From: .QUARTERLY PROGRESS REPORT No. TO, July 15, 1963, R. L. E.

XVII. MECHANICAL TRANSLATION*

V. H. Yngve
Carol M. Bosche
Elinor K. Charney
Ann Congleton
J. L. Darlington
Muriel S. Kannel
E. S. Klima
E. S. Lowry
J. D. McCawley
T. More, Jr.
W. K. Percival
A. C. Satterthwait
K. G. Sellin
J. S. Wright

A. LOGICAL EVALUATION OF ARGUMENTS STATED IN "FORMAT Q"

The COMIT program for logical translation and evaluation (Quarterly Progress Reports No. 68 (pages 174-175) and No. 69 (pages 165-168)) has been developed to the point at which one may submit for evaluation an entire argument written in a quasi-logical notation, "format Q." The program translates the argument into a strictly logical functional calculus notation, "format L," and then proceeds to test its validity by using the Davis-Putnam proof-procedure algorithm. The following excerpts from the machine output resulting from the translation and evaluation of a sample argument may be presented and briefly discussed. The sample argument, as it originally occurred in a logic textbook,[2]

> "Whoever belongs to the Country Club is wealthier than any member of the Elks Lodge. Not all who belong to the Country Club are wealthier than all who do not belong. Therefore not everyone belongs either to the Country Club or the Elks Lodge."

This argument was translated by hand into the following "format Q" representation; in this form it was submitted to the machine, which then proceeded to translate it into "format L," test it, and find it to be valid, in the time of 0. 7 minute, exclusive of compilation.

```
THE INPUT ARGUMENT IS ALL + X/A + SUCH + THAT + X/A + BELONGS +
TO + THE + COUNTRY + CLUB + IS + WEALTHIER + THAN + ALL + X/B +
SUCH + THAT + X/B + BELONGS + TO + THE + ELKS + LODGE + . + SOME +
X/C + SUCH + THAT + X/C + BELONGS + TO + THE + COUNTRY + CLUB +
IS + NOT + WEALTHIER + THAN + SOME + X/D + SUCH + THAT + X/D +
BELONGS + NOT + TO + THE + COUNTRY + CLUB + . + THEREFORE + SOME
+ X/E + SUCH + THAT + X/E + BELONGS + NOT + TO + THE + COUNTRY +
CLUB + IS + AN + X/E + SUCH + THAT + X/E + BELONGS + NOT + TO + THE
+ ELKS + LODGE + .
```

The sample argument consists of three sentences, the first two of which are premises and the third of which is the conclusion, since its first word is 'therefore'. The program proceeds to parse each sentence individually, in accordance with the grammar

This work was supported in part by the National Science Foundation (Grant G-24047).

described in Quarterly Progress Report No. 69 (pages 165-168). In the course of pars each word or sequence of words constituting what logicians call a "predicate," either simple or relational, is labelled with 'P' and given a numerical subscript. In our example, 'belongs to the Country Club' is labelled with 'P/.37', 'belongs to the Elks Lodge with 'P/.38', and 'is wealthier than' with 'P/.225'. For the negative predicates, like 'belongs not to the Country Club', the subscript '/NOT' is added to the appropriate 'P'

The structure of each parsed sentence corresponds to an equivalent structure in format L. The program contains a list of these equivalences and uses them to translate each sentence into format L. For all but the most simple sentences, the translation into format L involves several applications of these structural equivalences, since the first application usually results in a formula containing parts that are not yet in format L. These parts are then translated in turn into format L, and the results of these translations may themselves contain parts that need further translation. This translation "loop" is repeatedly executed until each sentence is entirely in format L. Any resulting formula that is not already in prenex normal form, with all the quantifiers on the left, is run through a subroutine that puts it into this form.

The three prenex formulae, corresponding to the three sentences of the argument, were printed out by the machine as follows:

```
(1)  THE PRENEX NORMAL FORM IS Q/A, ALL + Q/B, ALL +*( +*( +*( +
     P/.37, A + *) + AND/C + *( + P/.38, B +*) +*) + IMPLIES/C + *( + P/.225
     + *( + X/A + , + X/B + *) +*) +*) +

(2)  THE PRENEX NORMAL FORM IS Q/SOME, C + Q/SOME, D +*( +*( +
     *( + P/.37, C + *) + AND/C + *( + P/.37, NOT, D +*) +*) + AND/C + *( +
     P/.225, NOT + *( + X/C + , + X/D + *) +*) +*) +

(3)  THE PRENEX NORMAL FORM IS Q/SOME, E +*( +*( + P/. 37, NOT, E
     + *) + AND/C + *( + P/.38, NOT, E +*) +*) +
```

These formulae are then combined into a single formula, of implicational form, in which the conjunction of the premises is taken to imply the conclusion. The single formula is then put into prenex normal form and undergoes some added changes in format. All of the information contained in the subscripts, numerical or otherwise, is incorporated into the symbols themselves and the subscripts are eliminated. The principal reason for these changes in format is that the subsequent proof-procedure program, which was actually written before the translation program, was written without using any subscripts. This conversion of format has been greatly facilitated by the recent addition to the COMIT system of a provision for elevating any subscript on a symbol into a symbol in its own right (it may also work the other way around — a symbol may be turned into a subscript). The resulting formula, representing the input argument, was printed out by the machine as follows:

```
ENTERING PROOF PROCEDURE THE FORMULA IS (EA)(EB)AC)(AD)(EE)(((((P37A)
AND(P38B))IMPLIES(P225AB))AND(((P37C)AND(NOT(P37D)))AND(NOT(P225CD))))
IMPLIES((NOT(P37E))AND(NOT(P38E))))
```

The proof-procedure program, based on the Davis-Putnam algorithm, operates by reductio ad absurdum, that is, by negating the formula and attempting to derive a contradiction. For the sample argument, the negated formula consists of a sequence of quantifiers,

'(AA)(AB)(EC)(ED)(AE)',

followed by a matrix in conjunctive normal form,

'P37C AND NP37D AND NP225CD AND NP37A NP38B P225AB AND P37E P38E AND'.

The existentially quantified variables, 'C' and 'D', are replaced in the matrix by 'PAB' and 'QAB', respectively, which are distinct functions of 'A' and 'B', the universally quantified variables that precede 'C' and 'D' in the sequence of quantifiers. This gives the matrix

'P37PAB AND NP37QAB AND NP225PABQAB AND NP37A NP38B P225AB AND P37E P38E AND'.

In evaluating the sample argument, the program generated a sequence of 3^3 =27 "quantifier-free lines" on the basis of this matrix, by substituting the terms 'A', 'PAA', and 'QAA' for the variables 'A', 'B', and 'E' in all possible combinations. These 27 lines were found to contain a contradiction; thus the original formula is valid.

In the immediate future, it is hoped that the program described can be improved in some or all of the following ways:

(1) By mechanizing the translation from ordinary language into format Q, or at least from a restricted ordinary language into format Q.

(2) By expanding the list of quantifier-words, at present restricted to 'all', 'some', 'no', 'only', and 'the', so as to allow for numerical propositions. The program already permits 'at most n', where 'n' is a whole number equal to or less than 20, to be used as a quantifying expression. We next plan to program 'at least n' and 'exactly n', the latter of which will be treated as the conjunction of 'at most n' and 'at least n'.

(3) By expanding the grammar so as to admit a greater variety of sentence-types, at present restricted to sentences in which two "NPs" (noun phrases) are connected by a form of the verb 'to be', or by a "binary relational predicate," such as 'is wealthier than'. It eventually will be desired to handle relational predicates of greater degree, such as the ternary predicate illustrated by the construction 'A gives B to C'.

J. L. Darlington

References

1. M. Davis and H. Putnam, A computing procedure for quantification theory, J. Assoc. Computing Machinery _7_, 201-215(1960).

2. I. M. Copi, _Symbolic Logic_ (Macmillan and Company, New York, 1958), p. 140, no. 2.