



# National Physical Laboratory

TEDDINGTON · MIDDLESEX

Please reply to the DIRECTOR and quote our reference    Telegrams: Physics, Teddington    Telephone: Molesey 1380, Ext.

OUR REF: Ma. 8/29/01

Mathematics Division

YOUR REF:

16th February, 1960

Dear Peter,

I have just found your letter of February 4th in an envelope I thought to be empty! My apologies for the delay, and never mind if I have missed your time limit so that you are unable to incorporate my remarks.

I want to thank you for your valiant efforts on our behalf. I am largely satisfied with the result, even if different from what I had noted to be our "agreements". I have the following detailed comments:

1. Where reference to order of symbols in a program occurs it is rather unfortunate to refer to "from left to right". I suggest "in sequence as written" or some such wording. Page 15 line 3 and para 4.6.4 are O.K. p.11, 1.8, para 3.4.6, 3.5.3, 4.2.3.1, 4.5.5, 5.2.3.1, 5.3.3 are affected.
2. As regards  $\div$  I thought we admitted this for McCarthy's benefit, among others, and I have no strong feelings either way.
3. I agree with Rutishauser with regard to 4.2.4, and with his further wish that the transfer procedure equivalent to entier ( $E + 0.5$ ) should be understood to be automatically invoked whenever an expression  $E$  appears as a subscript. This of course contradicts 3.1.4.2 which would have to be altered.
4. I agree to remove the possibility of writing conditional expressions ending with an if expression.
5. I am satisfied with the incorporation of a list of parameters to be supplied by value, rather than the complementary "name" list.
6. I object to Rutishauser's amendment of 3.4.1. One may still insert brackets if one desires so, but this should not be obligatory. The order of precedence in 3.4.6.1 seems satisfactory. In particular I would not object to the construction  $S \leq 0 \vee P \leq Q$  (cf. Rutishauser's letter of 12th February).
7. I agree with Rutishauser that your syntax does not allow labelled conditional statements, and I don't see why these should be excluded.
8. I agree with Rutishauser to delete the last sentence within the brackets in 4.5.3.2.

/I

Mr. Peter Naur,  
Regnecentralen,  
Gl. Carlsbergvej 2,  
Valby,  
Denmark.



9. I would prefer to delete 4.5.4 rather than to patch it up - e.g. by inserting the word "largest" before "unconditional statement".

10. I also thought that the controlled variable of a for statement was automatically local to it (cf. 4.6.5).

11. p.2, line 2, for "forty" read "fifty". Line 20, delete "coming". Line 3 from end, for "case" read "ease".

12. p.3, line 7 from end, after "Lowering of the line between the brackets" add "and removal of the brackets".

13. To avoid the conflict between the two uses of the word "rules" on page 4, lines 24 and 27, I suggest the first be replaced by "instructions".

14. To avoid a similar conflict with regard to the word "self-contained" on page 4, lines 14 and 30, I suggest the first be deleted.

15. p.4, line 27, for "A declaration" read "Each declaration".

16. p.5, lines 11-13 are most obscure. Almost any rewording, in your own words, must be better.

17. p.5, definition on line 17 requires angle brackets enclosing the words "the null string of symbols".

18. At end of 2.1. add "(cf. sections 2.4 IDENTIFIERS, 3.2 PROCEDURE VALUES and 4.7 PROCEDURE STATEMENTS)".

19. In 2.3, lines 1 and 11, for "specificator" read "specifier" (as on p.25).

20. At end of 2.3, why underline the word "comment" and yet not admit it as a basic symbol? As a departure from the original Algol I regret this exclusion. Also, is it necessary to have a semicolon preceding "comment"? Also for "syntactically" read "semantically".

21. At end of 2.6.3 add "strings are used as actual parameters of procedures (cf. 3.2 PROCEDURE VALUES and 4.7 PROCEDURE STATEMENTS)".

22. Item 2.7, line 4, the word "type" should be within quotation marks.

23. Do I understand by the second paragraph of 2.9 that a rule is its own name? If so it would be briefer to say just that. Then I suppose a number, as a special case of an expression, is a rule - defining itself as its own value - and hence is also its own name. Likewise a label must be a rule, its own value, and its own name! I hope this terminological beanfeast is harmless!

24. Item 3, line 7, for "may contain" read "contains".

25. Item 3.2.3. The word "rules" is used in two very different senses. Suggest "restrictions on" instead of "rules for".

26. Item 3.2.5. "Recognized entities" should be more explicit.

27. The second alternative in the definition of "arithmetic if expression" in 3.3.1 is superfluous and should be omitted, since an arithmetic expression within parentheses is a primary, hence a factor, hence a term, hence a simple arithmetic expression.

/Likewise



Mr. Peter Naur, Regnecentralen, Denmark

✓ 28. Likewise the second alternative in the definition of "Boolean if expression" in 3.4.1 should be omitted.

✓ 29. Page 11, line 10, needs qualification. The first arithmetic expression following the Boolean might be part of a longer one, and it is the longest such "first" expression that is meant here.

✓ 30. Item 3.5.4, line 3, to read "assumes one of the positive values...".

✓ 31. Item 4.1.2 for "Unlabeled" read "unlabelled".

✓ 32. Item 5.3.5 uses the idea of the scope of a quantity, but 2.10 only defines the scope of a property of a quantity.

✓ 33. I am strongly opposed to 4.3.5. If this were to remain, then the implementation of the last sentence of 4.7.3.2 would involve (in the case of labels as parameters) searching for dummy go to statements which used labels that did not exist within the procedure body.

✓ 34. In accordance with Document 17, I would prefer to omit the whole of items 4.7.5 and 4.7.6 as unnecessary. If nevertheless I am overruled on this I have the following comments. On 4.7.5.3, it is not always possible to determine whether a formal identifier is used within the procedure body as an array identifier or not - e.g. example on p.26, the identifiers a and b - except of course from the specification part. On 4.7.6, I would say that a procedure statement is undefined if written outside the block B in whose head the procedure declaration appears. A global identifier I of the procedure body may be defined several times within different blocks included in B and in each such block a procedure statement may use the procedure with the corresponding definition of I.

✓ 35. Item 5, line 5, for "a dynamical" read "an".

✓ 36. Item 5.1.3. Add "For the semantics of own see the fourth paragraph of 5 above".

✓ 37. Item 5.3.3, last sentence, for "designational expression" read "value of the designational expression". Then the second sentence is not quite right, since the values may not actually be given in the switch list.

✓ 38. Item 5.3.4. At the end insert the word "involved".

✓ 39. If the specification part of a procedure declaration is necessary it should not be optional. Under the interpretation of our agreements presented by you it seems to be necessary.

✓ 40. I am sorry I have no comprehensive illustrative examples. Perhaps a separate volume of these would be in order.

I look forward to hearing from whoever produces a translator first for Algol 60. The proof of the pudding is in the eating.

Best wishes,

Yours sincerely,

*Mike*

M. WOODGER