

XXII. MECHANICAL TRANSLATION*

V. H. Yngve
D. G. Bobrow
Carol M. Bosche
Elinor K. Charney
J. L. Darlington
J. M. Dolan

R. S. Fabry
Ursula H. Funke
T. R. Hofmann
L. Jenkins
Muriel S. Kannel

E. S. Klima
Elizabeth K. Landers
W. A. Lea
F. J. H. Liu
D. M. Perlmutter
B. K. T'sou

RESEARCH OBJECTIVES

The primary objective of our research program is to find out how languages can be meaningfully manipulated and translated by machine, and to evaluate the quality that can be achieved with different approaches, the usefulness of the results and their costs. A further objective is to achieve a basic understanding of human communication and language use, and to add to the general knowledge of noncomputational uses of digital computing machinery.

We have always stressed a basic, long-range approach to the problems of mechanical translation. We are placing emphasis on completeness, when completeness is possible, and on finding out how to do a complete job if one is not now possible. This emphasis has led us into the study of many of the fundamental questions of language and translation. We are not looking for short-cut methods that might yield partially adequate translations at an early date — an important goal that is being pursued by other groups. We are seeking definitive solutions that will be permanent advances in the field, rather than ad hoc or temporary solutions that may eventually have to be discarded because they are not compatible with improved systems.

A broad and basic program of research is being carried out. In linguistic theory, a computer model of linguistic behavior is being studied which has already provided insight into the reasons why human languages are complex. Work has been progressing on linguistic descriptions for English, German, French, Arabic, Finnish, and Chinese. An experimental Arabic to English translation program has been completed. In the area of semantics, work is proceeding along several avenues in an attempt to program a machine to understand English. Language is also being studied from a logical point of view to clarify the semantic significance of certain difficult and crucial words. The nature of the translation relation is being given special emphasis. Computer programming languages are being studied.

V. H. Yngve

A. COMIT

The group has developed and is improving COMIT, a convenient large-scale computer programming system that greatly reduces the effort required to write programs related to the research. The most recent corrections to the original COMIT system were distributed in March 1963. At that time, a number of substantial improvements to the original COMIT had been suggested, and a new version, called COMIT II, had already been started. These improvements were in the areas of adding new language features, allowing FAP-coded machine-language subroutines to be used, increasing the speed of the system, and making it quick and easy to place COMIT in various monitor

*This work is supported in part by the National Science Foundation (Grant GN-244).

(XXII. MECHANICAL TRANSLATION)

systems. Since then, most of our effort on COMIT has been directed at producing COMIT II. We have allowed some use of the COMIT II system as it has developed. An independent system version of COMIT II with many improvements, except in the area of system compatibility, has been used experimentally since November 1963. A version with a good deal of system compatibility, but with fewer new language features was placed in the M. I. T. and MAC time-sharing systems in May 1964 and is ready for use in the M. I. T. FMS system now. Considerable effort has been spent in documenting for the user the new language features and the use of machine-language subroutines. We are well past the half-way mark for COMIT II.

Carol M. Bosche, R. S. Fabry, Muriel S. Kannel, V. H. Yngve

B. SOME RECENT DEVELOPMENTS IN LOGIC TRANSLATION

Our research into logic translation has resulted in some significant improvements in the COMIT logic translation programs that were described in previous issues of the quarterly report (Quarterly Progress Report No. 68, pp. 174-176; No. 69, pp. 165-168; and No. 70, pp. 265-268). Instead of the quasi-logical "format Q," which had to be used formerly when submitting arguments in functional logic, the program now accepts arguments formulated in a restricted form of ordinary English; and instead of several previously independent programs, there is now a single translation-and-evaluation system. Other new features include the program's ability to translate an input argument in four progressively refined ways (of which the first pertains to propositional logic and the last three to first-order functional logic), and the addition of a "level-of-analysis" routine that decides which of the four translations is likeliest to yield the simplest proof.

The programming of the new features has been facilitated by the time-sharing version of COMIT, which became available in May 1964, and has enabled us to incorporate the translation-and-evaluation program into the M. I. T. compatible time-sharing system, "CTSS." Use of CTSS has resulted in substantial reductions of the time needed to program and check out new routines, as well as in the introduction of considerable programming flexibility, because of greater operator control over the course of a running program. Two examples of the sort of flexibility that CTSS permits may be mentioned: (i) words that are not found in the program's dictionary may be typed directly into the workspace from the console complete with subscripts without the necessity of recompilation, and (ii) the operator may interrupt the logical evaluation whenever he thinks the level-of-analysis routine may have made the wrong decision and suggest another level-of-analysis for the program to try.

The input data that the program accepts consist of arguments, an argument being defined as a string of permissible sentences, one of which has 'therefore' as its first word. The sentence or sentences to the left of 'therefore' are taken as the premisses

of the argument, and those to the right are taken as the conclusion. It is frequently necessary to make verbal changes in the input sentences, so that they will conform to the grammatical restrictions imposed by the program. If any input sentence fails to parse, it causes an interruption or "break" in the program, which requires the operator to decide whether to continue, restart, or terminate the run. Our program employs a useful routine, "VREPLY," designed by Victor H. Yngve to increase the number of options when in break status. Using VREPLY, the operator may command a "dump," and thereby determine which sentence is causing the trouble. He may then further edit the refractory sentence and restart the program.

The grammar may be briefly described as follows. It permits sentences consisting of a subject and predicate, of which the predicate consists of a main verb, which may be followed by a predicate noun phrase or adjectival clause. The main verb may be a form of 'to be', or any transitive or intransitive verb. A subject or predicate noun phrase consists either of a single proper name, or of a principal noun preceded by one of five determiners, i.e., 'all', 'some', 'no', 'only', and 'the'. Also, the principal noun may be modified in various ways. It may be preceded by any number of adjectives, which may or may not be separated by 'and', and it may be followed by any number of relative clauses or prepositional phrases, all of which are assumed to modify the principal noun as long as they are separated by 'and'. Any relative clause or prepositional phrase not directly preceded by 'and' is assumed to modify a directly preceding noun. These relative clauses and prepositional phrases themselves ordinarily contain noun phrases, which in turn may contain further relative clauses or prepositional phrases, etc. There is thus no limit on the amount of qualification that may be appended to a noun.

The sentences permitted by the grammar may in turn be joined by truth-functional connectives, such as 'and', 'or', 'if', 'then', 'therefore', etc., to form compound sentences. In some cases, one or more sentences may be incomplete. If they are incomplete in certain predetermined ways, the parsing program is able to piece them out and make complete sentences out of them. For example, if the determiner of a noun phrase is lacking, the program provides a 'some'. If the principal noun is lacking, as in 'all who draw circles,' the program provides a 'one'. If the main noun phrase or verb phrase of a sentence is omitted, the program will borrow the main noun phrase from the nearest preceding complete sentence, or borrow the main verb phrase from the nearest following complete sentence. The main noun phrase may also be replaced by a personal pronoun, in which case the pronoun will be dealt with in one of three ways. (i) First-person pronouns are treated as proper names and are not eliminated. (ii) If the sentence is like 'If any one goes up the hill then he goes down the hill,' it is transformed into a sentence like 'All (one) who go up the hill go down the hill.' (iii) In any other sentences containing a third-person pronoun in place of the main noun phrase, the sentence is handled as if it had no main noun phrase, and the main noun phrase is borrowed from the

(XXII. MECHANICAL TRANSLATION)

nearest preceding complete sentence.

After the program has been started and an input argument as defined above has been typed in, the procedure may be described as follows. There is an initial dictionary look-up, followed by some preliminary transformations, such as 'do not + VERB= VERB+not.' Then the sentences are parsed, according to the grammar. The parsing routine employs a sort of predictive analysis, in the sense that, at any given stage in the parsing, the program searches for only those structures that could conceivably follow. The parsing routine also simplifies the sentence, by reducing all nouns to singular form, all verbs to the present infinitive, and some determiners to others, such as 'every' and 'each' to 'all'. It also pieces out the sentences, in the ways described above. A preliminary translation is then made into propositional logic, wherein each sentential clause is replaced by a single symbol, A/V, B/V, etc. Identical clauses are symbolized by the same letter, and clauses that differ only in that the main verb of one is the negation of the main verb of the other are denoted by contradictory symbols, e.g., A/V and A/V, NOT. At this stage, the decision whether to go further with the propositional logic analysis is made, the criterion being that, if one or more propositional letters are repeated in the premisses, then the propositional logic routine is carried out. A complete translation is made into a single propositional logic formula of implicational form,¹ which is tested for validity. If the answer "valid" is obtained, then the program is finished. If invalid, it proceeds to the more refined Analyses II, III and IV (Analysis I is the propositional logic routine).

Common to the three higher levels-of-analysis are the assumptions that all adjectives and most nouns (with an exception noted below) are unary predicates, and all proper names are individual terms. Every predicate is indicated by a 'P', and every individual term is indicated by an 'IND', distinctions between them being made by the use of different numerical subscripts. Within the P's, the additional distinction is made between P's with numerical subscripts of less than 500, which denote unary predicates, and P's with numerical subscripts of 500 or greater, which denote binary predicates or relations. If a P is negative, it is provided with the literal subscript 'NOT'. Where the choice of Analysis II, III or IV does make a difference is in the analysis of relative clauses and prepositional phrases. In II, every such clause or phrase is treated as a unary predicate and is denoted by a P/.n, where n is less than 500. In III and IV, prepositional phrases and relative clauses are analyzed relationally, the main preposition of the prepositional phrase and the main verb of the relative clause being denoted by terms P/.n, where n is 500 or greater. In the case of a prepositional phrase, the noun modifying the prepositional phrase is temporarily conjoined with the main preposition to form a single term, which is looked up in a special dictionary to see whether it can be relational. If so, as is the case for example with 'memberof' and 'friendof', the term thus fabricated, rather than the preposition itself, is taken as a binary predicate. The only

difference between III and IV pertains to the analysis of noun phrases of which the determiner is 'the', i.e., definite descriptions. In III, the entire noun phrase commencing with 'the' is taken to be a proper name, while in IV these P's are analyzed as asserting the unique existence of the object denoted by the predicate.

After a proof attempt in Analysis I fails, the program changes its strategy. Instead of straightway trying for a proof in Analysis II, and possibly failing, it tries to decide which of the three analyses, II-IV, is most likely to yield a proof. This is not so important in Analysis I, since quick and efficient decision procedures are available in propositional logic, but the functional logic part of our program, which is essentially a straightforward Herbrand expansion, is a proof procedure rather than a decision procedure, which means that it will find a proof if the argument is valid but it will not reject the argument if it is invalid. (The proof procedure program generates quantifier-free lines from the negated formula and attempts to prove them inconsistent, using a routine that is an amalgam consisting of one rule of the Davis-Putnam propositional logic algorithm and one rule of the Quine propositional logic algorithm.) The level-of-analysis routine, which attempts to decide which analysis is most promising, is based on the search for repetition between premisses and conclusion. It makes a list of the "terms" in the conclusion, where a "term" is either a propositional symbol, an individual name, a unary predicate, or a binary predicate. It then looks to see how many of the terms in the conclusion are repeated in the premisses. The repetition of just one term in Analysis I is sufficient to justify having a go at a proof in propositional logic. If no proof is available in propositional logic, the program will quickly find this out. In Analyses II-IV, however, it looks for the simplest analysis in which all terms in the conclusion are repeated in the premisses. If none of the four analyses are such that all of the terms in the conclusion are repeated in the premisses, then the highest level-of-analysis, i.e., Analysis IV, is selected. The level-of-analysis routine thus makes a guess as to which of the four analyses looks most promising. This guess, however, is not a final decision, since the operator may interrupt the logical computation at any time if he realizes that a more refined analysis is essential, or that a less refined analysis would yield a shorter proof. The logical formulae resulting from the different translations are written out into an output channel, and the operator may specify which of them he wishes the program to try simply by typing in the number of the desired formula. Or he may decide that the argument is probably invalid, and that it would be futile to spend any more time trying to prove it valid.

The rationale behind the procedure of starting the logical computation with the formula containing the greatest amount of repetition between premisses and conclusion is that repetition or recurrence of terms is intimately related to the validity of arguments. In fact, the most elementary example of a valid argument is "p therefore p," in which the conclusion merely repeats the single premiss, and the most elementary example of an

(XXII. MECHANICAL TRANSLATION)

invalid argument is "p therefore q," in which the conclusion is nowhere repeated in the premiss. Obviously, there are invalid arguments that do contain repetition of terms, e.g., "p therefore not-p," and valid arguments that contain no repetition of terms, though the latter all tend to be trivial arguments with contradictory premisses or tautological conclusions. These considerations, however, do not invalidate our procedure of starting the logical computation with the argument containing the most repetition, since all that we claim is that repetition of terms is a necessary, not a sufficient, condition of validity of a nontrivial argument, and that an argument has a greater chance of being valid the more repetition of terms it contains. Our procedure, furthermore, does seem to work in a number of cases, and if it does not work in a particular case, the operator can always interrupt and specify a different level-of-analysis.

We now present and discuss six valid arguments that have been recently submitted to the program. The examples were all taken from logic textbooks by I. M. Copi,^{2,3} though all but one required a certain amount of pre-editing. Four of them have been successfully translated and proved; the other two, though successfully translated, used up all of the 13,170 registers of free storage that the program makes available for logical computation, thereby causing the runs to terminate before the arguments could be proved. The four arguments that the program was able to prove required only a few hundred registers apiece (maximum 795). The two arguments that exhausted the free storage could probably be proved with the aid of additional storage that could be obtained by writing out parts of the expanded formula into one or more of the output channels that CTSS makes available, and reading the material back in as needed, though whether the arguments could then be proved in a reasonable time would remain to be seen.

- (1) If I buy a new car this spring or have my old car fixed, then I'll get up to Canada this summer and stop off in Duluth. I'll visit my parents if I stop off in Duluth. If I visit my parents they'll insist upon my spending the summer with them. If they insist upon my spending the summer with them I'll be there till autumn. But if I stay there till autumn, then I won't get to Canada after all! So I won't have my old car fixed.⁴

The pre-edited version of this argument, with a few deletions, is:

- (1') If I buy a new car or fix my old car then I'll get to Canada and stop in Duluth. If I stop in Duluth then I'll visit my parents. If I visit my parents then I'll stay in Duluth but if I stay in Duluth then I'll not get to Canada. Therefore I'll not fix my old car.

The conclusion, "I'll not fix my old car," is the negation of part of the premisses; this was enough to justify trying a proof in Analysis I, i.e., propositional logic, and a proof was in fact obtained. The formula is:

((((A)OR(B))IMPLIES((C)AND(D)))AND((D)IMPLIES(F)))AND(((F)IMPLIES(H))AND((H)IMPLIES(NOT(C))))IMPLIES(NOT(B)) .

(2) Whoever supports Ickes will vote for Jones. Anderson will vote for no one but a friend of Harris. No friend of Kelly has Jones for a friend. Therefore, if Harris is a friend of Kelly, Anderson will not support Ickes.⁵

The pre-edited version of this argument is:

(2') All who support Ickes will vote for Jones. Everyone whom Anderson will vote for is a friend of Harris. Jones is a friend of no one who is a friend of Kelly. Harris is a friend of Kelly. Therefore Anderson will not support Ickes.

Analysis I yielded:

A. B. C. D. Therefore E.,

which contains no repetition of terms between premisses and conclusion, and was therefore abandoned. Analysis II yielded:

(AA)((P/. 1, A) IMPLIES (P/. 2, A)) .
(AD)((P/. 3, D) IMPLIES (P/. 4, D)) .
(P/. 5 IND/. 0) . (P/. 6 IND/. 2) . Therefore (NOT (P/. 1 IND/. 3)) .

in which there are two terms in the conclusion, i.e., P/. 1 and IND/. 3. Of these, P/. 1 is repeated in the premisses, but IND/. 3 is not, so Analysis II was abandoned. Analysis III yielded:

(AB)((P/. 501 X/B IND/. 1) IMPLIES (P/. 500 X/B IND/. 0)) .
(AE)((P/. 500 IND/. 3 X/E) IMPLIES (P/. 502 X/E IND/. 2)) .
(AH)((P/. 502 X/H IND/. 4) IMPLIES (NOT(P/. 502 IND/. 0 X/H))) .
(P/. 502 IND/. 2 IND/. 4) . Therefore (NOT(P/. 501 IND/. 3 IND/. 1)) .

There are three terms in the conclusion, i.e., IND/. 1, IND/. 3, and P/. 501, all of which are repeated in the premisses. Analysis III was therefore tried, and resulted in a proof. The terms in the arguments above denote the following properties and individuals:

IND/. 0 = JONES; IND/. 1 = ICKES; IND/. 2 = HARRIS; IND/. 3 = ANDERSON;
IND/. 4 = KELLY; P/. 1 = SUPPORTICKES; P/. 2 = VOTEFORJONES; P/. 3 = ANDERSONVOTEFOR;
P/. 4 = FRIENDOFHARRIS; P/. 5 = FRIENDOFNOONEWHO-FRIENDOFKELLY;
P/. 6 = FRIENDOFKELLY; P/. 500 = VOTEFOR; P/. 501 = SUPPORT; P/. 502 = FRIENDOF .

After deciding on Analysis III, the program transliterated the argument in order to simplify the terms, transformed it into a single formula of conditional form, and put it into prenex normal form; this procedure yielded the formula:

(EX)(EY)(EZ)((((P2XC)IMPLIES(P1XB))AND((P1EY)IMPLIES(P3YD)))AND((P3ZF)IMPLIES(NOT(P3BZ))))AND(P3DF))IMPLIES(NOT(P2EC))).

(XXII. MECHANICAL TRANSLATION)

(3) All circles are figures . Therefore all who draw circles draw figures.⁶

No pre-editing was necessary for this example.

Analysis I yielded:

A. Therefore B.,

which contains no repetition of terms between premiss and conclusion, and was therefore abandoned.

Analysis II yielded:

(AA)((P/. 1, A) IMPLIES (P/. 2, A)) . Therefore (AD)((P/. 3, D) IMPLIES (P/. 4, D)) .

This also contains no repetition, and was likewise abandoned.

Analysis III yielded:

(AB)((P/. 1, B) IMPLIES (P/. 2, B)) . Therefore
(AE)(EF)((EG)((P/. 1, G)AND(P/. 500 X/E X/G)) IMPLIES ((P/. 2, F) AND
(P/. 500 X/E X/F))) .

The conclusion contains three terms, i.e., P/. 1, P/. 2, and P/. 500, the last of which does not occur in the premisses. Analysis IV yielded a result differing from Analysis III only in terminology. Analysis IV was tried, and resulted in a proof. The terms in these arguments denote the following properties and individuals:

P/. 1 = CIRCLE; P/. 2 = FIGURE; P/. 3 = DRAWSOMECIRCLE;
P/. 4 = DRAWSOMEFIGURE; P/. 500 = DRAW .

The transliterated prenex formula corresponding to the last formulation is:

(EX)(AY)(EZ)(AB)((P1X)IMPLIES(P2X))IMPLIES(((P1B)AND(P3YB))IMPLIES
((P2Z)AND(P3YZ))) .

(4) The architect who designed Tappan Hall designs only office buildings.
Therefore Tappan Hall is an office building.⁷

The pre-edited version of this argument is:

(4') The architect who designed Tappan-Hall designs only office-buildings.
Therefore Tappan-Hall is an office-building.

Analysis I yielded:

A. Therefore B.

This contains no repetition of terms, and was abandoned.

Analysis II yielded:

(P/. 1 IND/. 0) . Therefore (P/. 2 IND/. 1) .

This was abandoned for the same reason.

Analysis III yielded:

(AB)((P/. 500 IND/. 0 X/B) IMPLIES (P/. 2, B)) . Therefore (P/. 2 IND/. 1) .

The conclusion contains two terms, i.e., IND/. 1 and P/. 2, of which only P/. 2 recurs in the premisses. Analysis III was therefore abandoned.

Analysis IV yielded:

(EC)(AE)(AD)((P/. 3, C) AND (P/. 500 X/C IND/. 1)) AND ((P/. 3, E) AND (P/. 500 X/E IND/. 1)) IMPLIES (= X/C X/E)) AND ((P/. 500 X/C X/D) IMPLIES (P/. 2, D)) . Therefore (P/. 2 IND/. 1) .

Analysis IV was tried, and yielded a proof. The terms in these arguments denote the following properties and individuals:

IND/. 0 = THEARCHITECTWHODESIGN TAPPANHALL; IND/. 1 = TAPPANHALL;
P/. 1 = DESIGN ONLY OFFICE BUILDING; P/. 2 = OFFICE BUILDING; P/. 3 =
ARCHITECT; P/. 500 = DESIGN .

The transliterated prenex formula corresponding to the last formulation is:

(AX)(EY)(EZ)((P2X) AND (P3XB)) AND ((P2Y) AND (P3YB)) IMPLIES (=XY)) AND
(P3XZ) IMPLIES (P1Z)) IMPLIES (P1B)) .

- (5) There is a professor who is liked by every student who likes any professor at all. Every student likes some professor or other. Therefore there is a professor who is liked by all students.⁸

The pre-edited version of this argument is:

- (5') There is a professor whom every student who likes some professor likes. Every student likes some professor. Therefore there is a professor whom all students like.

Analysis I yielded:

A. B. Therefore C.

This was abandoned because of lack of repetition between terms.

Analysis II yielded:

(EA)((P/. 1, A) AND (P/. 2, A)) . (AH)((P/. 3, H) IMPLIES (P/. 4, H)) .
Therefore (EM)((P/. 1, M) AND (P/. 5, M)) .

The conclusion contains two terms, i.e., P/. 1 and P/. 5, of which P/. 5 is not repeated in the premisses, thereby causing Analysis II to be abandoned.

(XXII. MECHANICAL TRANSLATION)

Analysis III yields:

(EB)((P/. 1, B) AND (AC)((P/. 3, C) AND (ED)((P/. 1, D) AND (P/. 500 X/C X/D)))
IMPLIES (P/. 500 X/C X/B))) . (AI)(EJ)((P/. 3, I) IMPLIES ((P/. 1, J) AND
(P/. 500 X/I X/J))) . Therefore (EN)((P/. 1, N) AND (AO)((P/. 3, O) IMPLIES
(P/. 500 X/O X/N))) .

All three of the terms in the conclusion recur in the premisses, so Analysis III was chosen. The argument as formalized by Analysis III can be proved by hand computation, but it was not proved by our program because it exhausted the free storage. This is perhaps not surprising, since the argument thus formalized contains seven quantifiers. (5) is a very good example of an argument that could be easily solved with the proper heuristics. In fact, the second premiss, i.e., "Every student likes some professor," entails that the clause, "every student who likes some professor," in the first premiss may be reduced simply to "every student," and the first premiss may therefore be reduced to "There is a professor whom every student likes." The conclusion, "There is a professor whom all students like," is essentially no more than a restatement of the first premiss thus reduced, and the argument becomes trivial.

The terms in the arguments above denote the following properties and individuals:

P/. 1 = PROFESSOR; P/. 2 = ALLSTUDENTWHOLIKESOMEPROFESSORLIKE;
P/. 3 = STUDENT; P/. 4 = LIKESOMEPROFESSOR; P/. 5 = ALLSTUDENTLIKE;
P/. 500 = LIKE .

The transliterated prenex formula corresponding to the last formulation is:

(AX)(EY)(EZ)(EB)(AC)(ED)(AE)((((P1X)AND(((P2Y)AND((P1Z)AND(P3YZ)))IMPLIES
(P3YX)))AND((P2B)IMPLIES((P1C)AND(P3BC))))IMPLIES((P1D)AND((P2E)IMPLIES
(P3ED)))) .

- (6) Everything on my desk is a masterpiece. Anyone who writes a masterpiece is a genius. Someone very obscure wrote some of the novels on my desk. Therefore some very obscure person is a genius.⁹

The pre-edited version of this argument is:

- (6') Everything on my desk is a masterpiece. Everyone who writes a masterpiece is a genius. Some obscure one wrote some of the novels on my desk. Therefore some obscure one is a genius.

Analysis I yielded:

A. B. C. Therefore D.

This contains no repetition of terms, and was abandoned.

Analysis II yielded:

(AA)((P/. 1, A) IMPLIES (P/. 2, A)) . (AF)((P/. 5, F) IMPLIES (P/. 6, F)) .
 (EK)((P/. 7, K) AND (P/. 8, K)) . Therefore (ER)((P/. 7, R) AND (P/. 6, R)) .

in which the two terms in the conclusion, i.e., P/. 6 and P/. 7, are both repeated in the premisses. The argument, however, could not be proved in Analysis II, and Analysis III was therefore tried.

Analysis III yielded:

(AB)((EC)((P/. 3, C) AND (P/. 4, C)) AND (P/. 500 X/B X/C)) IMPLIES
 (P/. 2, B)) . (AG)((EH)((P/. 2, H) AND (P/. 501 X/G X/H)) IMPLIES (P/. 6, G)) .
 (EL)(EM)((P/. 7, L) AND ((P/. 9, M) AND (EN)((P/. 3, N) AND (P/. 4, N)) AND
 (P/. 500 X/M X/N))) AND (P/. 501 X/L X/M))) . Therefore (ES)((P/. 7, S)
 AND (P/. 6, S)) .

This formulation, like that of (5), though sufficient to prove the argument valid, caused the free storage to be exhausted and stopped the program short of a proof.

The terms in these arguments denote the following properties and individuals:

P/. 1 = ONSOMEMYDESK; P/. 2 = MASTERPIECE; P/. 3 = MY; P/. 4 = DESK;
 P/. 5 = WRITESOMEMASTERPIECE; P/. 6 = GENIUS; P/. 7 = OBSCURE;
 P/. 8 = WRITESOMENOVELONSOMEMYDESK; P/. 9 = NOVEL; P/. 500 = ON;
 P/. 501 = WRITE .

The transliterated prenex formula corresponding to the last formulation is:

(EX)(EY)(EZ)(EB)(AC)(AD)(AE)(EF)((((P2Y)AND(P3Y))AND(P7XY))IMPLIES(P1X))
 AND(((P1B)AND(P8ZB))IMPLIES(P4Z)))AND((P5C)AND(((P6D)AND(((P2E)AND(P3E))
 AND(P7DE)))AND(P8CD))))IMPLIES((P5F)AND(P4F)) .

Running times for the six examples above range from 7 seconds for (1) to 26 seconds for (6), for the translation alone. Times for logical computation, for those arguments proved valid, range from 5 seconds for (1) to 17 seconds for (3).

J. L. Darlington

References

1. The complete translation into a single formula of propositional logic is made according to the rules contained in the appendix of the memo "Translating ordinary language into symbolic logic," which is a joint memorandum of the Mechanical Translation Group of the Research Laboratory of Electronics and Project MAC, dated March 31, 1964 (available from either group). This memorandum gives a more detailed description than does the present report of the stages in the logical translation of a given sentence in ordinary English.
2. I. M. Copi, Introduction to Logic (Macmillan Company, New York, 1953).
3. I. M. Copi, Symbolic Logic (Macmillan Company, New York, 1954).
4. I. M. Copi, Introduction to Logic, p. 275, example 4.

(XXII. MECHANICAL TRANSLATION)

5. I. M. Copi, Symbolic Logic, p. 140, example 1.
6. Ibid., p. 140, example 3.
7. Ibid., p. 158, example 1.
8. Ibid., p. 140, example 4.
9. Ibid., p. 140, example 8.