Research Summaries

PHILIP BAGLEY, MANDALAY GREMS, Editors

Trajectory Computation in a Problem-Oriented Language

Appl. Math. Sec., The Boeing Co., Seattle, Wash.
Reported by: Clarice MacDonald (Jan. 1962)
Descriptors: **problem oriented language, simulation, FORTRAN, computer application, problem definition**

A new problem-oriented language, *XYZ*, is being used on an experimental basis for trajectory computation. In the experiment, *XYZ* is compared with FORTRAN to determine the relative advantages of the two systems for use in flight simulation work. The comparison is concerned with the steps of: problem definition, program design, coding, checkout, and production.

The problem selected for the experiment is the computation of the boost and free flight trajectory of a ballistic missile. The trajectory is two-dimensional treating the vehicle as a point mass traveling over a nonrotating spherical earth. Numerical integration is used in computing the boost path; closed form equations satisfying Kepler's laws are used to determine apogee, perigee, re-entry, and impact conditions.

The *XYZ* programs are written in a language very much like English. They are sets of glossaries and computing expressions. Each glossary contains a program dictionary with names, descriptions, synonyms, units designation, data values, scaling for printing, and outline expressions which state the organization of the problem solution. The computing expressions include algebraic expressions, text expressions of the "if . . ., then. . . , otherwise. . ." variety, table lookup and interpolation expressions, and output generating expressions.

The experiment indicates that use of the *XYZ* glossary features require precise problem definition and logical program design. Once the terms are defined and the organization fixed, coding is rapid and easy. The natural language, simple input-output, and lack of restrictions on symbols makes it possible to produce a readable, accurate program very quickly. Preliminary data on operating time for the experimental problem indicates that *XYZ* compiling time is about one-half 709/7090 FORTRAN time.

REFERENCE: Boeing Documents on *XYZ* Translator-Compiler.

Criteria for Acceptable Abstracts

Center for Research in System Dev., System Development Corp., Santa Monica, Calif.
Reported by: Harold Borko (Apr. 1962)
Descriptors: **abstracts, standards, heuristics, automatic abstracts**

Recently, a personal reaction to a published abstract started a study which may lead to criteria for abstracts. This particular abstract seemed inadequate to those who were familiar with the original journal article. This brought up the question of how the adequacy of an abstract is determined. Is it a matter of opinion, or is it a measure of excellence? Assuming it is the latter, are the means for measuring it specified?

These questions, and valid answers to them, had obvious implications for our research project. We therefore undertook the task of surveying approximately 400 scientific organizations which publish abstracts. We asked for the purpose of their abstracts—whether they were designed to serve as a substitute for the original article or to refer the reader to the original publication—and we also asked for a copy of the instructions they sent to their abstractors. We are still receiving responses, but it is of interest to note that a surprisingly large proportion of the journals have no formal set of instructions. They simply rely on "experience."

After these responses have been collected and analyzed, it is our intention to develop a set of requirements for an acceptable abstract. Abstracts for different purposes may have different requirements. Then we hope to derive a set of procedural rules, or heuristics, for writing abstracts. The validity of these rules will be checked and modified as they are used by human abstractors. Eventually we hope to program a computer to follow these rules, or a modification of them and produce abstracts automatically.

General Language Translation Schemata

Mechanical Languages Projects, The Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia, Pa.
Reported by: Peter Zilahy Ingerman* (May, 1962)
Descriptors: **mechanical languages, language translation, Backus normal forms, semiotics, syntax-oriented compilers**

In the past Backus Normal Form (BNF) has been defined only informally; however, a formal recognition definition has been evolved.

The translation technique being developed by this office is an effective recognizer for finite strings in any source language whose syntax is expressible in BNF. The translation algorithm has three inputs: the source statement (presumably in the source language); the head of the recognition tree of the source language; and a table defining the syntax of the source language in BNF.

If the input is not recognizable as being in the source language, an error indication is given; otherwise, the output of the recgnizer is a string of symbols representing the rules which were applied and their scopes, and is the recognition tree in prefix notation (although generated in reverse order).

In addition, this office is developing a processor which will accept the prefix tree from the preceding processor and a table describing the semantics and pragmatics of a specific target machine and will produce assembly language coding. These two routines (together with a small number of ancillary routines) comprise an effective translation scheme for a general class of languages of which Algol is a representant.

REFERENCES:

INGERMAN, PETER ZILAHY. A translation technique for languages whose syntax is expressible in extended Backus normal form. To be published as Technical Documentary Report under No. AFOSR2096.

FREEDMAN, HARRY A. A special-purpose assembly system for an algebraic language translator. In preparation.

* Research on Mechanical Languages is being carried out with joint support from the U.S. Air Force Office of Scientific Research (Contract AF-49(638)-951) and the National Science Foundation (Contract NSF G-14096). Principal Investigator is Dr. Saul Gorn.