Addressless Units for Carrying Out Loop-Free Computations

ANDRZEJ BLIKLE

Polish Academy of Sciences, PKiN, Warsaw, Poland*

ABSTRACT. The paper deals with the problem of how to design a computer so that the loop-free parts of computations performed by it are handled in the most efficient and rapid manner. In particular, the question is how to design a unit to carry out loop-free computations and how to design the unit-oriented language. Clearly, the sensitive area here is the storing and accessing of intermediate results. It is assumed that the unit reads the appropriate program step-by-step, performs successive operations serially and stores the intermediate results in a sequence as the computation progresses. Now a problem arises how the unit should extract intermediate results from the sequence when needed. It is shown that the optimal solution is addressless units (the unit-oriented programs are address-free formulas) which operate in double-ended-queue disciplines, i.e. in disciplines where the actually extracted intermediate result is always either the first or the last element of the stored sequence.

The paper follows and extends some ideas given by Z. Pawlak in a number of his papers. It gives the mathematical background and extentions of Pawlak's results which have been formulated originally in a rather descriptive and informal way. The reader is not assumed to be familiar with Pawlak's papers.

KEY WORDS AND PHRASES: iteration-free computation, loop-free computation, binary-ramificated tree, linear order, formal language, Polish notation, parenthesis-free notation, double-ended-queue discipline

CR CATEGORIES: 4.10, 4.21, 5.24, 5.32

1. *Introduction*

The problem to be considered in this paper is, generally speaking, the following: how to design a computer so that the loop-free parts of programs performed by it (e.g. arithmetical or Boolean expressions) are handled in the most efficient and rapid manner. It is assumed that the computer under investigation is equipped with a special unit and that loop-free parts of programs are always translated into a unit-oriented language. Now the problem is how to design such a unit and the unit-oriented language. We assume about the unit that it performs operations serially and stores the intermediate results in a sequence as the computation progresses. Clearly we want the following conditions to be satisfied:

(1) programs (expressions) of the unit-oriented language are compact and easy to write, and

(2) the memory used by the unit to store intermediate results involves a minimal amount of computation (time).

* Computation Centre.

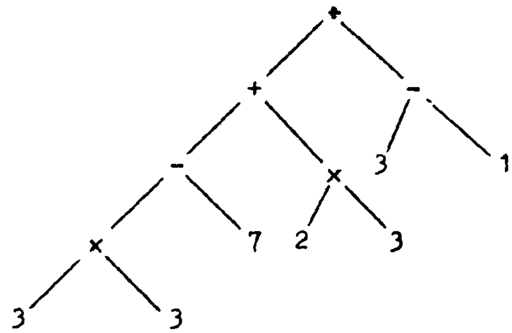

FIG. 1

Many interesting results concerning the solution of this problem have been given by Z. Pawlak [3–9]. Since the reader may not be familiar with these results, we summarize them here. Pawlak's principal ideas are the following: Every loop-free commutation has a tree structure where ramifications are labeled with symbols of operations and the "leaves" are labeled with initial arguments (data). For example, to the arithmetical expression

$$((((3 \times 3) - 7) + (2 \times 3)) \div (3 - 1)$$

corresponds the labeled tree in Figure 1. Now to perform the computation one should fix a succession of operations to be performed, i.e. one should fix an order in the set of ramifications of the tree. Clearly, not every order is possible (admissible) since, for example, in our case both multiplications, both subtractions and the addition must be performed before division, both multiplications and one subtraction must be performed before the addition, etc. Clearly, the problem of choosing an admissible order between operations is not the only one when designing a unit. The other, more difficult and subtle, is how to extract the intermediate result from the store the moment it is actually needed. Pawlak's ideas are now presented.

There is a class of admissible orders—Pawlak calls them addressless orders—that for every order ρ in this class there exists a simple algorithm $\mathfrak{A}_\rho$ which determines the location of intermediate results, provided the operations have been performed and the results have been stored in the succession given by ρ. The algorithm $\mathfrak{A}_\rho$ may be then used to design a ρ-decoder able to extract intermediate results from the store. Consider an example. One of Pawlak's addressless orders is called w (cf. Example 3 in Section 3). The w-program of the labeled tree in Figure 1 is

$$\div SS \quad +SS \quad -S7 \quad \times 33 \quad \times 23 \quad -31. \tag{1}$$

(The symbol S may be understood here as an abbreviation of "*store.*") The w-unit performs this program step by step from the right end to the left end considering in every step one subformula (called *instruction*) of the form

$$o_i\gamma_i\delta_i\,,$$

where o_i is a symbol of operation and γ_i, $\delta_i \in N \cup \{S\}$, where N is the set all numerals (names of numbers). Now, let *Top* denote the actually last-stored result (top of the store) and let x and y be two variables. Instruction $o_i\gamma_i\delta_i$ causes the following successive execution of operations:

(1) if $\gamma_i = S$, then $x := Top$ and erase Top
if $\gamma_i \in N$, then $x := [\gamma_i]$, where $[\gamma_i]$ denotes the number named γ_i,

(2) if $\delta_i = S$, then $y := Top$ and erase Top
if $\delta_j \in N$, then $y := [\delta_i]$, where $[\delta_i]$ denotes the number named δ_i,

(3) perform $[o_i](x, y)$, where $[o_i]$ denotes the operation named o_i, and put the result on top of the store.

In the case of program (1), successive states of the store are as follows:

scanned instruction	*content of the store after the instruction has been performed*
-31	2
$\times 23$	2, 6
$\times 33$	2, 6, 9
$-S7$	2, 6, 2
$+SS$	2, 8
$\div SS$	4

Note that the described w-unit realizes one of the so called double-ended-queue (or dequeue) disciplines. Double-ended-queue discipline means that intermediate results are stored in a sequence as the computation progresses and the required arguments are always taken in the same way either from the bottom or from the top of the stored sequence. For example, if we perform a binary operation $x \circ y$ and the stored sequence is $a_1, \cdots, a_n$, then we pick up the arguments in one of the following six ways: (1) $x := a_1$; $y := a_2$, (2) $x := a_2$; $y := a_1$, (3) $x := a_{n-1}$; $y := a_n$, (4) $x := a_n$; $y := a_{n-1}$, (5) $x := a_n$; $y := a_1$, (6) $x = a_1$; $y := a_n$. The way we pick up the arguments is fixed for the order. For example, the order w corresponds to (3). In general, every Pawlak order—or every addressless order as defined in this paper—corresponds to one of (1)–(6).

The present paper is concerned with the mathematical discussion of Pawlak's results. In fact, none of Pawlak's results was proved by him. In particular, he did not prove that the described ρ-units are well-defined (as, in fact, they are) or that the defined orders are the only addressless orders or the only dequeue-orders (as in fact, they are not).

Here I define formally six orders, four of which coincide with Pawlak's orders and two of which are new, and I prove them to be the only addressless orders in the binary (binary-operation) case. It is also proved that each addressless order corresponds to one dequeue discipline and conversely. The reason I deal in my investigations with the binary case is that the binary case is the most important and the simplest, and, on the other hand, it can be easily generalized to the n-ary case.

2. *Trees and Admissible Orders*

Let A be an arbitrary set and Q an arbitrary relation in A, i.e. $Q \subseteq A \times A$. By aQb we denote the fact that $(a, b) \in Q$. By the first and the second domain of Q we mean, respectively, the sets

$$D_1(Q) = \{a_1: (\exists a_2 \in A)(a_1 Q a_2)\},$$
$$D_2(Q) = \{a_2: (\exists a_1 \in A)(a_1 Q a_2)\}.$$

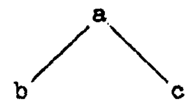

FIG. 2

Now we define the tree. This is clearly a very simple concept, but in this particular case we need more sophisticated definition, which distinguishes between the left and the right predecessor (argument) of every ramification. To this end by a tree we mean a system consisting of a finite set A of vertices and of three binary relations L, R, and $P = L \cup R$ in A called respectively the *left predecessor*, the *right predecessor* and the *predecessor*. For example, Figure 2 corresponds to the tree with $A = \{a, b, c\}$, $L = \{(a, b)\}$, $R = \{(a, c)\}$, $P = \{(a, b), (a, c)\}$. Here b and c are said to be the *left* and the *right predecessor*, respectively, of a, and a is said to be the *successor* of both b and c.

Definition 1: By a *tree* we mean a system $T = (A, L, R, P)$, where A is a finite set and L, R, and P are binary relations in A with the following properties:

(1) $P = L \cup R$.

(2) there is exactly one vertex ω in A with no successor in A; i.e., there exists an ω with $A - D_2(P) = \{\omega\}$. ω is called the *root* of T.

(3) for every a in $A - \{\omega\}$ there exists a sequence $a_1, \cdots, a_n$ in A (called the *chain* from ω to a) with $a_1 = \omega$, $a_n = a$, and $a_i P a_{i+1}$ for $i = 1, \cdots, n - 1$.

(4) $(\forall a \in D_1(P))(\exists a_1, a_2)[aLa_1 \;\&\; aRa_2]$.

(5) $D_2(L) \cap D_2(R) = \varnothing$.

(6) $(\forall a_1, a_2, a_3)[a_2Pa_1 \;\&\; a_3Pa_1 \Rightarrow a_2 = a_3]$.

(7) $(\forall a_1, a_2, a_3, a_4)[a_1La_2 \;\&\; a_1Ra_3 \;\&\; a_1Pa_4 \Rightarrow a_4 = a_2 \vee a_4 = a_3]$.

Vertices in $D_1(P)$ are called *ramifications* in T. Conditions (3) and (6) permit us to define in $D_2(P)$ the *successor function* S with

$$S(a_1) = a_2 \quad \text{iff} \quad a_2Pa_1 .$$

As it has been mentioned in the Introduction, we shall deal in the sequel with orders in sets of ramifications. To define these orders in a way uniform for all trees, we introduce now a concept of a universal tree and universal order.

Let $T^\infty = (A^\infty, L^\infty, R^\infty, P^\infty)$ be a system—fixed for the sequel of this paper—where A^∞ is an infinitely denumerable set and L^∞, R^∞, and P^∞ are binary relations in A^∞ with properties analogous to (1)–(7) in Definition 1 with the only exception that instead of (4) we have now:

$$(\forall a \in A^\infty)(\exists a_1, a_2)[aL^\infty a_1 \;\&\; aR^\infty a_2].$$

T^∞, which is clearly the maximal binary-ramificated infinite tree, will be referred to as the *universal tree*, and ω will denote always the root of T^∞.

A tree $T = (A, L, R, P)$ is said to be a *subtree* of T^∞ if

(1) $A \subseteq A^\infty$, $L \subseteq L^\infty$, $R \subseteq R^\infty$,

(2) $A - D_2(P) = A^\infty - D_2(P^\infty) = \{\omega\}$,

i.e. T is a subtree of T^∞ in a normal sense and the roots of T and T^∞ coincide.

In the sequel the term *tree* will always mean a subtree of T, and *set* (if not specified otherwise) will always mean a finite subset of A^∞.

By a *universal order* we mean any linear order in A^∞, i.e. any binary relation in A^∞ that is irreflexive, connected, transitive, and asymmetrical in A^∞.

By $[G, \rho]$, where G is a set and ρ a universal order, we mean the sequence $(g_1, \cdots, g_n)$, where $\{g_1, \cdots, g_n\} = G$ and $g_i \rho g_{i+1}$ for $i = 1, \cdots, n - 1$. The sequence $(g_1, \cdots, g_n)$ is called the *ρ-sequence associated with G.*

Consider now an arbitrary universal order ρ, an arbitrary set G and an arbitrary vertex g in G. By the *type of g in G*, in symbols $\tau_G(g)$, we mean a two-character word over the alphabet $\{S, d\}$ (S = store, d = data) defined as follows:

$$\tau_G(g) = \alpha\beta,$$

where

$$\alpha = S \quad (d), \qquad \text{if a } g_1 \text{ in } G \text{ with } gL^\infty g_1 \text{ does (does not) exist,}$$

$$\beta = S \quad (d), \qquad \text{if a } g_2 \text{ in } G \text{ with } gR^\infty g_2 \text{ does (does not) exist.}$$

By the *ρ-formula of G*, in symbols $F_\rho(G)$, we mean the string of symbols

$$\tau_G(g_1) \cdots \tau_G(g_n),$$

where $[G, \rho] = (g_1, \cdots, g_n)$.

Two sets G_1 and G_2 are said to be *ρ-similar* if $F_\rho(G_1) = F_\rho(G_2)$.

By the ρ-formula of a tree $T = (A, L, R, P)$, in symbols $F_\rho(T)$, we mean simply the ρ-formula of the set $D_1(P)$ of ramifications in T, i.e. $F_\rho(T) = F_\rho(D_1(P))$. By the *length* of $F_\rho(T)$ we mean the number of elements in $D_1(P)$.

Two trees T_1 and T_2 are said to be *ρ-similar*, if $F_\rho(T_1) = F_\rho(T_2)$.

ρ-formulas of trees, supplemented with symbols of operations, will be used in the sequel as programs to be scanned by appropriate units. As mentioned in the Introduction, these units read programs step by step from the right end to the left end. This implies the following definition of an admissible order.

Definition 2: A universal order ρ is called *admissible* if it is an extension of P, i.e. if for any a_1, a_2 in A^∞, $a_1 P^\infty a_2$ implies $a_1 \rho a_2$.

In other words admissible orders are these which guarantee that no attempt is made to perform an operation, corresponding to a ramification in the tree, until its arguments have been computed.

It is easy to see, that the family of admissible orders is not empty. Examples of such orders are given in Section 3 (cf. addressless orders in Example 3).

THEOREM 1. *For every admissible order ρ and any two trees T_1 and T_2, if T_1 and T_2 are ρ-similar, then $T_1 = T_2$.*

PROOF. Let ρ be an arbitrary admissible order and let $T_i = (A_i, L_i, R_i, P_i)$, for $i = 1, 2$, be two ρ-similar trees. Moreover, let $[D_1(P_1), \rho] = (g_1, \cdots, g_n)$ and $[D_1(P_2), \rho] = (h_1, \cdots, h_n)$. Clearly, $T_1 = T_2$ is equivalent to $D_1(P_1) = D_1(P_2)$. Thus let $D_1(P_1) \neq D_1(P_2)$. Therefore,

$$\text{neither} \quad D_1(P_1) \subseteq D_1(P_2) \quad \text{nor} \quad D_1(P_2) \subseteq D_1(P_1), \tag{2}$$

since, by similarity of T_1 with T_2, $D_1(P_1)$ and $D_1(P_2)$ have the same number of elements. On the other hand,

$$g_1 = h_1 = \omega, \tag{3}$$

since ρ is admissible. Now by (2) and (3) there exists the smallest i with

$$g_i \in D_1(P_1) - D_1(P_2).$$

Clearly, $i > 1$, therefore we can consider $S(g_i)$. Let $S(g_i) = g_k$. Since ρ is admissible, we have $k < i$; thus $g_k \in D_1(P_1) \cap D_1(P_2)$. We shall show now the equality

$$g_k = h_k. \tag{4}$$

To this end we prove, by induction on j, that $g_j = h_j$ for $j = 1, \cdots, i - 1$.

For $j = 1$ this is clearly true [see (2)]. Suppose $g_j = h_j$ for $j = 1, \cdots, m - 1$ where $m - 1 < i - 1$, and let $g_m \neq h_m$. Since $g_m \in D_1(P_2)$, then there exists an $s \leq n$ with $g_m = h_s$. Clearly, $s > m$, since in the opposite case $g_m = h_s = g_s$, which is impossible because $(g_1, \cdots, g_n)$ has no repetitions. Therefore,

$$h_{m-1} \, \rho \, h_m \, \rho \, h_s.$$

Hence, h_m is not in G, since the vertex ρ-next to $h_{m-1} = g_{m-1}$ in $D_1(P_1) \cap D_1(P_2)$ is g_m. Consider now $S(h_m) = h_p$. Clearly, $p < m$, since ρ is admissible, thus $h_p = g_p$. Hence $h_p P^\infty h_m$ and $g_p P^\infty h_m$ where $h_m \in D_1(P_2) - D_1(P_1)$, and therefore $\tau_{D_1(P_1)}(g_p) \neq \tau_{D_1(P_2)}(h_p)$ which contradicts the assumption that T_1 and T_2 are ρ-similar. Therefore $g_m = h_m$ which ends the proof by induction and hence also the proof of (4).

Since (4) is true, we have immediately $g_k P^\infty g_i$ and $h_k P^\infty g_i$ with $g_i \in D_1(P_1) - D_1(P_2)$ which implies the inequality $\tau_{D_1(P_1)}(g_k) \neq \tau_{D_1(P_2)}(h_k)$. This contradicts the assumption that T_1 and T_2 are ρ-similar. QED

Theorem 1 has an important consequence to the effect that for every admissible order ρ the function F is reversible, i.e. that for every admissible ρ we can theoretically construct a ρ-decoder (no matter how complicated) able to recognize trees when reading ρ-formulas. This ρ-decoder can be clearly used to design an appropriate ρ-unit. In general, such a ρ-unit is not very efficient, but, as is shown in Section 3, it becomes efficient if ρ is an addressless order. Let us develop this idea with more details.

Consider an arbitrary set $\boldsymbol{B}$ of objects (e.g. real or complex numbers, vectors, etc.) and an arbitrary set $\boldsymbol{O}$ of binary operations defined and with values in $\boldsymbol{B}$. Let now B and O denote some sets of names (symbols or words of a certain language) of elements in $\boldsymbol{B}$ and $\boldsymbol{O}$ respectively. Moreover for any b in B and o in O, let $[b]$ denote the object named b, and $[o]$ denote the operation named o.

By a *labeled tree* we mean any triple (T, q, p) where $T = (A, L, R, P)$ is a tree and q and p are total functions with

$$q\colon\ D_1(P) \to O,$$
$$p\colon\ A - D_1(P) \to B;$$

i.e., q associates with every ramification an operational symbol and p associates with every leave a symbol of data. Clearly, labeled trees correspond exactly to loop-free computations. An example of a labeled tree is given in Figure 1.

Now let ρ be an arbitrary order, let (T, q, p) be an arbitrary labeled tree, and let $[D_1(P), \rho] = (g_1, \cdots, g_n)$. By the ρ-*program* of (T, q, p), in symbols $\Pi_\rho(T, q, p)$, we mean the string

$$q(g_1)t(g_1) \cdots q(g_n)t(g_n),$$

where $t(g_i) = \gamma_i\delta_i$ and where, if $g_i L g^l$ and $g_i R g^r$, then

$$\gamma_i = \begin{cases} S, & \text{if } g^l \in D_1(P) \quad (g^l \text{ is a ramification}), \\ p(g^l), & \text{if } g^l \notin D_1(P) \quad (g^l \text{ is a leave}), \end{cases}$$

$$\delta_i = \begin{cases} S, & \text{if } g^r \in D_1(P) \quad (g^r \text{ is a ramification}), \\ p(g^r), & \text{if } g^r \notin D_1(P) \quad (g^r \text{ is a leave}). \end{cases}$$

Every substring of the form $q(g_i)t(g_i)$ is called an *instruction*. (1) (in Section 1) is an example of a ρ-program (for $\rho = w$) of the tree in Figure 1.

In order to describe ρ-units which operate on ρ-programs we need now an auxiliary concept. For every order ρ, let φ_ρ^L and φ_ρ^R be two functions defined as follows: for every tree $T = (A, L, R, P)$, if $[D_1(P), \rho] = (g_1, \cdots, g_n)$, then, for every $i \leq n$,

$$\varphi_\rho^L(T, i) = j \quad \text{iff} \quad g_i L g_j \quad \text{and} \quad g_j \in D_1(P),$$

$$\varphi_\rho^R(T, i) = k \quad \text{iff} \quad g_i R g_k \quad \text{and} \quad g_k \in D_1(P).$$

Consider now an arbitrary admissible order ρ. Clearly, F_ρ is reversible, thus Π_ρ is reversible too and consequently there exists a function Σ_ρ that with every ρ-program associates the corresponding unlabeled tree, i.e. that

$$\Sigma_\rho(\Pi_\rho[(T, q, p)]) = T.$$

Now we can describe the unit. We assume it is equipped with a device able to perform every operation in $\boldsymbol{O}$ and with a memory, squares numbered with successive nonnegative integers 0, 1, $\cdots$. Consider now an arbitrary admissible order ρ, an arbitrary labeled tree (T, q, p), and let

$$\sigma = o_1\gamma_1\delta_1 \cdots o_n\delta_n\rho_n$$

be the ρ-program of this tree. The unit reads σ, instruction by instruction, from the right end to the left end (i.e. from n to 1), interpreting every instruction $o_i\gamma_i\delta_i$ as follows: perform $[o_i](x, y)$ and put the result into the square number $n - i$ where

$$x = \begin{cases} \text{content of the square number } n - \varphi_\rho^L(\Sigma_\rho(\sigma), i) & \text{if } \gamma_i = S, \\ [\gamma_i], \quad \text{if } \gamma_i \in B \end{cases}$$

$$y = \begin{cases} \text{content of the square number } n - \varphi_\rho^R(\Sigma_\rho(\sigma), i) & \text{if } \delta_i = S, \\ [\delta_i], \quad \text{if } \delta_i \in B. \end{cases}$$

Example 1: Consider the labeled tree, say T, in Figure 3 and the following program of this tree:

$$\sigma = +SS \quad -S1 \quad \times 2S \quad +43 \quad \div 63.$$

Clearly, this program determines an order, say ρ, in the set of ramifications of the

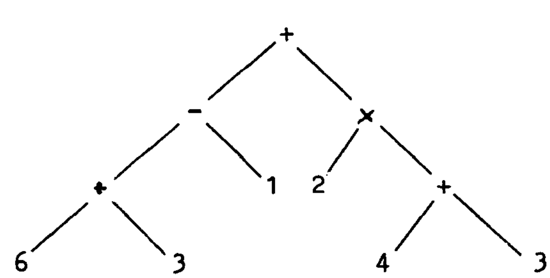

FIG. 3

TABLE I

i	scanned subformula	addresses of arguments		store					
		L	R	0	1	2	3	4	5
5	$\div$ 63	data	data	2					
4	$+$ 43	data	data	2	7				
3	$\times$ 2S	data	1	2	7	14			
2	$-$ S1	0	data	2	7	14	1		
1	$+$ SS	3	2	2	7	14	1	15	

tree. The functions φ_ρ^L and φ_ρ^R are now the following:

$$\begin{array}{ll}
\varphi_\rho^L(T, 1) = 2 & \varphi_\rho^R(T, 1) = 3 \\
\varphi_\rho^L(T, 2) = 5 & \varphi_\rho^R(T, 2) = ? \\
\varphi_\rho^L(T, 3) = ? & \varphi_\rho^R(T, 3) = 4 \\
\varphi_\rho^L(T, 4) = ? & \varphi_\rho^R(T, 4) = ? \\
\varphi_\rho^L(T, 5) = ? & \varphi_\rho^R(T, 5) = ?
\end{array}$$

where the symbol "?" is to be read as "undefined." Successive states of ρ-unit that performs σ are shown in Table I.

3. *Addressless Orders*

Efficiency and applicability of the ρ-unit described in the preceding section depends evidently on how the decoding functions

$$\psi_\rho^L(\sigma, i) = n - \varphi_\rho^L(\Sigma_\rho(\sigma), i),$$
$$\psi_\rho^R(\sigma, i) = n - \varphi_\rho^R(\Sigma_\rho(\sigma), i)$$

are complicated. Clearly, the most undesirable property of ψ_ρ^L and ψ_ρ^R is their dependence on the whole program σ, i.e. the fact that an appropriate ρ-decoder which realizes ψ_ρ^L and ψ_ρ^R have to scan at every instant the whole of σ, thus it can not be applied if σ is scanned instruction by instruction and only once. Evidently this property of ψ_ρ^L and ψ_ρ^R can make the idea of ρ-units unrealistic. On the other hand, we can ask if all admissible orders have this inconvenient property. Let us formulate this question in a precise way.

Let $\sigma = o_1\gamma_1\delta_1 \cdots o_n\gamma_n\delta_n$ be an arbitrary ρ-program (for some ρ) and for every $i \leq n$ let

$$\sigma|_i = o_i\gamma_i\delta_i \cdots o_n\gamma_n\delta_n .$$

Now we ask whether there exists an admissible order ρ, such that for every two ρ-programs σ_1 and σ_2 of length n and m respectively and for every $i \leq \min(n, m)$, if $\sigma_1|_{n-i} = \sigma_2|_{m-i}$, then

$$\psi_\rho^L(\sigma_{1,n-i}) \doteq \psi_\rho^L(\sigma_{2,m-i}),$$
$$\psi_\rho^R(\sigma_{1,n-i}) \doteq \psi_\rho^R(\sigma_{2,m-i}),$$

where $\doteq$ holds iff either both sides are undefined or both are defined and equal.

Clearly, for any admissible order ρ with the above property one can design a ρ-decoder able to determine the location of intermediate results, reading σ step by step in parallel with the unit. In other words, the decoder, in order to compute $\psi_\rho^L(\sigma, i)$ and $\psi_\rho^R(\sigma, i)$ needs to scan only $\sigma|_i$, and not the whole of σ as before.

As it turns out, admissible orders with this property exist. We call them *addressless orders*. In the sequel we shall prove the existence of exactly six addressless orders (for binary ramificated trees), we shall define all of them explicitly and show how the corresponding units can be designed.

Note now that in order to study addressless orders we need not consider labeled trees and ρ-programs, we can consider simply trees and ρ-formulas. To this end let us extend ψ_ρ^L and ψ_ρ^R to ρ-formulas in the following natural way:

Let ρ be an arbitrary admissible order, let T be an arbitrary tree with n ramifications, and let θ be the ρ-formula of T. Then, for every $i \leq n$,

$$\psi_\rho^L(\theta, i) = n - \varphi_\rho^L(F^{-1}(\theta), i),$$
$$\psi_\rho^R(\theta, i) = n - \varphi_\rho^R(F^{-1}(\theta), i).$$

Definition 3: An admissible order ρ is said to be an *addressless order* if for any two ρ-formulas θ_1 and θ_2 of length n and m, respectively, and, for any $i \leq \min(n, m)$, if $\theta_1|_{n-i} = \theta_2|_{m-i}$, then

$$\psi_\rho^L(\theta_1, n - i) \doteq \psi_\rho^L(\theta_2, m - i),$$
$$\psi_\rho^R(\theta_1, n - i) \doteq \psi_\rho^R(\theta_2, m - i).$$

Clearly, $\doteq$ means the same as previously and $\theta|_i$ is defined analogously as for ρ-programs. It is evident that the new definition of addressless order is equivalent to the previous one. On the other hand, it is more applicable in mathematical investigations. Now we shall introduce some auxiliary concepts to be used in the sequel.

Consider an arbitrary order ρ and an arbitrary set G (we recall that every set is understood as a finite subset of A^∞).

The set G is called *ρ-terminal*, if it has two following properties:

(1) if $[G, \rho] = (g_1, \cdots, g_n)$, then $\tau_G(g_1) \neq dd$ and $\tau_G(g_i) = dd$ for $i = 2, \cdots, n$,

(2) there exists a tree $T = (A, L, R, P)$ with $[D_1(P), \rho] = (h_1, \cdots, h_m)$ and $(h_{m-n+1}, \cdots, h_m) = (g_1, \cdots, g_n)$.

For example $\times 2S \quad +43 \quad \div 63$ is a subprogram that corresponds to the ρ-terminal set of the tree in Figure 3, where the order ρ is determined by the program.

Now with every admissible order ρ we associate two natural-valuated (i.e. with nonnegative integer values) functions f_ρ^L and f_ρ^R defined in the family of all ρ-terminal sets in the following way: for every ρ-terminal G if $[G, \rho] = (g_1, \cdots, g_n)$, then

$$f_\rho^L(G) = \begin{cases} i, & \text{if } g_1 L^\infty g_i, \\ 0, & \text{if the left predecessor of } g_1 \text{ is not in } G, \end{cases}$$

$$f_\rho^R(G) = \begin{cases} i, & \text{if } g_1 R g_i, \\ 0, & \text{if the right predecessor of } g_1 \text{ is not in } G. \end{cases}$$

The functions f_ρ^L and f_ρ^R are very helpful in designing an appropriate ρ-unit. For explanation consider an arbitrary tree $T = (A, L, R, P)$ with $[D_1(P), \rho] =$

$(g_1, \cdots, g_n)$ and with

$$\Pi_\rho(T) = o_1\gamma_1\delta_1 \cdots o_n\gamma_n\delta_n .$$

The unit scans $\Pi_\rho(T)$ instruction by instruction and, so long as $\gamma_i, \delta_i \in B$, it does not read in the store. Let now j be first, from the right side, (rightmost) with either $\gamma_j \notin B$ or $\delta_j \notin B$. Clearly, the set $\{g_j, \cdots, g_n\}$ is ρ-terminal; thus the position of the intermediate results corresponding to the jth step of computation can be given by f_ρ^L and f_ρ^R. In fact, if $\gamma_j = S$, then

$$\psi_\rho^L(\Pi_\rho(T), j) = n + 2 - j - f_\rho^L(\{g_j, \cdots, g_n\}),$$

and if $\delta_j = S$, then

$$\psi_\rho^R(\Pi_\rho(T), j) = n + 2 - j - f_\rho^R(\{g_j, \cdots, g_n\}).$$

Let now g_k be next, to the left of g_j, with either $\gamma_k \notin B$ or $\delta_k \notin B$. Clearly, $G = \{g_k, \cdots, g_n\}$ is not ρ-terminal, but, on the other hand, if we cancel in G the predecessors (or predecessor, if there is only one) of g_j, then the new set, say G', is ρ-terminal. Now we see that $f_\rho^L(G')$ and $f_\rho^R(G')$ determine the location of intermediate results required in the kth step of the computation, provided the previously implemented results (arguments in the jth step) have been erased and the store has been pushed down. For example, if $[G, \rho] = (g_5, g_6, g_7, g_8, g_9, g_{10}, g_{11})$, where the type of all g_i except g_5 and g_7 is dd and $g_7L^\infty g_8$ with $g_7R^\infty g_{10}$, then $G' = \{g_5, g_6, g_7, g_9, g_{11}\}$ and the store $(s_{11}, s_{10}, s_9, s_8, s_7, s_6)$ is first modified to $(s_{11}, \mathcal{E}, s_9, \mathcal{E}, s_7, s_6)$, where $\mathcal{E}$ means empty square, and then to (s_{11}, s_9, s_7, s_6).

THEOREM 2. *An admissible order ρ is an addressless order iff for any two ρ-terminal sets G_1 and G_2, if G_1 is ρ-similar with G_2, then $f_\rho^L(G_1) = f_\rho^L(G_2)$ and $f_\rho^R(G_1) = f_\rho^R(G_2)$.*

PROOF. (Necessity). Suppose ρ is an admissible order, and let there exist two trees $T_1 = (A_1, L_1, R_1, P_1)$ and $T_2 = (A_2, L_2, R_2, P_2)$ with $[D_1(P_1), \rho] = (g_1, \cdots, g_n)$ and $[D_1(P_2), \rho] = (h_1, \cdots, h_n)$, where $G_1 = \{g_{n-i}, \cdots, g_n\}$ and $G_2 = \{h_{m-i}, \cdots, h_m\}$ are ρ-terminal and ρ-similar, but either $f_\rho^L(G_1) \neq f_\rho^L(G_2)$ or $f_\rho^R(G_1) \neq f_\rho^R(G_2)$. Let $\tau_{G_1}(G_{n-i}) = \tau_{G_2}(h_{m-i}) = Sd$ (other cases are similar). Then there exist $l, p < i$ with $l \neq p$

$$g_{n-i}L^\infty g_{n-i+l}, \qquad h_{m-i}L^\infty h_{m-i+p} .$$

Therefore

$$\varphi_\rho^L(T_1, n - i) = n - i + l$$

and

$$\varphi_\rho^L(T_2, m - i) = m - i + p.$$

Let now $F_\rho(T_1) = \theta_1$ and $F_\rho(T_2) = \theta_2$. By assumption $\theta_1 |_{n-i} = \theta_2 |_{m-i}$ but, on the other hand,

$$\psi_\rho^L(\theta_1, n - i) = n - \varphi_\rho^L(F_\rho^{-1}(\theta_1), n - i) = l - i,$$
$$\psi_\rho^L(\theta_2, m - i) = m - \varphi_\rho^L(F_\rho^{-1}(\theta_2), m - i) = p - i,$$

hence ρ is not an addressless order.

(Sufficiency). Let ρ be an admissible order and let for any two ρ-terminal and ρ-similar sets G_1 and G_2, $f_\rho^L(G_1) = f_\rho^L(G_2)$ and $f_\rho^R(G_1) = f_\rho^R(G_2)$. We shall prove the following thesis:

For any two trees T_1 and T_2 with n ramifications and m ramifications, respectively, if $F_\rho(T_1) = \theta_1$ and $F_\rho(T_2) = \theta_2$, then for every $i \leq \min(n, m)$, if $\theta_1 |_{n-i} = \theta_2 |_{m-i}$, then

$$\psi_\rho^L(\theta_1, n - i) \doteq \psi_\rho^L(\theta_2, m - i) \quad and \quad \psi_\rho^R(\theta_1, n - i) \doteq \psi_\rho^R(\theta_2, m - i).$$

This thesis will be proved by induction on $n + m$. Let $n + m = 2$, i.e., $n = m = 1$. Then $F_\rho(T_1) = F_\rho(T_2) = dd$ and the thesis is evidently true.

Suppose the thesis is true for any $n + m < k$ and consider two trees $T_1 = (A_1, L_1, R_1, P_1)$ and $T_2 = (A_2, L_2, R_2, P_2)$ with $[D_1(P_1), \rho] = (g_1, \cdots, g_n)$ and $[D_1(P_2), \rho] = (h_1, \cdots, h_m)$, where $n + m = k$. Let now $i \leq \min(n, m)$ and let

$$\theta_1 |_{n-i} = \theta_2 |_{m-i}.$$

If $G_1 = \{g_{n-i}, \cdots, g_n\}$ and $G_2 = \{h_{m-i}, \cdots, h_m\}$ are ρ-terminal, then $f_\rho^L(G_1) = f_\rho^L(G_2)$ and $f_\rho^R(G_1) = f_\rho^R(G_2)$ and the thesis is clearly satisfied. Thus let G_1 and G_2 be non-ρ-terminal. Thus there exists k such that $G_3 = \{g_{n-i+k}, \cdots, g_n\}$ and $G_4 = \{h_{m-i+k}, \cdots, h_m\}$ are ρ-terminal. Let $\tau_{G_3}(g_{n-i+k}) = \tau_{G_4}(h_{m-i+k}) = Sd$ (other cases are analogous). Since G_3 and G_4 are clearly ρ-similar, $f_\rho^L(G_3) = f_\rho^L(G_4)$, i.e.

$$g_{n-i+k} L^\infty g_{n-p} \quad \text{and} \quad h_{m-i+k} L^\infty h_{m-p}$$

for some p with $p < k - i$. Note now that

$$(g_1, \cdots, g_{n-p-1}, g_{n-p+1}, \cdots, g_n)$$

and

$$(h_1, \cdots, h_{m-p-1}, h_{m-p+1}, \cdots, h_m)$$

are ρ-strings of ramifications of two trees say T_1^* and T_2^* with

$$F_\rho(T_1^*) |_{n-i-1} = F_\rho(T_2^*) |_{m-i-1}.$$

Hence, by the inductive assumption,

$$\psi_\rho^L(F_\rho(T_1^*), n - i - 1) \doteq \psi^L{}_\rho(F_\rho(T_2^*), m - i - 1)$$

and

$$\psi_\rho^R(F_\rho(T_1^*), n - i - 1) \doteq \psi_\rho^R(F_\rho(T_2^*), m - i - 1).$$

Therefore we have immediately

$$\psi_\rho^L(\theta_1, n - 1) \doteq \psi_\rho^L(\theta_2, m - i)$$

and

$$\psi_\rho^R(\theta_1, n - i) \doteq \psi_\rho^R(\theta_2, m - i)$$

which completes the proof of the thesis. Hence ρ is an addressless order. QED

Theorem 2 shows that, if we design a ρ-unit with ρ being an addressless order, we can base on f_ρ^L and f_ρ^R instead of ψ_ρ^L and ψ_ρ^R. Now we shall show how f_ρ^L and f_ρ^R behave for addressless orders. In particular, we show that every addressless order corresponds to some dequeue discipline (see Section 1).

THEOREM 3. *An admissible order ρ is an addressless order iff one of the following* *six conditions is satisfied:*

(*1*) *for every* ρ*-terminal set* G, *with* $[G, \rho] = (g_1, \cdots, g_n)$,
$$f_\rho^L(G) = 2, \qquad \text{if} \quad \tau_G(g_1) = Sd,$$
$$f_\rho^R(G) = 2, \qquad \text{if} \quad \tau_G(g_1) = dS,$$
$$f_\rho^L(G) = 2 \quad \text{and} \quad f_\rho^R(G) = 3, \qquad \text{if} \quad \tau_G(g_1) = SS;$$

(*2*) *for every* ρ*-terminal set* G, *with* $[G, \rho] = (g_1, \cdots, g_n)$,
$$f_\rho^L(G) = 2, \qquad \text{if} \quad \tau_G(g_1) = Sd,$$
$$f_\rho^R(G) = 2, \qquad \text{if} \quad \tau_G(g_1) = dS,$$
$$f_\rho^L(G) = 3 \quad \text{and} \quad f^R(G) = 2, \qquad \text{if} \quad \tau_G(g_1) = SS;$$

(*3*) *for every* ρ*-terminal set* G, *with* $[G, \rho] = (g_1, \cdots, g_n)$,
$$f_\rho^L(G) = n, \qquad \text{if} \quad \tau_G(g_1) = Sd,$$
$$f_\rho^R(G) = n, \qquad \text{if} \quad \tau_G(g_1) = dS,$$
$$f_\rho^L(G) = n \quad \text{and} \quad f_\rho^R(G) = n - 1, \qquad \text{if} \quad \tau_G(g_1) = SS;$$

(*4*) *for every* ρ*-terminal set* G, *with* $[G, \rho] = (g_1, \cdots, g_n)$,
$$f_\rho^L(G) = n, \qquad \text{if} \quad \tau_G(g_1) = Sd,$$
$$f_\rho^R(G) = n, \qquad \text{if} \quad \tau_G(g_1) = dS,$$
$$f_\rho^L(G) = n - 1 \quad \text{and} \quad f_\rho^R(G) = n, \qquad \text{if} \quad \tau_G(g_1) = SS;$$

(*5*) *for every* ρ*-terminal set* G, *with* $[G, \rho] = (g_1, \cdots, g_n)$,
$$f_\rho^R(G) = 2, \qquad \text{if either} \quad \tau_G(g_1) = Sd \quad \text{or} \quad \tau_G(g_1) = SS,$$
$$f_\rho^R(G) = n, \qquad \text{if either} \quad \tau_G(g_1) = dS \quad \text{or} \quad \tau_G(g_1) = SS;$$

(*6*) *for every* ρ*-terminal set* G, *with* $[G, \rho] = (g_1, \cdots, g_n)$,
$$f_\rho^L(G) = n, \qquad \text{if either} \quad \tau_G(g_1) = Sd \quad \text{or} \quad \tau_G(g_1) = SS,$$
$$f_\rho^R(G) = 2, \qquad \text{if either} \quad \tau_G(g_1) = dS \quad \text{or} \quad \tau_G(g_1) = SS.$$

PROOF. Sufficiency is obvious on account of Theorem 2. Necessity will be proved with the help of the following three lemmas.

LEMMA 1. *Let* ρ *be an arbitrary addressless order and* G *be a* ρ*-terminal set with* $[G, \rho] = (g_1, \cdots, g_n)$. *Let* G^* *consist of all elements in* G *except* g_1 *and both—or one, if there is only one—predecessors of* g_1. *If* $g_1 P^\infty g_i$, *then*

$$\text{either} \qquad g_i \rho g \quad \text{for all} \quad g \text{ in } G^* \qquad \text{or} \qquad g \rho g_i \quad \text{for all} \quad g \text{ in } G^*.$$

PROOF. Suppose that this is not the case, i.e. that $g_1 P^\infty g_i$ and there exist g_j, g_k in G with $j < i < k$. It is easy to see now that the sets

$$G_1 = \{g_1, \cdots, g_{j-1}, g_{j+1}, \cdots, g_n\}$$

and

$$G_2 = \{g_1, \cdots, g_{k-1}, g_{k+1}, \cdots, g_n\}$$

are both ρ-terminal and ρ-similar, but, on the other hand, if $g_1 L^\infty g_i$, then

$$f_\rho^L(G_1) = i - 1 \qquad \text{and} \qquad f_\rho^L(G_2) = i$$

and if $g_1 R^\infty g_i$, then

$$f_\rho^R(G_1) = i - 1 \qquad \text{and} \qquad f_\rho^R(G_2) = i,$$

which contradicts Theorem 2. QED

As a consequence of this lemma we immediately conclude the following:

COROLLARY 1. *For any addressless order ρ and any ρ-terminal set G, with $[G, \rho] = (g_1, \cdots, g_n)$,*

(1) *if g_i is the only predecessor of g_1 in G, then either $i = 2$ or $i = n$,*

(2) *if g_i, g_k are both predecessor of g_1 in G, then either $i = 2$ and $k = 3$, or $i = 3$ and $k = 2$, or $i = n - 1$ and $k = n$, or $i = n$ and $k = n - 1$, or $i = 2$ and $k = n$, or $i = n$ and $k = 2$.*

LEMMA 2. *For every addressless order ρ and for any two ρ-terminal sets G_1 with $[G_1, \rho] = (g_1, \cdots, g_n)$ and G_2 with $[G_2, \rho] = (h_1, \cdots, h_m)$ if g_1 and h_1 are both of the same type Sd or dS and $g_1P^\infty g_i$ with $h_1P^\infty h_j$, then*

$$\text{either} \quad i = 2 \quad \text{and} \quad j = 2 \quad \text{or} \quad i = n \quad \text{and} \quad j = m.$$

PROOF. Consider the case where $\tau_{G_1}(g_1) = \tau_{G_2}(h_1) = Sd$ and let $g_1L^\infty g_2$ and $h_1L^\infty h_m$ (cf. Corollary 1). Let moreover $m \geq n$. As it is easy to see, the sets G_1 and $G_2^* = \{h_1, h_{2+m-n}, \cdots, h_m\}$ are ρ-terminal and ρ-similar but

$$f_\rho^L(G_1) = 2 \qquad \text{and} \qquad f_\rho^L(G_2) = n,$$

which contradicts Theorem 2. QED

LEMMA 3. *Let ρ be an addressless order. If for any ρ-terminal set G, with $[G, \rho] = (g_1, \cdots, g_n)$,*

$$f_\rho^L(G) = 2 \qquad \text{for} \quad \tau_G(g_1) = Sd$$

and

$$f_\rho^R(G) = 2 \qquad \text{for} \quad \tau_G(g_1) = dS,$$

then one of the following cases is possible:

(1) *for every ρ-terminal G, with $[G, \rho] = (g_1, \cdots, g_n)$ and with $\tau_G(g_1) = SS$,*
$$f_\rho^L(G) = 2 \qquad \text{and} \qquad f_\rho^R(G) = 3;$$

(2) *for every ρ-terminal G, with $[G, \rho] = (g_1, \cdots, g_n)$ and with $\tau_G(g_1) = SS$,*
$$f_\rho^L(G) = 3 \qquad \text{and} \qquad f_\rho^R(G) = 2.$$

PROOF. Suppose that this is not the case, i.e. there exists a ρ-terminal set G with $[G, \rho] = (g_1, \cdots, g_n)$ and with $\tau_G(g_1) = SS$ where (cf. Corollary 1) either $f_\rho^L(G) = n$ and $f_\rho^R(G) = n - 1$ or $f_\rho^L(G) = n - 1$ and $f_\rho^R(G) = n$ or $f_\rho^L(G) = 2$ and $f_\rho^R(G) = n$ or $f_\rho^L(G) = n$ and $f_\rho^R(G) = 2$. Let the first be true. Then $G_1 = \{g_1, \cdots, g_{n-1}\}$ is ρ-terminal with $\tau_{G_1}(g_1) = dS$ and $f_\rho^R(G_1) = n - 1$ which contradicts the assumption that $f_\rho^R(G_1) = 2$. The other cases are similar. QED

Lemma 3 implies (1) and (2) of Theorem 3. It is easy to see that analogous lemmas can be proved for (3), (4), and (5). This completes the proof of Theorem 3.

By Theorem 3 the class of all addressless orders (if nonempty) can be divided into six disjoint classes of orders corresponding to the six dequeue disciplines. It is interesting to know if the six classes are nonempty and if they are one-or-more-element sets. The answer to the first question is positive and will be proved later. As to the second question, we shall show now that to every dequeue discipline corresponds at most one admissible order. In fact, we prove a more general theorem:

THEOREM 4. *Let ρ and ξ be two admissible orders. If for every set G which is simul-*

taneously ρ-terminal and ξ-terminal

$$f_\rho^L(G) = f_\xi^L(G) \qquad \text{and} \qquad f_\rho^R(G) = f_\xi^R(G),$$

then $\rho = \xi$.

PROOF. Let ρ and ξ be admissible orders which meet the assumption of the theorem. We shall show that ρ and ξ coincide in the set of ramifications of every tree. We prove this by induction on n, the number of ramifications in the tree.

For $n = 1$ the thesis is evidently true. Suppose it is true for any $n < k$, and let $T = (A, L, R, P)$ be a tree with

$$[D_1(P), \rho] = (g_1, \cdots, g_k).$$

Now let

$$G = \{g_j, \cdots, g_k\}$$

be ρ-terminal (i.e. j is the largest number with $\tau_{D_1(P)}(g_j) \neq dd$) and let $g_j P^\infty g_i$. Evidently, $j < i$ (since ρ is admissible), thus $\tau_{D_1(P)}(g_i) = dd$ and therefore

$$D = D_1(P) - g_i$$

is also the set of ramifications of some tree. Hence, by the inductive assumption,

$$[D, \rho] = [D, \xi] = (g_1, \cdots, g_{i-1}, g_{i+1}, \cdots, g_k).$$

Consequently, G is ξ-terminal since $g_j \xi g_i$ (ξ is admissible) and therefore, by the assumption of the theorem

$$f_\rho^L(G) = f_\xi^L(G) \qquad \text{and} \qquad f_\rho^R(G) = f_\xi^R(G).$$

Hence,

$$[G, \xi] = (g_j, \cdots, g_k),$$

and thus

$$[D_1(P), \xi] = (g_1, \cdots, g_k) = [D_1(P), \rho]. \qquad \text{QED}$$

By Theorems 3 and 4 there exist at most six addressless orders—at most one for each discipline. Now we shall complete this result by proving the existence of exactly six addressless orders, exactly one for each dequeue discipline. All the orders will be defined explicitly. First some auxiliary notions.

Let g_1 and g_2 be two vertices in A^∞. g_1 is said to be *left to* g_2, in symbols $g_1 \lambda g_2$, if there exist g_0, g_3 and g_4 in A^∞ with the following properties:

(1) $g_0 L^\infty g_3$ and $g_0 R^\infty g_4$,
(2) either $g_3 = g_1$ or there exists a chain (see Definition 1) from g_3 to g_1,
(3) either $g_4 = g_4$ or there exists a chain from g_4 to g_2.

g_1 is said to be *right to* g_2, in symbols $g_1 \lambda^* g_2$, if $g_2 \lambda g_1$.

g_1 is said to be *upwards of* g_2, in symbols $g_1 \beta g_2$, if the chain from ω (the root of the universal tree T^∞) to g_1 is shorter than the chain from ω to g_2.

g_1 is said to be *on the level* with g_2, in symbols $g_1 \alpha g_2$, if neither $g_1 \beta g_2$ nor $g_2 \beta g_1$.

An infinite sequence $g_1, g_2, \cdots$ of vertices is called a *left branch* if

(1) $g_1 \in D_2(R^\infty) \cup \{\omega\}$,
(2) $g_i L^\infty g_{i+1}$ for $i = 1, 2, \cdots$.

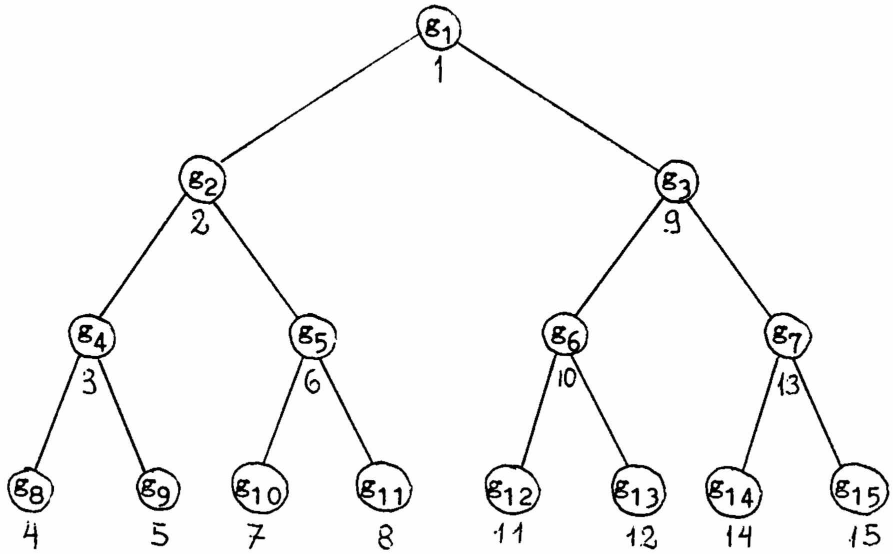

FIG. 4

g_1 is called the *head* of the branch, and the left branch with a head g is denoted by $B(g)$.

An infinite sequence $g_1, g_2, \cdots$ of vertices is called a *right branch* if

(1) $g_1 \in D_2(L^\infty) \cup \{\omega\}$,
(2) $g_i R g_{i+1}$ for $i = 1, 2, \cdots$.

g_1 is called the *head* of the branch and the right branch with a head g is denoted by $B^*(g)$.

As it follows therefore, no vertex except the root ω can be the head of both a left and a right branch.

Let ρ be an arbitrary order and let $B(g_1)$ and $B(g_2)$ be two left branches. By $B(g_1)\ \rho\ B(g_2)$ we mean that $b_1\ \rho\ b_2$ for any b_1 in $B(g_1)$ and any b_2 in $B(g_2)$. Analogously we define the meaning of $B^*(g_1)\ \rho\ B^*(g_2)$.

Example 2: Consider the tree in Figure 4. The vertices $g_2, g_4, g_5, g_8, g_9, g_{10}, g_{11}$ are left to g_3. The vertex g_3 is upwards of g_4–g_{15}. The string (g_1, g_2, g_4, g_8) is an initial segment of a left branch, but (g_2, g_4, g_8) is not since $g_2 \notin D_2(R^\infty) \cup \{\omega\}$. The string (g_3, g_6, g_{12}) is also an initial segment of a left branch, and (g_1, g_3, g_7, g_{15}) and (g_2, g_5, g_{11}) are both initial segments of right branches. The vertices g_4–g_7 are all on the same level.

Now we are ready to define all addressless orders. We denote them by p, w, v, p^*, w^*, and v^*, where p, w, and v are called *simple*, and p^*, w^*, and v^* are called *dual* orders. It should be emphasized here that p, w, p^*, w^* have been discovered by Pawlak [3–9].

Definition 4: (*simple horizontal order* p)

$$(\forall g_1, g_2)[g_1 p g_2 \Leftrightarrow g_1 \beta g_2 \lor (g_1 \alpha g_2 \ \&\ g_1 \lambda^* g_2)],$$

i. e. $g_1 p g_2$ iff either g_1 is upwards of g_2 or g_1 is on the level with g_2 and g_1 is right to g_2.

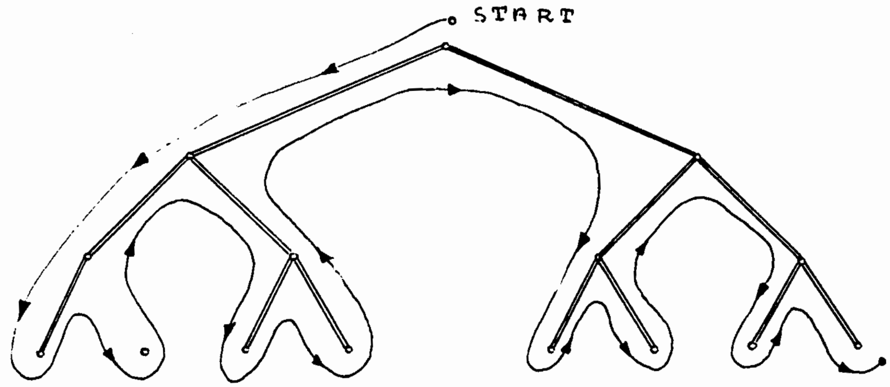

FIG. 5

Definition 5: (*simple vertical order w*)

$$(\forall g_1, g_2)\ [g_1 w g_2 \Leftrightarrow g_1 \lambda g_2 \ \vee\ (\sim g_1 \lambda g_2\ \&\ \sim g_1 \lambda^* g_2\ \&\ g_1 \beta g_2).$$

i. e. $g_1 w g_2$ iff either g_1 is left to g_2 or g_1 and g_2 belong to the same chain and g_1 is upwards of g_2.

Definition 6: (*simple branch-along order v*) By v we mean the smallest binary relation in A^∞ (i.e. the smallest subset of $A^\infty \times A^\infty$) with the following three properties:

(1) $(\forall g_1, g_2)\ [g_1 L^\infty g_2 \Rightarrow g_1 v g_2]$,
(2) $(\forall g \in D_2(R^\infty))\ [B(\omega)\ v\ B(g)]$,
(3) $(\forall g_1, g_2 \in D_2(R^\infty))\ [B(g_1) v B(g_2) \Leftrightarrow S(g_1) v S(g_2)]$.

We have defined here simple orders. To define the dual orders p^*, w^*, and v^* one needs to replace in Definitions 4, 5, and 6, respectively, all symbols λ by λ^* and conversely, all symbols L^∞ by R^∞ and conversely, and all symbols B by $B.^*$

Example 3: Consider the tree in Figure 4. The succession of indices corresponds to p^*, i.e. $(g_1, \cdots, g_{15})$ is a p^*-string. The external enumeration corresponds to w. The way we run over the vertices of a tree when enumerating them in w is shown in Figure 5. An example of v is given in Figure 6. The way we run over the vertices of a tree enumerating them in v is shown in Figure 7. Note that in this running over, only "right edges" can be cut short. Moreover, every edge may be cut only once and only if we meet it for the first time. Note that for better explanation we have numbered all vertices of trees. Clearly for program-writing purposes we number only ramifications.

We shall prove now that p, w, v, p^*, w^*, and v^* are addressless orders.

THEOREM 5.

(1) w satisfies (1) of Theorem 3,
(2) w^* satisfies (2) of Theorem 3,
(3) p satisfies (3) of Theorem 3,
(4) p^* satisfies (4) of Theorem 3,
(5) v satisfies (5) of Theorem 3,
(6) v^* satisfies (6) of Theorem 3.

PROOF. We shall consider only simple orders. The proofs for dual orders are similar.

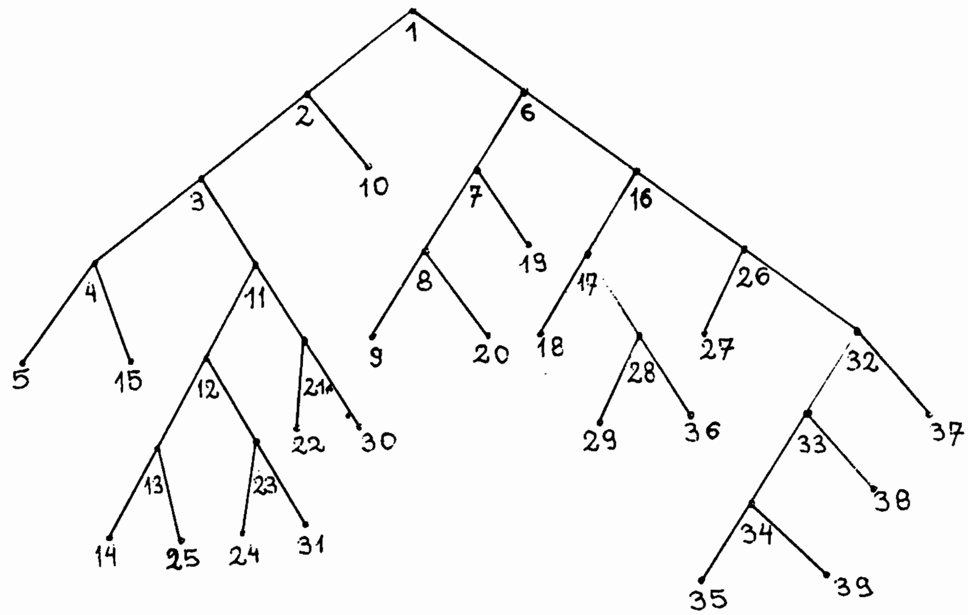

FIG. 6

ad (1): Let G be a w-terminal set with $[G, \rho] = (g_1, \cdots, g_n)$. By Definition 4, for any g_1 and g_2 in A^∞, if $g_1 L^\infty g_2$, then $g_1 w g_2$ and there is no g_3 with $g_1 w g_3 w g_2$. Hence, if $\tau_G(g_1) = Sd$ or $\tau_G(g_1) = SS$, then $f_w{}^L(G) = 2$. Let now $g_1 R^\infty g_i$ for some g_i in G. Since G is w-terminal, there is no vertex in $G - \{g_1\}$ that is simultaneously upwards of, and on the same chain with g_i, and there is no vertex left to g_i except eventually g_2, if $g_1 L^\infty g_2$. Hence, if $\tau_G(g_1) = dS$, then $f_w{}^R(G) = 2$; and if $\tau_G(g_1) = SS$, then $f_w{}^R(G) = 3$.

ad (3): Note first that, for any a_1 and a_2 in A^∞, if $S(a_1) \neq S(a_2)$, then

$$a_1 p a_2 \quad \text{iff} \quad S(a_1) p S(a_2).$$

Consider now a p-terminal set G with $[G, p] = (g_1, \cdots, g_n)$ and let $g_1 L^\infty g_i$ with $i < n$. Clearly, $g_1 R^\infty g_n$ does not hold, since in the converse case $g_n p g_i$ which is not true. Therefore, $S(g_i) p S(g_n)$, i.e.

$$g_1 p S(g_n). \tag{5}$$

Now let $T = (A, L, R, P)$ be a tree with $[D_1(P), p] = (h_1, \cdots, h_m)$, where $(h_{m-n+1}, \cdots, h_m) = (g_1, \cdots, g_n)$. Clearly, $S(g_n) \in D_1(P)$; thus—by (5)—$S(g_n) \in G$ and $S(g_n) \neq g_1$, which is clearly impossible since G is p-terminal. Hence $i = n$.

In the same way we prove other equalities of (3) in Theorem 3.

ad (5): Let G be a v-terminal set with $[G, v] = (g_1, \cdots, g_n)$. If $f_v{}^L(G) \neq 0$, then the equality $f_v{}^L(G) = 2$ can be proved in the same way as for w. Note now that, for any a_1, and a_2 in $D_2(R^\infty)$,

$$a_1 v a_2 \quad \text{iff} \quad S(a_1) v S(a_2).$$

In consequence, if $f_v{}^R(G) \neq 0$, then the equality $f_v{}^R(G) = n$ can be proved in the same way as for p. QED

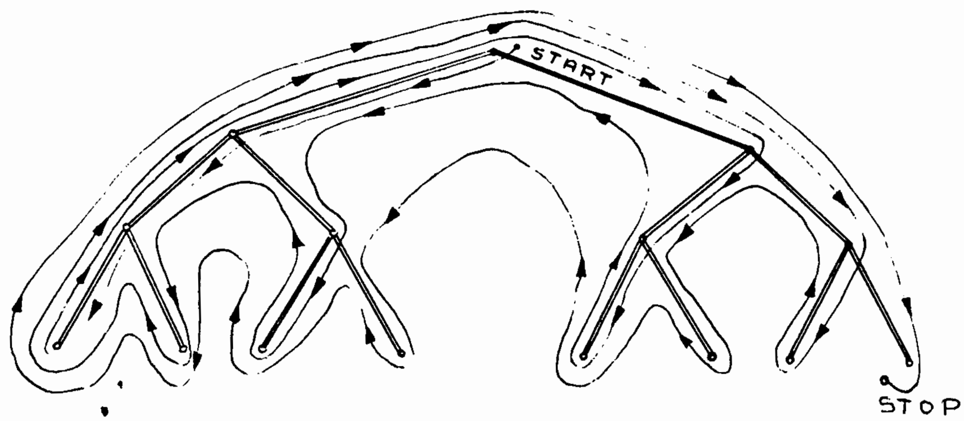

FIG. 7

4. *Units with Addressless Orders*

In Section 2 we have described a ρ-oriented unit which operates on ρ-programs of labeled trees and in Section 3 we have shown how the functions f_ρ^L and f_ρ^R can be interpreted and applied in designing an appropriate unit. In fact, if *read* is interpreted as *read and erase* and *erase* is always followed by *push down the store* (in order to make the stored sequence compact and starting at the square number 0), then f_ρ^L and f_ρ^R describe at every moment the location of required intermediate results. Now Theorems 3 and 5 can be applied in designing units which correspond to addressless orders. We shall consider only simple orders since dual are similar.

Let the unit under investigation be equipped with:

(1) device able to perform all operations of a set of operations (cf. Section 2),

(2) store for intermediate results where cells are numbered with successive nonnegative integers, and

(3) two registers x and y for storing actual arguments.

Now, let *Bot* denote the content of the cell number 0 (bottom of the store) and let *Top* denote the content of the rightmost nonempty cell on the store (top of the store). Suppose now a particular ρ-unit is scanning a ρ-program

$$\sigma = o_1\gamma_1\delta_1 \cdots o_n\gamma_n\delta_n .$$

Accordingly ρ is either w or p or v every instruction $o_i\gamma_i\delta_i$ effects the following succession of operations: For the *simple vertical order* w:

(1) if $\gamma_i = S$, then $x := Top$ and erase *Top*,
if $\gamma_i \neq S$, then $x := [\gamma_i]$;

(2) if $\delta_i = S$, then $y := Top$ and erase *Top*,
if $\delta_i \neq S$, then $y := [\delta_i]$;

(3) perform $[o_i](x, y)$ and put the result on the top of the store.

For the *simple horizontal order* p:

(1) if $\gamma_i = S$, then $x := Bot$, erase *Bot*, and push down the store,
if $\gamma_i \neq S$, then $x := [\gamma_i]$;

(2) if $\delta_i = S$, then $y := Bot$, erase *Bot* and push down the store,
if $\delta_i \neq S$, then $y := [\delta_i]$;

(3) perform $[o_i](x, y)$ and put the result on the top of the store.

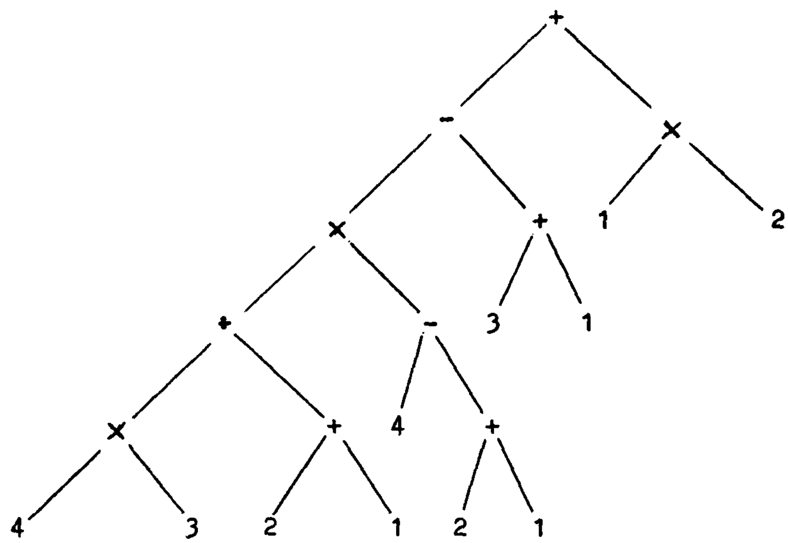

FIG. 8

TABLE II

$o_i\ \gamma_i\ \delta_i$	store							
	x	*y*	*0*	*1*	*2*	*3*	*4*	*5*
× 12	1	2	2					
+ 31	3	1	2	4				
+ 21	2	1	2	4	3			
− 4S	4	3	2	4	1			
+ 21	2	1	2	4	1	3		
× 43	4	3	2	4	1	3	12	
+ SS	12	3	2	4	1	4		
× SS	4	1	2	4	4			
− SS	4	4	2	0				
+ SS	0	2	2					

For the *simple branch-along order* v:

(1) if $\gamma_i = S$, then $x := Top$ and erase *Top*,
if $\gamma_i \neq S$, then $x := [\gamma_i]$;

(2) if $\delta_i = S$, then $y := Bot$ erase *Bot* and push down store,
if $\delta_i \neq S$, then $y := [\delta_i]$

(3) perform $[o_i](x, y)$ and put the result on the top of the store.

Example 4: Consider the labeled tree in Figure 8. The w-program of this tree is following:

$$+SS \quad -SS \quad \times SS \quad \div SS \quad \times 43 \quad +21 \quad -4S \quad +21 \quad +31 \quad \times 12.$$

Successive states of w-unit which performs this program are shown in Table II.

5. *Syntax of ρ-Programs and ρ-Formulas*

Let $\mathfrak{T}$ denote the set of all labeled trees and let, for every admissible order ρ,

$$\mathcal{P}_\rho = \Pi_\rho(\mathfrak{T}),$$

i.e. $\mathcal{P}_\rho$ is the set of all possible ρ-programs. The problem arises if the $\mathcal{P}_\rho$ are mutually disjoint for different ρ's; i.e. supposing John has written σ in $\mathcal{P}_\rho$, can Tom recognize ρ by studying the syntax (and only the syntax) of σ. The answer is negative, since—as we shall prove in this section—$\mathcal{P}_\rho = \mathcal{P}_\xi$ for any two admissible orders ρ and ξ.

To simplify the investigations we shall deal with formulas and trees instead of programs and labeled trees.

Let T denote the set of all trees and let for every admissible order

$$L_\rho = F_\rho(T);$$

i.e. L_ρ is the set of all possible ρ-formulas. Let $V = \{dS, Sd, SS, dd\}$ and let V^+ denote the set of all finite nonempty strings over V. In V^+ we define the following integer-valuated function H:

$$H(dS) = H(Sd) = 0,$$
$$H(SS) = -1,$$
$$H(dd) = 1,$$
$$H(\alpha_1\beta_1 \cdots \alpha_n\beta_n) = \sum_{i=1}^{n} H(\alpha_i\beta_i).$$

The intuitive interpretation of H is very simple. Let σ be a ρ-program (or a part of it) and θ a corresponding ρ-formula. $H(\theta)$ is simply the number of intermediate results stored actually by the unit after θ has been scanned, provided the unit erases when reads.

Let now L be the set of all θ in V^+ with the following properties:

(1) $H(\theta) = 1$,

(2) if $\theta = \alpha_1\beta_1 \cdots \alpha_n\beta_n$, then $H(\alpha_i\beta_i \cdots \alpha_n\beta_n) \geq 1$ for $i = 1, \cdots, n$.

THEOREM 6. *For every admissible order ρ, $L_\rho = L$.*

PROOF. Consider an arbitrary admissible order ρ and an arbitrary tree $T = (A, L, R, P)$. Let $[D_1(P), \rho] = (g_1, \cdots, g_n)$ and $F_\rho(T) = \theta$. We shall show that $\theta \in L$. The proof is by induction on n.

For $n = 1$, $\theta = dd$; thus $\theta \in L$.

Let the thesis be true for every $n < k$ and consider T with $[D_1(P), \rho] = (g_1, \cdots, g_k)$. Since ρ is admissible, $\tau_{D_1(P)}(g_k) = dd$ and hence $D_1(P) - \{g_k\}$ is also the set of all ramifications of some tree. Therefore, by the inductive assumption, $F_\rho(D_1(P) - \{g_k\}) \in L$. Let now

$$F_\rho(D_1(P) - \{g_k\}) = \alpha_1\beta_1 \cdots \alpha_{k-1}\beta_{k-1},$$
$$F_\rho(T) = \alpha_1^*\beta_1^* \cdots \alpha_k^*\beta_k^*,$$

and let $S(g_k) = g_i$. Clearly,

$$H(\alpha_j^*\beta_j^* \cdots \alpha_k^*\beta_k^*) = H(\alpha_j\beta_j \cdots \alpha_{k-1}\beta_{k-1}) + 1$$

for every $j > i$, and

$$H(\alpha_i^*\beta_i^*) = H(\alpha_i\beta_i) - 1.$$

Therefore,

$$H(\alpha_p^*\beta_p^* \cdots \alpha_k^*\beta_k^*) = H(\alpha_p\beta_p \cdots \alpha_{k-1}\beta_{k-1})$$

for every $p \geq i$. Hence $\alpha_1^*\beta_1^* \cdots \alpha_k^*\beta_k^* \in L$. In this way the inclusion $L_\rho \subseteq L$ is proved.

Now let ρ be an arbitrary admissible order and let $\theta = \alpha_1\beta_1 \cdots \alpha_n\beta_n \in L$. We shall show the existence of a tree T with $F_\rho(T) = \theta$. The proof is by induction on n.

For $n = 1$, $\theta = dd$; hence $\theta = F_\rho(\{\omega\})$.

Let the thesis be true for every $n < k$ and consider $\theta = \alpha_1\beta_1 \cdots \alpha_k\beta_k$ in L. Since θ is in L, there exists an $i < k$ with $\alpha_i\beta_i \neq dd$ and $\alpha_j\beta_j = dd$ for $j = i + 1, \cdots, k$.

Consider now the formula

$$\theta^* = \alpha_1\beta_1 \cdots \alpha_{i-1}\beta_{i-1}\alpha_i^*\beta_i^*\alpha_{i+1}\beta_{i+1} \cdots \alpha_{k-1}\beta_{k-1},$$

where

$$\alpha_i^*\beta_i^* = \begin{cases} dd, & \text{if } \alpha_i\beta_i = dS \text{ or } \alpha_i\beta_i = Sd, \\ dS, & \text{if } \alpha_i\beta_i = SS. \end{cases}$$

It is easy to see that $\theta^* \in L$, thus—by the inductive assumption—there exists a tree $T = (A, L, R, P)$ with $F_\rho(T) = \theta^*$. Let $[D_1(P), \rho] = (g_1, \cdots, g_{k-1})$ and consider g in A^∞ with $g_i L^\infty g$. Clearly, $D_1(P) \cup \{g\}$ is also the set of ramifications of a tree and, since, $g_i\rho g$ (ρ is admissible), we have immediately

$$F_\rho(D_1(P) \cup \{g\}) = \theta.$$

Hence, $\theta \in L$. QED

We can add here that L is a context-free language (cf. [2]) with the following grammar:

$$A \to BA, \qquad A \to dd, \qquad B \to dS,$$
$$A \to CAA, \qquad B \to Sd, \qquad C \to SS,$$

where A is the initial symbol. This thesis is proved in [1].

6. *Final Remarks*

Let us summarize the principal results of this paper.

(1) Addressless orders are the only orders that permit a step by step performance of addressless programs (Definition 3) and that guarantee a uniform accessing of intermediate results (Theorem 2).

(2) Addressless orders are the only orders that correspond to dequeue (double-ended-queue) disciplines; moreover, to each dequeue discipline corresponds exactly one addressless order (Theorems 3, 4, and 5).

(3) Two different addressless orders cannot be distinguished by studying the syntax of corresponding programs (Theorem 6).

All these results have been proved for the binary case, i.e. for the case where all trees are binary-ramificated. This restriction has been assumed to simplify mathematical investigations, however it can be rejected now without harming the truth of (1), (2), and (3). In order to generalize the results for arbitrary n-ary case we proceed as follows: First n-ary ramificated trees are defined which is clearly simple and needs no comments. Then the pair of functions ψ_ρ^L and ψ_ρ^R is replaced by an n-tuple $\psi_\rho^1, \cdots, \psi_\rho^n$ of analogous functions. This allows us to define n-ary addressless orders. In the next step the functions f_ρ^L and f_ρ^R are replaced by analogous $f_\rho^1, \cdots,$

f_p^n and an n-ary analogue of Theorem 3 is proved. The proof is based on the fact that Lemma 1 remains true for the n-ary case without any reformulation. The rest of the proof is similar. Also the proof of Theorem 4 needs only minor modifications. In effect we have a corollary to the effect that every n-ary addressless order corresponds to some n-ary dequeue discipline and vice versa, and that this correspondence is one-to-one. Since there are exactly $(n + 1)!$ n-ary dequeue disciplines, we claim therefore immediately that there exist exactly $(n + 1)!$ n-ary addressless orders. The generalization of Theorem 6 clearly needs an appropriate generalization of function H. This can be done as follows.

For every n-string $\alpha_1 \cdots \alpha_n$ (which generalizes the concept of the type $\alpha\beta$ of a vertex) with $\alpha_i \in \{d, S\}$ for $i = 1, \cdots, n$:

$$h(\alpha_1 \cdots \alpha_n) = 1 - m,$$

where $m = \text{Card}\{i: \alpha_i = S\}$.

ACKNOWLEDGMENTS. I wish to thank Professor Z. Pawlak for help and encouragement during this work. I am indebted to Doctors A. Ehrenfeucht and A. Rędziejowski for many valuable suggestions concerning the thesis of this paper. I also thank the referees of the *Journal of the ACM* for their constructive criticism of the earlier exposition of my results.

REFERENCES

1. BLIKLE, A. Formalization of parenthesis-free languages. *Zeitschr. Math. Logik Grundl. Math. 12* (1966), 177–186.
2. GINSBURG, S. *The Mathematical Theory of Context-Free Languages*, McGraw-Hill, New York, 1966.
3. PAWLAK, Z. Organization of the address-free digital computer for calculating simple arithmetical expressions. *Bull. Acad. Polonaise Sci., Ser. Sci. Tech. 8* (1960), 193–197.
4. PAWLAK, Z. New method of parenthesis-free notation of formulae. *Bull. Acad. Polonaise Sci., Ser. Sci. Tech. 8* (1960), 197–199.
5. PAWLAK, Z. On realization of recursive schemas in the address-free computer. *Bull. Acad. Polonaise Sci., Ser. Sci. Tech. 8* (1960), 685–690.
6. PAWLAK, Z. Organization of address-free computer B-100. *Bull. Acad. Palonaise Sci., Ser. Sci. Tech. 9* (1961), 229–238.
7. PAWLAK, Z. Some remarks on automatic programming of arithmetical formulae. *Bull. Acad. Polonaise Sci., Ser. Sci. Tech. 9* (1961), 317–321.
8. PAWLAK, Z. Realization of memory of partial results in certain parenthesis-free formalisms. *Bull. Acad. Polonaise Sci., Ser. Sci. Tech. 9* (1961), 487–491.
9. PAWLAK, Z. *Organizacja Maszyn Bezadresowych* (in Polish). Polish Scientific Publishers, Warsaw, 1965.

RECEIVED OCTOBER 1969; REVISED APRIL 1971