2.1.5.2.

FORTRAN to ALGOL TRANSLATOR.

TECHNICAL SPECIFICATION

PROVISIONAL

This description is liable to alteration

without notice.

Elliott Computing Division, Elliott Bros. (London) Ltd.
Elstree Way, Borehamwood, Herts.

ELS 2040

May 1963.

OPERATION

The FORTRAN to ALGOL translator is supplied in the form of a standard program.

After input the program will wait for a message from the operator. The first word of the message should be,

i) PROG if a whole FORTRAN program is to be translated, or

ii) SUBR if a single subroutine is to be translated.

The second word of the message indicates the input device to be used.

i) For an input on cards, place the deck in the input stacker of the card device, and type CARDS.

ii) For inout from magnetic tape, place the input spool on a transport switched to channel 1, and set to even parity high density. Writing should not be permitted. The word

MTAPE should be typed.

iii) For input from paper tape, place the leading edge in the reader switched to channel 1 and type the word PTAPE.

At the end of each job the computer will come to a data wait; if any further FORTRAN programs remain to be translated the entire operation may be repeated without reinput of the translator.

FORTRAN to ALGOL TRANSLATOR

FUNCTION

The translator accepts a program written in FORTRAN and produces an equivalent program written in ALGOL 60 which may be translated and executed by the Elliott 503 ALGOL programming system.

The translator also accepts a single SUBROUTINE or FUNCTION type subprogram written in FORTRAN, and produces an equivalent ALGOL 60 procedure, which may be subsequently incorporated in any ALGOL program for execution on the 503.

INPUT

The translator will accept FORTRAN programs in three forms.

1. IBM punched cards
2. Card images on IBM BCD magnetic tape
3. Elliott 8-channel paper tape.

When using paper tape, an exact copy should be made of the program listing; in particular, great care must be exercised in the correct tabulation of the first six columns of each line. The space character will normally be used to reproduce each blank column, but if all the first six columns are blank the tab character may be used instead.

SEQUENCE OF INPUT

If a FORTRAN program consists of several sub-programs, the FUNCTION and SUBROUTINE type subprograms must be

translated before the main program. Furthermore, if one subprogram calls another the called subprogram must be translated before the calling subprogram.

OUTPUT

After input and translation of each subprogram, an ALGOL version will be output on punch 1. At the end of translation of the main subprogram, a short tape containing certain global declarations will be produced on punch 2.

When using or printing up the ALGOL program, this second tape must be presented first.

RUNNING

During the running of an ALGOL program which has been translated from FORTRAN, the 'ignore overflow' key should be depressed.

SEPARATE COMPILATION

Any FORTRAN subprogram which does not use COMMON workspace may be independently translated into ALGOL.

The ALGOL procedures produced by the translation may be incorporated into ALGOL programs, which may themselves be the result of translation from FORTRAN.

COMPATIBILITY

FORTRAN is a Language which has no rigid definition; there are a great number of different versions implemented on different computers. All these versions are similar in broad outline, but usually there are a large number of unmentioned

and undetected incompatibilities between them. The version of FORTRAN accepted by the Elliott translator cannot therefore be exactly compatible with all previous versions, but it is designed to accept most programs written for any other version.

The following versions of FORTRAN are accepted in their entirety :

650 FORTRAN

1620 BASIC FORTRAN

1620 FORTRAN II

7090 BASIC FORTRAN

FORTRAN IV

The translator automatically detects whether FORTRAN II or FORTRAN IV is being used, and adapts itself accordingly.

The 503 does not possess any sense lights; the statements SENSE LIGHT i and IF (SENSE LIGHT i) n_1, n_2 will set markers internal to the computer; al programs containing these statements will still work, but will fail to communicate with the operator. With this reservation, the following versions of FORTRAN are accepted :

1401 FORTRAN

1410 FORTRAN

The range of real numbers on the 503 is twice as great as on certain IBM computers, and the range of integers extends to approximately 2.6×10^{11}. Consequently the overflow testing operations of a FORTRAN program may have different results on the 503.

With this reservation the following versions of FORTRAN are accepted :

7070 BASIC FORTRAN

7070 FULL FORTRAN

705 FORTRAN

7080 FORTRAN

709/90 FORTRAN II

The 503 does not possess a drum; orders referring to a drum will not be accepted. With this reservation, the following versions are accepted.

704 FORTRAN

704 FORTRAN II

ALGOL contains no facilities for double precision or complex arithmetic. With this exception, the forthcoming 7090 version of FORTRAN IV is accepted in its entirety.

In all cases where the results of arithmetic operations are not defined in FORTRAN, or defined differently on different computers, the methods used by the 7090 are followed, as far as is consistent with the greater range and accuracy of numbers held in the 503.

STORAGE ALLOCATION

The method of storage allocation used in FORTRAN differs from computer to computer; and the method used by ALGOL is different again. The difference will not be relevant in the majority of programs which make no essential use of knowledge

of the storage allocation methods of a particular implementation of FORTRAN.

Most cases where an inconsistent storage allocation is made in FORTRAN can be detected at time of translation into ALGOL, and others are detected at time of translation or running of the ALGOL program. In the former case, a warning message will be displayed, and the translator will automatically make a decision which is likely to be effective in nearly all programs which suffer from the particular defect.

The main course of difficulty will be an inconsistent "commoning" or "equivalencing" of variables. This occurs when two variables of different types are required to occupy the same locations, or when two arrays of different dimensions are required to overlap. All these cases will be detected and cause a display of message.

"Inconsistent commoning of A and B".

Where A and B are the indentifiers for the offending variables the usual remedy for this situation is to ignore the commoning, and allocate separate locations for each variable. In this case the message

"common ignored"

will be displayed. The same treatment will be applied to formal parameters placed in common storage.

Another possibility is that the variable will be consistent with another which has previously been allocated separate storage. In this case, the message

" commoned with C"

will be displayed, where C is the indentifier of the variable with which commoning has been successful.

This solution will produce valid results whenever the motive of commoning or equivalencing was merely to save space. If a more ulterior motive was involved, then recasting of the ALGOL program may be required.

In cases where FORTRAN storage allocation is poorly defined, or defined differently for different computers, the practice of 7090 FORTRAN IV is followed.

INCORRECT SCRIPTING

There are certain errors which occur in FORTRAN programs which are not detected by the FORTRAN system. The most serious of these are concerned with incorrect subscripting. A FORTRAN program may omit one of the subscripts of a two or three dimensional array; this will usually produce grossly inappropriate results, but it may occasionally be used to produce a working program. The ALGOL translator will reject this error when an attempt is made to translate the corresponding ALGOL program.

If during the running of a FORTRAN program the subscript of a variable "runs wild", and the computed address of the variable lies outside the range of storage specified in the DIMENSION statement

for that variable, the results are usually disastrous for the continued working of the program. Nevertheless, the error remains undetected, and the technique is occasionally used to produce a working program. Such programs will cease to work in ALGOL.

OTHER INCONSISTENCIES

It is generally recognised that no strict definition of the FORTRAN language has been made, and the only criterion for the "correctness" of a FORTRAN program is that it will be translated by a particular issue of a FORTRAN translator, and be executed correctly by a particular computer. No guarantee can be given that a new issue of the translator, or a change to a new machine will not cause the program to break down.

The nature of the inconsistencies liable to lead to breakdown is usually familiar to the programmer who uses them. In general, they are the result of deliberate experimentation in an attempt to "fool the system". Such programs cannot be generally interchanged even between computers which accept FORTRAN, and they cannot be transferred to the 503.

MANUAL INTERVENTION.

An inspection of the ALGOL program produced by the translator may reveal a few points at which the quality of the program could be improved.

In general there is no need to increase the efficiency of the program in terms of speed, but occasionally it may be necessary to alter the declarations in order to save space. There are two ways of doing this :

i) Alter on array declaration local to a procedure from own array to simple array.

ii) Remove one of the global array declarations into one of the procedures. This is particularly recommended to replace an inconsistent commoning in the FORTRAN program.

These alterations will be valid in cases where an array is referred to only in a single subprogram, and where the values of the array are irrelevant at the time of entry into the subprogram. The array name may also appear as a parameter in any subroutine called by the given subprogram, without affecting the validity of the alteration.