Algol conference in Mainz, 14. 16. dec. 1959.

Naur

Explaining notes concerning

THE PROPOSED SYNTACTICAL STRUCTURE.

The proposed structure is based on the concept "compound statement" (which may in principle consist only of a single statement, but which will in practise always comprise several single statements between begin end). In contrast to what is the case in the syntax described in the Zürich report, it is intended that the the compound statement be given the meaning of a closed unit with respect to (a) the meaning of the identifiers used within it and (b) the dynamical succession.

This concept of the compound statement as a closed unit is connected fundamentally with the idea of classifying identifiers as being local or global with respect to it. This is best explained by an example as follows:

```
begin
. . . . . .
a:  b := c + d/e ;
. . . . . .
f: local (a, c) ;
      begin
      . . . . . .
      b: a := c/d + e ;
      . . . . . .
      go to g
      end f ;
. . . . . .
g: d := h + c/d ;
. . . . . .
end
```

The illustrating example shows a larger compound statement, comprising a smaller one. The smaller one, which will be referred to as "compound f" extends from the label f to end f

The illustrating example shows a ~~syntactic~~ program, with one compound statement, labelled f and extending to end f, contained within another. We will consider the various identifiers and their relation to the compound statement. Consider first the identifiers a, and c, which are declared to be local to the compound. The fact that they are local to the compound implies (1) that the quantities represented by these identifiers inside the compound f have no relation to quantities represented by the same identifiers appearing outside the compound f. Thus the variable a within the compound f has no relation to the label a in the compound outside, and the two d's represent different variables. A further consequence of a and c being local is (2) that the values assigned to them will only be preserved as long as the compound is not left dynamically by a go to statement (such as go to g) or by a passage through end f.

All labels are considered to be local without needing to be declared so. This means that the label b is local, so that a go to b appearing outside the inner compound f to end f would have no meaning,

This again has the important consequence

again shows that the only way to enter into the inner compound from the outer one is through the begin (either directly or through a go to f in the outer compound).

In contrast to the local identifiers, the global identifiers are such which represent the same entity

In contrast to the identifiers which are local to the compound, those which are global to it represent the same entities inside the compound and in the next following compound outside. This is the case for any identifiers not declared or understood (labels) to be local. Thus the identifiers d, e, f, g, h all represent the same variables everywhere in the example.

Since the inner compound may again contain one or more complete compound statements, and likewise the outer compound may again be contained in a still larger compound, the above relations must all be understood to be recursive.

The reasoning leading to the above concept of the compound statement and the associated concepts local and global is the following.

Complicated numerical processes tend to split up into sections which are connected to the whole of the proces through certain common variables, but which besides make use of auxiliary variables of a completely local and temporary character. Instead of forcing the programmer at any stage of the work to keep track of exactly what identifiers are available as working variables and which still refer to useful information, it is natural to allow these variables to be defined at any stage and for exactly that part of the program in which they are relevant. This obviously will mean an extra burden for the compiler and the running program, since they have to administrate the "birth" and "death" of the local quantities. This becomes particularly ackward in case of arbitrary go to statements, and it is natural to compensate this by a restriction in the allowable go to. By the restriction that all labels are local we achieve that the dynamic passage into a compound, which will be accompanied by a calculation of the storage functions of arrays, will always take place in the same manner, while only the cancellations of the local identifications can take place in connection with a go to statement.

Simplifies analysis of loop structure. Helps optimalis