proceeds to a very detailed exposition of Yule's mathematical model of word distribution, which is followed on the next page by a discussion of the possible interpretations of de Saussure's term *valeur*. It is to be regretted that the section titles in many instances do not reflect their contents. Thus Section XV entitled "Information-theoretical interpretation of Chinese word structure" deals essentially with various methods for the binary coding of Chinese hieroglyphics which do not at all affect the Chinese word structure.

Some chapters of Herdan's book are an exposition of different aspects of probability theory and mathematical statistics, e.g. §§ 2.1, 2.2; 3.1 and 3.2 (p. 29), §§ 4.1, 4.2 (p. 41), etc.

There could have been no objection to this material as such, had it been more logically and more intrinsically connected with the central themes of the book.

Since all of the material is obviously intended as an aid for linguists, it should have been presented in a more popular way.

To sum up, Herdan's fundamental book which, like his previous publications, constitutes a kind of compendium on "statistical linguistics" will be of great use both to experienced linguists and to freshmen in the field of linguo-mathematical methods.

Moscow R. FRUMKINA
Institut jazykoznanija

William S. Cooper, *Set Theory and Syntactic Description* (= *Janua Linguarum*, series minor, XXXIV). The Hague, Mouton & Co., 1964. 52 pp. Gld. 8.—

The application of mathematical, logical, and set-theoretical notions to the grammatical description of natural languages has brought about a most beneficial revolution in linguistics. The first to use mathematical and logical concepts in a theory of grammatical description, was Louis Hjelmslev. Among the Bloomfieldians, it was Zellig S. Harris, who first introduced the conceptual apparatus of modern logic. During the last ten years the majority of the young linguists, headed, of course, by Noam Chomsky, and many of the older linguists, have been enthusiastically taking part in the development of what is sometimes called mathematical linguistics.

Unfortunately, however, the techniques of mathematical logic and set theory are not easy to master. They require a special training, to which linguists are forced to subject themselves, if they are to keep pace with recent developments. This explains in part the unwillingness of some linguists, especially in Europe, to accept, or even to appreciate, research done with the aid of mathematical or logical concepts.

This booklet by William Cooper serves a double purpose. It is "written for the reader who may not possess an extensive mathematical background, but who is nonetheless interested in the logical and mathematical foundations of the theory of grammatical description" (p. 5). On the other hand the "basic contention of this paper" is to provide evidence for the thesis that "syntactic description can *conveniently* be carried out within *elementary* set theory, introducing only a few special operations whose definitions are comparatively simple" (p. 51/2).

It cannot be sufficiently stressed that some knowledge of the principles of mathematical logic and set theory is indispensable for the grammatical linguist. There is, however, as yet, no book at hand, that gives an adequate selection of those logical and set-theoretical elements that are especially important in linguistics. As far as set theory is concerned, Cooper's paper certainly provides the linguist with a useful introduction, although I doubt if linguists without any previous training in the reading of mathematical texts, will be able to assimilate it even in the very simple form it has. In the beginning, at least, the uninitiated student needs coaching to learn how to read texts of a mathematical character. But, anyway, the publication of Cooper's study is fully justified, if only because it presents a not too difficult introduction to linguistically relevant elementary notions of set theory.

Cooper starts with a demonstration of the utility and importance of set theory (Chapter I), then gives an exposition of some simple set-theoretical concepts (Chapter II). In Chapter III Cooper introduces some "special operations" which he considers convenient for purposes of syntactic description, but which are generally not standard mathematical devices.[1] From Chapter IV onwards, the author finds himself in linguistic territory. He gives a set-theoretic characterization of natural languages and their grammars (Ch. IV). In Ch. V he presents thirteen examples of grammars of small subparts of English, of increasing complexity, and based upon the

[1] It should be noted that the definition of the concatenation operation, on p. 28, is erroneous: the right and left elements have been interchanged. The definition should read:

$$x^\frown y = x + \{\langle \beta, u\rangle : \langle \beta\text{-length}(x), u\rangle \; \varepsilon \; y\}.$$

general and special notions and operations that are explained in the Chapters II and III. Ch. VI is a defense of "the set theory thesis", the presentation of which was the second aim of the paper under review.

As a preparation to the reading of current mathematical-linguistic literature this book is, regrettably, incomplete. Nothing is said about the theory of deduction, or, more specifically, about the theory of algorithms. The reason for this limitation is, doubtless, the author's conviction that "to find a sufficiently flexible framework" for grammatical description "one need look no further than elementary set theory" (p. 16). That is, the isagogic value of this book is limited because of the author's set theory thesis. Cooper writes: "Some systems of linguistic analysis have already been formulated in terms of specialized mathematical notations of a restricted sort. If set theory is regarded merely as an addition to this list, in competition with the rest, the point of using an extremely general formalism such as set theory will have been lost. Set theory is intended to serve a purpose of a different sort from that of these other systems. Previous mathematically-formulated linguistic systems have been designed to provide a restricted mathematical framework suitable only for those specific notions central to the particular system of linguistic analysis. Set theory, by contrast, is intended more as a universal system which does not depend on any particular linguistic notions, but rather provides a foundation broad enough to support the existing systems, and hopefully future systems also" (p. 15). I am afraid, however, that this does not do justice to "previous mathematically-formulated linguistic systems".

The most notable example of a type of mathematically-formulated grammatical description is the Chomskian grammar, best known from Chomsky's *Syntactic Structures* (= *Janua Linguarum*, IV) ('s-Gravenhage, Mouton & Co., 1957[1], 1963[3]). Although it is not explicitly stated there, it must be recognized, that a Chomskian generative grammar is, essentially, a complicated algorithm.[2] An algorithm, as expounded by Rosenbloom,[3] is a deductive apparatus, or finite automaton, characterized by the following elements:

(1) a finite alphabet (c.q. a finite number of primitive terms)

(2) a finite number of initial strings (c.q. axioms)

(3) a finite number of rewriting rules (c.q. instruction rules, or productions).

[2] Cf. E. W. Beth, "Konstanten van het wiskundige denken", *Mededelingen der Koninklijke Nederlandse Akademie van Wetenschappen, Afd. Letterkunde*, Nieuwe Reeks, Deel 26, No.7 (Amsterdam, 1963), p. 249.

[3] Paul Rosenbloom, *The Elements of Mathematical Logic* (New York, Dover Publications Inc., 1950) (especially Chapter IV).

Strictly speaking, the theory of algorithms, developed mainly by E. L. Post, is part of set theory, since it is based upon the primitive terms of set theory and serves to characterize sets of a given sort. Algorithms may serve to characterize the set of theorems deducible from a set of axioms in primitive terms by means of a set of instruction rules. One notices that the structure of an algorithm is completely isomorphic to the structure of what is known in logic as a deductive theory, producing theorems from axioms and primitive terms. Algorithms are used also to characterize, by means of deduction, certain sets of numbers. In general, recursively enumerable, or canonical, sets are sets that can be characterized by means of an algorithm. As far as natural languages are concerned "Chomsky's work can quite naturally be described as an inquiry into the possibility of such an algorithm" (Beth, *l.c.*). In short, the theory of algorithms can be viewed as the general theory of deduction.

Therefore, if algorithmical grammars are among "previous mathematically formulated linguistic systems", it is certainly incorrect to say that these systems "have been designed to provide a restricted mathematical framework suitable only for those specific notions central to the particular system of linguistic analysis". On the other hand, since the theory of algorithms may be regarded as a branch of set theory, the author's contention that "the primitive technical terms needed for syntactic description are just the primitive technical constants of set theory" (p. 51), does not necessarily imply a rejection of algorithmical approaches. Yet the author apparently excludes algorithmical devices. He even seems unwilling to regard an algorithmical grammar as a competitor of his set-theoretic type of grammar. Let us overlook, however, this unwillingness, and see if Cooper's set-theoretic approach can stand the competition.

As in the algorithmical theory of grammar developed by Chomsky, a language L is considered a set of finite sequences, or strings, constructed out of a finite set of elements, the alphabet of L. Thus L is a subset of the set of all sequences constructible out of the alphabet of L. The sequences belonging to L are called grammatical sequences. The aim of a grammar Γ of L is to define, or to characterize, the set of grammatical sequences, separating them from the complement of L, which is the set of non-grammatical sequences in the same alphabet. "The function of a grammar, then", Cooper writes, "is to specify, or describe, some particular set of sequences" (p. 33). "Γ is a grammar of L if and only if Γ is a well-formed mathematical definition and Γ defines L" (p. 34).

But here their roads part. Cooper proceeds along the well-known lines of set-theoretic definition. He starts with the definitions of a number of primitive sets, then defines new sets out of the primitive ones, applying the operations given in the Chapters II and III, and so forth, until he arrives at the definition of a set L, composed out of the previously defined sets, and equivalent to some language. Let us take Grammar III (p. 38) in order to demonstrate this procedure:

"GRAMMAR III:

(1) $S = \{klif, mɔs, bɔg\}$
(2) $P = \{klifs, mɔsiz, bɔgz\}$
(3) $D = \{ðə, səm\}$
(4) $Q = D \cdot S$
(5) $Q_0 = D \cdot P$
(6) $Q_1 = Q + Q_0$
(7) $V = \{kəvirz, sərawndz\}$
(8) $V_0 = \{kəvir, sərawnd\}$
(9) $L_2 = Q \cdot V \cdot Q_1 + Q_0 \cdot V_0 \cdot Q_1$

The language L_2 contains 288 sequences".[4]

(The operation "·" (interconcatenation) is defined on p. 29 in the following way: "If A and B are sets of sequences, A · B is the set of all sequerces, which can be formed by concatenating a member of A onto the left end of a member of B". The "+" operation is set-theoretic union.)

Chomsky's grammar does not explicitly define a set, but can be considered a device for producing, in an automatic way, sequences of symbols, within an alphabet, from given "axiomatic" sequences, following a finite number of rewriting rules. The algorithmical grammar indirectly defines the sentences of a language by producing them all, and only these. Algorithms can be said to characterize sets of sequences, rather than to define them in set-theoretical terms.

Among linguists it is a commonly accepted opinion that natural languages are most adequately handled if they are treated as virtually infinite sets. There seems to be no limit to the number of sentences of a natural language. Given any sentence, one can always produce a new sentence that is longer than the original one. Any limit that a grammar might be made to set to the length of sentences in a natural language, is arbitrary. The grammatical apparatus presented by Cooper is very poor in this re-

[4] An unelegant feature of this grammar is, that it does not distinguish between different uses of "some". E.g. "some cliff surrounds the bog" is ungrammatical in one sense, but acceptable in another. A different objection can be raised against Grammar V (p. 40), which permits e.g. "now the cliff now covers the bog now", or "already the cliff now covers the bog already".

spect. All but one of his thirteen sample grammars define a finite number of sentences. Only Grammar VI allows for an infinite number, because of the definition of one infinite set, namely the set of all sequences formable from the two adjectives *old* and *green*, and the identity sequence O. This infinite set is defined by means of the special operation of exponentiation to the power infinity: "A^∞ is the set of all sequences formable from the members of A and the identity sequence by a finite number of applications of the concatenation operation. If A contains any members not equal to 0, A^∞ will contain an infinite number of (finite) sequences.

EXAMPLES: $\{t\}^\infty = \{0, t, tt, ttt, tttt, \ldots\}$;
$\{t, e\}^\infty = \{0, t, e, tt, ee, te, et, ttt, \ldots\}$" (p. 31).

This operation of raising to the power infinity is, in Cooper's exposition, the only means of defining an infinite set of sequences. Natural languages, however, constitute infinite sets by more sophisticated processes than the indeterminate stringing together of elements of a given set, say adjectives, as in the language of Grammar VI.

It is curious to note that all languages defined by the thirteen grammars of Chapter V are finite state languages.[5] Yet the theoretical resources of the apparatus set forth in the Chapters II and III are more powerful than to allow only for the definition of finite state languages. For instance, the model languages

L_1: *ab*, *aabb*, *aaabbb*, ..., and in general, all sentences consisting of n occurrences of *a* followed by exactly n occurrences of *b*, and only these;

L_2: *aa*, *bb*, *abba*, *baab*, *aabbaa*, ..., and in general, all 'mirror image' sentences consisting of a string X followed by X in reverse, and only these;

L_3: *aa*, *bb*, *abab*, *baba*, *aabaab*, ..., and in general, all sentences consisting of a string X followed by the identical string X, and only these,

which are not finite state languages, as Chomsky demonstrated,[6] can be defined in Cooper's terms as follows:

For L_1: $A = \{a\}^\infty$

$$\boxed{B = \{b\}^\infty}$$

$$L_1 = \{x^\frown y : x \in A, y \in B, \text{length}(x) = \text{length}(y)\}.$$

[5] Cf. Noam Chomsky, "Three Models for the Description of Language", *Institute of Radio Engineering. Transactions on Information Theory*, vol. IT-2, *Proceedings of the Symposium on Information Theory, September 1956.*

For L_2: $A = \{a, b\}^{\infty}$

$L_2 = \{x^\frown y : x \in A \text{ and } y = \bar{x}\}$.

A definition is required here of the reverse of a sequence ($\bar{s}$) (s), which may be formulated in the following way:

$\bar{s} = \{\langle \beta, s(\text{length } (s) - \beta + 1)\rangle : 1 \leqslant \beta \leqslant \text{length } (s)\}$.

For L_3: $A = \{a, b\}^{\sim}$

$L_3 = \{x^\frown x : x \in A\}$.

This implies that the devices of the Chapters II and III may be applied to construct grammars that are essentially more powerful than the thirteen sample grammars of Chapter V. Yet the question remains whether a Cooperian grammar is in any sense superior to an algorithmical one. To answer this question we may employ as a criterion the *decision problem* for sets of sequences, or languages.[7]

The DECISION PROBLEM for a set of sequences or strings consists in determining whether a given string belongs to the set or not. A DECISION PROCEDURE is a finite number of unequivocally defined steps, by which one determines whether a given string belongs to the set or not. The application of decision procedures does not require any creative intelligence. They can be executed by an automaton, or, as Rosenbloom says, by a happy moron. Sets for which there is a decision procedure (or, for which the decision problem has been solved), are called SOLVABLE, or RECURSIVE, sets. A set of strings that can be characterized by means of an algorithm is a CANONICAL set. Every solvable set is canonical. That is, for every solvable set an appropriate algorithm can be designed. But there are canonical sets that are unsolvable. For these sets no decision procedure can be found.

The languages L_1, L_2, and L_3 constitute solvable sets. Their decision procedures can be derived from their informal descriptions given above In general, all languages that can be defined by the descriptive devices set forth in the Chapters II and III are solvable sets, as can be readily seen from the following:

Since all finite sets are solvable, the finite sets definable in Cooper's terms are solvable. The only infinite sets in Cooper's apparatus are those defined by the operation of raising to the power infinity. These are solvable also, as one sees intuitively, because for any given sequence to belong

6 Noam Chomsky, "Three Models", 2.2, and *Syntactic Structures*, 3.2.

7 See for a detailed description of this problem and related subjects, Rosenbloom, *o.c.*, Ch. IV, and E. W. Beth, *The Foundations of Mathematics. A Study in the Philosophy of Science* (= *Studies in Logic and the Foundations of Mathematics*) (North-Holland Publishing Company, Amsterdam, 1959) (especially Chapter 21).

to a set x^∞ it is a necessary and sufficient condition that it consist of only members of x. This can be made out in a finite number of steps, since x is finite and any sequence of x^∞ is finite in length. More formally the solvability, or recursiveness, of all sets x^∞ can be demonstrated by reformulating the definition of x^∞, on p. 31, as follows:

$$x^\infty = \{y : y \in x^{\text{length }(y)}\}.$$

Since this is a recursive definition, all sets of sequences defined by this operation are recursive, hence solvable, sets.[8]

Thus it appears that the descriptive possibilities of Cooper's apparatus are essentially poorer than those of algorithmical systems, which provide descriptions of unsolvable canonical sets also.

This does not mean, of course, that set theory does not provide adequate descriptive devices for natural languages. First of all, it is still a moot point whether natural languages are in fact unsolvable sets. It is true that as yet no decision procedure has been found for natural languages. That is, there exists as yet no general automatic procedure for the grammatical analysis of given sentences of some language. But it is very well possible that such a procedure will be found some day. However, even if natural languages do constitute unsolvable sets, the set theory thesis is, as Cooper says, "almost trivially true, for set theory has been found adequate to the definition of virtually every known mathematical term, relation, and operation. ... Set theory in its broadest sense takes in all of known mathematics, and our hypothesis really says no more than that syntactic description can be carried out within present-day mathematics" (p. 51/2). But then the mathematical, or set-theoretical, framework has to include algorithmical devices.

Summarizing we may say that the paper under review provides linguists with a most valuable introduction to some basic notions of set theory, and gives at the same time a good training in the reading of simple set-theoretically formulated texts. On the other hand, the author's views upon grammatical problems seem to be rather naive, in that the sample grammars given in Chapter V do not exceed the limits of finite state grammars, although the descriptive techniques given in the Chapters II and III allow for more powerful grammars. They cannot compete, however, as they are, with existing algorithmical systems, since they do not permit the definition of unsolvable sets of sequences.

University of Groningen PETER A. M. SEUREN

[8] I am indebted to W. A. van der Moore of the University of Amsterdam for this simple and elegant proof.