Zeitschr. f. math. Logik und Grundlagen d. Math.
Bd. 9, S. 243—250 (1963)

ORGANIZATION OF ADDRESSLESS COMPUTERS WORKING IN PARENTHESIS NOTATION

by ZDZISŁAW PAWLAK in Warszawa (Poland)

Introduction

KALMÁR [1] and KÄMMERER [2] proposed organization of a digital computer working directly, without programing, in common used mathematical language with parenthesis. It is hoped, in this paper, to bring attention to some new solution of this problem, refering to authors ideeas of addressless computers [3, 4].

Processes

We shall be interested in this paper in exact definition of the processes performed by the computer under consideration. Notion of the process is a starting point of our investigation.

Process $\mathfrak{A} = \langle A, O, R \rangle$ is a finite set A of objects, a finite set O of operations, defined over the set A and an ordering relation R defined in O.

If the set O is well ordered by the relation R, we say that process $\mathfrak{A}$ is *sequential*, if the set O is partialy ordered by the relation R we say that process $\mathfrak{A}$ is *concurent*.

In the following, the elements of A are denoted by small Greek letters and the elements of O by capital Greek letters.

Process $\mathfrak{A}$ is *simple*, if:

1. For each $\Delta \in O$, there are three elements $L(\Delta)$, $R(\Delta)$, $V(\Delta)$, called *left argument* of the operation Δ, *right argument* of the operation Δ, and the *result* or *output* of the operation Δ respectively.
2. For each $\alpha \in A$, there is at least one operation $\Delta \in O$, such that $\alpha = L(\Delta)$ or $\alpha = R(\Delta)$ or $\alpha = V(\Delta)$.
3. There exists exactly one $\alpha \in A$, such that for none $\Delta \in O$ neither $\alpha = L(\Delta)$ nor $\alpha = R(\Delta)$ holds; α being called *final result* or *output* of the process $\mathfrak{A}$. Final result of the process $\mathfrak{A}$ will be denoted by "$\mathfrak{A}$".

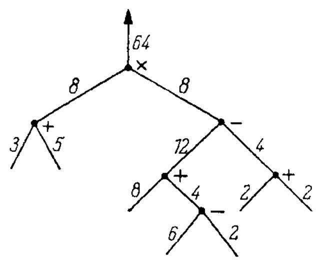
If $\alpha \in A$ and there is not such $\Delta \in O$ that $\alpha = V(\Delta)$, then α is *initial data* of the process $\mathfrak{A}$.

If $\alpha \in A$ and there are $\Delta, \Omega \in O$ such that $\alpha = V(\Omega)$ and $\alpha = L(\Delta)$ or $\alpha = R(\Delta)$ then α is called *partial result* of the process $\mathfrak{A}$.

Thus if the elements of A are natural numbers and the elements of O are arithmetical operations defined on natural numbers, such as addition, subtraction etc., then $\mathfrak{A}$ is called process of arithmetical computations.

In order to make the notion of process more evident let us observe the definition of simple process is similar to the definition of the tree in the theory of graphs. Operations are corresponding to the nodes of the tree and objects to the branches. But the difference between this two notions is in the fact that there is not the ordering relation over points in the tree. Thus the process and the tree are quite different ideas. But for the sake of simplicity we shall often use the notation of the tree instead — the process.

Example of simple process is given below.



I shall consider in this paper simple, sequential processes only.

Processes with order W and $\overline{W}$

If in the process $\mathfrak{A} = \langle A, O, R \rangle$ the set O is well ordered by the relation R, we will say that the process $\mathfrak{A}$ is ordered by the relation R, or that the process $\mathfrak{A}$ has the order R. I shall consider only two ordering relations denoted by W and $\overline{W}$.

Process $\mathfrak{A} = \langle A, O, R \rangle$ is called *partial process* of the process $\mathfrak{B} = \langle A', O', R \rangle$, in symbols $\mathfrak{A} \subset \mathfrak{B}$ if:

1. $A \subset A'$, $O \subset O'$.
2. If α, β, γ are left, right arguments and output of the operation Δ in $\mathfrak{A}$, then are also left, right arguments and output of the operation Δ in $\mathfrak{B}$.
3. For each $\alpha \in A$, if α is initial data in $\mathfrak{A}$, then α is also initial data in $\mathfrak{B}$.

If $\mathfrak{A} \subset \mathfrak{B}$, and Δ is final operation in $\mathfrak{A}$, then $\mathfrak{A}$ will be denoted by $\mathfrak{B}(\Delta)$.

Difference $\mathfrak{B} - \mathfrak{A}$ of simple processes $\mathfrak{B}$ and $\mathfrak{A}$, is process obtained from the process $\mathfrak{B}$ by removing from it all objects and operations belonging to the process $\mathfrak{A}$, except the final result of the process $\mathfrak{A}$.

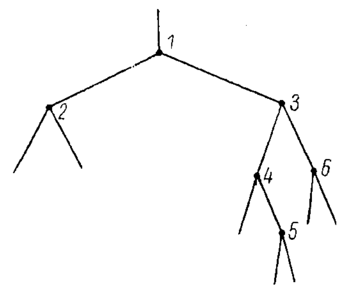
If $\alpha = V(\Delta)$ and $\alpha = L(\Omega)$ we shall writte $\Delta \to \Omega$, and if $\alpha = V(\Delta)$ and $\alpha = R(\Omega)$ we shall writte $\Delta \Rightarrow \Omega$. Now we may define the order W by such a numeration of operations, that if $\Delta, \Omega \in \mathfrak{A}$ and $\Delta \succ \Omega$, then $W(\Delta) > W(\Omega)$, where $W(\Delta), W(\Omega)$ are numbers associated with operations Δ, Ω, and $\Delta \succ \Omega$ denotes that operation Δ is performed befor the operation Ω, or the operation Ω is performed after the operation Δ.

The numeration W is as follows:

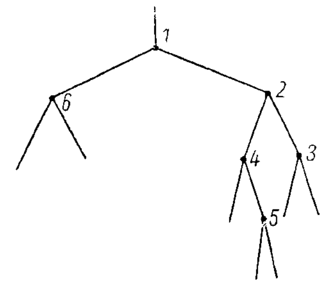
1. If Δ is final operation of the process $\mathfrak{A}$, then $W(\Delta) = 1$.
2. If $\Delta \to \Omega$, then $W(\Delta) = W(\Omega) + 1$.
3. If $\Delta \Rightarrow \Omega$, then $W(\Delta) = \operatorname{Max} W[\mathfrak{A}(\Omega) - \mathfrak{A}(\Delta)] + 1$.

$\operatorname{Max} W(\mathfrak{A})$ denotes greatest number of operation in the process $\mathfrak{A}$. Order $\overline{W}$ we obtain from the definition of order W by exchange the arrows $\to, \Rightarrow$ in points 2 and 3.

Examples of numeration W and $\overline{W}$ are given below.



Order W



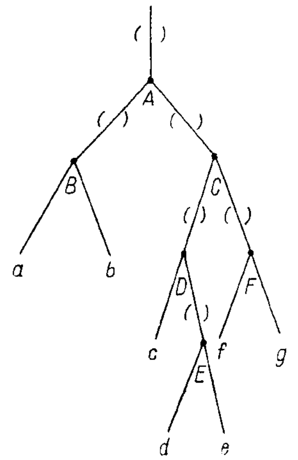
Order $\overline{W}$

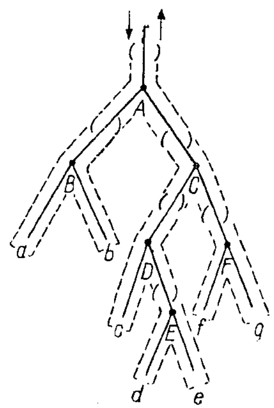
Thus the computation runs as follows:

Order W	Order $\overline{W}$
$2 + 2 = 4$	$3 + 5 = 8$
$6 - 2 = 4$	$6 - 2 = 4$
$8 + 4 = 12$	$8 + 4 = 12$
$12 - 4 = 8$	$2 + 2 = 4$
$3 + 5 = 8$	$12 - 4 = 8$
$8 \cdot 8 = 64$	$8 \cdot 8 = 64$

Programs of simple processes

Every linear representation of a tree is called *program*. Let N be the set of names of objects belonging to the set A, and let V denotes the set of names of operations belonging to the set O. Elements of V are denoted by capital Latin letters, data are denoted by small Latin letters and partial results are denoted by parenthesis as shown below.





String of symbols belonging to N and V is called program of the process if the symbols are occuring in the same order as in the diagram to the right, along the dotted line.

Thus the program of the above process is

$$((aBb)\,A\,((cD(dEc))\,C(fFg))).$$

Numeration of operations in the program may be obteined in the following way. Let us number successive left parenthesis in the program, with numbers $1, 2, \ldots, n$. With each left parenthesis number $i\,(1 \leqq i \leqq n)$, we associate function $K_i(\sigma)$, where σ denotes any symbol in the program.

Let σ' be the next symbol to the right after the symbol σ. Then the function $K_i(\sigma)$ may be defined recursively as follows:

1. $K_i(\sigma) = 1$, if σ is the left parenthesis number i;
2. $K_i(\sigma') = K_i(\sigma) + 1$, if σ' is the left parenthesis number j $(j \neq i)$, or symbol of data;
3. $K_i(\sigma') = K_i(\sigma) - 1$, if σ' is right parenthesis or symbol of operation.

It may be shown that the symbol of operation for which $K_i(\sigma) = 1$ has the number i according to the numeration W. Thus the function $K_i(\sigma)$ associates to each left parenthesis one symbol of operation with the same number as the corresponding parenthesis. Similar definition may be given for the order $\overline{W}$.

The following table gives the values of the function $K_i(\sigma)$ for the discused example.

Program	(	(	a	B	b	)	A	(	(	c	D	(	d	E	c	)	)	C	(	f	F	g	)	)	)
Number of parenthesis	1	2						3	4			5							6						
$K_1(\sigma)$	1	2	3	2	3	2	1																		
$K_2(\sigma)$		1	2	1																					
$K_3(\sigma)$								1	2	3	2	3	4	3	4	3	2	1							
$K_4(\sigma)$									1	2	1														
$K_5(\sigma)$												1	2	1											
$K_6(\sigma)$																			1	2	1				
Number of operation				2			1				4			5				3			6				

Numeration of operations according to the order W may be also obtained directly without the function $K_i(\sigma)$ in the following manner. With each left parenthesis we associate one symbol of operation. With the first left parenthesis from the right side we associate symbol of operation contained in this (and corresponding right) parenthesis. With each next left parenthesis we associate the nearest free to the right symbol of operation. Symbol of operation is free when it is not assigned to any parenthesis. Thus the above example of programm is represented now

$$\underset{1\,2}{((}aBb)\,A\,\underset{3\,4}{((}cD\underset{5}{(}dEe))\,C\underset{6}{(}fFg)))$$

where the arrows are marking relations between left parenthesis and corresponding symbols of operations.

There is also third method of determination order W of operations, called here reduction.

Let $\mathfrak{A} \subset \mathfrak{B}$. $\mathfrak{A}'$ will be called subprogram of the program $\mathfrak{B}'$, or in symbols $\mathfrak{A}' \subset \mathfrak{B}'$. Let $\mathfrak{A}'_{\mathrm{Max}}$ denote a subprogram such that the endoperation in $\mathfrak{A}'_{\mathrm{Max}}$ has the greatest number according to numeration W. If in the program $\mathfrak{B}'$, we replace subprogram $\mathfrak{A}'_{\mathrm{Max}}$ by the symbol $*$, then the resulting expression will be denoted by $\mathfrak{A}'_1$, and will be called reduced program of the rank 1. Reduced program

$\mathfrak{A}_1'$ may be reduced again, and we obtain reduced program of rank 2, $\mathfrak{A}_2'$ etc. In general $\mathfrak{A}_i'$ denotes reduced program of the rank i. $\mathfrak{A}_0'$ is unreduced program. It is easly to observe that by such definition of the operation of reduction, in the ith step of reduction the ith operation is reduced—according to the numeration W. Similar rule is holding for the order $\overline{W}$. Reduction of program given before runs as follows:

$$\begin{array}{ll} \mathfrak{A}_0' & ((aBb)\,A\,((cD(dEc))\,C(fFg))) \\ \mathfrak{A}_1' & ((aBb)\,A\,((cD(dEc))\,C^*)) \\ \mathfrak{A}_2' & ((aBb)\,A\,((cD^*)\,C^*)) \\ \mathfrak{A}_3' & ((aBb)\,A\,(^*C^*)) \\ \mathfrak{A}_4' & ((aBb)\,A^*) \\ \mathfrak{A}_5' & (^*A^*) \\ \mathfrak{A}_6' & * \end{array}$$

In the following we shall writte in the program all symbols of data together, after the parenthesis, in the same order as they are occuring in the program, so for example we have

$$[((B)\,A\,((D(E))\,C(F)))]\,(a, b, c, d, e, f, g).$$

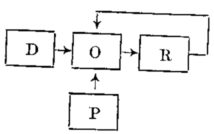
The expression in the brakets is called simplified program. Before given principles of numerations of operations, after small modification are also holding for simplified programs.

Organization of addressless computers

The addressless computer consist of:

D-data memory
P-programm memory
R-Partial results memory
O-operator
C-control

The simplified diagram of the machine is as follows:



ntrol is not shown in the figure.

Data memory is holding all data occuring in the process, in the same order as the program. Program memory P contains reduced program of the computa- 1. Partial results memory R holds all partial results during the computation. erator O is performing all operations occuring in the program. Control C is

computing function $K_i(\sigma)$ (or reduces the programm) and scanes all symbols of operations in the order W (or $\overline{W}$)—and scanes symbols of arguments corresponding to each scaned symbol of operation. The scaned symbol of operation is set in the operator O. If the scaned symbol of argument is symbol of data, proper data is taken from the memory D to the operator; if the scaned symbol of argument is symbol of partial result, as the argument the proper partial result is taken from the memory R to the operator O. Result of each operation is set in the memory R according to the following rule:

1. The first partial result is located in the location 1.
2. If both arguments of the performed operation are data, then the result of this operation should be located in successive free location.
3. If one argument of the performed operation is partial result, and one is data, then data is taken from the memory D, and as a second argument the last number written in the memory R is taken, and the result of this operation is written in the place of the number just taken.
4. If both the arguments of the performed operation are partial results as their values are taken two last numbers written in the memory R, and the result of this operation is written in the place of the last but one number in the memory R; the last number is erased.

References

[1] L. Kalmár, A practical infinistic computer, Infinitistic Methods. Proceedin of the Symposium on Foundations of Mathematics, Warsaw, 1959, pp. 347—362.
[2] W. Kämmerer, Ziffernrechenautomat mit Programmierung nach mathematischem Formelbild. Zeitschrift für Angewandte Mathematik und Mechanik **40** (1960).
[3] Z. Pawlak, Realization of Memory of Partial Results in Certain Parenthesis free Formalisms. Bull. Acad. Polon. des Sciences, Sèrie des sciences tech. IX, No. 8—1961, 487—490.
[4] Z. Pawlak, New Class of Mathematical Languages and Organization of Addressless Computers. Lecture held on the Colloquium "The foundations of mathematics, mathematical machines and their applications", Tihany (Ungary) 11—15 September 1962. To be published.

(Eingegangen am 26. November 1962)