

Peter Naur
No 137 p1

Exhibition Munich
IFIP Congress, Aug. 1962

THE ALGOL COMPILER FOR THE GIER.

In designing the Gier Algol compiler the primary considerations have been the generality of the language, speed of compilation, and thoroughness of error detection. However, in the course of the work it turned out to be possible to satisfy all reasonable requirements on the quality of the compiled program as well.

The compiler will accept all of ALGOL 60 except for the features mentioned below. In particular recursive procedures and the call-by-name feature are processed in full generality and identifiers may be of any length. The only restrictions, apart from obvious capacity limitations, are the omission of integers as labels, of own arrays and of calling arrays by value.

Input to the compiler is from 8-hole punched tape. The characters are those of the ALGOL 60 reference language except for the following adaptations:

negation	-	left string quote	{
space	⊥	right string quote	}
integer divide	:	implication	=>

Input and output from the compiled program are controlled by means of standard procedures. The more important output procedures are:

tryk Expects two or more parameters, the first of which should be a string defining the digit layout while the following should be arithmetic expressions whose values should be punched. Example:
tryk({dd.ddddd},p,s+t).

trykvr Punches one carriage-return-line-feed character.

trykml Punches the number of SPACE characters supplied as parameter. Examples:
trykml(5); trykml(p+q);.

The full input-output facilities are described in A Manual of the DASK ALGOL Language (Regnecentralen 1961).

Apart from the language specifications the most important design considerations has been the heavy burden of the two-level store of Gier and its very limited high speed store (1024 words). This storage structure has led to the design of a fully automatic, dynamic, administration of the transfer of program from drum to core. In this system the complete program is stored on the drum and only those tracks which have been referred to most recently during the run of the program will also be present in the core memory. Again the number of available program track places in the cores will be determined dynamically from the amount of space not reserved for variables. In this manner the system will always try to make full use of the core memory. The administrative system handling the dynamic allocation of program and variables permanently occupies 200 words of the cores.

All variable parts of programs are stored in a stack in the core memory. The method used for referring to variables follows the principles described by E.W.Dijkstra. Data stored on the drum cannot be handled at present. Standard procedures for performing output to and input from drum will be included in the near future, however.

The compiler works in 9 passes. In this manner it has been possible to accommodate the program of each pass, including its tables, in the core memory. Thus during each pass the only drum transfers necessary are those of the input to and the output from the pass. The total resulting compilation speed is about 2 seconds per final drum track of about 60 instructions. The function of the 9 passes is as follows:

- Pass 1: Conversion to reference language.
- Pass 2: Identifier matching. About 450 identifiers can be accommodated, depending on their lengths.
- Pass 3: Analysis and check of delimiter syntax.
- Pass 4: Collection of declarations and specifications. (A reverse scan).
- Pass 5: Distribution of declarations and specifications. Storage allocation of variables.
- Pass 6: Conversion to inverse Polish form and type checking.
- Pass 7: Allocation of working variables. Specification of machine instructions for expressions.
- Pass 8: Final addressing of internal program references. Slicing of program in drum track length. Production of final machine code. (A reverse scan).
- Pass 9: Placing of program drum tracks on their correct places on the drum.

Error detection is completely integrated with the compilation. Erroneous spots detected in early scans will be removed from the program and the compiler will then continue to detect further errors. However, since error checking is completed in pass 6 no program containing errors will be taken through passes 7, 8 and 9.

The complete compiler occupies about 5000 words on the drum. Thus in the current compile-and-go version there are about 7800 words available for the compiled program. Other versions giving more space for the final program might easily be produced, however.

The compiler, which is the second ALGOL 60 compiler to be completed at Regnecentralen, was designed and written during February-August 1962 by Jørn Jensen, Peter Naur, Peter Kraft, Henning Christensen and Paul Lindgreen, assisted during the last phases by Knut-Sivert Skog.