Connection Science
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/ccos20

Incremental Syntactic Tree Formation in Human Sentence Processing: a Cognitive Architecture Based on Activation Decay and Simulated Annealing

GERARD KEMPEN [a] & THEO VOSSE [a]

[a] NICI, University of Nijmegen, Montessorilaan 3, Nijmegen, HR, 6525 Phone: +31(80)512621 E-mail: kenpen@hnykun53

To cite this article: GERARD KEMPEN & THEO VOSSE (1989): Incremental Syntactic Tree Formation in Human Sentence Processing: a Cognitive Architecture Based on Activation Decay and Simulated Annealing, Connection Science, 1:3, 273-290

To link to this article: http://dx.doi.org/10.1080/09540098908915642

Connection Science, Vol. 1, No. 3, 1989

Incremental Syntactic Tree Formation in Human Sentence Processing: a Cognitive Architecture Based on Activation Decay and Simulated Annealing

GERARD KEMPEN & THEO VOSSE

A new cognitive architecture is proposed for the syntactic aspects of human sentence processing. The architecture, called Unification Space, is biologically inspired but not based on neural nets. Instead it relies on biosynthesis as a basic metaphor. We use simulated annealing as an optimization technique which searches for the best configuration of isolated syntactic segments or subtrees in the final parse tree. The gradually decaying activation of individual syntactic nodes determines the 'global excitation level' of the system. This parameter serves the function of 'computational temperature' in simulated annealing. We have built a computer implementation of the architecture which simulates well-known sentence understanding phenomena. We report successful simulations of the psycholinguistic effects of clause embedding, minimal attachment, right association and lexical ambiguity. In addition, we simulated impaired sentence understanding as observable in agrammatic patients. Since the Unification Space allows for contextual (semantic and pragmatic) influences on the syntactic tree formation process, it belongs to the class of interactive sentence processing models.

KEYWORDS: Sentence processing, syntax, simulated annealing, agrammatism, segment grammar, computational psycholinguistics.

1. Introduction

One of the long-standing issues in psycholinguistic research concerns the syntactic tree formation process which takes place during sentence production and comprehension. Two basic problems are usually distinguished:

- By what kinds of trees (or treelike structures) is the syntactic makeup of sentences represented in the human cognitive system?
- What are the inner workings of the cognitive module(s) responsible for building syntactic structures?

In previous papers (De Smedt & Kempen, 1987; Kempen, 1987; Kempen & Hoenkamp, 1987; Van Wijk & Kempen, 1987) we have proposed a partial solution in the

Gerard Kempen & Theo Vosse, NICI, University of Nijmegen, Montessorilaan 3, 6525 HR Nijmegen, The Netherlands; Tel: +31 (80) 512621; Bitnet: kempen@hnykun53. An earlier version of this paper appeared in J. Montangero & A. Tryphon (Eds) (1989) *Language and Cognition*. Geneva: Jean Piaget Archives Foundation. We thank Koenraad De Smedt, Herman Kolk, Henk Haarmann, Pieter Bison, Gerben Abbink, Gary Dell and Kathryn Bock for their comments and suggestions.

form of a new type of grammar. For overviews of this approach we refer to Levelt (1989, Ch. 7) and De Smedt (1990). The work to be reported below was motivated by dissatisfaction with the *cognitive* (or functional) *architecture* (in Pylyshyn's, 1984, sense) that was presupposed. What is missing is a set of assumptions from which quantitative predictions regarding error rates, reaction times, processing loads, etc., can be derived.

This paper aims to stop this gap. We propose a tree formation mechanism that does enable us to simulate dynamic characteristics of syntactic tree formation in human language users. Section 2 explains our versions of unification—the formal operation which serves to compose syntactic trees out of elementary 'syntactic segments'. In Section 3 we describe the dynamic model called 'Unification Space'. Section 4 is devoted to the results of a series of Monte Carlo simulation studies testing the behavior of the Unification Space as a psychologically plausible syntactic parser. Finally, in Section 5 we draw some comparisons with recent psycholinguistic parsing models.

2. A Unification-based Tree Formation Formalism

Kempen (1987) introduced a formalism for constructing linguistically plausible syntactic trees out of so-called segments. A segment consists of two nodes connected by an arc. Usually depicted in vertical orientation, it is said to have a *foot*, the bottom node, and a *root*, the top node. Nodes are labeled after syntactic *categories*, i.e. word classes (noun, verb, adjective, proposition) or phrases (sentence, noun phrase, prepositional phrase). Names of syntactic *functions* serve as arc labels (subject, object, head, modifier, determiner). The sentence of Figure 1 consists of eight segments, seven of which are of different type. Segments whose foot is a word class are called 'lexical' or 'terminal'.

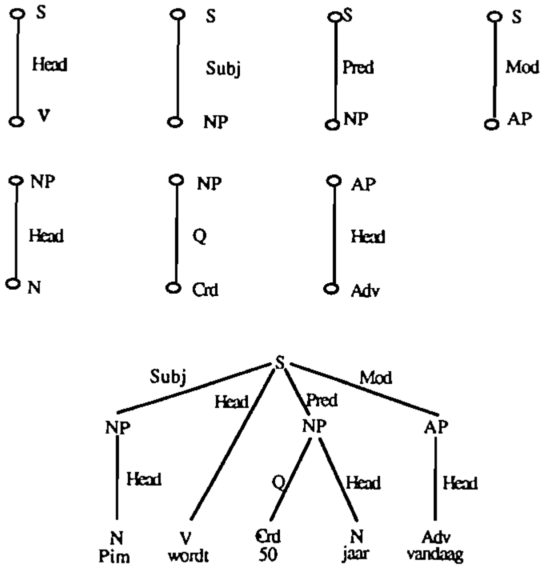


Figure 1. Syntactic segments and syntactic tree corresponding to Dutch equivalent of 'Pim has his 50th birthday today' (literally 'Pim becomes 50 years [old] today').

The composition of syntactic trees out of segments is controlled by a single elementary operation called *unification* (Kay, 1985). It is a rather simple operation, reminiscent of set union. In the present context, the set members are attributes, values and attribute-value pairs (i.e. features). Attributes are atomic character strings. A value is an atom or a list. In the latter case, list members represent alternative values (no duplications allowed). For example, feature f_1

gender = (masculine feminine)

specifies two possible values for the gender attribute. This expression can be *unified* with f_2

gender = masculine

and with f_3

gender = (masculine feminine neuter)

but not with f_4

gender = neuter.

In the last case the intersection of value sets is empty. In the first two cases, unification returns the non-empty intersection of value sets (without doubles). More precisely,

unify(f_1,f_2)→gender = masculine
unify(f_1,f_3)→gender = (masculine feminine)
unify(f_1,f_4)→[unification fails]

Two feature matrices (i.e. unordered sets of features) can be unified in case of successful unification of all features shared by both matrices. The result is a feature matrix containing all non-shared features plus the unification of shared features. See Figure 2 for an example.

$$\begin{array}{ccc} m_1 & m_2 & m_3 \\ \left[\begin{array}{lll} \text{person} & = & \text{(first second third)} \\ \text{case} & = & \text{nominative} \\ \text{number} & = & \text{(singular plural)} \end{array}\right] & \left[\begin{array}{lll} \text{person} & = & \text{third} \\ \text{number} & = & \text{singular} \\ \text{category} & = & \text{pronoun} \end{array}\right] & \left[\begin{array}{lll} \text{person} & = & \text{third} \\ \text{case} & = & \text{nominative} \\ \text{number} & = & \text{singular} \\ \text{category} & = & \text{pronoun} \end{array}\right] \end{array}$$

Figure 2. Unification of m_1 and m_2 yields m_3.

After having introduced the basic ingredients—segments and unification—we can proceed to the essence of the tree formation formalism. First, we will associate feature matrices with a segment's root, arc, and foot. Figure 3 gives two examples, one corresponding to the English words *he/him*, the other one representing the subject branch of finite clauses. Second, we will assume that unification of feature matrices is equivalent to merging the corresponding nodes. For Figure 3 this implies concatenation of segments as depicted in Figure 4. Notice that nominative case has been selected for the NP node, ruling out realization of lemma HE as *him*.

If the roots of two segments are unified, the result is furcation. Figure 5 presents the lexical segment corresponding to the English auxiliary verb DO, and the result after furcation with the subject segment of Figure 3.

Segments are retrieved from the *lexicon* in response to words recognized (in sentence comprehension) or concepts to be verbalized (in sentence formulation). A

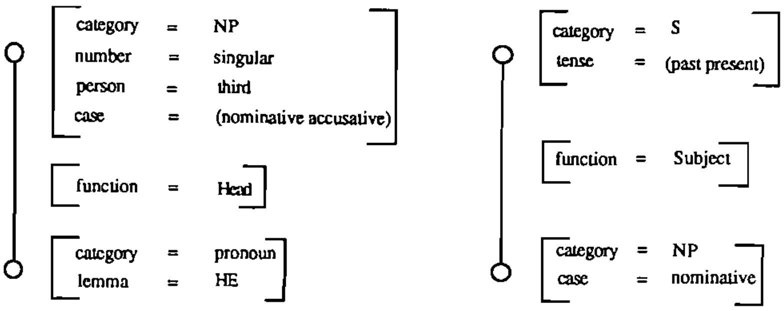


Figure 3. Two segments plus feature matrices.

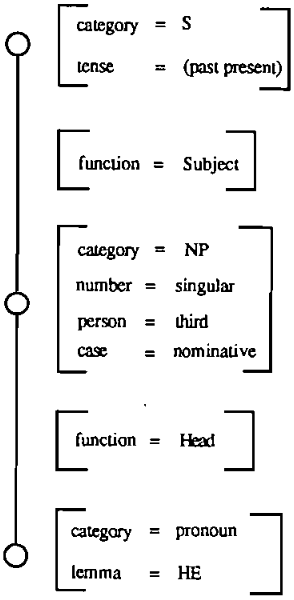


Figure 4. Concatenation of the segments of Figure 3.

lexical entry specifies a single segment or a 'mobile' consisting of several segments. Examples of the latter are provided by verbs and prepositions whose selection entails the presence of additional constituents (subject, direct object, prepositional object). The lexical entry for the Dutch verb *worden* (English, *to become*; see Figure 6) lists, among other things, obligatory subject and predicate constituents and one or more optional modifiers.

Compound lexical entries such as the one depicted in Figure 6 are mobiles rather than trees because the left-to-right order of segments has not been fixed yet. This is done by separate ordering rules at a later stage of the formulation or parsing process.

We finally make the assumption that the lexicon is the only source of segments appearing in a syntactic structure. In other words, no segments are imported by phrase-structure, transformational or any other type of rules. Thus the grammar is fully 'lexicalized'.

We cannot discuss any further details here, although we realize that the outline given barely scratches the surface of what a linguistically acceptable tree formation

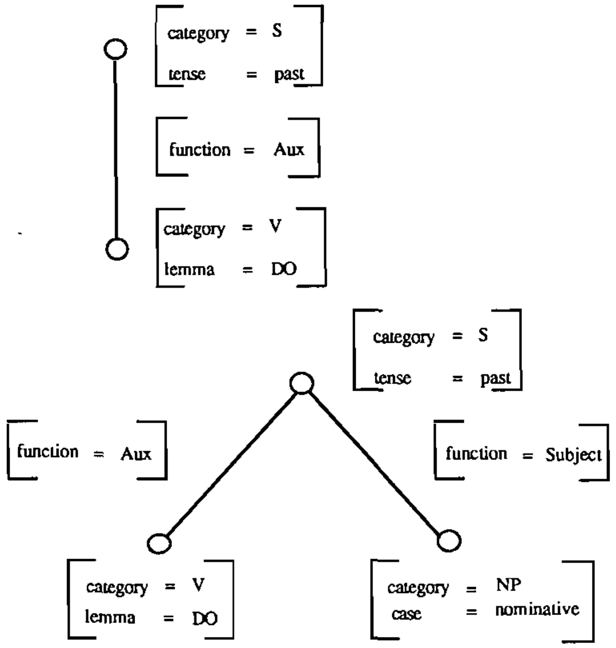


Figure 5. Furcation of auxiliary 'did' with subject segment of Figure 3 (e.g. *Did he . . .*).

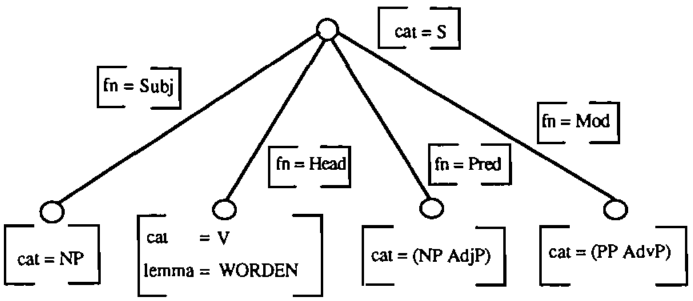


Figure 6. Incomplete lexical entry for Dutch copula verb *worden* (cf. Figure 1).

formalism is supposed to accomplish. We refer to De Smedt (1990) for a full description of the Dutch-language sentence generator he has implemented on the basis of 'segment grammar' (SG).

3. The Unification Space

We now introduce a cognitive architecture suitable to support computations as required by the SG formalism. Being biologically inspired, the architecture features properties such as

- *parallelism* (several processing units are active simultaneously)
- *homogeneity* (all processors carry out the same, very small set of simple primitive operations)
- *locality* (a processor only affects the state of nearby processors)
- *control by activation level* (a processor's functioning depends on the current amount of activation it possesses)

But, contrary to a popular trend, the architecture is not connectionist. In the area of syntax—parsing as well as generation (Fanty, 1985; Kalita & Shastri, 1987; Selman & Hirst, 1987; Howells, 1988)—connectionism has not met with a great deal of success. Whether this state of affairs can change is a controversial issue: compare Fodor & Pylyshyn (1988) with Touretzky (1986), Touretzky & Hinton (1988) and Elman (1989).

Our source of inspiration, or basic metaphor, has not been the network of nerve cells but the biosynthesis of proteins and the physical-chemical processes determining the three-diminsional structure ('conformation') assumed by a protein as a function of its amino acid sequence. See Reeke (1988) for a survey of computer simulation techniques used in this area, particularly those based on energy minimization (simulated annealing) and molecular dynamics.

Imagine syntactic tree formation taking place in a 'mental testtube' containing the segments and mobiles retrieved from the lexicon. Assume furthermore that their nodes continuously attempt to combine with other nodes they hit upon, and that the likelihood of a successful combination—or, rather, unification—depends not only on their feature composition but also on their level of activation. Whenever two nodes actually unify, they merge with one another and join together the segments (or trees/mobiles) they belong to, thus effectively creating a larger tree. Unifications are not granted the life everlasting, though. Depending on the strength of the original bond and due to activation decay, merged node may separate again, thereby clearing the way for other—maybe stronger—unifications. This dynamic process of coalescence, disintegration and reintegration will gradually come to rest accordingly as the segments in the testtube succeed in connecting up with one another in bonds of sufficient strength to lend stability to the resulting tree structure.

Let us try to make these intuitions more precise and develop them into a *computational model of a psychologically plausible parser*. We begin by stating what we consider to be the three core features.

(A) *Activation decay*. When a lexical entry is aroused and retrieved, the foot and root segments are assigned an activation level which gradually decays over time. Since lexical entries are entered into the *Unification Space* (henceforth, the fancy name of the 'mental testtube') immediately upon recognition of the input words they correspond to, it follows that the nodes of more recent segments are generally more active.

(B) *Stochastic parse tree optimization*. Typically, a node in the Unification Space can successfully unify with more than one other node. The search for the best possible unification partner is modeled by a stochastic optimization technique known as simulated annealing (cf. Hinton & Sejnowski, 1986). Two nodes that, given their feature composition, *could* unify successfully, actually unify with probability $p(U)$, which depends, among other things, on their activation level (more active nodes are more likely to unify) and on the grammatical 'goodness of fit' (cf. the 'strength' parameter introduced in paragraph (4) below). Similarly, the model assumes that nodes which have indeed unified may nevertheless break up again with probability $p(B)$. This probability is higher for nodes with lower levels of activity and/or goodness of fit. The combined effect is a bias in favor of syntactic (sub)trees arising from highly active nodes and containing very strong bonds (i.e. representing good grammatical fit).

(C) *Global excitation*. Since node activation is subject to spontaneous decay, all unifications would ultimately be undone as a consequence of rising $p(B)$. What we need is a mechanism for intercepting a parse tree as soon as a satisfactory or even optimal configuration of segments has been discovered. In a standard version of simulated annealing this effect would be reached by making both $p(U)$ and $p(B)$ dependent on a global 'temperature' variable T. The value of T at any point in time is controlled by a predefined 'annealing schedule' according to which T is gradually lowered from an initially high value. The scheme gives the objects involved sufficient time to find a (near-)optimal configuration and finally 'freezes' them in that position.

We are *not* going to introduce a separate annealing schedule. Instead, we call upon the spontaneous activation decay which we already postulated for nodes in the Unification Space. We define a parameter E (for global excitation value) whose value equals the summed activation levels of all nodes currently populating the Unification Space. In the model, E fulfills a role analogous to T in standard simulated annealing: in the absence of any other changes its value decreases monotonically due to activation decay in the individual nodes. Finally it causes syntactic trees to freeze in a (close to) optimal configuration. Apart from greater simplicity, this approach has at least two advantages over a predefined annealing schedule. First, E is responsive to sentence length as its value will tend to rise when more and more segments enter the system. The ensuing expansion of the search space of potential unifications will be matched by a more avid exploration of that space (i.e. higher $p(U)$ and $p(B)$). Second, the value of E tends to drop after a successful unification (because the number of nodes is reduced by one), whereas a break-up has the opposite effect. This factor causes the Unification Space to stabilize more rapidly after unifications with a better grammatical fit. In sum, we obtain an *adaptive* annealing schedule.

In the remainder of this section we describe in detail the version of the Unification Space architecture that we have implemented. We wish to emphasize that this version is not the only possible elaboration of the general principles outlined in the preceding part of this section. Several decisions we had to take are somewhat arbitrary, in particular those concerning parameter values. (In the final section we will return to this issue.) Appendix A contains a listing of the control structure of the simulation model.

(1) Spoken or written words recognized in the input sentence are stored in an input buffer for a limited period of time T_B. The individual words are read out from left to right, one-by-one, at fixed intervals $T_W < T_B$. The lexical entry corresponding to a word, i.e. a segment or a mobile, is immediately entered into the Unification Space. In case of lexical ambiguity, all entries listed in the lexicon are entered simultaneously.

(2) Upon entry in the Unification Space, every node of a segment/mobile is allotted a certain amount of activation. Activation levels, which are subject to spontaneous decay, may assume values between 0 and 1. The activation level of a node, designed by A_0, is initialized to a value associated with the lexical entry. However, in the simulations presented in Section 4, we have uniformly chosen 1 as initialization value.

In paragraph (7) below we will see that segments which come loose from a tree after a break-up, are removed from the Unification Space. A *lexical* segment thus eliminated—that is, a segement whose foot is a word class—will, however, reenter the Unification Space without delay if the corresponding word is still surviving in the input buffer. The initial activation level of reentering nodes, too, is set to the value listed in the lexicon.

(3) The probability of two nodes n_i and n_j hitting upon each other after they entered the Unification Space is completely random. Upon impact, the legality of unification $U(n_i, n_j)$ is checked, exclusively on the basis of the feature composition of

n_i and n_j. (We assume that the unification operation is always applied correctly, without any performance degradation.)

The amount of activation of the single node n_u which results from unification of nodes n_i and n_j with activation levels A_i and A_j equals

$$A_u = A_i + A_j - A_i A_j. \tag{1}$$

If two or more lexical entries corresponding to an ambiguous word are simultaneously present in the Unification Space, and a node of one of these is involved in a successful unification, then all other entries are immediately and completely removed.

(4) One of the factors determining probability $p[U(n_1, n_2)]$ is the function S—see equation (2) below. S (strength) yields as its value a real number between 0 and 1 representing the grammatical goodness-of-fit—in a broad sense—of the concatenation/furcation of segments involved in the unification. S is sensitive to semantic, pragmatic and lexical factors (cf. selection restrictions) as well as to those syntactic factors which are not taken into account by the unification operation. By allowing for semantic and pragmatic influences upon the syntactic tree formation process we explicitly subscribe to an 'interactive' view of syntactic processing (Marslen-Wilson, 1975; Altmann & Steedman, 1988; Taraban & McClelland, 1988; see also Section 5).

Our computer simulations admit of two syntactic contributions to the value of S. First, S serves to coerce word order in the parse tree towards that of the input sentence. The unification operation defined in Section 2 creates mobiles rather than trees and has no way of knowing whether the subsequently applied ordering rules can produce a word sequence identical to the input sentence. Our version of function S therefore returns low values for proposed unifications which entail a mismatch with input word order. (This would happen, for example, when 'John loves Mary' would be parsed as if Mary were the subject and John the direct object, for the sentence corresponding to that interpretation has a different word order: 'Mary loves John'.) Second, we have built into S a slight dislike of attaching anything to the footnode of a modifier segment: the current goodness-of-fit value is multiplied by 0.9.

The value returned by $S(n_i, n_j)$, that is, s_u, will be associated with node n_u produced by $U(n_i, n_j)$ and become one of the paramaters in equation (3).

(5) The events in the Unification Space are critically determined by global parameter E, for 'global excitation level', in ways similar to temperature in simulated annealing. E values rise accordingly as the Unification Space is inhabited by a larger number of segments with more active nodes. But E depends on external factors as well, in particular on mental effort (reducing E) and pathological conditions (increasing E).

At high E levels, the structures in the Unification Space are unstable: unifications can take place at a high rate, but break-ups occur frequently, too. With decreasing excitation levels, the structures stabilize gradually until a steady state is reached with an unchanging, 'frozen' configuration of segments. Alluding to the terminology of physical chemistry, we will call a stable, frozen configuration a 'conformation'. The parsing process succeeds if and when the Unification Space comes to rest with a unique conformation covering all input words. This conformation is the correct parse tree (or *a* correct parse tree, if the input is ambiguous). If several conformations—detached from each other—are present in the steady state, or if no conformation is reached at all (oscillation), the parsing process fails.

(6) Probability $p[U(n_i, n_j)]$ referred to in paragraph (4) above, is given by:

$$p[U(n_i, n_j)] = S(n_i, n_j)\ [w_A (A_i A_j)^{w_E^{E}} - w_A + 1] \tag{2}$$

where w_E and w_A ($0 \leqslant w_E \leqslant 1$ and $0 \leqslant w_A \leqslant 1$) are weights tempering the influence of E and A, respectively.

(7) Trees are susceptible to disintegration. The segment with root n_r and foot n_f breaks up with probability

$$p[B(n_r, n_f)] = \exp(-s_f A_f' A_r' / kE) \tag{3}$$

where k is a constant and $A_i' = A_i(1-\alpha)+\alpha$. In our simulations we have chosen $k=1/35$ and $\alpha=1/5$. Foot nodes will have the default strength $s=1$, unless it has been set to a lower value by function S (see paragraph 4).

Disintegration of a tree implies that the broken segment is removed from the Unification Space. A segment or subtree attached to that segment's foot, however, is allowed to stay as a separate structure, detached from the tree to which it belonged.

(8) Although we hypothesize that in the Unification Space many unifications may occur in parallel, we have devised a sequential simulation program. Successive processing *cycles* are supposed to start at equidistant points in time t, $t+1$, $t+2, \ldots,$ etc. Every cycle brings together two randomly selected nodes and tries to unify them. In addition, various bookkeeping activities take place. To begin with, the global excitation level is updated during every cycle:

$$E_{t+1} = \sum_{n_i \in \text{U-space}} A_i C + E_t C/5 \tag{4}$$

That is, the new E value is partly determined by the activation levels of the nodes that currently inhabit the Unification Space. It also depends on the previous E level and, most importantly, on 'chaos' parameter C. (Its value, which is kept constant during successive cycles, lies between 0.1 and 2.) C is supposed to aggregate external influences upon the global excitation level: its value is raised by neuropathological conditions such as in aphasia, and lowered as a consequence of increased mental effort invested in syntactic processing.

In our computer implementation, E_{t+1} is set to maximum E level $E_{\max} = 10C$, whenever equation (4) returns a higher value.

(9) Another bookkeeping duty is to take care of spontaneous decay of activation in nodes. The decay rate, which is a function of global excitation level, is given by

$$D = c_1 c_2^{wE} \tag{5}$$

where $c_1 = 0.975$, $c_2 = 0.995$ and $w = 0.80$. For higher values of E, D will be lower and speed up decay:

$$A_{i,t+1} = DA_{i,t} \tag{6}$$

Our computer program uses a somewhat more complex version of (6) which embodies activation spreading. A small portion (about 0.1%) of the activation of neighboring node (mother, daughter(s)) is added up to $A_{i,t+1}$. We skip this detail, which we believe has only a marginal effect on the simulation results.

In the next section we present the results obtained with our computer implementation of the Unification Space model.

4. Simulation Results

A parsing architecture pretending to psychological plausibility may set out to prove this claim by simulating the basic syntactic phenomena of sentence processing. These include such relatively uncontroversial facts as

- the difficulty of analyzing triply center-embedded clauses, whereas their triply righthand-embedded counterparts are fairly easy—compare examples (1) and (2) below—in conjunction with the absence of a similar contrast in doubly embedded clauses: compare examples (3) and (4).

- Right association (Kimball, 1973), e.g. the adverb *yesterday* in (5) is attached to the subordinate rather than to the main clause (*came yesterday* instead of *said yesterday*).
- Minimal attachment (cf. Frazier, 1987), e.g. the garden-path quality of sentence (7) necessitating reanalysis of *chased* as a past participle in a reduced relative clause.
- Lexical ambiguity: the slightly increased processing difficulty imposed by words having more than one meaning (cf. sentence (8), where *can* is either an auxiliary or a main verb, and *fish* either a noun or a verb).

(1) C3: The rat the cat the dog bit chased escaped.
(2) R3: The dog bit the cat that chased the rat that escaped.
(3) C2: The rat the cat chased escaped.
(4) R2: The cat chased the rat that escaped.
(5) RA: John said he came yesterday.
(6) MA: The horse raced past the barn yesterday.
(7) NMA: The horse raced past the barn fell.
(8) LA: They can fish.

Our primary goal was to investigate whether the Unification Space architecture was indeed capable of generating these phenomena. Furthermore, we have attempted to simulate the sentence comprehension impairment in two groups of aphasic patients, one suffering from a more severe degree of agrammatism than the other. We were inspired to do so by a computer modeling study by Haarmann & Kolk (1988). The Dutch-language sentence materials and the performance data stem from their work. A representative sample is listed in (9) through (12).

(9) Active: De man groet het meisje.
(The man greets the girl)
(10) Locative: De man loopt achter her meisje.
(The man walks behind the girl)
(11) Passive1: Het meisje wordt door de man gegroet.
(The girl is by the man greeted)
(12) Passive2: Het meisje wordt gegroet door de man.
(The girl is greeted by the man)

The first type of passive does not occur in English but is quite common in Dutch. Notice that the relation expressed in each sentence is reversible. This forced the patients to take syntactic structure into account while trying to fulfill their task: to chose the correct alternative from two pictures, one matching with the input sentence and the other one representing the inverse relation.

In preparation of the computer simulations we implemented a miniature lexicon containing all lexical entries—segments, mobiles, feature matrices, etc.—needed to build SG parse trees for the 12 sentences (depicted in Appendix B). In order to satisfy the requirements of strength function S we wrote a small set of ordering rules (both Dutch and English) enabling S to compare word order in the input buffer with word order in the nascent parse tree. The simulations were run with *a fixed set of parameter values which were not allowed to vary across sentences.* Actually, we varied only one parameter systematically (subjecting all sentences to the same schedule, of course): chaos parameter C. This choice derives from the hypothesized relationship between this variable and severity of agrammatism. The values assigned to C covered the complete trajectory from 0.1 to 2 in steps of 0.1.

All sentences were subjected to 100 stochastic (Monte Carlo) simulation runs for each value of *C*. The only difference between Dutch and English runs concerned word order rules and lexicon. The final preparatory step was a series of test runs which helped us to find reasonable values for the various parameters. These estimates, mentioned in the numbered paragraphs of Section 3 and in Appendix A, were obtained by hand without statistical parameter fitting techniques.

As a simple statistic representing the model's behavior in response to an input sentence we chose the percentage of correct parses (i.e. unique conformations including all input words) obtained during a set of 100 runs. More specifically, for the non-ambiguous sentences we counted the number of times the Unification Space settled down on a conformation identical to the correct parse tree.[1] In the case of ambiguous sentences (5) and (8) we focused on the model's preference for either of the two correct parses and determined their frequencies of occurrence. Because sentence (8) was so easy that correct conformations were obtained in all runs for all *C* values, we had recourse to a secondary performance index: the average number of cycles within a run that was needed to reach a conformation.

We will now present the simulation results for each of the five sentence processing phenomena. Since the model's behavior was fairly constant over lower levels of parameter *C*, we report percentages averaged over *C* levels 0.1 through 0.5, unless indicated otherwise. For more detailed information we refer to Figure 7.

Center- versus righthand-embedding (Figure 7A). The percentages of correct parses showed the predicted pattern:

C2: 98	C3: 50
R2: 97	R3: 82

That is, the model did not have any trouble in parsing double center or righthand embeddings. In contrast, triply center-embedded clauses appear to be very difficult, considerably harder than their righthand-embedded counterparts.

For higher *C* values the difference between R3 and C3 disappears, as shown in Figure 7A. This is a consequence of R3 being two words longer than C3. Towards the end of the parsing process, the left-hand flank of the R3 parse tree is more prone to disintegration (break-ups) than the C3 tree. A similar length effect was evidenced by C2/R2 sentence pair, although setting in at *C* values in excess of 0.9 (not shown in the figure).

Right association (Figure 7B; RA and NRA denote right association and non-right association, respectively). Sentence (5) was correctly parsed in nearly 100% of the simulation runs. In 74% of these, the adverb *yesterday* got attached to the lower (embedded) clause. The clear right association preference is interpretable as a 'recency' effect. When the *yesterday* segment enters the Unification Space, activation in the complement S node (a foot) has just been boosted due to its unification with the S node (a root) in the *came* segment.

Minimal attachment (Figure 7C). The non-minimally attached (NMA) sentence appeared to present substantial problems, as expected. A correct parse was obtained in only 8%, whereas the minimally attached (MA) sentence yielded a successful analysis in 77% of the runs. The fact that MA, although being relatively simple, failed to reach the 100% score is due to the ambiguity of *raced*. The English lexicon specified two readings: as a finite verb and as a past participle. Selection of the latter reading in effect blocked interpretation of MA as a clause, yielding an NP analysis.[2]

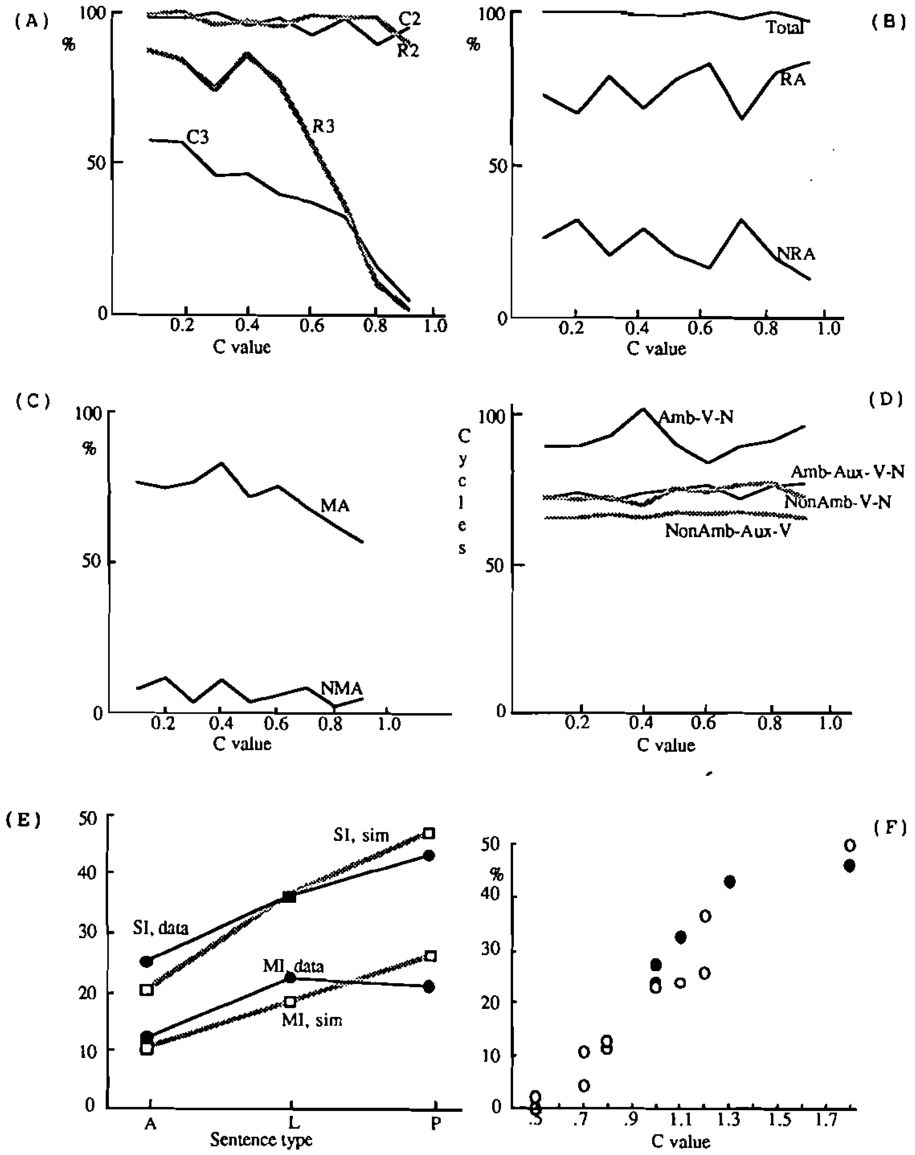


Figure 7. Essential simulation results (for explanation see text).

Lexical ambiguity (Figure 7D). Sentence (8) is ambiguous between Aux-V (auxiliary-main verb) and V-N (main verb-object noun) readings. No preference for either analysis was established (Aux-V, 48%; V-N, 52%). However, it took the model more cycles to come to a halt in case of the V-N reading. This difference is related to the slightly higher number of nodes in the N-V than in the Aux-V parse tree (see Appendix B) and the ensuing higher level of global excitation.

In order to assess the effect of lexical ambiguity *per se*, we had the computer execute two additional runs with *can* and *fish* disambiguated. That is, during those runs the lexicon contained only one entry for each of these words, either Aux-V or V-N. Figure 7D demonstrates clearly that non-ambiguous runs (curves labeled NonAmb-

Aux-V and NonAmb-V-N) are taken to completion in fewer cycles than their ambiguous counterparts.

Agrammatic sentence comprehension (Figures 7E and 7F). Haarmann & Kolk (1988) were the first to develop a computational model of agrammatic sentence understanding which takes degrees of severity of the aphasic impairment into account. Their study is a successful attempt to simulate two sets of empirical data originally collected by Schwartz *et al.* (1980) and Kolk & Van Grunsven (1985), respectively. The former authors enumerate the comprehension scores obtained by five severely impaired (SI) English-speaking aphasics on Active, Locative, and Passive2 sentences: see examples (9), (10), and (12) above. The latter authors deal with 11 mildly impaired (MI) Dutch-speaking patients who were presented with Active, Locative and Passive1 sentences (examples (9), (10), and (11)). The error percentages—averaged over patients—are shown by the continuous lines of Figure 7E. Sentence type P stands for either Passive1 (MI) or Passive2 (SI). In Figure 7F, the vertical dimension represents level of severity and shows error scores for 16 individual patients averaged over three sentence types (open circles: Dutch, MI; filled circles: English, SI).

The simulation runs for sentences (9) through (12) provided us with parsing probabilities for all sentence types (A, L, P1, P2) and C levels (0.1, 0.2, . . ., 2). We converted these probabilities to 'predicted' error percentages by taking guessing into account.[3] Remember that the patients had to select one from a pair of pictures. Then we utilized a least-squares method to estimate, for each individual patient, the C value yielding a minimal distance between his/her A, L, and P scores on the one hand, and the predicted A, L, and P error scores corresponding to that C value on the other.

The correlation between these C values and the patients' overall severity scores was 0.97 (Spearman's rank correlation coefficient; see scatter diagram 7F). The dotted lines of Figure 7E shows mean A, L, and P error percentages for the two groups of 'simulated patients'. The fit between observed and simulation data is very satisfactory.

5. The Unification Space as an Interactive Model of Parsing

The positive results of our simulation study qualify the Unification Space as a plausible cognitive architecture for the syntactic aspects of sentence processing. Actually we are not aware of any competing parsing model of comparable psycholinguistic validity.[4] Nevertheless, we immediately admit that the model leaves much to be desired. For example, we do not know which assumptions underlying the specific equations and parameters are indeed necessary. Better motivations are desirable here, and variants of the model need to be explored. However, we do consider as essential the three core features of the architecture listed in the first half of Section 3.

An equally important task is extension of the model with a sizeable lexicon and grammar, and with a semantic interpretation component. We are working in these directions, as well as towards a syntactic *generator* based on the Unification Space architecture.

Because contextual (i.e. semantic and pragmatic) factors are allowed to influence the value of strength function S in the unification and break-up equations, the Unification Space is an *interactive* parsing architecture. Recent empirical evidence obtained by Altmann & Steedman (1988), Altmann (1988) and Taraban & McClelland (1988) indeed appears to favor interactive models at the expense of simpler non-interactive 'syntax-first' architectures as advocated by Frazier (1987), Frazier & Fodor (1978) and Rayner *et al.* (1983). We believe the human syntactic processor is susceptible to contextual influences *as well as* to syntactic preferences (based on current activation levels, or built into the strength function). In the 'null-context' the latter are more likely to surface than in biasing contexts.

However, it is notoriously hard to create experimental conditions which expose syntactic processes without any lexical, semantic or pragmatic distortions. To make things even worse, syntactic preferences may counteract each other. A possible example is provided by the sentence material used by Altmann (1988) who compared the difficulty of interpreting sentences including an object complement clause (C) or a relative clause (R).

C: The psychologist told the woman *that he was worried about* her marital problems.

R: The psychologist told the woman *that he was worried about* to visit him again.

The italicized passage shared by the sentences creates the possibility of a garden-path. In the Unification Space two preferences would be in competition:

- a bias against attaching anything to modifier segments, i.e. against analyzing the *that* clause as an NP modifier, and
- a preference for unifying with more active nodes, i.e. with the recent NP node dominating *the woman* rather than with the object-S node introduced earlier as part of the lexical entry for *told*.

As a matter of fact, in his carefully designed experimental study, Altmann did not observe any symptoms of syntactic preferences either way. The Unification Space can provide an account for this finding by assuming a balance between rivaling syntactic preferences. It remains to be seen, however, whether the current parameter settings indeed produce this balance.

In conclusion, models of human sentence processing should be developed and evaluated against a broad spectrum of psychological and linguistic phenomena. From the perspective of this criterion, the Unification Space appears to represent a promising cognitive architecture.

Notes

1. In our implementation, the Unification Space is bound to reach a steady state. If unification/disintegration fails to converge on a unique parse tree and keeps oscillating, the number of cycles will exceed parameter T_B. This causes the input buffer to be emptied, so that the Unification Space will lose segment after segment (no replacement!). Global excitation will then soon drop below minimum $E_{\text{threshold}}$.
2. Inspection of the MA and NMA parse trees in Appendix B reveals that, from the SG viewpoint, '(non-)minimal attachment' is a misnomer: NMA does not involve more nodes or segments than MA. As already indicated in Section 3, the preference for so-called minimal attachment in the present pair of constructions derives from a slight bias against unifications involving the foot of modifier segments. Other famous (non-)minimal attachment examples such as MA′ and NMA′ would receive a different treatment in the Unification Space.

   MA′ The spy saw the cop with binoculars.
   NMA′ The spy saw the cop with a revolver.

   The final PP would be attached (via a modifier-PP segment) to either the NP dominating *the cop* or to the S node. Preferences for either attachment point presumably relate to the *S* (strength) values that would be assigned to the NP or the S nodes on the basis of semantic goodness of fit.
3. Three further details:

   (a) If P denotes the parsing probability, then the predicted error probability equals $(1-P)/2$.

   (b) Instead of parsing probability P we used corrected probability P_{corr}, computed as follows. The probability for some input sentence to be parsed correctly during a set of runs with, say, $C=0.6$ is denoted by $P(C=0.6)$. Then, the corrected probability for that C value, i.e. $P_{corr}(C=0.6)$, is given by $[P(C=0.5)+3P(C=0.6)+3P(C=0.7)+P(C=0.8)]/8$. Haarmann & Kolk provide a theoretical motivation for assuming a binomial distribution of severity parameter values in a patient.

(c) A few patients scored worse than chance in one or more picture matching tasks. We changed such error percentages to 50.

4. It is also interesting to note that the model easily satisfies a neurophysiological plausibility criterion proposed by Feldman (1986)—the '100 step rule'—in the following sense. The parsing process typically settles down on a parse tree in less than 100 cycles after the last word of the sentence has been entered into the system (see Figure 7D for an example; every 30th cycle a new word is entered into the Unification Space). This applies to shorter as well as longer sentences unless they contain garden paths, triple center-embeddings, etc.

References

Altmann, G. (1988) Ambiguity, parsing strategies, and computational models. *Language and Cognitive Processes*, **3**, 73-97.

Altmann, G. & Steedman, M. (1988) Interaction with context during human sentence processing. *Cognition*, **30**, 191-239.

De Smedt, K. (1990) *Incremental sentence generation. A computer model of grammatical encoding.* Dissertation, University of Nijmegen.

De Smedt, K. & Kempen, G. (1987) Incremental sentence production, self-correction, and coordination. In G. Kempen (Ed.) *Natural Language Generation: New Results in Artificial Intelligence, Psychology and Linguistics*. Dordrecht: Kluwer.

Elman, J.L. (1989) Structured representations and connectionist models. *Proceedings of the Eleventh Annual Conference of the Cognitive Society*. Hillsdale, NJ: Erlbaum.

Fanty, M.A. (1985) *Context-free parsing in connectionist networks*, Report TR 174. Rochester, NY: Computer Science Department, University of Rochester.

Feldman, J.A. (1986) *Neural representation of conceptual knowledge*, Report TR 189. Rochester, NY: Computer Science Department, University of Rochester.

Fodor, J.A. & Pylyshyn, ZW. (1988) Connectionism and cognitive architecture: a critical analysis. *Cognition*, **28**, 3-37.

Frazier, L. (1987) Theories of sentence processing. In J. L. Garfield (Ed.) *Modularity in Knowledge Representation and Natural-language Understanding*. Cambridge, MA: MIT Press.

Frazier, L. & Fodor, J.D. (1978) The sausage machine: a new two-stage parsing model. *Cognition*, **6**, 291-325.

Haarmann, H.J. & Kolk, H.H.J. (1988) *A computer model of the temporal course of agrammatic sentence understanding: the effects of variation in severity and sentence complexity.* Paper, NICI, University of Nijmegen.

Hinton, G.E. & Sejnowski, T.J. (1986) Learning and relearning in Boltzmann machines. In D. E. Rumelhart, J. L. McClelland and the PDP Research Group (Eds) *Parallel Distributed Processing. Explorations in the Microstructure of Cognition.* Cambridge, MA: MIT Press.

Howells, T. (1988) VITAL: a connectionist parser. *Proceedings of the Tenth Annual Conference of the Cognitive Science Society*. Hillsdale, NJ: Erlbaum.

Kalita, J. & Shastri, L. (1987) Generation of simple sentences in English using the connectionist model of computation. *Proceedings of the Ninth Annual Conference of the Cognitive Science Society*. Hillsdale, NJ: Erlbaum.

Kay, M. (1985) Parsing in functional unification grammar. In D. R. Dowty, L, Karttunen & A. M. Zwicky (Eds) *Natural Language Parsing. Psychological, Computational and Theoretical Perspectives*. Cambridge: Cambridge University Press.

Kempen, G. (1987) A framework for incremental syntactic tree formation. *Proceedings of the Tenth International Joint Conference on Artificial Intelligence (IJCAI '87)*. Milan.

Kempen, G. & Hoenkamp, E. (1987) An incremental procedural grammar for sentence formulation. *Cognitive Science*, **11**, 201-258.

Kimball, J.P. (1973) Seven principles of surface structure parsing in natural language. *Cognition*, **2**, 15-47.

Kolk, H.H.J. & Van Grunsven, M.M.F. (1985) Agrammatism as a variable phenomenon. *Cognitive Neuropsychology*, **2**, 347-384.

Levelt, W.J.M. (1989) *Speaking: From Intention to Articulation*. Cambridge, MA: MIT Press.

Marslen-Wilson, W.D. (1975) Sentence perception as an interactive parallel process. *Science*, **189**, 226-228.

Pylyshyn, Z.W. (1984) *Computation and Cognition. Toward a Foundation for Cognitive Science*. Cambridge, MA: MIT Press.

Rayner, K., Carlson, M. & Frazier, L. (1983) The interaction of syntax and semantics during sentence processing: eye movements in the analysis of semantically biased sentences. *Journal of Verbal Learning and Verbal Behavior*, **22**, 358–374.

Reeke, G.N. (1988) Protein folding: computational approaches to an exponential-time problem. *Annual Review of Computer Science*, **3**, 59–84.

Schwartz, M.F., Saffran, E.M. & Marin, A.S.M. (1980) The word order problem in agrammatism. I. Comprehension. *Brain and Language*, **10**, 249–262.

Selman, B. & Hirst, G. (1987) Parsing as an energy minimization problem. In L. Davis (Ed.) *Genetic Algorithms and Simulated Annealing*. Los Altos, CA: Morgan Kaufmann.

Taraban, R. & McClelland, J.L. (1988) Constituent attachment and thematic role assignment in sentence processing: influences of content-based expectations. *Journal of Memory and Language*, **27**, 597–632.

Touretzky, D.S. (1986) Representing and transforming recursive objects in a neural network, or "Trees *do* grow on Boltzmann machines". *Proceedings of the 1986 IEEE International Conference on Systems, Man, and Cybernetics*. Atlanta, GA.

Touretzky, D.S. & Hinton, G.E. (1988) A distributed connectionist production system. *Cognitive Science*, **12**, 423–466.

Van Wijk, C. & Kempen, G. (1987) A dual system for producing self-repairs in spontaneous speech: evidence from experimentally elicited corrections. *Cognitive Psychology*, **19**, 403–440.

Appendix A. Simulation model of the Unification System

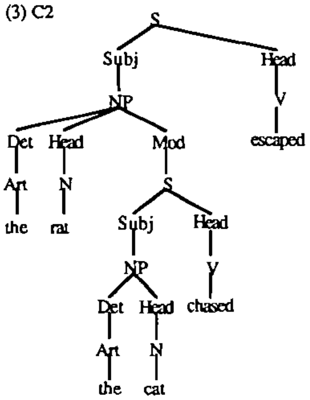
For explanation of this algorithm see paragraphs numbered (1) through (9) of Section 3.

```
Repeat until U-space contains exactly one conformation (stable tree) or $E$ has been below $E_{\mathrm{threshold}}$ for $T_{\mathrm{stop}}$
             cycles
1 Compute global excitation $E$: equation (4)
2 Randomly select two nodes $n_i$ and $n_j$
2.1 If $U(n_i, n_j)$ succeeds then
2.1.1 Compute $p[U(n_i, n_j)]$: equation (2)
2.1.2 If $Rnd<p[U(n_i, n_j)]$ then
2.1.2.1 Apply $U(n_i, n_j)$
2.1.2.2 Compute $A_u$: equation (1)
2.2.2.3 Remove alternative lexical entries (cf. paragraph 3)
3 For each $n_i \in$ U-space do
3.1 Compute new acitivation: formulae (5) and (6)
4 For each $n_i \in$ U-space do
4.1 For each segment with root $n_r$ and foot $n_f$ do
4.1.1 Compute $p[B(n_r, n_f)]$: equation (3)
4.1.2 If $Rnd<p[B(n_r, n_f)]$ then
4.1.2.1 Remove that segment
5 If $T_w$ cycles have passed since the current input word was fetched from the input buffer and input buffer
  is not empty then
5.1 Retrieve from lexicon the lexical entry/entries corresponding to next input word and insert it/them into
    U-space
6 If during the current cycle a lexical segment was removed from U-space due to disintegration (step
  4.1.2.1) and the corresponding word is still present in the input buffer (parameter $T_B$; cf. paragraph 1 and
  footnote 1) then
6.1 Retrieve from lexicon its lexical entry/entries and inset it/them into U-space
Notes: $E_{\mathrm{threshold}}=C/10$ (cf. paragraph 8)
       $T_{\mathrm{stop}}=100$ cycles
       $T_W=30$ cycles
       $T_B=1500$ cycles
       $Rnd$ generates a random real number between 1 and 0.
```

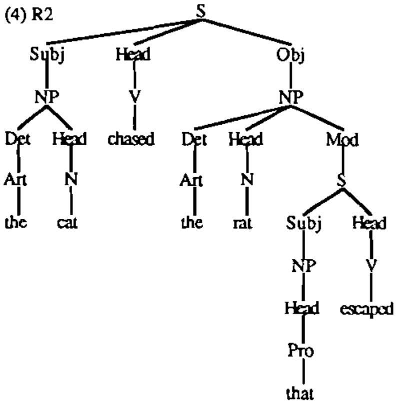
Appendix B. SG Parse Trees Corresponding to Sentences (3)–(12)

As a consequence of our tree drawing program, the labels of segment arcs (e.g. Subj, Head, Mod) are represented as separate nodes. Words rather than word class labels are used in the figure as terminal nodes. Apart from that, the tree diagrams are equivalent to that of Figure 1. The parse trees of sentences (1) and (2)—not shown here—are expansions of (3) and (4), respectively.

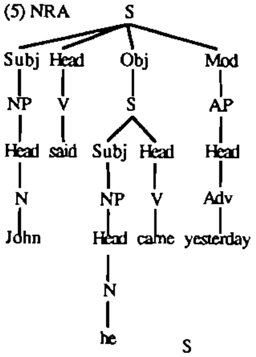
(3) C2



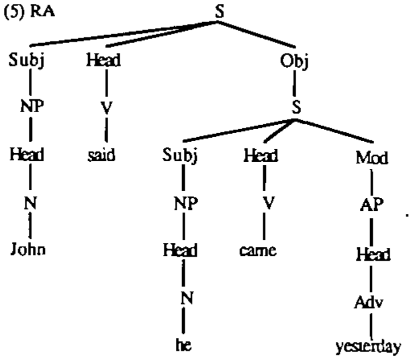
(4) R2



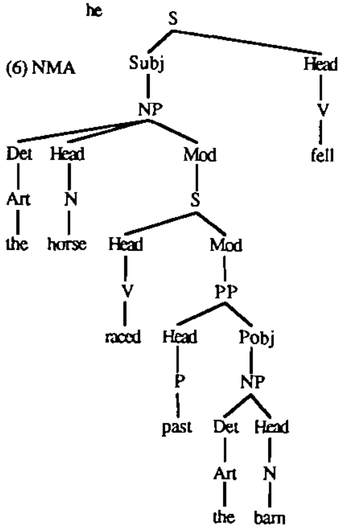
(5) NRA



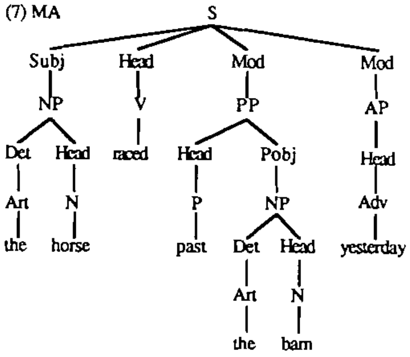
(5) RA



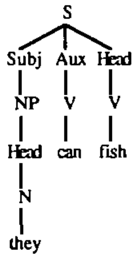
(6) NMA



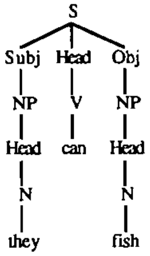
(7) MA



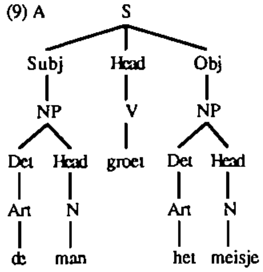
(8) Aux-V



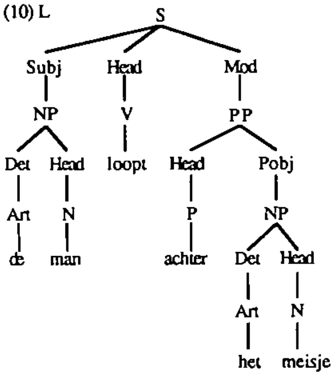
(8) V-N



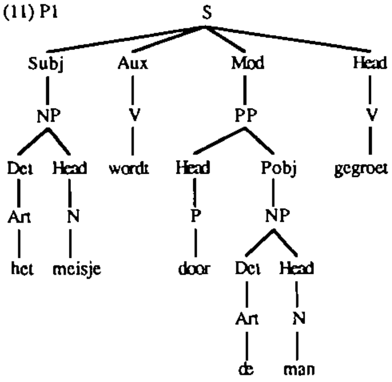
(9) A



(10) L



(11) P1



(12) P2