

PROCEDURE STATEMENTS
-----Perlis
Samelson

Green

4.3. Semantics

A procedure statement serves to initiate (call for) the execution of a procedure block (cfr. procedure declaration section). The execution however, is effected as though all formal input parameters listed in the formal part of the procedure declaration heading were assigned the values of the actual parameters in the corresponding positions in the procedure statement ; and the values of all formal output parameters listed in the procedure declaration are assigned to the actual corresponding output parameters listed in the procedure statement.

4.4. Formal description of the correspondence

The input-part-output-part of the procedure statement must be identical in form with the formal part of the corresponding procedure declaration. In this way there is defined a one-to-one correspondence between the parameters in the input-output lists of the procedure statement and the identifiers in the formal part of the procedure declaration. This one-to-one correspondence together with the input-output specifications given in the procedure heading (section) give complete information concerning the admissibility of parameters employed in any procedure call. Rules covering all admitted cases are given as follows :

4.5. Identity of parameters

If a formal output parameter is identical to a formal input parameter, this identity must be preserved in the call. The corresponding actual parameter as well as any other parameter entered both as input and output parameter in the procedure call must obviously meet the requirements of both input and output parameters.

4.6. Actual parameters.

All actual parameters must be defined in a compound statement where the procedure statement occurs.

4.7. Global parameters of the procedure compound

Identifiers which are global to the procedure compound and which have been used there must not have been redefined (by declaration) in the compound where the procedure statement occurs.

4.8. The effect of a procedure call may be specified in the following symbolic form :

Let the procedure declaration be :

procedure NAME (u, V, A, F, P) = : (Z, A', L, S) ;

array A, A' [d : d'] ; function F (X) ; label L ; switch S [m] ;

procedure P (V, B) = : (W) ; array B [k : k'] ;

begin Σ end NAME ;

Let the procedure call be :

NAME (t, E, C, G, R) = : (q, D, J, SWITCH)

Then the effect will be as though the call will have been replaced by :

LOCAL u, V ; ARRAY A, A' [e : e'] ¹⁾ ;

function F (x) := G (x) ;

procedure P (V, B) = : (W) ; array B [k : k'] ;

begin R (V, B) = : (W) ; ~~return~~ end P ;

begin u := t ; 2)
V := E ;
A $\Leftarrow$ C ³⁾ ;

<procedure compound> ^{4, 5, 6, 7)}

LABEL : end NAME ;

Procedure
declaration

Foot notes -

- 1) e, e' are the dimensions specified in the array declaration for C. *and D*
- 2) E is an arithmetic expression whose type corresponds to that declared for the formal parameter V in the procedure declaration heading.
- 3) Symbolic description of an appropriate ALGOL compound for effecting the transfer of the values of the elements of the array C to become the values of the elements of the array A.
- 4) Each statement in the procedure compound of the form

Z := <Text> ;

becomes

q := Z := <Text> ;

Each statement in the procedure compound of the form

$A' [i] := \langle \text{text} \rangle ;$

becomes

$D [i] := A' [i] := \langle \text{text} \rangle ;$

5) All occurrences of the label L and the switch S are replaced by J and $SWITCH$, respectively.

~~6) All occurrences of return are replaced by go to LABEL.~~

7) It is as though each call requires a copy of $NAME$ by the above r replacement. All own declarations in each of these copies refer to the same set of variables. This is accomplished by some meta-ALGOL technique.

AP/js

1.14.60