliff, N. J.: Prentice-Hall).

Zwicky, A. M., J. Friedman, B. C. Hall, and D. E. Walker
1965 "The MITRE Syntactic Analysis Procedure for Transformational Grammars", *Information System Language Studies* 9 (The MITRE Corporation, Beford, Mass.).

AUTHOR INDEX

SUBJECT INDEX

(Italicized numbers refer to definitions)

Postscript

What has become of formal grammars in linguistics and psycholinguistics?

The aim of this postscript cannot be to review the theory and language science applications of formal languages and automata, as developed since the mid 1970s. That would require more than a three-volume work. I will, rather, touch upon just a few developments that seem to me of special relevance to linguists and psycholinguists. I will do this under the three main headings I used in *Formal Grammars.*

Formal languages and automata

Of special linguistic relevance has been the construction of tree grammars and tree automata. The original grammar types in the Chomsky hierarchy, as well as the corresponding automata were string handling devices. Their inputs and outputs were strings of symbols. Their structural descriptions consisted of the derivation or recognition trees as they emerged in the stepwise application of the rules. The newly introduced tree grammars and tree automata operate on trees, not on strings. In that sense they are operations on structural descriptions. A tree grammar generates a set of trees. The 'frontier' of a tree is its bottom string of symbols, which consists of (at least one) terminal and/or non-terminal nodes. The tree set generated by a tree grammar is the set of 'completed' trees derived from one or more special S-rooted initial trees. A tree is completed if its frontier consists of terminal elements only. The language generated by this tree grammar is the 'yield' of this tree set, i.e., the set of its (terminal) frontiers.

The strong generative power of the grammar is the set of terminal trees generated for this language. The theory of tree automata and grammars originated from Büchi (1960). A recent overview is presented in Comon et al. (2007).

The generative power of types of tree grammars does not simply match the power of types in the Chomsky hierarchy. An interesting equivalence holds

between context-free languages and the languages generated by so-called regular tree grammars (cf. Gékseg & Steinby 1997). More generally, however, the power of tree grammars (and automata) straddles the power levels in the Chomsky hierarchy, which has interesting applications in linguistic theory (see below).

Ellis (1971) introduced the notion of probabilistic tree grammars and automata, which generate/accept probabilistic languages. He showed that context-free probabilistic languages (as defined in *Formal Grammars* II, 3.4) can be fully characterized by probabilistic tree automata.

Linguistic applications

An appropriate level of generative power

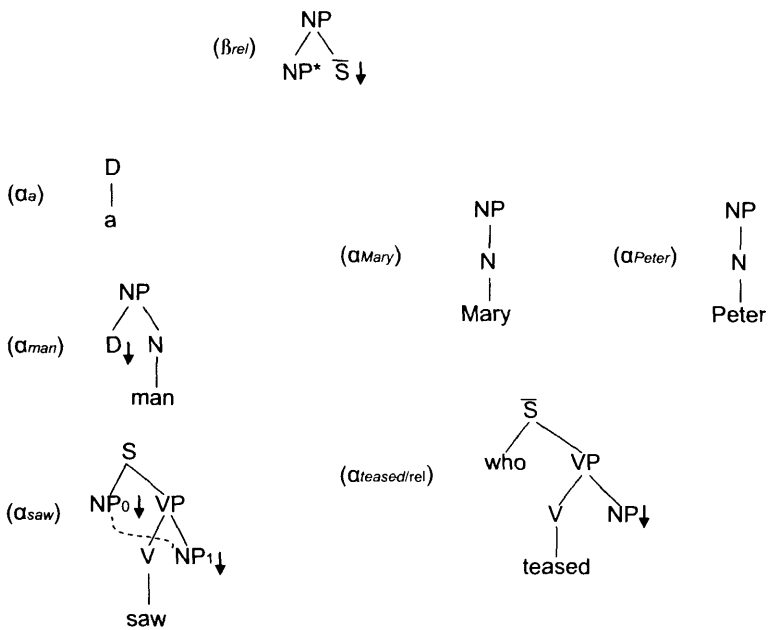
A perennial issue in formal linguistics has been the characterization of the 'right' level of grammatical power for natural language grammars. A major motivation for Chomsky's original work on formal grammars had been to show that finite state automata or regular grammars cannot characterize natural languages. Here the recursive self-embedding property of natural languages transcended the capacity of this type of system. Context-free grammars faired a lot better, but reached their limits in dealing with crossed and other long-distance dependencies. This was all comprehensively reviewed in *Formal Grammars* II. Initially, the move to transformational grammars seemed to be a promising one for handling such problems. However, Peters & Ritchie's proof (1973, see *FG* II, chapter 5) that the then most advanced transformational grammar, Chomsky's *Aspects* model, has the generative capacity of a Turing machine (see *FG* II, chapter 5), showed that simple or 'natural' solutions were not yet around. In his interview with Huybregts and van Riemsdijk, Chomsky (1982, p. 15) remarked:

> "The systems that capture other [than the context-free WL] properties of language, for example transformational grammar, hold no interest for mathematics. But I do not think that that is a necessary truth. It could turn out that there would be richer or more appropriate mathematical ideas that would capture other, maybe deeper properties of language than context-free grammars do. In that case you have another branch of applied mathematics which might have linguistic consequences. That could be exciting" [1]

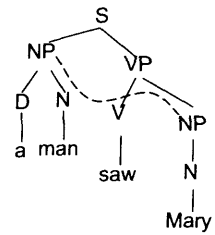
Such exciting formalisms were, then, about to emerge. There exists now a class of equivalent grammars, called 'mildly context-sensitive grammars' (MCSGs), among them linear indexed grammars, head grammars, combinatory categorial grammars

1. I thank Aravind Joshi for this reference.

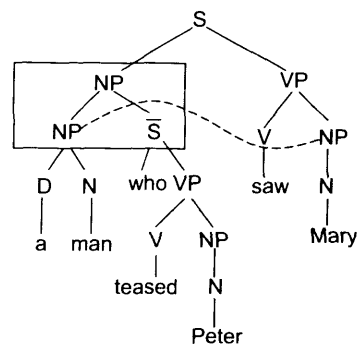
and tree adjoining grammars, that seem to have just the right level of generative capacity to overcome the linguistic limitations of context-free grammars without 'over-generating' to the level of context-sensitive grammars, or worse, of recursively enumerable languages. Let us, for a moment, consider the case of the tree adjoining grammars (TAG) conceived of by Joshi and his co-workers (cf. Joshi & Schabes 1997 and further references there). A TAG has a set of 'elementary trees', consisting of 'initial trees' and 'auxiliary trees'. In the following example seven such elementary trees from a TAG are exemplified (adapted from Joshi & Schabes, p. 75):



New trees are generated from the elementary trees by two operations, *substitution* and *adjoining*. Substitution is the operation by which the root of an initial tree (marked α above) substitutes for a node in the frontier of some tree which is marked for substitution (indicated by ↓ on that node). Substitution is allowed when certain constraints are met. In this simple example the only constraint is that the node labels of the marked node and the replacing root node match. (In the full grammar a Boolean comparison of feature sets is to be performed). So, for instance, the NP root node of the *Mary* tree can substitute for the $NP_1\downarrow$ node of the saw tree. The following tree is entirely derived by substitution, using four of the initial trees above. The order of substitution is irrelevant. The terminal frontier of the tree is the sentence *a man saw Mary*.



Because a TAG has only a finite set of initial trees (to be motivated below), substitution only allows for the derivation of a finite set of derived trees. Recursion in a TAG is handled by the other operation, adjoining. It requires the insertion of an auxiliary tree (marked β above). An auxiliary tree has some leaf node (marked *) that is identical to its root node. That is the case for the β*rel* tree above, which has a root node NP and a leaf node NP* in its frontier. This tree can be (does not *have* to be) inserted into the above derived tree by detaching the NP tree dominating *a man*, attaching the *auxiliary* tree to the 'freed' NP node and finally attaching the detached a man tree to the starred NP node of the inserted auxiliary tree. The result is the derived tree below (with the inserted tree indicated by a square frame).



Substituting the S node by the *teased* tree and substituting the *Peter* tree in its NP node, results in a tree dominating the sentence *a man who teased Peter saw Mary.* Repeated adjunction of the auxiliary tree generates sentences such as *the man who teased Mary who teased Peter saw Mary*, etc.

This is just enough example[2] to notice that syntactic dependences (such as the c-command relation between NP_0 and NP_1 in the initial *saw* tree above, marked by a dotted line both there and in the two following derived trees) can be defined

2. without any linguistic pretensions.

within the elementary trees. Recursion is exclusively realized by adjunction. Adjunction can move the elements between which a dependency relation exists arbitrarily far apart, thus accounting for long-distance dependences. TAGs elegantly separate the syntactic functions of dependency and recursion.

Like the other MCSGs, TAGs can handle crossed dependences, such as in Dutch complement constructions (e.g. ... *dat Jan Piet Marie zag laten zwemmen – that John Peter Mary saw let swim*, where John sees, Peter lets and Mary swims) (Stabler, 2004), which are problematic for a context-free account, and they can handle other similar phenomena. Their generative power is 'slightly' more than context-free, but less than context-sensitive. TAG grammars have equivalent accepting automata, called 'embedded push-down automata' (EPDA) and a parser has been defined which, incrementally, parses a string in the TAG's language 'from left to right', ultimately delivering the appropriate tree for the sentence. Parsing of TAGs, and probably more generally of MCSGs, is polynomial (in the worst case with $O(n^6)$ time and $O(n^4)$ space), hence hardly worse than for context-free grammars.

A closely related, but alternative tree adjoining grammar, 'Performance Grammar' (PG) has been introduced by Kempen & Harbusch (1998). Like in TAG, its trees are 'lexicalized' (see below) and it has essentially the same substitution operations as in TAG. However, there is no adjoining operation. Instead, the grammar has a topological or 'linearization' component which can handle recursion and long distance dependences. Each (initial) lexical tree is combined with a linear array, which is a topology for the ordering of the lexical item's arguments. This topology goes back to the time-honored tradition in Germanic linguistics of distinguishing syntactic Vorfeld (forefield), Mittelfeld (midfield), and Nachfeld (endfield). The key recursive property of PG, which at the same time generates the appropriate linearization, is that constituents may move out of their 'own' array and receive a position in an array located at a higher level. The linearization operation is implemented as a finite state automaton. This mechanism provides PG with the generative power of MCSGs. At the same time, it elegantly handles the relatively free word order of German (Harbusch & Kempen 2002).

Lexicalization of grammars

Another major development since the 1970s is the 'lexicalization' of grammars. The first context-free, and also transformational grammars, were rule systems fully abstracted from the ultimate lexical insertions. A rule such as VP $\rightarrow$ V + NP applies whatever the ultimate lexical insertions (such as *saw* and *Mary*). The lexicon was the linguist's last concern, as it were. This approach was already challenged by the generative semanticists (see below), but the major later innovation was to

characterize lexical items as syntactic structures, which would then 'bottom-up' interact or 'unify' to generate syntactic phrase structures. *Formal Grammars* (II, p. 94) already pleaded for such a move, but the idea caught the (psycho)linguistic community's imagination by Kaplan and Bresnan's (1982) work on their Lexical Functional Grammar (LFG). Still, the core notion had been around in categorial grammar since it was created by Ajdukiewicz (1935). This formalism is extensively discussed in *Formal Grammars* II 4.2). Each word has a specific syntactic category, which allows it to 'hook up' (or unify) with other words that have some 'hookable' category, more or less like Lego pieces.

The lexicalization of grammars was a major asset for psycholinguistic applications. Language users have a huge mental lexicon and they know the syntactic affordances of these lexical items. It is a natural idea that the listener, who recognizes one word after another, indeed, on-line, 'unifies' their syntax, thus incrementally building up the phrase structure of the incoming sentence. That has been Steedman's motivation all along in developing his Combinatory Categorial Grammar (CCG) (cf. Steedman 2000), which mediates both the on-line semantic interpretation and the interpretation/generation of prosody. My own book *Speaking* (1989) makes extensive use of LFG.

Joshi and Schabes' (1997) version of TAG is completely 'lexicalized'. This means, first, that all elementary trees have at least one lexical term in their frontier; it is their 'lexical anchor' (this is the case for six of the seven elementary trees above). It means, second, that all lexical items of the language figure as anchor in a finite number of elementary trees (at least once). This explains the earlier finite generative power of the substitution operation in TAG and also in PG. This type of lexicalization also provides a natural way of handling fixed expressions, such as *kick the bucket*. They have their own V-rooted elementary tree with a multiple lexical anchor. This is presently receiving interesting applications in language acquisition research (see below).

Semantics

Many such fixed expressions, in particular idioms, violate the 'principle of compositionality' (PC), which says that the meaning of the whole is a function of the meaning of the parts and the way they are syntactically combined. Five to ten percent of the words we speak are part of some fixed expression (Sprenger et al. 2006), which means that the principle could still have wide application in language use. In fact, it is basic to all formal semantics. The state of semantics discussed in *FG* II 3.3 was the then raging conflict between the generative semantics and the interpretative semantics approach. Both incorporated PC but in quite different ways. In the first

approach syntax was itself semantic. Deep structures were as much syntactic as semantic representations (hence they were fully lexicalized) and transformations were, presumably, meaning preserving or 'paraphrastic'. Soon a 'prelexical syntax' developed, in which lexicalized subtrees could be transformationally replaced by other subtrees, in particular by unitary lexical items (for instance replacing 'cause to become not alive' by 'kill'). This obviously raised the power of generative semantics to Turing machine level.

In interpretative semantics the underlying deep structures were purely syntactic entities, but could receive semantic interpretation after lexical insertion. Here an 'autonomous' semantics had to be developed which would provide the 'logical form' associated with some deep or (later) surface syntactic structure. However, it soon turned out (see *FG* II, p. 109) that transformations could not preserve the semantic interpretation of quantifiers in deep or underlying structure (*John sings and dances* can be paraphrased as *John sings and John dances*, but *One boy sings and dances* cannot be paraphrased as *One boy sings and one boy dances*).

It is both beyond the aim of this postscript and beyond my competence to sketch the developments in formal semantics since these early beginnings. A few remarks should suffice. The generative semantics approach survived a rather dramatic history of upheavals, ultimately producing a broad and formal treatment of meaning in language as it is represented in the language user's mind. That has, in its later developments, largely been Pieter Seuren's achievement, now available as Seuren (2009). It was in generative semantics and in Harman (1970) that a truth-functional semantics was first introduced in generative linguistics. Seuren's (1969) 'operators' (quantifiers, modal, tense and other operators) were truth-functional operations on their arguments, the so-called nuclei. The nuclei are the elementary propositions, which can be negated, questioned, etc. Harman extended the operator approach to the nuclei themselves, defining the main verb as an operator on the other phrases as arguments. It is now commonplace to analyze linguistic expressions as function-argument structures, but that idea was totally absent in early interpretative semantics. Although the truth-functional approach is now basic to any formal semantics, the cognitive perspective which has always been essential in generative semantics, has made the latter a laboratory for studying the many other aspects of meaning involved in the listener's on-line interpretation of language. Among them are the fascinating complexities of presupposition, discourse, anaphora, metaphor and lexical meaning. Here, Seuren (2009) provides a rich source for psycholinguists.

The early autonomous approach was completely transformed under the influence of Montague's (1970) truth-functional approach to natural language semantics. It was in particular Barbara Partee who managed to fuse the Chomskyan

and Montagovian traditions, using lambda extraction to handle variable binding (see her own wonderful account of these and later developments in Partee 1997). Basic to Montague's handling of compositionality is the 'rule-by-rule' correspondence between syntax and semantics. Each syntactic composition of 'smaller' or 'lower-level' syntactic entities goes with a semantic interpretation of the higher-level entity in terms of the semantic interpretations of the lower-level units. This homomorphism between syntax and semantics has found wider application, for instance in TAG semantics. There it is not the phrase-structural relations of the derived tree that receive semantic interpretation. Rather, each substitution or adjoining application goes 'synchronously' with a corresponding semantic interpretation. It is therefore the derivational history (represented in a 'derivation tree') that provides the step by step correspondence to semantic operations that generate the 'logical form' of the linguistic expression. Although Partee always intended the formal semantics developed in the Montagovian tradition to be a theory of meaning in the mind, it didn't really conquer the hearts of psycholinguists. The 'possible worlds' framework and its somewhat daunting formal rendering, rightly or wrongly, always remained somewhat unapproachable for the psycholinguist studying the *process* of 'on-line' semantic interpretation in the language user's mind.

A third major approach has been Jackendoff's (2002). Coming from the interpretative tradition, but doing away with Chomsky's syntactocentrism, he developed, in much detail, a semi-formal cognitive theory of grammar with three parallel generative components, a conceptual, a syntactic and phonological one. In *Speaking* (1989) I gratefully used a version of Jackendoff's conceptual component. The system is still 'interpretative' in that it handles semantic interpretation by means of 'correspondence rules' that hold between conceptual and syntactic structures (just as phonological interpretation is handled by correspondence rules between phonological and syntactic structures). Jackendoff handles a great variety of meaning aspects, which have obvious psycholinguistic applications.

Many of these meaning aspects have conversational impact. A major challenge is to sort out how semantics and pragmatics interact in conversational implicature and anaphora. Levinson (2000) advocated a strong pragmatic stance here, which is probably less amenable to formalization than, for instance, Seuren's more formal semantic approach to these matters.

Probabilistic grammars and linguistic intuitions.

When I wrote *Formal Grammars*, probabilistic grammars were generally avoided by linguists as 'not done'. "It must be recognized that the notion of 'probability of a sentence' is an entirely useless one, under any known interpretation of this term",

Chomsky wrote (see *FL* II, p. 174). Mine was the only text for linguists around that treated them. The only linguistic example of a probabilistic context-free grammar I could find at the time was the one that Patrick Suppes, always averse to current dogma, had written for a child language corpus (see *FG* II 6.2). The next one I came across was the probabilistic CFG Wolfgang Klein published, as early as 1974, for handling a large corpus of untutored second language (German) acquisition data. Meanwhile, however, stochastic approaches to grammars, automata, parsers, inference devices, automatic translation have exploded. A landmark publication was Charniak (1993), in which a wide range of (in some cases still potential) linguistic applications of probabilistic grammars was treated. Since then, the computational analysis of ever larger natural language corpora has stimulated the further development and use of probabilistic tree grammars and automata, which Ellis (1970) had initiated. Shabes (1992), for instance, introduced probabilistic TAGs, with Resnick (1994) applying them to natural language parsing. For a more recent review of stochastic tree approaches to natural language processing, see Knight & Graehl (2005).

One issue addressed in *FL* II, chapter 1, was the status of linguistic intuitions. At that time, generative linguistics was, as an empirical science, largely intuition-based. The grammaticality judgment played the essential role in telling 'grammatical' from 'ungrammatical' strings. In *FL* III I argued that grammaticality judgments are the outcome of metalinguistic judgment, a psychological process whose workings were still largely in the dark. And worse, I could provide some empirical evidence for their alarming unreliability. I analyzed various causes of this unreliability and proposed a range of empirical procedures to improve on this empirical weakness in linguistic practice. It didn't help much. Nor did Bard et al.'s (1996) careful proposal to use easily applicable magnitude estimation. Many linguists still mark strings as ungrammatical (by '*'), without providing their empirical reasons for doing so. Their tacit assumption is that they are dealing with god-given 'grammaticality', not with human 'acceptability'.

Luckily, the use of large corpora has meanwhile reduced the importance of grammaticality judgments. But it also raised the new issue whether some consistent relation exists between grammaticality judgments and statistical corpora data. Bresnan's (2006) study of linguistic intuitions seems to show that there is, indeed, a close relation between naturalness/acceptability judgments and corpus frequency data. But that requires judgments to be made in the appropriate textual context. In one experiment she had subjects rate the 'naturalness' of two alternative sentential continuations of short texts from a natural language corpus. The two sentences both contained a dative verb construction, but differed in whether the construction was prepositional or double object (for instance: *because he brought*

the pony to my children versus *because he brought my children the pony*). It turned out that the naturalness judgments were highly predictable from the syntactic probabilities in the corpus model. What about really 'ungrammatical' sentences? In a second experiment subjects judged dative constructions that linguists usually mark as ungrammatical, such as *the dealer pushes someone the pot.* The stochastic corpus model, however, predicted contexts in which such sentences would appear, even with higher probability than the 'grammatical' alternatives (such as *the dealer pushes the pot to someone*). Again naturalness judgments followed the corpus model, not the linguists' judgments. Bresnan's conclusion was that grammaticality judgments reflect implicit knowledge of syntactic probabilities.

Others, however, observed systematic disagreement between judged grammaticality and corpus probability. There is even talk about a 'grammaticality-frequency gap'. Kempen and Harbusch (2005, 2008) compared available judgment data for various German word order patterns (German allows for six different orderings of subject and objects in double object sentences) to frequencies of occurrence in two text corpora. One surprising finding was that lower rated word orders never occurred in the corpus. (It was unlikely that corpus size was an important factor here.) Another was that similarly highly rated word order types turned out to have quite different corpus frequencies. Extensive analyses of these data led the authors to make specific claims about the grammaticality judgment process. Judgers will normally try to internally generate the target sentence. If it works, it will be judged (highly) 'grammatical'. If it doesn't work, the subject will generate a sentence with the same semantic gist and then judge its *similarity* to the target sentence. In this way, highly unlikely sentences can still (by similarity to likely sentences) be judged as (somewhat) grammatical although they (or their type) didn't make it into the corpus. In other words, this similarity factor would deserve careful control, in addition to all the other reliability undermining factors I discussed in *FL* III, chapter 1. The Kempen & Harbusch studies cannot be directly compared to Bresnan's, because the grammaticality judgments were made on sentences in isolation. One wonders in particular whether the grammaticality-frequency gap will also appear in her data when the same experimental sentences are judged in isolation.

My one original psychological contribution to the study of linguistic intuitions in *Formal Grammars* is a mathematical theory of syntactic relatedness intuitions (*FG* III, 27-65). We have, for instance, the strong intuition that in the sentence *John ordered a pizza* the syntactic relation between *a* and *pizza* is much stronger than between *John* and *a* or between *John* and *pizza.* The mathematical theory relates such cohesion intuitions to the structural descriptions grammars adduce to sentences. That makes it possible to test the descriptive adequacy of (different kinds of) grammars in an entirely new way. The initial empirical tests, reported

in *FL*, showed that transformational dependency grammars excelled on this test (see Schils 1983 for more extensive data and analyses). The method has also been successfully applied by Fodor et al. (1980) to distinguish between alternative 'underlying' structural descriptions. Take the two sentences (1) *the captain persuaded the passengers to leave* and (2) *the captain expected the passengers to leave.* Here one would expect the syntactic cohesion between *captain* and *passengers* to be stronger in (1) than in (2). This is because in the former, but not in the latter, the two items are in the same underlying clause (the *captain persuaded the passengers S*). And this was indeed found in the rating experiment. Fodor et al. could then use this sensitive procedure to test whether causative verbs (as in *John killed Mary*) have an underlying structure like (2) (i.e., *John caused Mary to die*), where *John* and *Mary* do not share a clause, or rather the simple 'non-definitial' one *John killed Mary*, just like for *John liked Mary*, where they do. The rating results were crystal-clear: the latter was the case. There is no evidence for a 'definitial' underlying structure of causative verbs. It is my impression that cohesion judgments are more reliable, less vulnerable than grammaticality judgments. They are, moreover, *direct* tests of descriptive adequacy, as opposed to grammaticality judgments, which concern strings, not structures. In short, intuitions of syntactic cohesion should still be embraced by linguists.

Psycholinguistic applications

Incrementality

An essential feature of modern theories of speaking and speech comprehension is incrementality. Speakers work with quite restricted 'look-ahead' (Levelt 1989). And the evidence is overwhelming that listeners interpret speech largely 'on line' as it comes in. All relevant knowledge (phonetic, phonological, morphological, syntactic, semantic, pragmatic) is in no time applied to any next incoming signal. Interpretations are rarely (but not never!) delayed or revised. In the early 1970s this insight was not yet around. Tom Bever's 'garden path' sentence *the horse raced past the barn fell* was on everybody's mind. Incrementality severely restricts the nature of adequate processing models. For instance, Miller and Chomsky's (1963) initial approach of modeling language comprehension as the grammar-equivalent automaton cannot guarantee incrementality, and they were aware of that. The push-down automaton for a context-free language, for instance, would time and again stack up its push down store, thus delaying structural decisions.

Various solutions have been proposed for grammars to handle incrementality in language use. The very first one (to my knowledge) was the Incremental

Procedural Grammar by Kempen and Hoenkamp (1987). This was still a string grammar, but designed to account for the speaker's incremental sentence generation. I used it in *Speaking* (1989). But then, tree grammars took over. Kempen & Harbusch's (1998) Performance Grammar (PG) was, again, explicitly constructed for handling incrementality, in both speaker and listener models. Incrementality is naturally implemented in the linearization component of the grammar, which is essentially a finite state device operating on trees. Meanwhile, various applications of PG have seen the light. One recent example is the modeling of the speaker's generation of clausal coordination and coordinate ellipsis, with all of its gapping and reduction complexities (Kempen, 2009). Another one (Vosse & Kempen 2008) is the implementation of PG in an incremental, but parallel parser (called SINUS). It is parallel in that it can simultaneously entertain different unification alternatives for the same lexical input. At any moment these alternatives are in different states of activation and activations are continuously adapted as new input arrives. The final parse of a sentence corresponds to the configuration of 'winning' (most highly activated) unifications at the end of the sentence. There is no backtracking in the sense of retracing to an earlier point in the sentence and from that point onward selecting an analysis/interpretation that was not entertained before. The claim is that states of activation of unifications are reflected in on-line measures of comprehension load, such as ERP and eye tracking data.

As already noticed above, there exist incremental parsers for TAG. Ferreira (2000) introduced the TAG architecture in her model of the speaker's syntactic production. Ferreira et al. (2004) used it in their account of listener's processing of disfluences in speech. Joshi (1985) himself used aspects of TAG for modeling incremental code switching between Marathi and English. Webber et al. (2003) used TAG in their study of anaphora. See Joshi (2004) for an overview of TAG applications.

Hale (2001) modeled incremental, 'eager' parsing by way of a probabilistic context-free grammar (based on a sample of the Treebank Corpus), implemented in Stolcke's (1997) probabilistic Early parser. For each next word in the sentence this algorithm computes 1 minus the so-called 'prefix-probability', that is the amount of disconfirmation of (probabilistic) expectations that word provides. That is the word's 'surprise value', which can serve as a measure for the effort it takes to 'eagerly' or fully exploit the information provided by that word. This measure peaks when reaching the word *fell* in *the horse raced past the barn fell.* More generally, it provides detailed predictions for word-by-word reading latencies. Levy (in press) supplies a rich application of his own, equivalent 'surprise' measure to a range of linguistic cases and experimental data.

Learnability

All formal work on the learnability of grammars, grammatical inference, goes back to Gold's (1967) seminal paper. Under Gold's specific definitions, the somewhat shocking finding was that only finite languages are learnable from so-called 'text presentation', which is an enumeration (infinite for infinite languages) of the sentences of the language. Learnability of infinite languages only exists under 'informant presentation', any enumeration of both the grammatical and the ungrammatical strings over the language's vocabulary (and marked for their (un) grammaticality). With that type of presentation, languages in the Chomsky hierarchy up to context-sensitive (and in addition primitive recursive ones) are learnable. These formal results (reviewed in *FL* I, chapter 8) substantially sharpened the reasoning about Chomsky's *Language Acquisition Device* (LAD), the potential mechanism that would enable any child to infer a grammar for its native language from the linguistic (or other) input received. The fact that children usually do acquire their native language argues for the existence of such a device. I thoroughly treated these matters in *FL* III, Chapter 4. The major issues discussed there are as relevant today as they were in the early seventies. Just to mention some of them: If learnability requires informant presentation, how much negative evidence is (in whatever way) presented to children? The dominant view at that time was: none. No child is told: 'the utterance you (or I) just produced is ungrammatical'. And the fast conclusion was: because a natural language is not learnable, it must be largely innate. That is Chomsky's Universal Grammar (UG). Meanwhile, convincing evidence has been obtained for 'negative evidence' provided by adults to children. Chouinard and Clark (2003), for instance, reported evidence of systematic corrections by adults of children's utterances and evidence of children's attending to, acknowledging and incorporating these corrections.

Another issue was and is: noise ruins learnability. If only one sentence doesn't show up in the limit or if, in informant presentation, one ungrammatical string is marked as grammatical, learnability breaks down in Gold's algorithm. The child's language input is, obviously, quite noisy; how to deal with that? Horning (1969) was the first to conceive of a procedure for selecting or 'learning' a probabilistic grammar from stochastic text presentation. As I reviewed in *FL* I 8.4, probabilistic non-ambiguous context-free grammars are learnable this way (under Horning's definition). One advantage of statistical learning algorithms is that they can be noise-resistant. Meanwhile the statistical modeling of language learning has made substantial advances, in particular by the work of Valiant (1984) and Haussler (1996). Neural network modeling has become a major new statistical approach to issues of language inference (cf. Elman (2005). But there, the perennial problem is that learnability is at best demonstrated by computer simulation; it is never *proven*

within this paradigm. It doesn't meet Gold's golden standard. For reviews of these and other issues in learnability, see Jain et al. (1999), Pullum (2003) and Scholz and Pullum (2006).

A potentially important type of linguistic input to language learning children is the prosody of the utterance. That prosody reveals to some extend the syntactic structure of the utterance. That would make tree automata interesting devices in the modeling of grammatical inference. Another good reason for using tree automata in modeling children's early speech corpora is the ubiquitous use of constructions. These are complete holophrases (such as *in there*) or phrases with just one or a small number of variable positions (such as *where N go?*). As mentioned above, tree grammars can naturally handle such fixed expressions by way of elementary trees with multiple lexical anchors. Borensztaijn et al. (2008) have applied this to the Adam, Eve and Sarah corpora from Brown (1973). The developmental notion here is that lexical anchors in the initial multiple-anchored trees are replaced by variables in the course of development. The fixed construction becomes less and less 'fixed', slowly approximating the adult lexical trees, which usually have a single lexical anchor plus a number of variable positions marked for substitution.

Conclusion

When I wrote *Formal Grammars*, the world of formal paradigms in linguistics and its applications was still surveyable, if not simple. That paradisaical situation is long gone. No single linguist or psycholinguist can now oversee the richness of formal devices used in the theoretical and empirical study of natural language and its uses. Using them has become team work. And, at least in psycholinguistics, the use of formal devices has become eclectic. Although trends still come and go, no single approach achieves the aura of being 'the right one'. That is, by and large, a healthy situation. Still, the drive to do things formally right will always be with us, students of language and its uses.

References

Ajdukiewicz (1935). Die syntaktische Konnexität. *Stud. Philos.*, *1*, 1–27.

Borensztajn, G., Zuidema, W., & Bod, R. (2008). Children's grammars grow more abstract with age - Evidence from an automatic procedure for identifying the productive units of language. *Proceedings of the 30th Annual Meeting of the Cognitive Science Society (CogSci2008)*, Washington DC, July 2008.

Bresnan, J. (2006). Is knowledge of syntax probabilistic? Experiments with the English dative alternation. Proceedings of the International Conference on Linguistic Evidence, Tuebingen, 2–4 February 2006, *Roots: Linguistics in search of its evidential base.* Series: Studies in Generative Grammar, edited by Sam Featherston and Wolfgang Sternefeld. Berlin: Mouton de Gruyter.

Büchi (1960). Weak second-order arithmetic and finite automata. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik, 6,* 66–92.

Charniak, E. (1993). *Statistical language learning.* Cambridge, MA: MIT Press.

Chomsky, N. (1982). *The generative enterprise. A discussion with Riny Huybregts and Henk van Riemsdijk.* Dordrecht: Foris Publications.

Chouinard, M. and Clark, E. (2004). Adult reformulations of child errors as negative evidence. *Journal of Child Language,* 30, 637–669.

Comon, H., Dauchet, M., Gilleron, R., Jacquemard, F., Lugiez, D., Löding, Ch., Tison, S., & Tommasi, M. (2007) *Tree Automata Techniques and Applications.* E-Book http://tata.gforge.inria.fr/

Gécseg, F. & Steinby, M. (1997). Tree languages. In: G. Rozenberg & A. Salomaa (Edss.). *Handbook of formal languages.* Vol. 3 *Beyond words,* 1–68.

Gold, E.M. (1967). Language identification in the limit. *Information and Control, 10,* 441–474.

Ellis, C.A. (1971). Probabilistic tree automata. *Information and Control, 19,* 401–416.

Elman, J. L. (2005). Connectionist models of cognitive development: Where next? *Trends in Cognitive Sciences,* 9, 111–117.

Ferreira, F. (2000). Syntax in language production: An approach using Tree-Adjoining Grammars. In L. Wheeldon (Ed.), *Aspects of language production.* Cambridge, MA: MIT Press.

Ferreira, F., Lau, E.F., Bailey, K.G.D. (2004). Disfluencies, language comprehension, and Tree Adjoining Grammars. *Cognitive Science, 28,* 721–749.

Fodor, J.A., Garrett, M.F., Walker, E.C.T. & Parkes, C.H. Against definitions. *Cognition, 8,* 263–367.

Hale, J. (2001). A probabilistic Earley parser as a psycholinguistic model. In *Proceedings of NAACL, 2,* 159–166.

Harbusch. K. & Kempen, G. (2002). A quantitative model of word order and movement in English, Dutch and German complement constructions. In: *Proceedings of the 19th International Conference on Computational Linguistics (COLING-2002),* Taipei (Taiwan). San Francisco: Morgan Kaufmann.

Haussler, D. (1996). Probably aproximately correct learning and decision-theoretic generalizations. In: P. Smolensky, M.C. Mozer, & D.E. Rumelhart. (Eds.), *Mathematical perspectives on neural networks,* 651–718. Mahwah, NJ: Erlbaum.

Horning, J.J. (1969). A study of grammatical inference. *Technical Report CS 139. Stanford Artificial Intelligence Project).*

Jackendoff, R. (2002). *Foundations of language.* Cambridge, MA: MIT Press.

Jain, S., Osherson, D.N., Royer, J.S., & Sharma, A. (1999). *Systems that learn: An introduction to learning theory.* Second edition. Cambridge, MA: MIT Press.

Joshi, A. (1985). Processing of sentences with intra-sentential code-switching. J.Horeck (Ed.), *Proceedings of COLING 82.* Amsterdam: North-Holland Publishing Company.

Joshi, A. K. (2004). Starting with complex primitives pays off: complicate locally, simplify globally. *Cognitive Science, 28,* 637–668.

Joshi, A.K. & Shabes, Y. (1997). Tree-adjoining grammars. In: G. Rozenberg & A. Salomaa (Eds.). *Handbook of formal languages.* Vol. 3 *Beyond words*, 69–123.

Kaplan, R.M. & Bresnan, J. (1982). Lexical-functional grammar: A formal system for grammatical representation. In: J. Bresnan Ed.), *The mental representation of grammatical relations*, 173–281. Cambridge MA: MIT Press.

Kempen, G. & Harbusch, K. (1998). Tree Adjoining Grammar without Adjoining? The case of scrambling in German. *Proceedings of TAG+4 - Fourth International Workshop on Tree Adjoining Grammars and Related Formalisms.* Philadelphia, Pennsylvania/USA: IRCS Report # 98-12, 80–83.

Kempen, G., & Harbusch, K. (2002). Performance Grammar: A declarative definition. In: A. Nijholt, M. Theune & H. Hondorp (Eds.), *Computational Linguistics in The Netherlands 2001.* Amsterdam: Rodopi.

Kempen, G. & Harbusch, K. (2005). The relationship between grammaticality ratings and corpus frequencies: A case study into word order variability in the midfield of German clauses. In: S. Kepser & M. Reis (Eds.), *Linguistic Evidence—Empirical, Theoretical, and Computational Perspectives.* Berlin: Mouton De Gruyter.

Kempen, G. & Harbusch, K. (2008). Comparing linguistic judgments and corpus frequencies as windows on grammatical competence: A study of argument linearization in German clauses. In: Steube, Anita (Ed.), *The discourse potential of underspecified structures*, 179–192. Berlin: DeGruyter.

Kempen, G. & Hoenkamp (1987). An incremental procedural grammar for sentence formulation. *Cognitive Science, 11*, 201–258.

Kempen, G. (in press 2009). Clausal coordination and coordinate ellipsis in a model of the speaker. *Linguistics.*

Klein, W. (1974). *Variation in der Sprache. Ein Verfahren zu einer Beschreibung.* Kronberg: Scriptor Verlag.

Knight, K. & Graehl, J. (2005). An overview of probabilistic tree transducers for natural language processing. In: *Proceedings of the Sixth International Conference on Intelligent Text Processing and Computational Linguistics (CICLing).* Lecture Notes in Computer Science. Springer Verlag, 2005.

Levelt, W.J.M. (1989). *Speaking. From intention to articulation.* Cambridge, MA: MIT Press.

Levinson, S.C. (2000). *Presumptive meanings.* Cambridge, MA: MIT Press.

Levy, R. (in press). Expectation-based syntactic comprehension. *Cognition.*

Miller, G.A. & Chomsky, N. (1963). Finitary models of language users. In: R.D. Luce, R.R. Bush & E. Galanter, *Handbook of mathematical psychology.* New York: John Wiley.

Montague, R. (1970). English as a formal language. Reprinted in: R. Thomassen (Ed.), (1974), *Formal philosophy: selected papers of Richard Montague*, 188-221. New Haven: Yale University Press.

Partee, B.H., with H. Hendriks (1997). Montague grammar. In: J. van Benthem & A. ter Meulen (Eds.). *Handbook of logic & language.* 5–91. Amsterdam: Elsevier and Cambridge, MA: MIT Press.

Peters, P.S. & Ritchie, R.W. (1973). On the generative power of transformational grammars. *Information Sciences, 6*, 49–83.

Pullum, Geoffrey K. (2003) 'Learnability.' *The Oxford International Encyclopedia of Linguistics.* Second edition, 431-434. Oxford: Oxford University Press.

Resnick (1992). Probabilistic tree-adjoining grammars as framework for statistical natural language processing. In: *Proceedings of the 15th International Conference on Computational Linguistics (Coling'92)*. Nantes.

Schabes, Y. (1992). Stochastic tree-adjoining grammars. In: *Proceedings of the 15th International Conference on Computational Linguistics (Coling'92)*. Nantes.

Scholz, B.C. and Pullum, G.K. (2006) Irrational nativist exuberance. In: R. Stainton (ed.), *Contemporary Debates in Cognitive Science*, 59–80. Oxford: Basil Blackwell.

Seuren, P.A.M. (1969). *Operators and nucleus. A contribution to the theory of grammar.* Cambridge: University Press.

Seuren, P.A.M. (2009). *Language from within.* 2 Vols. Oxford: Blackwell Publishers.

Sprenger, S.A., Levelt, W.J.M., & Kempen, G. (2006). Lexical access during the production of idiomatic phrases. *Journal of Memory and Language, 54*,161–184.

Stabler, E.P. (2004). Varieties of crossed dependencies: structure dependencies and mild context-sensitivity. *Cognitive Science, 28*, 699–720.

Steedman, M. (2000). *The Syntactic Process.* Cambridge, MA: MIT Press.

Stolcke, A. (1979). Linguistic knowledge and empirical methods in speech generation. *AI Magazine, 18*, 25–31.

Suppes, P. (1970). Probabilistic grammars for natural language. *Synthese, 22*, 95–116.

Valiant, L. (1984). A theory of the learnable. *Communications of the ACM, 27*, 1134–1142.

Vosse, Th. & Kempen, G. (2008). Parsing verb-final clauses in German: Garden-path and ERP effects modeled by a parallel dynamic parser. *Proceedings of the 30th Annual Meeting of the Cognitive Science Society (CogSci2008)*, Washington DC, July 2008.

Valiant, L. (1984). A theory of the learnable. *Communications of the ACM, 27*, 1134–1142.

Webber, B., Stone, M., Joshi, A.K. & Knott, A. (2003). Anaphora and discourse structure. *Computational Linguistics, 29*, 545–588.