

SUPPLEMENT TO THE ALGOL 60 REPORT

(Numerische Mathematik 2, 106-136; Acta polytech. Scand., 284;  
Comm. A.C.M., 3, 299-314)

Edited by M. Woodger.

Introduction.

A meeting of some of the authors of ALGOL 60 was held on 2nd - 3rd April 1962 in Rome, Italy, through the facilities and courtesy of the International Computation Centre. The following were present:

<u>Authors</u>	<u>Advisers</u>	<u>Observer</u>
F.L. Bauer	M. Paul	W.L.v.d.Poel
J. Green	R. Franciotti	(Chairman, IFIP TC 2.1,
C. Katz	P.Z. Ingerman	Working Group ALGOL)
R. Kogon (representing J.W. Backus)		
P. Neur		
K. Samelson	G. Seegmuller	
J.H. Wegstein	R.E. Utman	
A.v. Wijngaarden		
M. Woodger	P. Landin	

The purpose of the meeting was to correct known errors in, attempt to eliminate apparent ambiguities in, and otherwise clarify the ALGOL 60 Report. Extensions to the language were not considered at the meeting. Various proposals for correction and clarification that were submitted by interested parties in response to the Questionnaire in ALGOL Bulletin No. 14 were used as a guide.

This report constitutes a supplement to the ALGOL 60 Report which should resolve a number of difficulties therein. Not all of the questions raised concerning the original report could be resolved. Rather than risk hastily drawn conclusions on a number of subtle points, which might create new ambiguities, the committee decided to report only those points which they unanimously felt could be stated in clear and unambiguous fashion.

Questions concerned with the following areas are left for further consideration by Working Group 2.1 of IFIP, in the expectation that current work on advanced programming languages will lead to better resolution:

1. Side effects of functions.
2. The call by name concept.
3. Own: static or dynamic.
4. For statement: static or dynamic.
5. Conflict between specification and declaration.

The authors of the ALGOL 60 Report present at the Rome Conference, being aware of the formation of a Working Group on ALGOL by IFIP, accepted that any collective responsibility which they might have with respect to the development, specification and refinement of the ALGOL language will from now on be transferred to that body.

### Misprints.

The following corrections refer to misprints in the ALGOL 60 Report as published in the Communications of the A.C.M. and Numerische Mathematik.

Section 3.3.4.3. (C.A.C.M. and Num.Math.)-

For '(the denominator has i factors)' -  
read '(the denominator has -i factors)'

Section 3.4.1. (C.A.C.M.)

The formula for <implication> should read  
'<implication> ::= <Boolean term> | <implication>  $\supset$  <Boolean term>'

Section 4.2 2. (C.A.C.M. and Num. Math.)

The fourth example should begin with upper case S.

Section 4.7.1. (C.A.C.M.)

The formula for <parameter delimiter> should read  
'<parameter delimiter> ::= , | <letter string> : ('

### Amendments.

Section 1, paragraph 3, last sentence.

Replace by

'A sequence of statements may be enclosed between the statement brackets begin and end to form a compound statement'.

Section 1, last two sentences of paragraph 4 and whole of paragraph 5.

Replace by

'A sequence of declarations followed by a sequence of statements and enclosed between begin and end constitutes a block. Every declaration appears in a block in this way and is valid only for that block. A program is a block or compound statement which is not contained within another statement and which makes no use of other statements not contained within it.'

Section 1, footnote 1

For 'is said to be undefined'

read 'is left undefined or said to be undefined'.

Section 2.1, footnote (Comm. A.C.M. edition)

Add

'The reference language symbol for a space is  $\square$ . For typographical reasons the symbol # is used here instead.'

Section 2.3, last paragraph, following end

Replace by

'By equivalence is here meant that any of the three structures shown in the left hand column may, in any occurrence outside of strings, be replaced by the symbol shown on the same line in the right hand column without any effect on the action of the program. It is further understood that the comment structure encountered first in the text when reading from left to right has precedence in being replaced over later structures contained in the sequence'.

Section 2.7., second paragraph.

Replace by

'The scope of a quantity is the set of statements and expressions in which the declaration of the identifier associated with that quantity is valid. For labels see section 4.1.3'.

Section 3.3.3.

For '(cf. section 5.4 Procedure declarations)'

read '(cf. section 5.4.4. Values of function designators)'

Section 3.3.4.2.

For 'defined as follows' read 'mathematically defined as follows'.

Section 3.4.1.

Replace the formula for <relation> by

'<relation> ::= <simple arithmetic expression>

<relational operator><simple arithmetic expression>'

Section 4.1.1.

Add after the formula for <block> the following

'<program> ::= <block> | <compound statement>'

Sections 4.1.1. and 4.5.1.

Replace the formulae for <unconditional statement> and <statement> respectively by the following:

'<unconditional statement> ::= <basic statement> |

<compound statement> | <block>'

'<statement> ::= <unconditional statement> |

<conditional statement> | <for statement>'

Section 4.1.3., paragraph 2, last sentence.

Replace by 'A label separated by a colon from a statement, i.e. labelling that statement, behaves as though declared in the head of the smallest embracing block, i.e. the smallest block whose brackets begin and end enclose that statement. In this context a procedure body must be considered as if it were enclosed by begin and end and treated as a block.'

Section 4.2.1.

Replace the formula for <left part> by

'<left part> ::= <variable> := | <procedure identifier> :='.

Section 4.2.3.

Replace the first sentence by

'Assignment statements serve for assigning the value of an expression to one or several variables or procedure identifiers. Assignment to a procedure identifier may only occur within the body of a procedure defining the value of a function designator (cf. section 5.4.4.).'

Section 4.2.4., first four statements.

Replace by

'The type associated with all variables and procedure identifiers of a left part list must be the same. If this type is Boolean, the expression must likewise be Boolean. If the type is real or integer, the expression must be arithmetic. If the type of the arithmetic expression differs from that associated with the variables and procedure identifiers, appropriate transfer functions are understood to be automatically invoked'.

Section 4.2.4.

Add 'The type associated with a procedure identifier is given by the declarator which appears as the first symbol of the corresponding procedure declaration (cf. section 5.4.4).'

Section 4.3.4

Add 'A go to statement may, however, lead from outside into a compound statement'.

Section 4.5.1.

Replace the formula for

<if statement> by  
'<if statement>::= <if clause><unconditional statement>'

and the formula for

<conditional statement> by  
'<conditional statement>::= <if statement> |  
                                  <if statement>else<statement> |  
                                  <if clause><for statement> |  
                                  <label>:<conditional statement>'

Section 4.7.3.

Replace the final colon by

'at the time of execution of the procedure statement:'.

Section 4.7.3.1.

Replace the last sentence by

'The effect is as though an additional block embracing the procedure body were created in which those assignments were made to variables local to this fictitious block with types as given in the corresponding specifications (cf. section 5.4.5.). As a consequence, variables called by value are to be considered as non-local to the body of the procedure, but local to the fictitious block (cf. section 5.4.3).'

Section 4.7.3.3.

Add 'If the procedure is called from a place outside the scope of any non-local quantity of the procedure body the conflicts between the identifiers inserted through this process of body replacement and the identifiers whose declarations are valid at the place of the procedure statement or function designator will be avoided through suitable systematic changes of the latter identifiers.'

Section 4.7.5.1.

Replace by 'If a string is supplied as an actual parameter in a procedure statement or function designator, whose defining procedure body is an ALGOL 60 statement (as opposed to non-ALGOL code, cf. section 4.7.8), then this string can only be used within the procedure body as an actual parameter in further procedure calls. Ultimately it can only be used by a procedure body expressed in non-ALGOL code.'

Section 4.7.5.4.

For 'a switch identifier or a procedure identifier,'  
read 'a switch identifier or a procedure identifier or a string,'

Section 4.7.6.

Delete this section.

Section 5, first sentence.

Replace by 'Declarations serve to define certain properties of the quantities used in the program, and to associate them with identifiers.'

Section 5, second sentence.

For 'declaration for' read 'declaration of'.

Section 5, paragraph 3.

For 'lose their significance again'  
read 'lose their local significance'.

Section 5.3.3., first sentence.

Replace by 'A switch declaration defines the set of values of the corresponding switch designators.'

Section 5.3.5.

Replace by

'If a switch designator occurs outside the scope of a quantity entering into a designational expression in the switch list, and an evaluation of this switch designator selects this designational expression, then the conflicts between the identifiers for the quantities in this expression and the identifiers whose declarations are valid at the place of the switch designator will be avoided through suitable systematic changes of the latter identifiers.'

Section 5.4.3.

Add 'The procedure body always acts like a block, whether it has the form of one or not. Consequently the scope of any label labelling a statement within the body or the body itself can never extend beyond the procedure body. In addition, if the identifier of a formal parameter is declared anew within the procedure body (including the case of its use as a label as in section 4.1.3), it is thereby given a local significance and actual parameters which correspond to it are inaccessible throughout the scope of this inner local quantity'.

Section 5.4.4.

Replace by 'For a procedure declaration to define the value of a function designator there must, within the procedure body, occur one or more explicit assignment statements with the procedure identifier in a left part; at least one of these must be executed, and the type associated with the procedure identifier must be declared through the appearance of a type declarator as the very first symbol of the procedure declaration. The last value so assigned is used to continue the evaluation of the expression in which the function designator occurs. Any occurrence of the procedure identifier within the body of the procedure other than in a left part in an assignment statement denotes activation of the procedure'.

Section 5.4.5. Change second sentence to read

'In this part no formal parameter may occur more than once. Specifications of formal parameters called by value (cf. section 4.7.3.1) must be supplied and specifications of formal parameters called by name (cf. section 4.7.3.2) may be omitted.'