

Alan Perlis

Feb 1960

The following represent my and A.Holt's comments on the report draft.

Holt is Turanski's replacement.

page/line	change	to	read
-----------	--------	----	------

✓ 4/16	recursive		iterative
--------	-----------	--	-----------

7:2.8./4 to end 2.8.

Certain of the quantities are said to possess values at some instant in program time. The value of a quantity X at time t means:

X	value of X at time t
---	----------------------

numeral	corresponding number
---------	----------------------

variable	a number at time t
----------	--------------------

array designator	a sequence of numbers at time t
------------------	---------------------------------

expression	the result of evaluating the expression at time t
------------	---

procedure designator	the result of evaluation as specified in the procedure body at time t.
----------------------	--

Not all procedures have values.

Reference to time dependance may be omitted wherever the context permits.

8:2.9	<u>delete</u>
-------	---------------

8:2.10./1 /2	<u>delete</u>
-----------------	---------------

The scope of a quantity is the block in which that quantity is declared to possess that property.

8/1-	identifier	designator
------	------------	------------

9:3.2	VALUES	DESIGNATORS
-------	--------	-------------

/1-	value	designator
-----	-------	------------

3.2.3./1	The values of procedure designators are....	
----------	---	--



which result from the application, to fixed  
sets of actual parameters, of given sets of  
rules.....The rules governing specifica-  
tion of actual  
.....analysis, which will

3.2.4./3

10:3.2.5.

/1

Transform procedures

for all occurrences of the stem 'transfer'

'read' transform .. procedures which transform  
quantities of one type into another may be  
defined. Such procedures may be called transfer  
procedures.

:3.3.1./4 value

designator

occurrences of

simple

unconditional



0:3.3.1.

Insert after 3.3.1. Table of most complex forms

$$f = A_1^{A_2}$$

$$t = f \times f \times \dots \times f$$

$$A_u = t + t + \dots + t$$

$$A_{if} = \text{if } B \text{ then } (A)$$

$$A = \text{if } B \text{ then } (A) \text{ else } A$$

Peter: the letters f,t,etc. could be printed to the left of the left margin of  
the syntax list. This will help the reader see what is going on.

11: occurrences of simple unconditional

12:3.4./5 procedure value procedure designator

13:3.4.4./2 " "

13:3.4.3./2 computing a logical value-computing a logical value -- that for  
the Boolean expression.



14: and thereafter

all occurrences

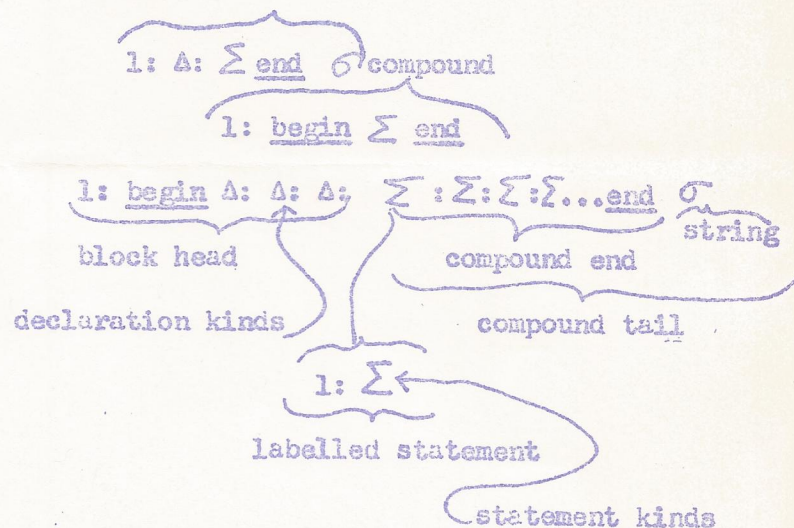
switch value

switch designator

15: 4.1.1.

Peter: the following picture will

clarify the syntax of compound statements. block



16: 4.1.3/6,7,11

entity

quantity

/10

to

in

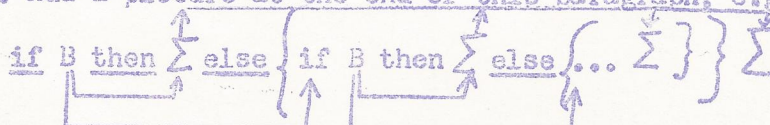
4.2.4./ all occurrences

transfer

transform

17: 4.5. Peter: for clarity add syntax definition of <statement>

18: 4.5.3.2. Add a picture at the end of this paragraph, e.g.,





19: 4.6.3/3,4

delete

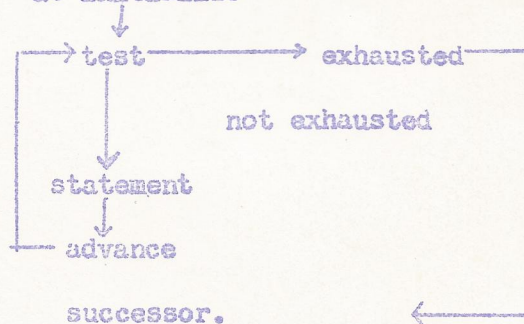
while both .....remain true.

5,6,7

replace by

A for clause is in effect a sequence of assignment statements. The picture represents such a for clause. Initialize means perform the first assignment of the for clause. Advance means perform the next assignment of the for clause. Test determines if the last assignment has been done. If so, control is transferred to the successor of the for statement. If not, the statement following the for clause is executed.

1: initialize



20:4.6.5./lff

Upon exit out of the statement S through a go statement the value of the controlled variable will be the same as it was immediately preceeding the execution of the go to statement.

4.6.6. replace

The statement covered by a for clause may be complex, and may contain go-to statements leading to some statement outside the scope of the for-clause. At this point in computation time the for-clause is not yet exhausted and, for the time being its state remains unchanged. then:



1. If the computation progresses to a new go-to statement which leads back into the compound under the for-clause, the for statement is continued just as if no interruption had taken place.
2. If the computation leads back to the beginning of the for statement it is re-initialized and begun again.
3. If the computation does not lead back to the for statement (either to its interior or its beginning) then the for statement computation remains incomplete.

4.7.3./4	the language	ALGOL
.1./1	replacement	assignment
/2	identifiers	parameters
21:/2,4.7.4./3.	4.7.5.2./1	
4.7.5.4./1,4.7.5.5./2,6	identifier	parameter
4.7.3.2	list is ...names of the	list is replaced, throughout ..., by the corresponding actual parameters,
4.7.3.3./2	finally the	finally the effect is as though....
21/4.7.5./4	a correct statement ....	<div> <div>.... as above, is executed in place .....</div> <div>procedure statement.</div> <div>a correct ALGOL statement.</div> </div>
4.7.5.3.		<p>If the formal parameter is an array identifier and called for by value, the dimension of the actual parameter must be identical to that of the formal parameter and the local array created during the call will have the same ... actual array.</p> <p>the heading of the procedure declaration).</p>
22:4.7.6	Positional	<p>relation of procedure statement to the corresponding procedure body.</p> <p>For each non-local identifier of the body of a procedure P, there must be a block in whose heading it is declared.</p>



4.7.8./3	
5.	<u>All occurrences of</u>
	identifiers
2	
/14	
/17	
/20	
23:5.1.3./2	simple variables
2	
24:5.2.4	
5.2.4.2./2	global
/4	<u>delete</u>

All procedure statements involving P must be positionally within all of these blocks. body not expressed in ALGOL code evidently...

quantities  
the program. The scope of a declaration is the block in whose heading it lies. (Two distinct quantities may have the same identifier if declared in different blocks. As control passes into a block the significance of a given identifier may consequently change.)

ment) all quantities which are declared ...block become undefined.

effect: upon a reentry into.....

All quantities of a program must be declared. Identifiers which use standard function of transform procedures will not be declared.

identifiers

arithmetic expressions any position which can be occupied by a real declared variable may be occupied by an integer declared variable.

lower bound upper bound expressions.

non-local

fixed



5.3.3./3

identifier

designator

.4./2

will be evaluated using the current value  
of all variables every time the item  
.....is referred to.

.3.5./2

For each local identifier occurring in designational expressions in a switch declaration there must be a block in whose heading it declared. All designational expressions referring to this switch declaration must be positionally within all of these blocks.

Peter: Will we not have to append a new character to the list of declarators?

Thus, do we not need an allocation constraint called array limits?

25:5.1./3,4,5

<formal identifier> <formal parameter>

/12

<specification part> <specifier>...

note: concatenate on the right

26:5.4.3./7

body to represent formal parameters.

Those formal parameters called for as  
values in the ...

/10

the values of actual parameters.

/11

which do not represent formal parameters  
will be either local or non-local...

/13

global

non-local

5.4.5/3

identifiers

parameters

Peter: against not being optional feature.

McCarthy asked me to communicate his satisfaction with the report with the  
exception of 3:/11 where we would prefer

...for stating and communicating processes.-

Otherwise we think you did a magnificent job and will go along with what you deem the best

compromise of the various entries.

A.J.Perlis,

for A.Holt and J.McCarthy