Quarterly Progress Report of the Research Laboratory of Electronics;
M.I.T., July, 1958

XIV. 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. A COMPILER-INTERPRETER FOR MECHANICAL TRANSLATION

It has been said that an automatic digital computer can do anything with symbols that we can order it to do. If we are interested in telling a digital computer to translate a certain text from one language into another language, we are faced with two tasks. Our first task is to find out how to translate the text. Our second task is to "tell" the computer how to do it. The first task is a problem for skilled linguists. In connection with the second task, we have been developing a programming language in which the linguist can conveniently "tell" the computer to do things that he wants it to do. We have completed a description of the programming language, and the programming necessary for converting a program written in the programming language into a 704 computer program is under way.

la the past, it has been the custom for the linguist who wanted to try out a certain approach to mechanical translation to ask an expert programmer to program his material rather than to learn how to program it himself. In addition to the usual inconveniences and difficulties that attend communication between experts in two separate fields, this practice has a further disadvantage: Neither the linguist nor the programmer has been able to be fully effective. The linguist has not been aware of the full power of the machine, and the programmer, not being a linguist, has not been able to use his special knowledge of the machine with full effectiveness on linguistic problems.

The availability of an adequate programming language promises to be of great assistance in linguistic research in mechanical translation. The linguist can write the results of his research in a notation especially devised to fill his needs. The programming for the computer is taken care of once and for all by an automatic programming system. Thus the expense, time, and effort needed to program each linguistic approach separately is saved, and, even more important, the linguist is given direct access to the machine. He becomes more fully aware of its potentialities, and his work is greatly facilitated.

Besides serving for the programming of linguistic problems, our programming language can also be of assistance to other potential users of computers. It is a very convenient programming method for problems that involve a considerable amount of manipulation of arbitrary symbols and very little, if any, arithmetic. It is quite easy, for example, to program algebraic manipulations.

V. H. Yngve, G. H. Matthews

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