June 1969 Brief B69-10187

NASA TECH BRIEF

JFLIP-JPL FORTRAN Language with Interval Pre-Processor

JFLIP and TMG are a FORTRAN pre-processor and a Syntax-Directed-Compiler used to describe the language in which the former is written. They were written for those who write in FORTRAN IV and who desire greater language flexibility and power. It is also useful for those who require assistance in converting FORTRAN IV decks to run on the UNIVAC 1108.

The problem:

A method was needed for those who write in FORTRAN IV and who desire greater language flexibility and power.

The solution:

The JFLIP and TMG system is a Fortran pre-processor and a Syntax-Directed-Compiler to describe the language in which the former is written. JFLIP relaxes many of the restrictions common in most FORTRAN Compilers.

How it's done:

The JFLIP system permits the input of IBM 1620 FORTRAN IID, IBM 7094 FORTRAN II, IBM 7094 FORTRAN IV, IBM 360 FORTRAN IV, and most of the coding conveniences present in CDC FORTRAN IV, SDS FORTRAN II, and SDS FORTRAN IV. The output of the processor is IBM 7094 FORTRAN IV. Since IBM 7094 FORTRAN IV is a proper subset of UNIVAC 1108 FORTRAN V, output from JFLIP will also be suitable for the UNIVAC 1108.

The primary advantages offered by JFLIP include: 1) mixed mode arithmetic, 2) generalized array subscripting, 3) input and output statements compatible with most FORTRAN systems, 4) generalized FORMAT statements, 5) many syntax relaxations, 6) generalized DATA statements, 7) generic functions, 8) extensions to the IF statement, and 9) the inclusion of interval variables as another data type.

Under normal operation of the JFLIP system, the user obtains a listing of his source program, followed by a cross reference dictionary which alphabetically lists all his source program symbols (variables, constants, key-words, statement numbers and literals). If the user requests a listing of the generated program and/or trace of the compilation process, this output will be interspersed with the source listing. If the user requests punching of the generated program, this output will be the first to be written on the system punch file. In all cases the generated program with control cards is written on SYSCK1 for subsequent input to the IBM 7094 FORTRAN IV compiler.

TRANS-MOGRIFICATION (TMG) is a compiler writing program which permits the programmer to completely define a programming language or a translator to his specifications. Description of the new language is written in TMGL (TMG language) which is described in detail in the documentation.

The TMG system is basically a syntax-directed compiler, meaning that the system has a syntax table which directs the operation. Thus, by changing the syntax table, the system can be made to operate in a different manner.

Physically, the TMG system is comprised of three separate decks. The first deck (TMG) is a collection of routines which analyzes the input and produces the output deck. The second deck (TMGIOP) is the Input-Output Package for TMG. The third deck (TMGLAN) contains the syntax rules by which analysis is performed on the program. After analysis, TMGLAN creates a new syntax table in accordance with the rules provided by the input text.

(continued overleaf)

After analysis of the source language is complete and a new syntax table is produced, the new table replaces TMGLAN and analysis is performed in accordance with the source language.

Note:

1. Inquiries should be made to:

COSMIC
Computer Center
University of Georgia
Athens, Georgia 30601
Reference: B69-10187

Patent status:

No patent action is contemplated by NASA.

Source: D.A. Germann, P.H. Knowlton, and H.L. Smith
NASA Pasadena Office
(NPO-10835)