

REGNECENTRALEN  
Borups Alle 45  
Copenhagen N  
Denmark

ALGOL-Bulletin no. 13  
18.8.1961.  
Supported by:  
Rask-Ørsted Fondet  
Copenhagen

Temporary address of editor:  
Peter Naur  
Computation Center  
University of North Carolina  
Chapel Hill  
North Carolina  
U.S.A.

13.1. Summary by P. Naur of the discussion in the

U.S. ALGOL 60 MAINTENANCE GROUP DURING AUGUST 1960 TO JUNE 1961.

According to the rules quoted in AB 10.1.11 the discussion within the U.S. has been conducted by letters sent directly from each contributor to all other members of the group. The following summary is based on the letters collected at the Computation Center, University of North Carolina.

Serial no. 1. From H.H.Bottenbruch and A.A.Grau. 19.9.60.  
Subject: The then corresponding to a given else (AB 10.1.3.2).

The authors propose the following:

Clarification. Whenever in an ALGOL program, ambiguity in meaning is possible because else may be associated with any of several if clauses in a set of nested statements, the else must be associated with the last preceding if clause involved in the ambiguity. Thus, this if clause, the statement between it and else, the else, and the statement following else, together constitute a conditional statement. The statement parentheses (begin, end) may of course be used to indicate a different association.

Serial no. 2. From Bill Wattenburg. 19.9.60.  
Subject: Conditional Statements.

A detailed proposal for changes of the ALGOL 60 Report to allow both conditional and unconditional statements following the then of an if statement. This includes rules for a unique association of any else with an if clause as well as changes of the syntactic formulae. A rule for the meaning of a for clause controlling a conditional statement is also given.

Serial no. 3. From H.H.Bottenbruch and A.A.Grau. 29.9.60.  
Subject: Evaluation of functions which change the values of global parameters.

The authors propose not to remove the ambiguity at all, but simply to add the following:

Clarification. Whenever in an ALGOL program, ambiguity in meaning is possible in the evaluation of arithmetic expressions because a function contained in that expression changes the values of global variables, that ALGOL program is not defined.

Serial no. 4. From Peter Zilahy Ingerman. 5.10.60.  
Subject: The three proposals, serial no.s 1, 2, and 3 (see above).

The author favors serial no. 1 over serial no. 2 because of its greater simplicity. He considers the point to be a minor one, and thinks that changes to the ALGOL report should be kept to a minimum.

The author opposes the proposal of serial no. 3, which he finds is a misinterpretation of the original proposal approved by the Group (quoted in AB 10.1.10.5, Items 4, 5 and 6).

Serial no. 5. Application from RCA concerning membership. 6.10.60.

Serial no. 6. Application from Carnegie Institute of Technology. 30.9.60.

Serial no. 7. From F.L.Bauer and K.Samelson. 20.10.60.  
Reprinted in full in AB 11.1.1.

Serial no. 8. Application from Henry C. Thacher, Argonne National Lab. 21.10.60.

Serial no. 9. From Henry C. Thacher. 21.10.60.  
Subjects: 1. Extension of ALGOL to allow initialization of own variables.  
2. Extension to allow a synonym assignment.

The author claims that own variables cannot be fully exploited because means for initialization are lacking. He proposes to remedy this by changing the type declaration to include an initialization.

In order to improve the convenience of the language he further proposes to include a synonym declaration, which would make it possible to refer to a non-local quantity of a block by a different identifier inside the block than the identifier used outside.

The proposals are discussed and illustrated in detail.

Serial no. 10. From Peter Zilahy Ingerman and Kirk Sattley. 7.11.60.  
Subject: Proposed symbols for use in ALGOL translators.

The authors give a list of letter combinations, corresponding to all basic symbols and metasyntactic classes of ALGOL 60, for use in describing ALGOL translators.

Serial no. 11. Application from Royal McBee Corporation. 7.11.60.

Serial no. 12. Change of membership for Bendix Computer Division. 12.12.60.

Serial no. 13. Application from duPont. 18.11.60.

Serial no. 14. From Henry C. Thacher. 1.1.61.  
Subject: Policy on changes to ALGOL 60.

The full text of this letter goes as follows:

It is suggested that the following policy on changes to ALGOL 60 be adopted by the ALGOL 60 Maintenance Group:

1. For the present, changes to ALGOL 60 which would have the effect of invalidating programs acceptable under the syntax and semantics of the 1960 report shall not be approved unless they are necessary to eliminate logical inconsistency or ambiguity. Removal of ambiguities shall be accomplished in such a way that actual changes in the report are minimized.

2. Changes to ALGOL 60 which will have the effect of invalidating existing programs shall, however, be considered to determine their utility, their implementability, and their effects upon the validity of existing programs. If found acceptable, they may be given tentative approval, to be confirmed when the time comes for an extensive revision of ALGOL.

3. Changes to ALGOL 60 which would not have the effect of invalidating programs acceptable under the syntax and semantics of the 1960 report may be approved whenever it can be determined that they meet the following criteria:

- a. They are logically consistent with the present language.
- b. They either extend the scope of algorithms which can be described by ALGOL, or increase the convenience of ALGOL as a programming language, or permit improvements in the object code which would be produced by a compiler.
- c. No superior method of achieving the same end is apparent.

This statement of policy is intended to serve as a compromise between two opposing arguments as to the best way to secure widespread acceptance of ALGOL. The first position, as set forth by Bauer and Samelson, is that a language in a constant state of flux cannot be expected to gain acceptance. Their position is supported by the considerable investment in preparing compilers, and in publishing collections of tested algorithms.

The second position is that a language which cannot describe common computing and data processing procedures is unlikely to gain full acceptance. Although experience in using ALGOL 60 is still limited, it appears that such techniques as multiprecision arithmetic operations, symbol manipulation, and many others will be, if not impossible, at least very clumsy, to describe in the present language.

There will thus be a strong pressure toward development of extended languages which can cope with various tasks of this type, and unless the ALGOL Maintenance Group is sympathetic toward the needs of such workers, there is a likelihood of a second Babel.

By allowing the extension of the scope of ALGOL 60 without undue delay, while maintaining the integrity of the basic structure, it should be possible for ALGOL to compete successfully with independently developed languages.

Although the acceptance of extensions to ALGOL 60 would pose a certain problem to compiler writers and to publishers, it is not believed that this problem is as serious as it appears at first. It is, indeed, likely that very few compilers will be able to accept ALGOL 60 with all its extensions. However, this is likely to be true for basic ALGOL 60 as well. It will always be possible to write a basic ALGOL program which cannot be compiled or run on a given machine because of equipment limitations. For this reason, as well as because of the need to translate to an appropriate hardware representation, a preliminary editing of published programs will always be necessary. If this preliminary editing reveals the presence of sections which cannot be accepted by the particular compiler, there will be two alternatives:

- a. If the section is one which can be expressed, even if less effectively, by basic ALGOL, this substitution can be made.
- b. If the section cannot be expressed in basic ALGOL, it must be hand coded to machine language. In this case, the extension allows a more widely acceptable description of a procedure which is outside the scope of ALGOL 60.

The approval of extensions to ALGOL 60 need not harm the effectiveness of the Taschenbuch. To the extent that they increase the number of algorithms which can be described, they will make the book more useful, and the extensions may well be described either in the algorithms where they are introduced, or in supplementary numbers.

An example of a type of extension which I feel should not be postponed unduly is the proposed string statements of Wegstein, Youden, and Galler, as well, of course, as Argonne Proposal no. 9.

Serial no. 15. From Henry C. Thacher. 20.1.61.  
Subject: Initialization of own variables (cf. serial no. 9).

It is pointed out that the initialization of own variables might be achieved through the use of a go to statement to an undefined switch designator.

Serial no. 16. From Peter Zilahy Ingerman. 6.3.61.  
Subject: Own variables.

The question is raised whether own variables will retain their values from one use of a program to the next, thus for example counting the number of uses of the program.

Serial no. 17. Application from IBM. 4.4.61.

Serial no. 18. Application from Dartmouth College, Thomas E. Kurtz. 18.4.61.

Serial no. 20. From Jean E. Sammet. 10.4.61.  
Subject: Combining ALGOL and COBOL.

This letter was accompanied by a paper entitled A METHOD OF COMBINING ALGOL AND COBOL. This was presented at the Western Joint Computer Conference, Los Angeles, May 9 - 11, 1961. The stated goal is to achieve a combination of the two languages with a minimum change in either.

Serial no. 21. From F.L.Bauer, H.H.Bottenbruch, A.A.Grau, K.Samelson, J.Wegstein. 20.4.61.  
Subject: Policy on changes to ALGOL 60.

The full text of this letter goes as follows:

The existence of the ALGOL Maintenance Group has caused some concern among translator constructors and prospective users of the language. See for instance the letter of R. Bemer on page A12, Comm. ACM, Vol. 4, No. 3.

It must be admitted there are some doubts concerning the interpretation of certain minor points of the ALGOL report. For some time it was considered a matter of great importance to have these ambiguities resolved. In practice this has turned out to be unimportant. The algorithms which have been published so far in the Communications are unambiguous. Also it is very simple to change an ALGOL-program which is ambiguous into an unambiguous program, e.g. by inserting parentheses. The dubious points are furthermore completely insignificant as far as the power of the language is concerned.

We therefore propose that this committee adopt the following general attitude towards ALGOL maintenance:

The members of this group will adhere to the ALGOL language as defined in the ALGOL 60 Report. Translators should be constructed in such a way that ALGOL programs which are unambiguously defined by the report will be correctly translated. ALGOL programs which are ambiguous are not defined. For several years to come this committee will not propose any changes of or additions to the ALGOL language. Now is the time to implement ALGOL 60 and gain experience with it as a programming tool.

The ALGOL Maintenance Group should assist ALGOL Compiler implementors and users in interpreting and explaining the ALGOL 60 language and serve as an exchange for information on ALGOL Compiler implementation and use.

Serial no. 22. From A.J.Perlis, Arthur Evans, Jr., H.R.Van Zoeren. 2.5.61.

Subject: The previous letter from F.L.Bauer, et al. (20.4.61)

Full text:

The undersigned heartily approve of the sentiment expressed in the letter referred to.

Serial no. 23. From E.D.P. Gross, Jr. 5.5.61.

Subject: The above letter from F.L.Bauer, et al. (20.4.61)

Full text:

This facility concurs with the sentiment expressed in the Memo referred to above. In fact our construction of an ALGOL Compiler is predicated upon the general sentiment of the Memo.

Serial no. 24. Application from T.E.Cheatham, Jr., Computer Associated, Inc. 8.5.61.

Serial no. 25. From Robert J. Hunn, duPont. 10.5.61.

Subject: The above letter from F.L.Bauer, et al. (20.4.61)

Supports the letter referred to.

From Henry C. Thacher, Jr. 16.5.61.

Subject: Problem-Oriented Languages Without a Compiler.

The letter was accompanied by a 9 page paper presented at the 1961 meeting of POOL, the LGP-30 Users Group. The paper describes the experience obtained in using published ALGOL programs as the bases for machine codes, without the use of a compiler. Conclusions:

1. Problem oriented languages such as ALGOL are generally better adapted than either standard mathematical notation supplemented by English texts, or conventional flow charts, or machine coding for communicating and describing computational procedures.

2. The ALGOL language, at least in its more important features, is not difficult to learn.

3. Very little training is needed to allow the interpretation of published algorithms. These can be a considerable savings to installations either with or without the services of a numerical analyst.

4. Even if translators which can handle most of ALGOL are available, hand-coding from an ALGOL statement will always have important applications.

5. The ALGOL language can be of considerable assistance in designing numerical programs and in documenting machine codes.

Serial no. 26. From Peter Zilahy Ingerman.

31.5.61.

Subject: Miscellaneous Business to Date.

1. The letter from F.L.Bauer, et al. (20.4.61), referred to as the Oak Ridge Proposal. The author writes:

With respect to this proposal, we would like to make the following suggestions:

There seem to be two distinct functions to be served by ALGOL: that of communication of algorithms among human beings, and that of providing universal compiler language for communication of algorithms to machines.

The first function demands a language without ambiguity but does not require that any particular attention be paid to the resolution of difficulties which the second function causes to arise. Hence, the Oak Ridge Proposal clearly advocates the maintenance of the first function but, unfortunately, seems to minimize the second function by failing to provide compiler writers with an official decision regarding the ambiguities in the language. Note that we do not consider the assertion in the Oak Ridge proposal that 'anything which is ambiguous is undefined' to be a suitable answer. Hence, we feel that we must vote 'no' to the Oak Ridge proposal as it stands, although we agree with the spirit of the proposal.

We propose as an alternative, with apologies to Oak Ridge for the plagiarism:

'The members of this group will adhere to the ALGOL language as defined in the ALGOL-60 Report. Translators should be constructed in such a way that ALGOL programs that are unambiguously defined by the report will be clearly translated. The committee will prepare immediately a list of ambiguities at present in the ALGOL language so that these ambiguities may be avoided by algorithm writers who prefer quiet to contention.

Anyone who constructs a quasi-ALGOL translator, often a recognizably important activity, is requested to be explicit about his departures from canonical ALGOL, since such departures should be interpreted by the committee as changes or additions which are in some sense desirable.

The committee will prepare during the next several years its official recommendations towards the resolution of ambiguities in ALGOL 60 and any changes or additions thereto. These recommendations will be made available to any implementor or user upon request but will not be considered in any sense final until such time as they are officially recommended as changes to ALGOL'.

We hope this is in the spirit of the Oak Ridge proposal. It is offered only to avoid the embarrassment of having compiler writers give up in disgust because there is no 'official ALGOL' to translate.

2. Acceptance of new members.

3. The Jean Sammet proposal concerning combining ALGOL and COBOL (10.4.61). The author rejects the proposal, but advocates the appointment of a small joint subcommittee for further study of the matter.

4. Acknowledgement of receipt of various papers, including a letter from W. Borsch-Supan to the editor of Datamation, pointing out 13 different errors in the articles by Dr. Ivan Flores in Datamation, Sep/Oct, and Nov/Dec 1960.

5. Comments on a letter from Mr. Nagao, Japan.

6. Proposal for a meeting around the time of the 16th National Conference of the ACM in Los Angeles, Sept. 5 - 8, 1961.

Serial no. 27. From H.H.Bottenbruch and A.A.Grau.  
Subject: Letter from P.Z.Ingerman, serial no. 26.

9.6.61.

Full text:

We concur in your suggestion that a list of ambiguities in ALGOL 60 should be drawn up and circulated. These should be accompanied by devices that avoid them. They might also be published in the Communications of the ACM.

We may also say that expositions of ALGOL are in preparation at Mainz, Oak Ridge, and elsewhere, both on the elementary and on the programmer's level.

There is some background behind the memorandum of which you may not be fully aware. It was originally hoped that the ALGOL project would provide the universal communication, publication, and programming tool sorely needed to give scientific computation cohesion. Therefore, the ALGOL Maintenance Group must give attention to practical and economic, as well as theoretical, considerations. The construction of a compiler for ALGOL, together with the necessary adjuncts such as diagnostic routines and operating system, is a relatively expensive activity. We cannot expect manufacturers to take the language seriously if a decision of this committee could overnight cause their work to become obsolete. Here at Oak Ridge we experienced just this in the transition from ALGOL 58 to ALGOL 60, and so we know what this means. Indeed, the changes made have been a most serious setback for ALGOL, as shown by the SHARE actions and R.W.Bemer's letter in the Communications, though no one will argue that most of the changes were not definite improvements. We must assure compiler builders that this will not be repeated.

In the memorandum, we had in mind compiler builders and not so much algorithm writers. A compiler once written is usually hard to change; programmers find it relatively easy to adopt new features. It follows that the counterproposal made in your letter is not in the spirit of the memorandum.

The fact that the committee exists has already created the impression in some quarters that little is left in ALGOL 60 unless ambiguities are resolved. We insist, however, that this is not so; on the contrary, the avoidance of ambiguous and unclear features leaves a powerful and flexible language.

We even contend that the resolution of the ambiguities would add little to the power and practical usefulness of the language. Also, the existence of ambiguities does not endanger program exchange if translators are written in such a way that they reject (or correct) ambiguous programs.

We may consider some typical ambiguities and difficulties:

1. The use of a for clause with a conditional statement. This is satisfactorily avoided by the introduction of statement parentheses.

2. The use of globals in functions. This is contrary to all ordinary mathematical usage. Its avoidance is automatic for scientists and engineers who are conversant with mathematical language and wish to make use of machines. It poses problems only for those who would like to use programming tricks.

3. The use of recursive procedures. In numerical work, these have been little used. Before they could be incorporated fully into ALGOL, considerable additional research must be undertaken. This cannot be done satisfactorily under the pressure of committee meetings, discussions, and decisions.

4. The handling of strings. Provision already exists for the use of strings as parameters in output procedures, which accomplishes the same thing as in Fortran. It is quite satisfactory for scientific numerical computation. ALGOL, it may be recalled, was not designed as a compiler building language.

We are seriously concerned about the fate of ALGOL. The ALGOL project should not be destroyed by the wish of some to continue the design of languages. They can do that apart from ALGOL maintenance. Only experience with ALGOL will indicate the direction that the resolution of ambiguities and improvements must take. We shall never have this if ALGOL is not given a fair chance. This requires ALGOL compilers. It is absolutely imperative, therefore, from the practical point of view, that compiler builders and algorithm writers concentrate on the clear and unambiguous part of ALGOL and that a moratorium for a considerable period be declared on changes in and additions to the language as given in the ALGOL 60 Report.

Serial no. 28. Change of representation of RCA. 19.6.61.

Serial no. 29. Change of representation of University of North Carolina. 13.7.61.

From J. Wegstein. 20.7.61.

Subjects:

1. Up-to-date membership list. Compared with the list in AB 10.1.10.1 there are the following changes:

Delete W. Feurzeig and Kirk Sattley, Chicago LAS.

Replace Thomas P. Saboski, North Carolina, by Miriam Shoffner, Lee Butler, and Robert Caviness.

Delete Mart. I. Bernstein, Rand.

Add the following:

Robert E. Hux	A	Radio Corporation of America
Kenneth A. Brons		EDP, Applied Programming
Raymond Dash		Cherry Hill, Bldg. 204-2
J. Bruce Paterson		Camden 8, New Jersey
Jordan B. Rabin		

Prof. Alan J. Perlis	AF	Computation Center
Arthur Evans, Jr.		Carnegie Institute of Technology
Harold R. Van Zoren		Pittsburgh 13, Pennsylvania

Prof. F.L. Bauer	A,AL	Institute fur Angewandte Mathematik
Prof. K. Samelson		Johannes Gutenberg-Universitat
		Mainz, Germany

Dr. Henry Thacher, Jr.	U	Argonne National Laboratory
		9700 South Cass Avenue
		Argonne, Illinois

E. D. P. Gross, Jr.	AF,U	Computer Center
		Royal McBee Corporation
		1031 New Britain Avenue
		West Hartford 10, Connecticut



Robert J. Hunn	U	Engineering Service Division E. I. DuPont de Nemours and Co. Wilmington, Delaware
Rainer Kogon Rex Franciotti	AL	IBM Corporation 112 East Post Road White Plains, New York
Thomas E. Kurtz	AL,U	Mathematics Department, Dartmouth College Hanover, New Hampshire

2. Log of formal correspondence.

3. The Frank Wagner Committee. The text of this item is:

On May 15, the President of the ACM sent a letter to Mr. Frank Wagner, Mr. George Forsythe, Mr. Joe Wegstein, and Mr. Robert Bemer. It said:

'This is to confirm your appointment to an ad hoc committee which has the responsibility to make a recommendation to the ACM Council relative to the situation on ALGOL. It seemed to me that the critical point discussed at the council meeting was the question of the relationship between the publication language of ALGOL 60 and the subsets which various groups are using as legal inputs to their respective systems.'

I would very much appreciate any advice that you can send me on what to recommend to the ACM Council.

4. Published Algorithms. The Sept. Comm. will contain the 69th published algorithm.

5. Suggestion for a meeting in Los Angeles on the 5th September.

6. Call for a vote. A vote has been requested on the last three paragraphs of the letter from F.L. Bauer, et al., dated 20.4.61, beginning: We therefore propose . . .

From John W. Carr, III.

24.7.61.

Subject: Points 5 and 6 of the previous letter from J. Wegstein.

The Univ. of North Carolina concurs in the suggestion for a meeting and votes yes.

From Jean E. Sammet.

28.7.61.

Subject: Business matters and vote on Oak Ridge Proposal.

1. The author concurs in holding a meeting in Los Angeles.

2. The text concerning the Oak Ridge Proposal reads:

Having disposed of this trivia, I can now address myself to the more important matter of this Oak Ridge proposal of April 20, 1961. In response to your request for a vote, I must cast a very loud NO. I feel that the policy suggested in the memo by Henry Thatcher dated January 1, 1961 is a much more useful and desirable one. My reasons are based essentially on the premise that problems do not disappear just because they are ignored. I consider the most objectionable sentence in the Oak Ridge proposal to be the one stating 'For several years to come this committee will not propose any changes of or additions to the ALGOL Language'. This seems to negate the very purposes of having a Maintenance Committee.

There seem to be four major types of activities in which any maintenance group can and should engage, in addition, the dissemination of information:

- (1) removal of inconsistencies
- (2) clarification of ambiguities
- (3) addition of more capability
- (4) changes of the specifications

The amount of time and emphasis devoted to each of these is a function of the system being maintained and the basic interests of the people involved.

It is perfectly clear by now that there are ambiguities and inconsistencies in ALGOL. This is to be expected, since the state of the art does not yet permit the development of language specifications without these unpleasant characteristics. This basic fact cannot be ignored. To maintain the respect of those who wish to use ALGOL 60, it seems essential that the Maintenance Committee at least have the appearance of a group attempting to correct at least this type of deficiency.

The 4 main items of any maintenance group were listed in what I consider decreasing order of importance for most situations. Thus, the addition of more capability to the system - whether the system be a language, a program, or a piece of hardware - is generally the last thing to which attention should be directed (unless of course the system is too weak to do its job. That is certainly not the case with ALGOL.) In the case of a language, it seems to me that the most important function of any maintenance group is to remove inconsistencies whereas it can afford to take its time about adding more features, and should certainly hesitate a long time before making changes which invalidate past work. With regard to ALGOL 60, there have been cited in various documents some inconsistencies and a large number of ambiguities, usually followed by the remark that these cases are not really important anyhow. This seems to be just ducking the issue; either a problem exists or it doesn't, and in the former case, it should be solved. There are several things which the ALGOL Maintenance Committee can do which will not in any way disturb the integrity of the concept that the language should not be continually changing. The most important item is to ferret out all of the inconsistencies and ambiguities and correct them by legally established procedures. From long and bitter experience in another effort of a similar kind, I know that the correction of an inconsistency or an ambiguity often requires changes in remote areas of the specifications. This fact requires that these corrections be grouped so that as many possible could be made at one time. This helps avoid the likelihood of compounding the errors. The same statements apply with equal force to the clarification of ambiguities.

The important factor in all of this is the timing. It must be emphasized that there is a difference between doing things slowly and carefully and not doing them at all. The ALGOL Maintenance Committee seems to be headed in the latter direction, whereas it could so easily be taking the steps which are necessary to improve the usage and acceptance of ALGOL as a universal language. The making of changes and additions to the present specifications is indeed something that should be viewed with extreme caution, but again this does not mean that nothing constructive can be done. Specific illustrations of things which could be investigated are the addition of an I/O package specifically suited to ALGOL, and the complete detailing of the changes which would have to be made to permit combining with COBOL. The task of cleaning up the inconsistencies and ambiguities is not an easy one, and the principle of doing this in a reasonable time scale should become an established one.

If the current motion is defeated - and I hope very strongly that it is - then I suggest that several working subcommittees be established, with specific tasks and some reasonable and realistic deadlines. This is the only way progress is ever made.

From Rainer Kogon.

31.7.61.

Subject: The call for a vote of J. Wegsteins letter of 20.7.61.

The author turns the attention to the Suggested Conclusions of AB 11.1.6. He further thinks that to cast a meaningful vote on the proposal it would be necessary to clear up the following remark taken from the 8. June 1961 Minutes of the American Standards Association Subcommittee X3.4 on Common Programming Languages:

' . . . The ALGOL maintenance situation was summarized to the effect that ambiguities and inconsistencies are so few and minor, that psychologically it is not desirable to make changes to ALGOL 60 until ALGOL 62 comes along.'

13.2. Comment from M. Woodger, England, on

26.4.61.

#### STRINGS AS ACTUAL PARAMETERS.

The ALGOL 60 Report Section 4.7.5.1 is not strictly true.

A counterexample has arisen in connection with the recognition of strings on an input tape. SEEK is a procedure whose body is expressed in code and whose heading is:

```
procedure SEEK(S); string S;
```

This procedure is used within the body of a procedure READ as a global quantity, the procedure declaration for READ beginning thus:

```
procedure READ(X,Y); integer X; string Y;  
begin SEEK(Y); ...
```

When a procedure statement such as READ(3, '3') is executed the string '3' is an actual parameter, and the body of the procedure READ is an ALGOL 60 statement, although invoking the procedure SEEK whose body is not.

I would recommend the deletion of section 4.7.5.1, as a revision of the Report; the question is already sufficiently dealt with in section 4.7.8.

13.3. Suggestion from R.W.Hockney, England, concerning

3.5.61.

#### INCLUSION OF COMPLEX AND OTHER DECLARATIONS IN ALGOL.

I feel that the early suggestion by A. Van Wijngaarden and E.W. Dijkstra (AB 7.3.5) for the inclusion of complex and other declarations in ALGOL should be followed up, in order to make the language more attractive for engineering calculations.

ABS 12, which has been distributed, suggests an extension to ALGOL 60 on these lines and I feel that further discussion and, if possible early standardisation, on such an extension would be beneficial to users in Industry. Comments are invited.