XIII. MECHANICAL TRANSLATION[*]

V. H. Yngve
J. R. Applegate
A. N. Chomsky
U. C. Dickman
M. Dyck
E. S. Klima
K. C. Knowlton
R. B. Lees
G. H. Matthews

A. SOME PROPERTIES OF PHRASE STRUCTURE GRAMMARS

As stated in earlier reports, we consider a language to be a set of finite strings (sentences) in a finite vocabulary, and a grammar to be a device that gives a recursive enumeration of the sentences of the language. Consider a grammar G of the following form: G is based on a finite vocabulary $V = V_T \cup V_N$ (V_T, V_N disjoint) and a finite irreflexive relation "rewrite ϕ as ψ" (symbolized by $\phi \to \psi$), where ϕ and ψ are strings in V, and ϕ is not a string in V_T. There is a symbol $S \in V_N$ and a symbol $\# \in V_T$ with the property that no sentence in the generated language is of the form $\phi \# \psi$, where ϕ and ψ are not null (see definitions below).

To simplify the discussion, we introduce the following notation: Capital letters will represent strings in V_N; lower-case letters will represent strings in V_T (or the identity); and Greek letters will represent arbitrary strings. Early letters of the alphabet will be used for single symbols; late letters, for strings; and μ and ν will be arbitrary variables.

We say that ψ *follows from* ϕ if $\phi = \chi_1\omega_1\chi_2$, $\psi = \chi_1\omega_2\chi_2$, and $\omega_1 \to \omega_2$. $(\phi_1, \ldots, \phi_n)$ is a *ϕ-derivation* if $\phi = \phi_1$ and for $i < n$, ϕ_{i+1} follows from ϕ_i. A ϕ-derivation is *terminated* if it is not a proper initial subsequence of another ϕ-derivation. L_G is the (terminal) language *generated* by G if $L_G = \{x|$ there is a #S#-derivation terminating in $x\}$. ϕ *represents* ψ if (i) for some ω_1, ω_2, $\omega_1 \phi \omega_2 \to \omega_1 \psi \omega_2$; or (ii) ϕ represents χ, and χ represents ψ; or (iii) $\phi = \omega_1\omega_2\omega_3$, $\psi = \omega_1\omega_4\omega_3$, and ω_2 represents ω_4; or (iv) $\phi = \omega_1\omega_2$, $\psi = \chi_1\chi_2$, and ω_i represents χ_i. G is *equivalent* to G^* if they generate the same language. A derivable string is one which is a line in a #S#-derivation.

It is known that any recursively enumerable set can be represented as a terminal language in the sense defined above. As might be expected, grammars of this type will not, in general, have linguistic significance; that is, in general, it will not be possible to derive from the grammar a meaningful structural description of the output sentences. To insure significance, we may impose various restrictions on these grammars:

Restriction 1: If ψ follows from ϕ, then there are unique strings A, ω_1, ω_2, ω_3 with the property that $\phi = \omega_1 A \omega_2$, $\psi = \omega_1\omega_3\omega_2$, and ω_3 is not the identity element.

Restriction 2: If $\phi \to \psi$, then for some A, $\phi = A$, and ψ is not the identity.

Grammars that meet these restrictions give decidable languages only. They have some linguistic significance in the sense that a structural description with many of the formal properties of traditional phrase structure can be assigned to each sentence generated. This linguistic interpretation has received some study (1). We shall see, however, that neither of these restrictions is quite appropriate.

[*] This work was supported in part by the National Science Foundation.

THEOREM 1. Suppose that G meets restriction 1 and that X, B are strings of G. Then we can construct a grammar G^* that will meet restriction 1, which is equivalent to the grammar G' formed by adding the rule XB → BX to G.

Suppose also that $X = A_1 \ldots A_n$. Choose $\bar{A}_1, \ldots, \bar{A}_n, \bar{B}$ to be all new and distinct from one another. Let $\sum$ be the sequence of rules:

$$A_1 \ldots A_n B \rightarrow \bar{A}_1 A_2 \ldots A_n B$$

$$\bar{A}_1 A_2 \ldots A_n B \rightarrow \bar{A}_1 \bar{A}_2 A_3 \ldots A_n B$$

$$\vdots$$

$$\rightarrow \bar{A}_1 \ldots \bar{A}_n \bar{B}$$

$$\rightarrow \bar{A}_1 \ldots \bar{A}_n \bar{A}_n$$

$$\rightarrow \bar{A}_1 \ldots \bar{A}_{n-1} \bar{A}_{n-1} \bar{A}_n$$

$$\rightarrow \bar{A}_1 \ldots \bar{A}_{n-2} \bar{A}_{n-2} \bar{A}_{n-1} \bar{A}_n$$

$$\vdots$$

$$\rightarrow \bar{A}_1 \bar{A}_1 \bar{A}_2 \ldots \bar{A}_n$$

$$\rightarrow B \bar{A}_1 \ldots \bar{A}_n$$

$$\rightarrow B A_1 \bar{A}_2 \ldots \bar{A}_n$$

$$\vdots$$

$$\rightarrow B A_1 \ldots A_n$$

where the left-hand element of each rule is the right-hand element of the immediately preceding one. It can be shown that if $(\phi_1, \ldots, \phi_n = x)$ is a #S#-derivation of G' that follows the rules of Σ, then there is another #S#-derivation of G' terminating in x in which these rules are applied only in the sequence Σ. Consequently, G^* formed by adding the rules of Σ to G is equivalent to G', and it clearly meets restriction 1.

Now consider the grammar G with the following characteristics:

$$V_T = \{a, b, c\}; \quad V_N = \{S, S', S'', A, \bar{A}, B, \bar{B}, C, D, E, F\}$$

Rules: (I) (a) $S \rightarrow CDS'F$ (b) $S' \rightarrow S''S'$ (c) $\left\{\begin{array}{l} S'F \rightarrow AF \\ S'F \rightarrow BF \end{array}\right\}$

(d) $\left\{\begin{array}{l} S''A \rightarrow AA \\ S''B \rightarrow AB \end{array}\right\}$ (e) $\left\{\begin{array}{l} S''A \rightarrow BA \\ S''B \rightarrow BB \end{array}\right\}$

(II) (a) $\begin{Bmatrix} CDA \rightarrow CE\bar{A}A \\ CDB \rightarrow CE\bar{B}B \end{Bmatrix}$ (b) $\begin{Bmatrix} CE\bar{A} \rightarrow \bar{A}CE \\ CE\bar{B} \rightarrow \bar{B}CE \end{Bmatrix}$ (c) $E\mu\nu \rightarrow \nu E\mu$

(d) $E\mu\# \rightarrow D\mu\#$ (e) $\mu D \rightarrow D\mu$

(III) $CDF\mu \rightarrow \mu CDF$

(IV) (a) $\begin{Bmatrix} \bar{A} \rightarrow A \\ A \rightarrow a \\ \bar{B} \rightarrow B \\ B \rightarrow b \end{Bmatrix}$ (b) $CDF\# \rightarrow CDc\#$ (c) $CDc\# \rightarrow Ccc\#$

(d) $Ccc\# \rightarrow ccc\#$

where μ, ν range over A, B, and F.

It can be shown that the only #S#-derivations of G that terminate in strings of V_T follow this procedure:

(1) The rules of (I) are applied as follows: (a) once, (b) n - 1 times (for some n ≥ 1), (c) once, and then (d) or (e) a total of n - 1 times. The result is a string:

$\#CD\mu_1\ldots\mu_nF\#$ $(\mu_i = A \text{ or } B)$

(2) The rules of (II) are applied as follows: (a) once, and (b) once, to give

$\#\bar{\mu}_1CE\mu_1\ldots\mu_nF\#$ (where in general, $\bar{\mu}_i = \bar{A}$ if $\mu_i = A$, $\bar{B}$ if $\mu_i = B$)

then (c) n + 1 times and (d) once, to give

$\#\bar{\mu}_1C\mu_2\ldots\mu_nFD\mu_1\#$

then (e) n times, to give

$\#\mu_1CD\mu_2\ldots\mu_nF\mu_1\#$

(3) The rules of (II) are applied as in (2), to give

$\#\bar{\mu}_1\bar{\mu}_2CD\mu_3\ldots\mu_nF\mu_1\mu_2\#$

⋮

(n+1) the rules of (II) are applied as in (2), to give

$\#\bar{\mu}_1\ldots\bar{\mu}_nCDF\mu_1\ldots\mu_n\#$

(n+2) the rule (III) is applied n times, to give

$\#\bar{\mu}_1\ldots\bar{\mu}_n\mu_1\ldots\mu_nCDF\#$

(n+3) the rules of (IV) are applied, (a) 2n times, (b), (c), (d) once each, to give

$\#\nu_1\ldots\nu_n\nu_1\ldots\nu_nccc\#$, where $\nu_i = a$ if $\mu_i = A$, b if $\mu_i = B$.

Any other sequence of rules will fail to lead to a string of V_T. Notice that the form of the terminal string is completely determined by step (1) above. The number of applications of (Ib) determines its length; the choice in (c) and the choices of (d) or (e) determine its form. Rules (II), (III), and (IV) merely copy the output of (I) (and convert it into terminal form, suffixing ccc). By Theorem 1 there is a grammar equivalent to G that meets restriction 1. Consequently, we have Theorem 2.

THEOREM 2. Let L be the language that consists of all and only sentences #xxccc#, where x is some sequence of a's and b's. Then there is a grammar of L that meets restriction 1.

It is also possible to construct a somewhat more complex grammar that meets this restriction and gives the language $\{\#xx\#\}$.

THEOREM 3. Let L be as in Theorem 2. Then there is no grammar of L that meets restriction 2.

PROOF. Suppose that G is a grammar of L that meets restriction 2. We can assume that for each $A \in V_N$ there are infinitely many x's with the property that A represents x (if there are not, A can be eliminated from the grammar completely). Now consider all sentences of the form $\#a^n b^m a^n b^m ccc\#$. Evidently there are infinitely many derivations of such sentences in which, for some letter A, the next to last line of the derivation is #xAyccc#, for some x, y (not both null). By considering the various possibilities for x and y in these cases we can now see that G will give infinitely many sentences not of the form #zzccc#, since for each w represented by A there will be a terminal sentence #xwyccc# for each derivable string #xAyccc#. Therefore, G is not a grammar of L.

We see, then, that restrictions 1 and 2 have essentially different effects. Restriction 2 excludes rules of the form: Rewrite ϕ as ψ in the context $\omega_1 --- \omega_2$. These contextual rewriting rules are permitted in grammars that meet restriction 1. Clearly rules of this type are needed in grammar. But the extra power of grammars with contextual rewriting rules must be considered an inadequacy of these grammars, very much as in the case of grammars that meet no restrictions. The reason is that the rules of the form $XB \rightarrow BX$ which are permitted, essentially, by these grammars, have no celar linguistic meaning – at least in terms of phrase structure. That is, the relation "represents" may be reflexive in the case of these grammars, and this conflicts with the desired linguistic interpretation. It seems necessary, for linguistic significance, to find some restriction that is weaker than 2 but stronger than 1. It would be interesting, for example, to see whether or not there is a natural way to impose on grammars a restriction that will insure the irreflexivity of "represents."

N. Chomsky

References

1. Cf. A. N. Chomsky, Three models for the description of language, Trans. IRE, vol. IT-2, no. 2, pp. 113-124, Sept. 1956. Also, Syntactic Structures (Mouton and Company, The Hague, 1957).