

Algol, Mainz, 14. dec 1959

Procedure declarations . A procedure declaration declares a ~~compound~~ statement to be a closed unit (a procedure or a function) which may be regarded as a single compound operation (in the sense of a generalized function) depending on a certain fixed set of ~~input~~ parameters, yielding a fixed set of results, and having a fixed set of possible exits defining possible successors. The parameters, the results, and the successors may be specified either through entities ^{either explicitly} ~~declared~~ ^{understood} (explicitly or implicitly) (to be global for the compound statement constituting the procedure, or may be specified through formal input and output parameters, which will be substituted by actual parameters according to the rules given under procedure statements.

Form:1 (procedure)

procedure I(P₁) =: (P₀) ; ~~xxxxxx~~ D₁ ; D₂ ; D₃ ; D_n ;
G₁ ; G₂ ; ~~xxxx~~ Σ ~~xxx~~

Form 2 (function ^{in the form of a procedure})

function I (P₁) ; D₁ ; D₂ ; D_n ;
G₁ ; G₂ ; ~~xxxx~~ Σ ~~xxx~~

Here, I is the identifier of the procedure. ~~xxxxxx~~ P represents an ordered ^{identifiers representing} list of formal input parameters, while P₀ is the list of formal output parameters, which include any exits required by the procedure. Either or both of the ~~parts~~ ^{parts} ~~(P₁) and P₀~~ may be ~~absent~~ ^{absent} in which case the notation (-) is used in place of (P).

The D₁ to D_n are ~~xxxxxx~~ groups of symbols in form similar to declarations giving complete information concerning the input and output parameters. These are to be constructed in the following manner. The group D₁ should give ~~xxxxxx~~ ^{information} for each formal variable which is not a ~~single~~ ^{simply a preliminary} variable. If necessary the declarations D₁ will use dummy ~~variables~~ ^{identifiers}. The possible types of information will be of the form:

type (I, I, I, 232)

label (F, I, ... I) $\rightarrow$ array (I, I, ... I[d:d], I, I, I[d:d],)

switch (I := (d,d,...d), I := (d,d,...d))

function (I(d,d,d,d), I(d,d,d,d))

procedure (I(d,d,d,d) =: (d,d,...d), I(d,d,...) =: (d,d, ...d))

where the d's are dummy ~~variables~~ ^{identifiers}. Again D₂ will provide ~~xxxx~~ ^{ordinary} information for all d's appearing within D₁ which are not ~~single~~ ^{variables}, written in the same form as the information for D₂, etc.

The G₁ is an optional declaration of the form

global (I, I, I)

indicating that the ~~xxxxxx~~ entities referred to in the list of identifiers will be found in the compound statement to which the procedure declaration applies, and that these entities will be used in the compound statement constituting the procedure denoted by the same identifiers.

an optional group of symbols giving information in the above manner
The G_2 is a set of declarations for other variables within the compound ~~the~~ ^{concerning} the statement. (At least one)

The operational end of the procedure must be indicated by a return statement.
The entry into the procedure is the first statement following begin. In the compound statement ~~for~~ ^{for} ~~or~~ ² form (function) a value must be assigned within the statement ~~procedure~~ by an assignment statement "I := E", where I is the identifier naming the ~~procedure~~ ^{function}.

global identifiers

~~A formal input parameter~~

~~The identifier is~~

~~The list of formal input identifiers~~

A formal input identifier may represent

- a) an ~~single~~ ordinary variable
- b) a type declared "
- c) an array
- d) a function
- e) a procedure
- f) ~~the~~

A formal output identifier may represent

- a) an ordinary variable
- b) a type declared "
- c) an array
- d) a label.
- e) a switch