BULLETIN DE L'ACADÉMIE
POLONAISE DES SCIENCES
Série des sciences techniques
Volume IX, No. 11 — 1961

Realization of Certain Class of Recursive Formulae in the Addressless Computer

by

Z. PAWLAK

Presented by P. SZULKIN on June 17, 1961

In computation we often have to evaluate formulae like these

$$a_0+a_1+a_2+ \dots +a_n,$$

$$a_0 \cdot a_1 \cdot a_2 \cdot \dots \cdot a_n,$$

$$a_0+x\,(a_1+x\,(a_2+x\,(\dots\, x\,(a_{n-1}+xa_n)\dots),$$

$$a_0+a_1\,(y-x_0)+a_2\,(y-x_0)\,(y-x_1)+ \dots +a_n\,(y-x_0)\,(y-x_1)\dots(y-x_{n-1}),$$

where one operation or a group of operations is applied several times to different variables.

The general scheme for this class of formulae may have the form

i. $f(0, x, y, \dots, z, a', b', \dots, c') = g\,(x, y, \dots, z, a', b', \dots, c')$,

ii. $f(n+1, x, y, \dots, z, a', b', \dots, c') = h\,(n, x, y, \dots, z, a', b', \dots, c', f(n, x, y, \dots, z, a', b', \dots, c')$.

where $x, y, \dots, z$ are arbitrary numbers and $a, b, \dots, c$ are addresses of numbers and the symbol $'$ denotes operation on addresses such that, if a is a_i, then a' is a_{i+1}, or, in other words, the operation $'$ increases the address by one, with each iteration, starting with address a_0. We may also say that $'$ denotes the taking of the value of the next variable, assuming that the considered set of variables is ordered. Thus, besides arithmetical operations we allow, in the scheme under discussion, to carry out the operations on variables which are, of course, not of the same logical type as the arithmetical operations.

Applying the commonly used symbolism, we may present the above examples as

i. $z_0 = a_0$,	i. $z_0 = a_0$,
ii. $z_{i+1} = z_i+a_i$;	ii. $z_{i+1} = z_i \cdot x+a_i$;
i. $z_0 = a_0$,	i. $z_0 = a_0$,
ii. $z_{i+1} = z_i \cdot a_i$;	ii. $z_{i+1} = z_i \cdot (y-x_i)+a_i$.

It is noteworthy to point out the dual significance of indices in the above formulae: z_i denotes the i-th value of the variable z, and a_i denotes the value of the i-th variable a_i. However, in the formula

$$a_i = b_i + c_i,$$

all the indices have the same meaning. All variables with indices denote addresses. It is also possible to interpret univocally all indices in the recursive schemes, as subsequent values of corresponding variables. Thus in this case a_i is assumed to be the i-th value of the variable a. Starting to organize the computer capable to operate with recursive formulae containing indices, it is necessary to decide which interpretation of indices will be most suitable, in order to comply with given requirements.

If the number of iterations is given in advance, indices are superfluous, and we may write, for example,

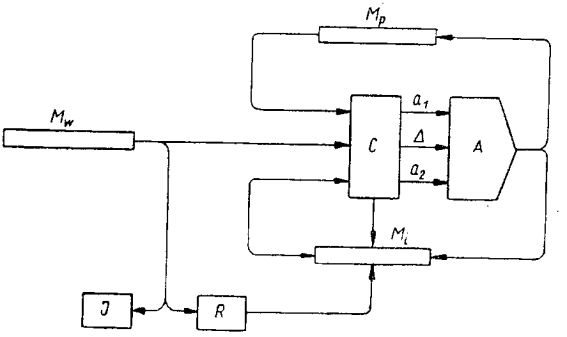
$$z = z \cdot (y - x') + a',$$

where x' and a' are variables taking with each new iteration a new, previously prescribed, value, and the sign $=$ denotes now the transfer of the final result under a given address. The substitution of an initial value for z may be written in any manner. Assuming the parenthesis-free symbolism with "blank spaces" (see, e.g. [1]), the above example will be

$$z = \pi n - yx' \cdot z* + *a',$$

where πn denotes n times repetition of the formula.

The scheme of the corresponding computer is presented in the Figure. The computer consists of a non-addressable working memory M_w, partial results memory M_p, addressable indexed variables memory M_i, arithmometer A, control circuit C, iteration counter J, and final address result register R.



Iteration counter J holds the number of iterations n; the final result is to be placed in the memory M_i according to the address in the register R. Control circuit C scanes subsequent symbols of the formula in the memory M_i, sets the arithmometer to the proper operation and selects values of both arguments according to the rule:

i. If the argument is the initial data, denoted in the formula by a variable without a dash, take the value from the memory M_w.

ii. If the argument is the initial data denoted in the formula by a variable with a dash, take its value from the memory M_t.

iii. If the argument is a partial result, take the value from the memory M_v.

If a scanned variable contains a dash, the control circuit C increases the value of the indices at this variable by one.

If there are not many variables with indices in the formulae, special counters to represent the value of the indices may be used, and in the opposite case a special place in each word in the memory M_w, to represent the value of an index, may be applied.

In the case when variables with indices are to be interpreted as subsequent values, but not addresses, the organization of the machine is similar to the one presented above with only this difference that, instead of memory M_t, a non-addressable memory is used, for example, paper tape.

INSTITUTE OF MATHEMATICS, POLISH ACADEMY OF SCIENCES
(INSTYTUT MATEMATYCZNY, PAN)

REFERENCES

[1] Z. Pawlak, *On realization of recursive schemes in the address-free computer*, Bull. Acad. Polon. Sci., Sér. sci. techn., **8** (1960), 685.