

Report on SUBSET ALGOL 60 (IFIP)

Introduction

The present report contains the results of the work of IFIP Working Group 2.1 (WG 2.1) on establishing a subset of the algorithmic language ALGOL 60 as defined in the Revised Report on the Algorithmic Language ALGOL 60 [Numerische Mathematik 4, 420 (1963), Communications of the ACM 6, 1 (1963), and The Computer Journal 5, 349 (1963)].

The meaning of a subset of ALGOL 60

By a subset of ALGOL 60 is here meant a language such that every program written in the subset language is automatically also a program written in ALGOL 60 and has the same meaning in both languages.

The purposes of establishing a subset

The main incentive to the work is the realization that the generality of some of the features of ALGOL 60 and the disagreement concerning the exact meaning of others have proved a considerable discouragement to some of the groups who have contemplated implementing the language and have caused most of the existing implementations to be based on subsets defined locally. Clearly, if no attempt is made to avoid it, this development will lead to the use of any number of languages, all subsets of ALGOL 60, but in many cases mutually incompatible ones.

This state of affairs means a weakening of the efforts of the authors of ALGOL 60 toward establishing a common language, which may still be avoided if a subset which avoids the above-mentioned difficulties were defined and recommended by the official working group.

In lending its support to this development the IFIP/WG 2.1 is not blind to the fact that to a certain extent it may reduce the effort spent on implementing the full ALGOL 60 language, and thus may decrease the effect of this language on users and implementors. However, this negative effect is counteracted not only by the positive benefits of the subset already mentioned, but also by the fact that the impact of the full ALGOL 60 on the computing world has already been very considerable, as evidenced by the attention given to it at recent meetings and in the literature.

The development of the subset

The members of WG 2.1 wish to acknowledge the extensive and valuable work on the establishment of subsets done by other bodies. The present SUBSET ALGOL 60 is, in fact, to a large extent based on such work, as is evident from the following historical notes.

The work of WG 2.1 on defining the subset was begun during a meeting in Munich in August 1962. During this meeting a detailed comparison of two existing subsets, the ALCOR subset [ALGOL-Manual der ALCOR-Gruppe. Elek-

tronische Rechenanlagen, 3, 206 (1961)], and SMALGOL [Smalgol-61. Communications of the ACM, 4, 499 (1961)], was made, and tentative decisions on a number of the characteristics of the subset were taken. By September 1963, when the next meeting of WG 2.1 took place in Delft, a proposal for a subset prepared by the European Computer Manufacturers Association (ECMA) was kindly made available to WG 2.1. This proved to be particularly valuable to the WG 2.1 because of the care with which its stipulations had been phrased and was in fact used verbatim for many of the definitions of the present subset.

The form of the definition of the subset

The description of the subset consists of (1) the corrections necessary to convert the Revised Report on the Algorithmic Language ALGOL 60 into a report defining the SUBSET ALGOL 60, and (2) explanatory remarks describing the effect of the defining changes in an informal manner only.

Definition of SUBSET ALGOL 60 in terms of the Revised ALGOL 60 Report.

<i>Section</i>	<i>Subset definitions</i>	<i>Explanation</i>
2.3.	Delete from definition of declarator " own ". ¹	The own concept is not included in the subset.
5.	Delete first two sentences of fourth paragraph.	
5.1.1.	Replace the last two metalinguistic formulae by: "⟨type declaration⟩ ::= ⟨type⟩ ⟨type list⟩".	
5.1.3.	Delete last sentence.	
5.2.1.	Replace the last formula by: "⟨array declaration⟩ ::= array ⟨array list⟩ ⟨type⟩ array ⟨array list⟩".	
5.2.5.	Delete "even if an array is declared own ".	
4.7.5.	Add section 4.7.5.6: "No call of the procedure itself may occur during the execution of the statements of the body of any procedure, nor during the evaluation of those of its actual parameters, the corresponding formal parameters of which are called by name, nor during the evaluation of expressions occurring in declarations inside the procedure"	Recursive procedures and recursive use of procedures are not included.
5.4.4.	Delete last sentence.	
3.5.1.	Delete " ⟨unsigned integer⟩".	Integer labels are not provided for.
3.5.5.	Delete.	

¹ [Where typewritten copy uses underlining, this is replaced by boldface type in printed copy. — Ed.]

- 5.4.5. Replace third sentence by: "Specifications of all formal parameters if any must be supplied". Complete specification parts are required.
- 4.7.5.5. Replace by: "Kind and type of actual parameters must be the same as those of the corresponding formal parameters, if called by name".
- 2.1. Delete: " $|A|B| \dots |Y|Z$ ".
Delete: ",", or extended ... delimiter)".
Add: "Note: If a particular implementation requires capitals rather than small letters, one must regard them as a hardware representation for the small letters". Only one case of letters is provided for.
- 2.4.3. Replace: "They may be chosen freely" by: "Identifiers may be chosen freely; but the effects due to the occurrence of two different identifiers the first six basic symbols of which are common are undefined". In the subset identifiers are differentiated only up to six leading basic symbols.
- 3.3.4. Replace the words: "the following rules" of the last sentence by: "a set of rules. However, if the type of an arithmetic expression according to the rules cannot be determined without evaluating an expression or ascertaining the type or value of an actual parameter, it is **real**. These rules are". In the subset the type of an arithmetic expression will be in certain cases **real** where it will be **integer** in ALGOL 60. Thus arithmetic will be less precise in some cases.
- 4.3.5. Delete. The effect of a **go to** statement involving an undefined switch designator is undefined in the subset.
- 5.4.4. Add to text: "A function designator must be such that all its possible uses in the form of a procedure statement are equivalent to dummy statements".
- 3.3.4.3. Insert between "... rules" and "(": "with the exception that, if both the basis a and the exponent i are of **integer** type, then the exponent has to be an unsigned integer, otherwise the result is undefined". Exponentiation with integer basis and exponent is restricted in the subset.
- 2.3. Delete: " $| \div$ ". The so-called integer division is

- 3.3.1. Delete: “| ÷”
- 3.3.5.1. Delete: “÷”
- 3.3.4.2. Replace: “The operations ... both denote” by: “The operation $\langle \text{term} \rangle / \langle \text{factor} \rangle$ denotes”.
Delete last sentence.
- 4.6.1. Replace: “ $\langle \text{for clause} \rangle \dots \mathbf{do}$ ” by: “ $\langle \text{for clause} \rangle ::= \mathbf{for}$ $\langle \text{variable identifier} \rangle := \langle \text{for list} \rangle \mathbf{do}$ ”.
- 5.3.1. Replace: “ $\langle \text{switch list} \rangle ::= \langle \text{designational expression} \rangle | \langle \text{switch list} \rangle, \langle \text{designational expression} \rangle$ ” by: “ $\langle \text{switch list} \rangle ::= \langle \text{label} \rangle | \langle \text{switch list} \rangle, \langle \text{label} \rangle$ ”.
- 3.5.1. Replace the last two formulae by: “ $\langle \text{designational expression} \rangle ::= \langle \text{label} \rangle | \langle \text{switch designator} \rangle$ ”.
- 3.5.3. Delete: “In the general case ... is already found.”. Replace “selects one of the designational expressions ... a recursive process” by: “selects one of the labels contained in the switch list of the switch declaration. The selection is obtained by counting these labels from left to right”.
- 5.3.3. Replace: “These values ... its associated integer” by: “These values are given as labels entered in the switch list. With each of these labels there is associated a positive integer 1, 2, ..., obtained by counting the items in the list from left to right. The value of the switch designator corresponding to a given value of the subscript expression (cf. section 3.5. Designational expressions) is the label in the switch list having this given value as its associated integer”.
- 5.3.4. Delete.
- 5.3.5. Replace by: “Influence of scopes. If a switch designator occurs outside the scope of a label in the switch list, and an evaluation of this switch designator selects this label, then a pos-
- not included in the subset.
- The controlled variable in a for clause is restricted in the subset to be a variable identifier.
- In the subset the designational expressions in a switch list are restricted to be labels only.
- In the subset only unconditional and unparenthesized designational expressions are provided for. See 5.3.1.

sible conflict between the identifier used to denote this label and an identifier whose declaration is valid at the place of the switch designator will be avoided by a suitable change of this latter identifier.”.

- 4.7.3.2. Replace: “after enclosing this ... syntactically possible” by: “this actual parameter being an identifier, or string, otherwise the name replacement is undefined”.
- 4.7.5. Insert after: “... ALGOL statement” and before “.”: “in the sense of this subset”.
- 5. Insert after “... any one block head” and before “Syntax”: “The identifier associated with a quantity declared in a declaration may not occur denoting that quantity more than once between the **begin** of the block in whose head that declaration occurs and the semicolon which ends that declaration, excepting the case where this occurrence is the occurrence of a procedure identifier in the left part list of an assignment statement in the sense of section 5.4.4.”.
- 5.4.3. Add: “No identifier may occur more than once in a formal parameter list.”.

In name replacement (call by name) the actual parameter can only be an identifier or a string.

This report has been reviewed by IFIP/TC2 on Programming Languages in May 1964 and has been approved by the Council of the International Federation for Information Processing. Reproduction of this report for any purpose is explicitly permitted only in full. In making reference to this report the name IFIP SUBSET ALGOL 60 must be mentioned. IFIP does not authorise the language described in this report to be referred to as ALGOL without adding the word SUBSET.

International Federation for Information Processing
Working Group 2.1 on ALGOL
Chairman: W. L. VAN DER POEL,
Technological University Delft,
Delft, Netherlands