XVII. MECHANICAL TRANSLATION*

Prof. V. H. Yngve	Dr. G. H. Matthews	J. Gough, Jr. (visiting)‡
Prof. J. R. Applegate	Dr. B. Ulvestad (visiting)†	K. C. Knowlton
Prof. A. N. Chomsky	Ursula C. Dickman	R. B. Lees

RESEARCH OBJECTIVES

The basic objective of our program is to devise techniques by which languages can be translated by machine. There are two aspects of the program: the linguistic and the mechanical. The two must progress side by side because the methods of each influence the results of the other. Moreover, a balance must be maintained between short-range programs which aim at the immediately practicable, and long-range research which may add to the basic understanding needed for future progress.

Work is continuing on an analysis of German sentence structure from the point of view of translation into English. In addition, work is being done on the theory of language and on general considerations of machine capabilities and how they impinge on the syntactic problems.

V. H. Yngve

A. ON CHARACTERIZATION OF FINITE-STATE LANGUAGES

A finite-state grammar G is determined by a set of internal states $S_0, \ldots, S_n$ and transition symbols $W_0, \ldots, W_m$. S_0 is the initial state; W_0, the identity element. Each state S_i can be represented as a set of triples (i, j, k). Such a triple indicates that when the grammar is in the state S_i, the symbol W_j can be produced to switch the grammar into the state S_k. Suppose the grammar G passes through a sequence of states, beginning and ending with S_0 and containing no other occurrence of S_0. It will then have produced the sentence consisting of the string of symbols in the order in which they were selected at the successive transitions. A finite-state language is any language generated by a finite-state grammar. This grammar can be represented in a familiar way by a state diagram. Two finite-state grammars are said to be equivalent if they produce the same language. Given any finite-state grammar G, it is possible to construct an equivalent grammar G^* with the property that for any two states S_i, S_j, there is a pair $[k, l]$, with $(i, k, l) \in S_i$ and $(j, k, l) \notin S_j$. Hence each state of G^* can be represented unambiguously as a set of pairs that indicate the transition symbol and the following state. Each pair will be called a "grammatical rule." We consider, henceforth, only grammars of this form.

Clearly, the loops in the state diagram are of particular relevance in characterizing any finite-state language. Given a finite-state grammar G, define a cycle as a sequence of states $\left(S_{a_1}, \ldots, S_{a_m}\right)$, with $m \geq 2$, $a_1 = a_m$, and for $1 < i < m$, $a_1 \neq a_i \neq 0$.

---

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

†From University of Bergen.

‡From Georgia Institute of Technology.

S_{a_1} will be called the initial state of the cycle. A basic cycle is a cycle whose initial state is S_o. The sequence of cycles $(C_o, \ldots, C_n)$ is a chain if C_o is a basic cycle, each C_i contains the initial state of C_{i+1} $(0 \leq i < n)$, and no C_j contains the initial state of C_{j-i} $(1 \leq i \leq j \leq n)$. It is a completed chain if no sequence $(C_o, \ldots, C_n, C_{n+1})$ is a chain. Clearly, G has only a finite number of cycles and a finite number of chains. We can use these chains as the basis for constructing a grammar, with a particularly simple structure, equivalent to G.

Construct G^* in the following manner. Suppose that the completed chains of G are $H_1, \ldots, H_p$. Construct states $T_1, \ldots, T_p$, with each T_i containing $[0, i]$ as its sole member. These "absorbing states" will be used to index and to identify occurrences of states in different chains. Next, set an arbitrary one-one correspondence between finite sequences of integers and integers greater than p, to be used for purely notational purposes. If $(b_1, \ldots, b_k) \longleftrightarrow b$ under this correspondence, we shall use "$(b_1, \ldots, b_k)$" and "b" interchangeably in characterizing states and grammatical rules.

Now suppose that $H_i = (C_o, \ldots, C_m)$ is the i^{th} completed chain of G, and that $(a_o, \ldots, a_m)$ is the sequence of indices of the initial states of $C_o, \ldots, C_m$, respectively. Suppose that $C_j = \left(S_{j_1}, \ldots, S_{j_n}\right)$ $(0 \leq j \leq m)$. For each $1 \leq k < n$, construct the state $T_{(i, a_o, \ldots, a_j, j_k)}$ containing $[0, i]$ and each grammatical rule $[r, (i, a_o, \ldots, a_j, j_{k+1})]$ so chosen that $[r, j_{k+1}] \in S_{j_k}$. Having carried out this construction for each chain, identify the states labeled $T_{(x, a_o, \ldots, a_j, a_j)}$ and $T_{(y, a_o, \ldots, a_j)}$. Let $T_{(x, 0)}$ be the initial state of G^*.

In G^* the analysis of G into chains is made explicit. We can establish that G^* is equivalent to G.

Suppose that we recursively define the notation

$$a_1(a_2, \ldots, a_m)a_{m+1} \tag{1}$$

where the a_i's are strings or, again, are notations of the form

$$x_1(x_2, \ldots, x_t)x_{t+1} \tag{2}$$

and so on, in the following way. Let Q_1 be the set of sequences $(b_1, \ldots, b_{n+1})$, where $b_1 = a_1$, $b_{n+1} = a_{m+1}$, and each b_i $(2 \leq i \leq n)$ is one of $a_2, \ldots, a_m$. Let Q_2 be the set of sequences formed from the sequences of Q_1 by expanding the b_i's that are not already strings in the same manner, and so on. Then for some r, Q_r will be a set of sequences $(z_1, \ldots, z_s)$, in which each z_i is a string. Each of these sequences represents the string $z_1 \ldots z_s$, and expression 1 represents the set of these strings.

It is clear that for any G^* constructed in the aforementioned manner, the language generated by G^* can be represented completely in a finite manner in this notation. In

fact, the representation can be read off directly from the state diagram. By virtue of the equivalence that has been stated, we obtain the result that any finite-state language can be represented by a finite number of finite notations in the form of expression 1. Thus a finite-state language can be said to have an elementary sort of bracketing or "phrase structure" imposed on it.

A. N. Chomsky