

Ruhshauer
BauerSEMANTICS OF PROCEDURE STATEMENTSCase 1 :

formal input parameter : $f \neq$ single variable .

formal specification : type F (f) (special case : no specification)

actual parameter : $E \neq$ expression in variables which are defined at the level where the procedure declaration is written.

formal identifier will be replaced by $g \neq$ unique identifier of type F which is local.

Execution is effected as if \$)

$g := E$; procedure compound

Case 2 :

formal input parameter : $f \neq$ identifier defining an array

formal specification : type F array (f [$\langle$ array list $\rangle$]). (special case : type F lacking)

actual parameter : a partly open expression list

The number of open subscript positions must be the same as the number of subscript positions for f .

declaration for actual parameter : type A array (a [$\langle$ array list $\rangle$])

formal identifier f will be replaced by $g \neq$ " unique " identifier of type A , at the same time an appropriate array declaration-derived from the array declaration for a - must be provided for g .

Execution is effected as if \$

for ... begin

for ... begin

g [.....] = a [..... , l, l ,]

.....

end

end ;

procedure compound.

where the variables running in the for-statements correspond to the empty position of the arrays g and a, 1,1 indicating the prefixed positions of the array a.

§)

All problems of type-transition are covered by the rule that the assignment(s) must be permitted according to the semantics of assignment statement and formation of expressions.

Case 7

formal output parameter : f 2 single variable

mutatis mutandis the same rules hold as in case 1 with the following change in the execution

The assignment a : = g immediately follows before any return.

Case 8

formal output parameter f 2 identifier representing an array.

mutatis mutandis as in case 2, the changes in the execution mutatis mutandis as in case 7.

1.13.60

Case 3-6 remain

<u>Add</u>	<u>Heinz, Fritz</u>
For 7	5
Against 4	4
N	
Mer.	