Zurich Conference on
"U n i v e r s a l l a n g u a g e"

S Y M B O L L I S T 30.5.58

Syllabels

list
I ⌣ E , E , ... E

simple variable
V ⌣ I

subscripted variable
V ⌣ I (E , E , ... E)

function
F ⌣ I [P , P , ... P]

expression and Boolean expression
E
B composition rules see "First draft of syntactical structure" and 21.)

tree variable
T ⌣ I { E ; E ; ... E }

statement label
L ⌣ I
L ⌣ G

designational expression
D ⌣ L
D ⌣ I (E)

parameters
P see 31. for possible notation
see also 34.

identifier
I ⌣ λ δ δ δ ... δ

integer
G ⌣ ξ ξ ξ

number
N ⌣ ξ ξ ξ ... ξ . ξ ... ξ ξ 10 ± G

string of symbols
S ⌣ χ χ ... χ χ where χ ≠ ▷

Quantifier statement Statements
Q ⌣ F B,
FOR V = t
E

Statements Σ

assignment statements

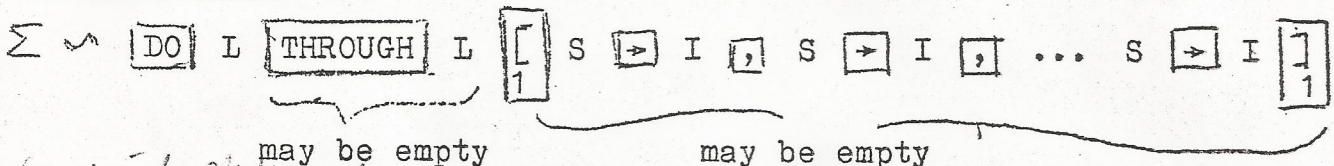
Σ ⤳ V ⟸ E , V ≥ B

~~E ⟹ V~~ F(,,):= E

GO TO -statements

Σ ⤳ GO TO D

DO -statements



may be empty may be empty

Compound statements

Σ ⤳ BEGIN Σ . Σ END

labelled statements

Σ ⤳ L : Σ

~~L : BEGIN Σ Σ ... Σ END~~

number ≥ 1

IF -statements

Σ ⤳ IF B ; 1 Σ

~~IF B ; 1 BEGIN Σ Σ Σ END~~

number ≥ 1

FOR -statements

Σ ⤳ FOR V ⟸ 1 | Σ | ~~Σ Σ ... Σ END~~

⤳ FOR V ⟸ E (3 E) 3 E , E (3 E) 3 E , ... E (3 E) 3 E | Σ |

tree assignment statements

Σ ⤳ T ⟸ B

STOP- and RETURN -statements

Σ ⤳ STOP

⤳ RETURN

procedure call statements

Σ ⤳ I [Q] ⟹ [Q]

where Q is defined like in procedure definitions

Declarations Δ

Function declaration F( ) := E

Array declaration

Δ ⌣ ARRAY I [1 , 1 ... 1]

ARRAY(1, 1, ... 1[,], 1, ... 1[ ])

Switch declaration

Δ ⌣ Switch I := (D , D , ... D)

Tree declaration

Δ ⌣ TREE I { G } , ... I { G }

Symbol classification

Δ ⌣ see ACM proposal TYPE (, 1, 1, [ ],)

Procedure declaration Π

Π ⌣ PROCEDURE I (Q) ⇒ (Q) I (Q) ⇒ (Q) ...

may be empty may be empty

I [Q] ⇒ [Q] Σ Σ ... Σ END

may be empty

where Q ⌣ P , P , P ... P , P

Basic symbols

Letters

Reference language

λ any true alphabet lower and upper

digits

$\sim$ 0 through 9

operators

$\omega \sim$	+	
	−	
	×	
	/	
	↑	
	∨	
	∧	
	¬	
	=	
	≠	
	>	
	≤	
	<	
	≥	

separators

			appears in
$\sigma \sim$	.		numbers
	,		
	⇐	:=	assign ment
	⇒	=:	assignment
	:	::	after label
	;	)	tree variabels
	1	)	if statements
	⇐|	!=	switch declar
	|⇐		for statements
	→		do statement
	≀		array declar
	17		
	)		

in PL rising $_{10}$xx to $\times 10^{xx}$

Exponent sign

if line comm., any separator in place of ;

Statement delimiter (line convention in publ)

TYPE		appears in
GOTO	GO TO	
DO		
FOR		
IF		
END		
BEGIN		
THROUGH	)	
STOP		
RETURN		
ARRAY		
SWITCH		
TREE		
PROCEDURE		
COMMENT	S,	
brackets		
β ~	↑ ↓	exponentiation
( )		expressions
⟨ ⟩	[ ]	subscripted var
[ ]	( )	functions and procedures
{ }	( )	trees
( )	( )	designational expressions
1 1	( )	switches
2 2	( )	for statements
3 3	( )	do statements
⌈ ⌉	[ ]	tree declar
⌊ ⌋		