Eidg. Technische Hochschule
Institut für angewandte Mathematik
Zürich
Prof. Dr. H. Rutishauser

ZÜRICH, February 26, 1960

P. Naur
Gl. Carlsbervej 2

Valby, Copenhagen

DENMARK

Dear Sir,

I thank you for your letter of February 18 which fully satisfies me.

However, I got a mimiographed letter from A.J. Perlis just at the same time and this contains among other things (which I leave to ~~him~~ you to treat) quite a new proposal in form of a comment to 4.6.6. I am opposed to this, but it shows at least that the discussion on the for statement is not quite settled. In addition I made some experience in this direction which I want to bring to your attention:

1. The writing of step and until makes the "heading" of the for statement clumsy and difficult to read, but since in publication language we can always use the brackets instead of those worddelimiters I do not care.

2. The fact that all controlled variables of for statements have to be declared is very annoying in programs with many loops. Since this cannot be changed in publication language I am somewhat concerned in this case.

3. The availability of the controlled variable after a go to out of the loop is a quite singular rule in ALGOL, and I understand that it was made for the benefit of users. In the meantime, however, I found that the need for the value of controlled variable after exit from the loop is not so very urgent, especially since it is available only after irregular exit (isn't this funny?).

I am asking therefore if it were not better and simpler to define:

"The controlled variable of the for statement is automatically local to the for statement and the corresponding declaration (real or integer according to context) is understood."

With this rule the controlled variable would be unavailable outside the for statement; if it was needed after a go to outside this had to be done as follows:

```
for K : = ....  begin
        ⋮
        S : = K ;
    go to outside ;
        ⋮
    end
```

This only inconvenience for the user would be only slight compared with the writing of the nasty declarations which have to be written now. I think therefore, that the rule mentioned above could be justified the more as it would simplify matters considerably and would lead to a very clear situation. I would not oppose to saying that the controlled variable is automatically own to the for statement and this I think would satisfy also A.J. Perlis. It is clear to me that such a last minute change will be very difficult and will cause you a lot of trouble. If for this reason or because of opposition the change is impossible, then the present form of the draft report is my second choice.

Enclosed you find a draft of the Runge-Kutta example. I beg you to check the consistency with the present definition of ALGOL (but of course not the program as such).

As soon as the agreement on your report is complete I shall make 200 copies and mail them to you.

Very sincerely yours,

H. Rutishauser

Example *)

```
procedure RK(x,y,n,FKT,eps,eta,xE,yE,first) ; value x,y ;
integer n ; Boolean first ; array y,yE ; procedure FKT ;
comment : RK integrates the system y'_k = f_k(x,y_1,y_2,...,y_n) (k=1,2,..n)
of differential equations with the method of Runge-Kutta with automa-
tic search for appropriate integration step. Parameters are :
The inital values x and y[k] for x and the unknown functions y_k(x).
The order n of the system. The procedure FKT(x,y,n,z) which represents
the system to be integrated, i.e. the set of functions f_k. The tole-
rance values eps and eta which govern the accuracy of the numerical in-
tegration. The end of the integration interval xE. The output parame-
ter yE which represents the solution at x=xE. The Boolean variable first,
which must always be given the value true for an isolated or first entry into
RK. If however the functions y must be available at several meshpoints
x_0,x_1,...,x_n, then the procedure must be called repeatedly (with x=x_k ,
xE = x_{k+1}, for k=0,1,..,n-1) and then the later calls may occur with
first = false which saves computing time. The input parameters of FKT
must be x,y,n , the output parameter z represents the set of deri-
vatives z[k] = f_k(x,y[1],y[2],..,y[n]) for x and the actual y's ;
begin
    array z,y1,y2,y3[1:n] ; real x1,x2,x3,H ; Boolean out ;
    integer k,j ; own s, Hs ;
    procedure RK1ST(x,y,h,xe,ye) ; array y,ye ;
    comment : RK1ST integrates one single RUNGE-KUTTA-step
    with initial values x,y[k] which yields the outputparameters
    xe = x+h and ye[k], the latter beeing the solution at xe.
    Important:the parameters n,FKT enter RK1ST as global entities ;
    begin
        array w[1:n], a[1:5] ; integer k,j ;
        a[1]:= a[2] := a[5] := h/2 ; a[3] := a[4] := h ;
        xe := x ;
        for k := 1 step 1 until n do  ye[k] := w[k] := y[k] ;
```

---

*) This RK-program contains some new ideas which are related to ideas of S.Gill, A process for the step by step integration of differential equations in an automatic computing machine. Proc. Camb. Phil. Soc Vol 47 (1951) p.96.
and E.Fröberg On the solution of ordinary differential equations with digital computing machines. Fysiograf. Sällsk. Lund. Förhdl. 20 Nr 11 (1950) p.136-152.
It must be clear however, that with respect to computing-time and roundoff-errors it may not be optimal, nor has it been actually been tested on a computer.

```
        for j := 1 step 1 until 4 do
        begin
            FKT(xe,w,n,z) ;
            xe := x + a[j] ;
            for k := 1 step 1 until n do
            begin
                w[k]  := y[k] + a[j]×z[k] ;
                ye[k] := ye[k] + a[j+1]×z[k]/3 ;
            end k ;
        end j ;
    end RK1ST ;
begin of program :
    if first then begin H := xE-x ;  s:= o end else H := Hs ;
    out := false
AA: if(x+2.o1×H-xE > o) ≡ (H > o) then
    begin Hs := H ; out := true ;  H := (xE-x)/2 end  if ;
    RK1ST(x,y,2×H,x1,y1) ;
BB: RK1ST(x,y,H,x2,y2)  ;   RK1ST(x2,y2,H,x3,y3) ;
    for k := 1 step 1 until n do
        if comp(y1[k],y3[k],eta) > eps  then go to CC ;
    comment : comp(a,b,c) is a procedure, the value of which is the ab-
    solute value of the difference of the mantissae of a and b, after
    the exponents of these quantities have been made equal to the lar-
    gest of the exponents of the originally given parameters a,b,c ;
    x := x3 ;  if out then  go to DD ;
    for k := 1 step 1 until n do  y[k] := y3[k] ;
    if s=5 then begin  s:= o ;  H := 2×H end if ;
    s := s+1  ;   go to AA ;
CC: H := o.5×H ; out := false ; x1 := x2 ;
    for k := 1 step 1 until n do  y1[k] := y2[k] ;
    go to BB ;
DD: for k := 1 step 1 until n do  yE[k] := y3[k]
end RK
```