Computer Science Journal of Moldova, vol.**19**, no.1(55), 2011

Multiset-based Tree Model for Membrane Computing

D. Singh C.M. Peter

Abstract

In this paper, we introduce a new paradigm - multiset-based tree model. We show that trees can be represented in the form of wellfounded multisets. We also show that the conventional approach for this representation is not injective from a set of trees to the class of multisets representing such trees. We establish a one-to-one correspondence between trees and suitable permutations of a wellfounded multiset, which we call *tree structures*. We give formal definitions of a *tree structure* and a *subtree structure* of a tree structure. Finally, we represent membrane structures in the form of tree structures – a form in which membrane structures can suitably be represented at programming level.

Keywords: wellfounded multiset, saw-like structure, multiset-based tree structure, membrane structure.

1 Introduction

A tree is an *acyclic* connected graph (having one source or trunk and several exits or leaves). It can also be defined as a partial order relation over a finite set with the smallest element. Note that if there is only one edge from a source node, then we have only one branch on the node.

Trees have served as handy tools in solving problems involving decisions and the flow of information. Thus they form a fundamental concept in graph theory. They are used in many disciplines such as Mathematics, Management, Economics, Commerce, Biology, Computer Science, Statistics and Probability, just to mention a few.

In Managerial accounting ([7], p. 502), a tree diagram in the form of an organizational chart was used to illustrate the organization of Aloha Hotels and Resorts. In Computer Science [1], the Huffman tree is used to compress bits so as to reduce the amount of storage that is necessary in a storage media. In Biology, ([12], p. 57) exploits a tree structure to represent the basic characteristics of meiosis involving one chromosome duplication followed by two nuclear and cell divisions. In Probability [8], the concept of *tree measure* is applied to represent a sequence of repeated throws of a coin with either side labelled head or tail. In English language ([6], p. 24), the classification of nouns is represented in the form of a tree. As some membranes can contain several other membranes [10], trees can also be suitably used to study membrane structures and hence in comprehending membrane computing.

Note that it may be relevant to count the number of nodes in an n-nary tree – a tree with ($n \geq 2$) number of branches emerging from each of the major branches after them (which is called a *tail* or a node) or a general tree, for that matter, with n varying from branch to branch.

In the recent years, the representation of a tree in the form of a wellfounded (cardinality–bounded) multiset has been arguably used. It is observed that the said representation is not *injective* from a set of trees to the class of multisets representing such trees. In order to achieve injection, we devise various permutations of the wellfounded multiset in consideration along with a suitable rule. We begin with a binary tree and generalize the approach to an n-nary tree as well as a *general* tree.

2 Aptness for the Use of Trees and a Multiset Environment

Unlike other graphs, trees are quite innovative especially in one of the recently researched areas of computer science – molecular and membrane computing. This is because among all other forms of graphs (e.g., a *loop* or a *multigraph*), only trees can suitably represent membrane structures (without intersections, loops or parallel edges). This singular but essential ability of trees is what has inspired us to study them and to see how they can help us in contributing to the improvement of membrane computing.

There is no provision for a diagrammatic or pictorial input of membrane structures at programming level. Thus, one aim of this paper is to demonstrate how membrane structures can be represented at programming level by devising a discrete approach for such a representation. It is in line with this that we have employed multiset as an environment in representing trees, and consequently membrane structures.

3 Some Basic Concepts

Definition i: Multiset.
A multiset (mset, for short) is a collection of objects in which, unlike a crisp (Cantorian) set, objects are allowed to repeat finitely in most of the application areas; although infinite multiplicities are also dealt with in a theoretical development (see [2] and [3] for details).

A multiset is represented in several ways. The use of square brackets to represent a multiset is quasi-general. Thus, a multiset containing one occurrence of a, two occurrences of b, and three occurrences of c is notationally written as $[[a, b, b, c, c, c]]$ or $[a, b, b, c, c, c]$ or $[a, b, c]_{1,2,3}$ or $[a, 2b, 3c]$ or $[a.1, b.2, c.3]$ or $[1/a, 2/b, 3/c]$ or $[a^1, b^2, c^3]$ or $[a^1b^2c^3]$. For convenience, the curly brackets are used in place of the square brackets. In fact, the last form of representation as a string, even without using any brackets, turns out to be the most compact one, especially in computational parlance. The following schematic representation of a multiset as a numeric-valued or count function abounds, particularly in the foundational development of multiset theory and its application:

A multiset is a mapping from some ground or generic set of objects into some set of numbers. For example, a multiset $\alpha = [x, y, z]_{1,2,3}$ is a mapping from a ground set D to $\mathbb{N}$, the set of non negative integers, defined by

$$\alpha(t) = \begin{cases} 1, & \text{if } t = x \\ 2, & \text{if } t = y \\ 3, & \text{if } t = z \\ 0, & \text{for all the remaining } t \in D. \end{cases}$$

In other words, a multiset α drawn from a ground set D can be represented by a cardinal-valued function $C_\alpha : D \to \mathbb{N}$.

In general terms, for a given ground set D and a numeric set T, we call a mapping $\alpha : D \to T$,

$$\begin{cases} \text{a set, if } T = \{0,\ 1\}\,; \\ \text{a multiset, if } T = \mathbb{N},\ \text{the set of natural numbers;} \\ \text{a signed multiset (hybrid or shadow set), if } T = \mathbb{Z}, \text{the set of integers;} \\ \text{a fuzzy (or hazy) set if } T = [0,1] \subseteq \mathbb{R},\ \text{a two}-\text{valued Boolean algebra.} \end{cases}$$

In view of the above definition, a multiset A can also be represented by a set of pairs as follows:

$A = \{\langle m_A(x_1), x_1\rangle, \ldots, \langle m_A(x_j), x_j\rangle, \ldots\}$ or

$A = \{m_A(x_1) \cdot x_1, \ldots, m_A(x_j) \cdot x_j, \ldots\}$ or

$A = \{n_1/x_1, \ldots, n_j/x_j, \ldots\}$, where $m_A(x_j) = n_j =$ the count or the multipliciy of x_j in A.

Note that there are other forms of representing a multiset (see [2], [3] and [11] in particular).

Definition ii: Submultiset.

Given a multiset M over a domain set D, a multiset A over D is called a submultiset of M written as $A \subseteq M$ or $M \supseteq A$ if $m_A(x) \leq m_M(x)$ for all $x \in D$, where $m_A(x)$ and $m_M(x)$ are the multiplicities of x in the multisets A and M respectively. Also if $A \subseteq M$ and $A \neq M$, then A is called a proper submultiset of M. A multiset is called the ancestor in relation to its submultiset (see [11], for details).

Definition iii: Dressed epsilon.

For any object x occurring as an element of a multiset A i.e., $m_A(x) > 0$, we write $x \in_+ A$, where $\in_+$ (dressed epsilon) is a binary predicate intended to be 'belongs to at least once', as $\in$ is 'belongs to only once' in the case of sets. Also, $x \in_+^k A$ implies 'x belongs to A at least k times', while $x \in^k A$ means x belongs exactly k times to A. $x \notin A$ means 'x does not belong to A' ([11], for details).

Definition iv: Partial ordering.
A binary relation $\lessdot$ on a set X is called a *partial order* on X if $\lessdot$ satisfies the following axioms:

1. $x \lessdot x$ for all $x \in X$. (Reflexivity)
2. $x \lessdot y$ and $y \lessdot x \Rightarrow x = y$ for all $x, y \in X$. (Anti symmetry)
3. $x \lessdot y$ and $y \lessdot z \Rightarrow x \lessdot z$ for all $x, y, z \in X$. (Transitivity)

The set X is said to be *partially ordered* with respect to $\lessdot$. We note that for some pair of elements x, y in X, neither $x \lessdot y$ nor $y \lessdot x$ may hold. If $x \lessdot y$ or $y \lessdot x$ for all x, y in X then X is said to be *totally ordered* or linearly ordered or a chain (see [5] for details).

Definition v: The Dershowitz-Manna ordering on multisets.
We follow the Dershowitz-Manna ordering on multiset. Let $(S, >)$ be a partially ordered set and $M(S)$ be the set of all finite multisets with elements taken from the set S. Then a partial order $\gg$ on $M(S)$ can be defined as follows:

Let M, $M' \in M(S)$. Then $M \gg M'$ if for some multisets $X, Y \in M(S)$, with $\emptyset \neq X \subseteq M$, we have $M' = (M \backslash X) \cup Y$ and $(\forall y \in_+ Y)(\exists x \in_+ X) \quad x > y$.

For example, let $S = \mathbb{N}$, the set of natural numbers including 0 with the usual ordering $>$, then under the corresponding multiset ordering $\gg$ over $\mathbb{N}$, the multiset $\{3, 3, 4, 0\}$ is greater than each of the three multisets $\{3, 4\}$, $\{3, 2, 2, 1, 1, 1, 4, 0\}$ and $\{3, 3, 3, 3, 2, 2\}$. (see [4] for details).

Definition vi: Wellfounded multiset.
A wellfounded multiset is a multiset with an irreflexive and transitive

ordering defined on it, such that its every submultiset has a minimal element; in other words, no infinite descending chain occurs. We shall follow the Dershowitz-Manna ordering on multisets over a set of natural numbers which has been proved wellfounded (see for the proof in [4]).

Definition vii: An n-nary tree.
Given a non-negative integer n, an *n-nary tree* is a tree which has exactly n number of branching on each of its branches. In the case where the number of branching varies from branch to branch on the tree, we call such a tree a *general tree* or simply a tree.

Definition viii: A binary tree.
A *binary tree* is an n-nary tree for $n = 2$.

Definition ix: A subtree.
A *subtree* is a subgraph which is a tree.

See [4] for details of the aforesaid definitions.

4 The Binary Tree

Dershowitz and Manna ([4]) demonstrated the termination of a program to count the tips of a binary tree using a wellfounded multiset ordering. We describe in brief one of the examples they considered.

> *Consider a simple program to count the number of tips – terminal nodes (without descendents) – in a full binary tree. Each tree y that is not a tip has two subtrees, left(y) and right(y).*

Typically, a binary tree can be schematically represented as in Figure 1.

In Figure 1, the *tree trunk* is represented by the largest integer in the labelling of the tree. One of the branches on a y-shaped (or fan-shaped) subtree is called an *axis.* The part of the tree which continues from the base to a tip without a gap is called a *chain.* The y-shaped subtree at the bottom of the tree is called the *base* of the subtree.

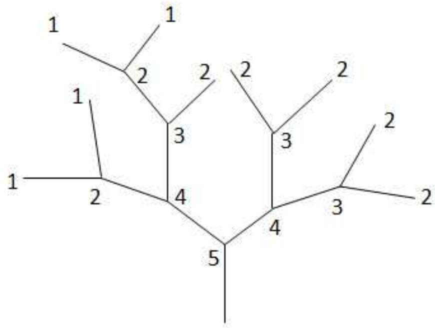


Figure 1. A binary tree

Notice that in the diagram above the integer label of a branch is less than the integer label of the node upon which the branch rests. One of the advantages of representing a tree in this way is reflected in membrane structures. Thus, the sizes of membranes in a membrane structure can be used in place of the integer labels, yet retaining the tree representation of membrane structures. This fact is vindicated in this paper when we shall be applying tree structures to membrane computing.

5 Conventional Approach to Representing a Tree by a Wellfounded Multiset

A *conventional method* of representing a tree by a wellfounded multiset seems to have first appeared in [4]. The conventional method entails that in a wellfounded multiset, any element-multiset representing a subtree which is built upon another element-multiset representing another subtree needs to be smaller than the one upon which it is built. Moreover, there is no permutation governing the arrangement of the element-multisets of the wellfounded multiset. Rather, one rule inherent in the representation is that the smaller and larger element-multisets have exactly one element in common. This common element is also the largest element in the smaller element-multiset and cannot be the largest element in the larger element-multiset. The action of picking an element-multiset to represent a subtree on the tree is done exhaustively. Though no definition of this method has been given in [4], we give the following formal definition to capture the concept.

Definition x.
Formally, given a multiset S over a domain set D, a multiset $T(S)$ whose elements are submultisets of S is a *conventional representation of a tree* if and only if it satisfies the following properties:

1. There exists $z \in^1 T(S)$ such that $z = max\{y : y \in_+ T(S)\}$,
2. For each $u \in_+ T(S)$ where $u \neq max\{y : y \in_+ T(S)\}, \exists w \in_+ T(S)$ with $w \gg u$ and $x_0 \in_+ S$ such that $x_0 \in^1 u \cap w$, $x_0 = max\{x : x \in_+ u\}$ and $x_0 \neq max\{x : x \in_+ w\}$.

In the above definition, the first condition is called *the base condition* and z is called the *base* of a *tree*. The second condition is called *the join condition* and x_0 is the *join* between two *subtrees*.

6 Wellfounded Multiset Representation of a Binary Tree (Use of the Conventional Approach)

A binary tree is represented in the form of a wellfounded multiset whose elements are multisets containing only three elements. For example, the wellfounded multiset $\{\{322\}, \{211\}\}$ represents the binary tree in Figure 2.

Each y-shaped subtree is represented by an element say a in the multiset; a itself is a multiset with three elements of the form $\{a_1, a_2, a_3\}$. We show that the *conventional* approach of the representation is not injective from a set of trees to the class of wellfounded multisets representing such trees.

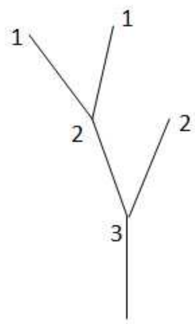


Figure 2. A two-element binary tree

Consider $N = \{\{544\}, \{433\}, \{432\}, \{322\}, \{322\}, \{322\}, \{211\}, \{211\}\}$. Each element of N has three elements and represents a y-shaped subtree of the tree in Figure 1.

It is easy to see that N is ordered according to the Dershowitz-Manna ordering on multisets.

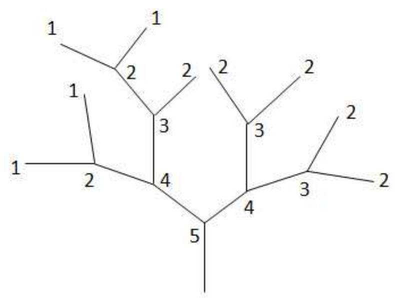
The element $\{544\}$ represents the first y-shaped subtree with tail 5. The two 4's in $\{544\}$ show that we can locate two elements in N, each of which has 4 as its largest element. The two elements are $\{433\}$ and $\{432\}$ which give rise to two y-shaped subtrees with tails labelled 4 on the trunk labelled 5.

The three 3's in these two elements combined show that we can locate three other elements in N each of which has 3 as its largest element. These elements are three $\{322\}$'s, which give rise to three y-shaped subtrees with tails labelled 3 on the branch labelled 4.

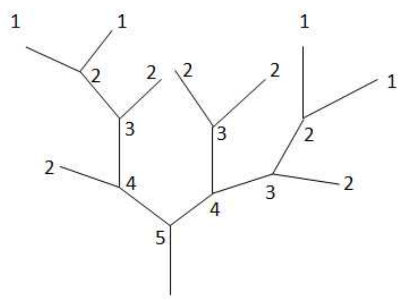
Next are the six 2's in the three elements. But, since there are only two $\{211\}$'s in N each of which has 2 as its largest element, there can only be two y-shaped subtrees each of whose tail is labelled 2.

Also, the 2 in $\{432\}$ shows that we can locate an element in N which has 2 as its largest element. This element is $\{211\}$, and can give rise to a y-shaped subtree with tail labelled 2.

It can be observed that there arises a decidability problem as to how and on which of the two types of branches, one of which is labelled 4 and three of which are labelled 3, should the two subtrees with tail labelled 2 be built upon. If one of the two $\{211\}$'s is built on any one of the three $\{322\}$'s while the other $\{211\}$ is built on $\{432\}$, we get the binary tree A in Figure 3 below. If on the other hand each of the two $\{211\}$'s is built on each of any two of the three $\{322\}$'s, we get the binary tree B in Figure 3.



Binary Tree A



Binary Tree B

Figure 3. Two different binary trees generated by the multiset N

Thus, our ordered multiset N can yield two different binary trees and so the conventional method does not give room for a wellfounded multiset representation of a binary tree in a one-to-one manner.

The above assertion can be generalised in the case of an n-nary tree. It can similarly be shown for a tree with n varying from branch to branch on the tree, called a *general* tree.

7 The Saw Rule

We demonstrate below by considering a suitable permutation of a partially ordered multiset how to represent a rule in which each submultiset of elements is arranged in an *uphill* manner, while at the same time the multiset of all the elements in the system is arranged in a *downhill* manner, called a *saw-like permutation*.

An *uphill* multiset of elements is a permutation of a partially ordered multiset of elements which ends with the largest element in the multiset. A *downhill* multiset of elements is a permutation of a partially ordered multiset of elements which begins with the largest element in the multiset. Thus, all the elements in an uphill multiset of elements or in a downhill multiset of elements may neither be ascending nor descending.

This rule is called a *saw-like permutation* because it creates a resemblance of a wood saw blade when viewed pictorially using vertical bars to represent the elements of a permutation of a partially ordered multiset according to their sizes as in the following figure:

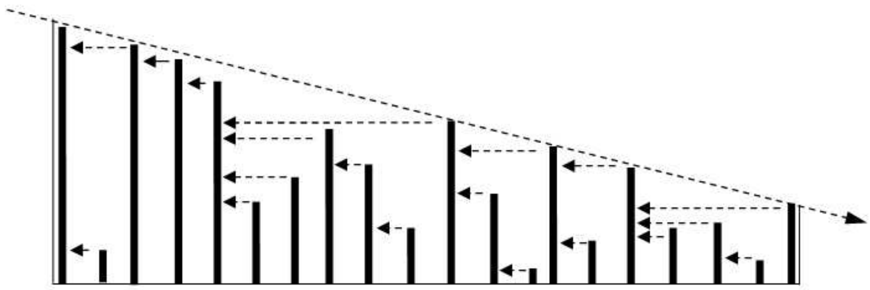


Figure 4. A saw-like structure illustrating a saw-like permutation

The scheme can be interpreted as follows: Each bar in a multiset of ascending bars is attached to the longest bar immediately before the multiset. In other words, any two bars in which one is attached to another must not have any bar longer than the attached bar in between them. The bars represent the element-multisets of a permutation of a partially ordered multiset. The left arrow pointing from one bar to another indicates attachment from the bar on its right to the bar on its left. In other words, each bar on the right side of an arrow represents a branch while the corresponding bar on the left represents a node on the tree. We shall often refer to the element-multisets of a wellfounded multiset as simply elements.

While going through the elements of the partially ordered multiset, we pick the first and largest element represented by the longest bar. This is followed by the smallest element that can be attached directly to the largest element. This in turn is followed by another element only larger than the smallest element, and can be attached directly to the largest element. Any element that must be attached to an element which has already been attached to the largest element must come after the element upon which it is to be attached, even if it happens to be smaller than the first element attached directly to the largest element.

We continue in this way until we have exhausted all the elements that can be attached to the largest element. In Figure 4 above, only two bars have been attached to the first (largest) bar. There is no doubt that these two bars are trivially in an uphill order according to their heights.

This gives us a submultiset of elements immediately following the largest element in an uphill order. Among all the submultisets in an uphill order which have been attached to an element, this one happens to be the largest. The last but not the least element (the third bar in Figure 4 above) in this submultiset turns out to be the second largest element in the partially ordered multiset, and it is the element upon which the second submultiset of elements in an uphill order will be attached starting with the smallest element directly attachable to it. In Figure 4, the second largest bar (which is the third bar) has only one bar attached to it. This one bar can be regarded as a submultiset in an uphill order containing a singleton multiset, though a very trivial case.

Next is the third largest submultiset of elements in an uphill order. We continue in this way until all the elements in the partially ordered multiset belong to a group of elements in an uphill order. Note that in Figure 4 above, the fifth bar has four bars attached to it, including the eighth bar, which in turn has two bars attached to it. These four bars are clearly in ascending order, whereas the six bars (the four bars with the two bars) make up the multiset of bars attached to the fifth bar in an uphill order. This is a non trivial example of bars arranged in an uphill order.

Every element on the saw-like permutation is called a *bar*; every bar which has an element attached to it is called a *column*; a column which is followed by a non-singleton submultiset of descending bars each of which is less than the column and at least one of which has a bar attached to it, is called a *pillar*. A submultiset of consecutive elements from the same object whose multiplicity is more than one is called a *platform*. The first bar on the saw-like permutation is called its *base*.

Having demonstrated how the *saw-like permutation* is used to arrange the elements of a partially ordered multiset in the above discussion, we now give a formal definition of the representation of a partially ordered multiset *in order of a saw-like permutation* in the following definition:

Definition xi.
A permutation $P(S)$ of a partially ordered multiset S of order n is said to be *in order of a saw-like permutation* if and only if $y_1 > y_i$ in $P(S)$, for all $i = 2, 3, \ldots, n$.

If such a permutation exists, we say that $P(S)$ is arranged in order of a saw-like permutation or $P(S)$ is a saw-like permutation of the elements of S. We shall see later in this paper that this construction is of immense help in defining a tree structure.

8 Representation of a Tree by a Saw-like Permutation of a Wellfounded Multiset (The Saw Rule)

To resolve the aforesaid issue of the representation not being injective from a set of trees to the class of multisets representing such trees as indicated in section six above, we demonstrate by considering *saw-like permutations*, how to represent a tree by a permutation of a wellfounded multiset. We called this *the saw rule*. We show below how to represent a binary tree using *the saw rule* and also how the injection is achieved using this rule. In this case, consider the following illustrations to bring our point home:

Let us consider the following two permutations N_1 and N_2 of N:

$N_1 = [\{544\}, \{423\}, \{322\}, \{211\}, \{433\}, \{322\}, \{322\}, \{211\}]$ and

$N_2 = [\{544\}, \{423\}, \{211\}, \{322\}, \{211\}, \{433\}, \{322\}, \{322\}]$

The element $\{544\}$ of N_1 represents the first y-shaped subtree (called the *base*) with tail 5. The two 4's in $\{544\}$ show that we can locate at most two elements in N_1 each of which has 4 as its largest element. The two elements are $\{423\}$ and $\{433\}$.

The element $\{423\}$ is smaller than $\{544\}$ and so can give rise to a y-shaped subtree with tail labelled 4.

The 3 in $\{423\}$ shows that we can locate at most an element in N_1 which has 3 as its largest element, and since the next element $\{322\}$ is smaller than $\{423\}$ it can give rise to a y-shaped subtree with tail labelled 3.

The two 2's in $\{322\}$ show that we can locate at most two elements each of which has 2 as its largest element. Since there is only one such element which has 2 as its largest element and smaller than $\{322\}$ then it can give rise to a subtree with tail labelled 2.

The next element $\{433\}$ is larger than $\{322\}$ and so no subtree representing $\{433\}$ can arise on $\{322\}$. However, $\{544\}$ is the smallest element larger than $\{433\}$, going backwards. Since only one of the two elements having 4 as their largest element has its subtree on $\{544\}$, there can arise another subtree with tail labelled 4 on $\{544\}$ (since $\{544\}$ has two 4's in it).

The two 3's in $\{433\}$ show that we can locate at most two elements each of which has 3 as its largest element. Since the next two elements satisfy this condition, and are smaller than $\{433\}$, then there can arise two subtrees with tails labelled 3.

The last two 2's in the last $\{322\}$ show that we can locate at most two elements each of which has 2 as its largest element. Since we have only one of such elements which is $\{211\}$ having 2 as its largest element, there can arise only one subtree with tail labelled 2 coinciding with only one of the axes labelled 2.

The permutation N_1 of N can be seen to have constructed the one and only tree A in Figure 3 above and no other. Similarly, N_2

determines only the binary tree B in Figure 3. We call N_1 and N_2 *tree structures.* The following figures are saw-like structures of N_1 and N_2 equivalent to the one in Figure 4, for a clear understanding of the concept.

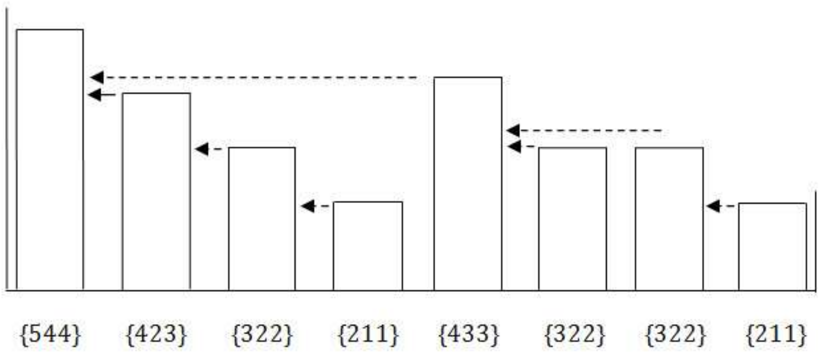


Figure 5. A saw-like structure illustrating N_1

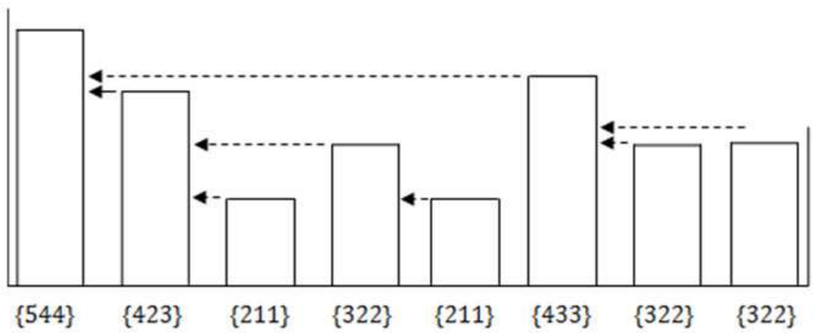


Figure 6. A saw-like structure illustrating N_2

Notice in Figures 5 and 6, that a bar (representing a subtree on the tree) can only be attached to a larger bar (representing a node on the tree) where the larger bar is a multiset containing the largest element of the smaller multiset representing the smaller bar. Such largest element must not be the largest in the larger bar. Notice also that a bar has not been attached to another bar of equal size (or height in this case).

Not all wellfounded multisets, even with the *conventional method*, can represent a tree. This is seen from the multiset $[\{544\}, \{432\}, \{211\}, \{432\}, \{211\}, \{211\}]$ since the supposed subtree structure $[\{211\}]$ of the last element $\{211\}$ does not have a branch to rest upon. The *saw rule* ensures, firstly, that a wellfounded multiset can suitably represent a tree; secondly, that every tree can be represented in the form of a wellfounded multiset (this is the only property that the conventional method ensures) and thirdly, that this representation is injective from a set of trees to the class of multisets representing such trees.

Remark.
It is important to observe that in order to construct a tree using a unique permutation of a given wellfounded multiset, the use of *the saw rule* in building an element upon a preceding element is not only that the subsequent element be smaller than the preceding one, but also that its largest element must belong to the preceding element, and that the multiplicity of such subsequent element in an uphill submultiset of elements must not be greater than the number of occurrences of such largest element in the preceding element.

For example, in $[\ldots\{543\}, \{211\}\ldots]$, the subtree representing $[\{211\}]$ cannot be built upon the subtree representing $[\{543\}]$ since the largest element 2 of $\{211\}$ does not belong to $\{543\}$. But in $[\ldots\{543\}, \{411\}\ldots]$, $[\{411\}]$ can be built upon $[\{543\}]$ since the largest element 4 of $\{411\}$ is in $\{543\}$. Again, we cannot build any subtree using $[\ldots\{543\}, \{411\}, \{411\}, \{411\}\ldots]$ since 4 does not appear up to three times in $\{543\}$. However, $[\ldots\{5444\}, \{411\}\ldots]$ represents a subtree. Such an element as 4 in this case is called a *join* in a *tree structure*. We use square brackets for a *tree structure* since a tree structure is a list – an ordered sequence of elements with repetitions allowed.

To avoid misinterpretation, we shall not use a tree and a tree structure interchangeably. As mentioned in the introduction, a tree is an **acyclic connected graph**. On the other hand, a *tree structure* is a

multiset representation of a tree by exploiting the *saw rule*, e.g., N_1 and N_2 above. Notwithstanding the fact that there is a one-to-one correspondence between the two, this distinction is useful especially when we shall be applying the concept of tree structures to membrane computing. The same argument goes for a subtree and a subtree structure.

Having discussed the representation of a binary tree in order of a saw-like permutation of a wellfounded multiset (called a *tree structure*), we consider it necessary to generalize our discussion to capture the concepts of an n-nary tree structure (corresponding to a tree with exactly n branches on each node of the tree) and a *general tree* (corresponding to a tree with n varying from branch to branch on the tree). Therefore, based on the discussions above, we now give the following formal definitions:

Definition xii.
Let S be a partially ordered multiset over a domain set D. A permutation $P(\tau(S))$ of a wellfounded multiset $\tau(S)$ of cardinality m, whose elements are submultisets of S is called an *n-nary tree structure* if and only if it satisfies the following properties:

1. For all $y \in_+ P(\tau(S))$ $\exists\, n \in \mathbb{N}$ such that $C(y) = n$, where $C(y)$ denotes the cardinality of y.

2. $y_1 \gg y_i$ in $P(\tau(S))$ for all $i = 2, 3, \ldots, m$.

3. For all $i, j \in \mathbb{N}$ with $y_i \gg y_j$ for $i < j$, if $\nexists\ k \in \mathbb{N}$ such that $y_i \gg y_k \gg y_j$ for $i < k < j$ then $\exists\ x_0 \in^1 y_i \cap y_j$ such that $x_0 = max\{x : x \in_+ y_j\}$ and $x_0 \neq max\{x : x \in_+ y_i\}$, where $y_i, y_k, y_j \in_+ P(\tau(S))$, $i = 1, 2, \ldots, m-1$, $k = 2, 3, \ldots, m-1$ and $j = 2, 3, \ldots, m$.

   In other words, it says that y_i *yields* y_j via x_0 or y_j is an immediate successor of y_i or y_i is an immediate predecessor of y_j, and we denote this by $y_i \overset{\{x_0\}}{\twoheadrightarrow} y_j$.

4. For each $z \in P(\tau(S))$ and for a given $x \in_+ S$ the multiset $Y = \{y : z \overset{\{x\}}{\twoheadrightarrow} y\}$ is such that $C(Y) \leq m_z(x)$, where $C(Y)$ is

the cardinality of Y and $m_z(x)$ is the multiplicity of x in z.

In the above definition, the first condition is known as *the equal cardinality condition*. The second condition is known as *the base condition* (or *the saw rule condition*) and y_1 is *the base* of the tree structure. The third condition is the *join condition* and x_0 is *the join* between two subtree structures. The fourth condition is *the parallelism condition*. The parallelism condition ensures that the multiplicity of a join in a node is greater than or equal to the number of subtree structures (or branches) joinable to the node using such join.

Definition xiii.
Let S be a partially ordered multiset over a domain set D, a permutation $P(\tau(S))$ of a wellfounded multiset $\tau(S)$ of cardinality m, whose elements are submultisets of S, is called a *general tree structure* if and only if it satisfies the following properties:

1. $y_1 \gg y_i$ in $P(\tau(S))$ for all $i = 2, 3, \ldots, m$.

2. For all $i, j \in \mathbb{N}$ with $y_i \gg y_j$ for $i < j$, if $\nexists\ k \in \mathbb{N}$ such that $y_i \gg y_k \gg y_j$ for $i < k < j$ then $\exists\ x_0 \in^1 y_i \cap y_j$ such that $x_0 = max\{x : x \in_+ y_j\}$ and $x_0 \neq max\{x : x \in_+ y_i\}$, where $y_i, y_k, y_j \in_+ P(\tau(S))$, $i = 1, 2, \ldots, m-1$, $k = 2, 3, \ldots, m-1$ and $j = 2, 3, \ldots, m$.

   In other words, it says that y_i *yields* y_j via x_0 or y_j is an immediate successor of y_i or y_i is an immediate predecessor of y_j, and we denote this by $y_i \overset{\{x_0\}}{\twoheadrightarrow} y_j$.

3. For each $z \in P(\tau(S))$ and for a given $x \in_+ S$ the multiset $Y = \{y : z \overset{\{x\}}{\twoheadrightarrow} y\}$ is such that $C(Y) \leq m_z(x)$, where $C(Y)$ is the cardinality of Y and $m_z(x)$ is the multiplicity of x in z.

The first condition is *the base condition*. The second condition is *the join condition* and the third condition is *the parallelism condition*.

In the above two definitions, the tree structure $P(\tau(S))$ is said to be built over the multiset S with D as its domain. We define the root set of the tree structure $P(\tau(S))$ as the set $R = \{x \in D : x \in_+ y \; \forall \; y \in_+ P(\tau(S))\}$. If there is no confusion about which multiset S is intended, we simply write τ for $P(\tau(S))$.

The process by which a subtree structure yields another subtree structure in a tree structure is called a *succession.*

That is, if $y_1 \overset{\{x_1\}}{\twoheadrightarrow} y_2$ and $y_2 \overset{\{x_2\}}{\twoheadrightarrow} y_3$ then y_3 is a successor (not an immediate successor) of y_1 and we write $y_1 \overset{\{x_1,x_2\}}{\twoheadrightarrow} y_3$ (y_1 yields y_3 via the set $\{x_1, x_2\}$).

There are immediate successions between y_1 and y_2 and between y_2 and y_3. There is also a succession between y_1 and y_3, however, this is not an *immediate succesion.*

Definition xiv.
Given a tree structure τ over a multiset S, a tree structure σ over S is called a *subtree structure* of τ if and only if $\forall \; y, z \in_+ \sigma$ and $x \in_+ S$ such that $z \overset{\{x\}}{\twoheadrightarrow} y, \exists \; a, b \in_+ \tau$ with $y \subseteq a$ and $z \subseteq b$ such that $b \overset{\{x\}}{\twoheadrightarrow} a$.

In other words, a tree structure σ is a subtree structure of a tree structure τ if and only if it *inherits* all its immediate successions from the tree structure τ.

The following are some immediate consequences of this definition: A subtree structure of a tree structure may not necessarily be a submultiset of the tree structure and vice versa. There are subtree structures of a tree structure whose members are not elements of the tree structure. For example, for the tree structure $\tau = [\{544\}, \{423\}, \{322\}, \{211\}, \{433\}, \{322\}, \{322\}, \{211\}]$, the subtree structure $[\{54\}, \{43\}, \{32\}]$ of τ is not a submultiset of τ since all of its elements do not belong to τ. The submultiset $[\{544\}, \{322\}, \{211\}]$ of τ does not represent any subtree structure of τ, since it does not form a tree structure. The multiset $[\{544\}, \{433\}, \{21\}]$ is neither a submultiset of τ nor a subtree structure of τ while the multiset $[\{544\}, \{423\}, \{322\}]$ is both a submultiset of τ and a subtree structure of τ.

9 Tree Structure Based Representation of Membrane Structures

In this section, we apply the aforesaid technique to represent membrane structures. Membrane structures can be represented in the form of a tree ([10], p. 8). Let us consider the schematic representation of a membrane structure as in Figure 7. It is customary to label the largest membrane with the number 1, the next larger membrane with the number 2 and so on.

In our example, in order to have an intuitively clearer representation, we shall identify the membranes by their membrane sizes. For instance, a membrane labelled 2 will be identified by the membrane size M_2.

The schematic representation of the membrane structure (μ) in Figure 7 can be discussed as follows: M_1 contains M_2, M_3, M_5 and M_8; M_2 contains M_4 and M_6; M_3 contains M_7 while M_8 is empty. Also the ordering relations $M_2 > M_3 > M_5$ and $M_6 > M_4$ hold.

Let $S = \{M_i : i = 1, 2, \ldots, 9\}$. The size of a membrane is defined as the sum of the multiplicities of all the objects in the membrane ([9], p. 6). The domain set of the multisets consisting of the sizes of membranes in a membrane structure as elements, is wellfounded with the usual ordering, being a subset of the set $\mathbb{N}$ of natural numbers. Thus, S is wellfounded and it follows that a multiset having elements of S as elements of its elements is also wellfounded ([4]).

We note that the membrane sizes may change during the process of *transition*. In particular, the above relations may not hold especially for elementary membranes. However, this does not change the fact that the tree structure representations still apply. Figure 8 is the tree representation of the membrane structure in Figure 7.

The tree structure representation of the tree in Figure 8 is denoted by μ.

$$\mu = [\{M_1M_2M_3M_5M_8\}\{M_3M_7\}\{M_2M_4M_6\}].$$

We now present the above representation in greater details. The contents of the membranes are represented by a letter such as a_{ij} of

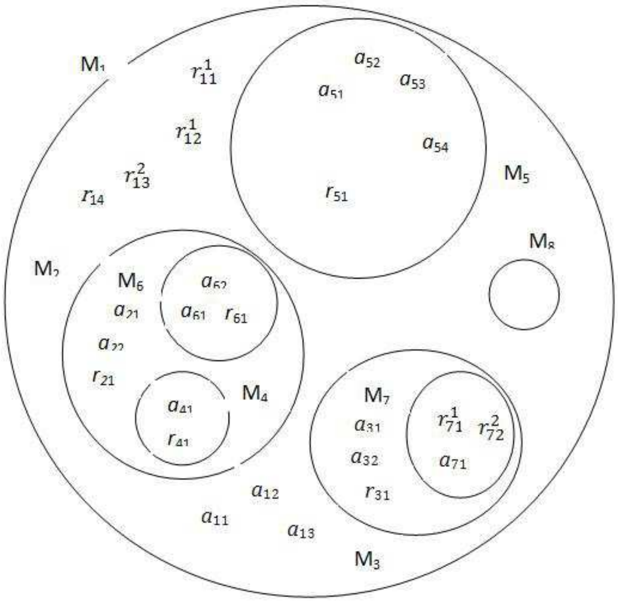


Figure 7. A membrane structure

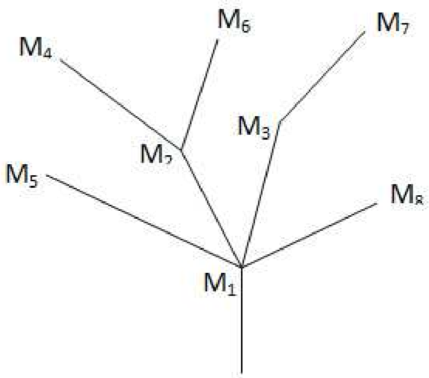


Figure 8. A tree representation of the membrane structure in Figure 7

the j^{th} element in the i^{th} membrane, while the m^{th} rule in the k^{th} membrane is represented by r^l_{km}, l being the rules' priorities if any. The membrane structure can further be expanded to show the contents of each membrane in the initial configuration of the system.

$$\begin{aligned}\mu \;=\; & [\{M_1[a_{11}a_{12}a_{13}r^1_{11}r^2_{12}r^3_{13}]M_2[a_{21}a_{22}r_{21}]M_3[a_{31}a_{32}r_{31}]\\ & M_5[a_{51}a_{52}a_{53}r_{51}]M_8[]\}\{M_3M_7[a_{71}r^1_{71}r^2_{72}]\}\\ & \{M_2M_4[a_{41}r_{41}]M_6[a_{61}a_{62}r_{61}]\}].\end{aligned}$$

The subscripts used for the labelling of the elements are just for illustration purpose and will not appear in the example we shall give below. The contents of membrane M_i have been grouped in the square brackets immediately following the membrane. If a membrane is contained in M_i, such containment follows the rule governing attachment in the tree structure. An empty membrane is denoted by an empty square bracket. Elementary membranes appear only once in the tree structure.

The first membrane is the skin membrane and is the only non-elementary membrane which is allowed to appear once in the tree structure. Any other membrane which is neither the skin membrane nor an elementary membrane will appear more than once, since it contains some other membranes.

10 Computation (An Example)

The following is the example, given in ([9], pp. 10-11), of a transition in a (cooperative) super-cell system. In this example we substitute *membrane structures* by *tree structures* (saw-like permutations of wellfounded multisets). Also, membranes are represented by subtree structures having both objects and rules (with associated rule priorities) as members. However, a membrane or a susbtree structure may exist having only rules as its members. The original tree structure representing the membrane structure is the initial configuration prior to the transitions.

Let us consider the following super-cell system of degree 4.

$\Pi = (V, \mu, M_1, \ldots, M_4, (R_1, \rho_1), \ldots, (R_4, \rho_4), 4)$,
$V = \{a, b, c, d\}$,
$\mu = [\{M_1 M_2 M_4\}\{M_2 M_3\}]$,
$M_1 = [aacr^1_{11} r^1_{12} r_{13}]$,
$M_2 = [ar_{21}]$,
$M_3 = [cdr_{31}]$,
$M_4 = [r_{41}]$,
$R_1 = \{r_{11} : c \to (c, in_4), r_{21} : c \to (b, in_4), r_{13} : a \to (a, in_2)b, dd \to (a, in_4)\}$,
$\rho_1 = \{r_{11} > r_{13}, r_{12} > r_{13}\}$,
$R_2 = \{r_{21} : a \to (a, in_3), ac \to \delta\}$,
$\rho_2 = \emptyset$,
$R_3 = \{r_{31} : a \to \delta\}$,
$\rho_3 = \emptyset$,
$R_4 = \{r_{41} : c \to (d, out), b \to b\}$,
$\rho_4 = \emptyset$.

$$C_0 : \mu = [\{M_1[aacr^1_{11} r^1_{12} r_{13}] M_2[ar_{21}] M_4[r_{41}]\}\{M_2 M_3[cdr_{31}]\}]$$

In the initial configuration C_0 we can apply a rule in membrane M_1 and one in membrane M_2. If we use the rule $c \to (b, in_4)$ in membrane M_1, the rule $b \to b$ can be applied non-stop and the computation will never end. Therefore, we will not use the rule $c \to (b, in_4)$, but the rule $c \to (c, in_4)$. Since both these rules can be applied and they have priorities over the rule $a \to (a, in_2)b$, this latter rule cannot be used. Hence, the object c is sent from membrane M_1 to membrane M_4 and at the same time the object a is sent from membrane M_2 to membrane M_3.

$$C_1 : \mu = [\{M_1[aar^1_{11} r^1_{12} r_{13}] M_2[r_{21}] M_4[cr_{41}]\}\{M_2 M_3[acdr_{31}]\}].$$

No rule can be applied on c in membrane M_1, hence the rule $a \to (a, in_2)b$ can be used. It will be used for both copies of a in membrane

M_1, and so two copies of a will be sent to membrane M_2 and two copies of b will remain in membrane M_1. At the same time, the rule $a \to \delta$ will be used in membrane M_3, dissolving it, and the rule $c \to (d, out)$ will be used in membrane M_4, sending a copy of d to membrane M_1. As a result of these operations, membrane M_1 will contain the string bbd, membrane M_2 will contain the string $aacd$, while membrane M_4 will contain no string; membrane M_3 no longer exists, therefore the rule $a \to (a, in_3)$ in membrane M_2 is useless for now.

$$C_2 : \mu = [\{M_1[bbdr^1_{11}r^1_{12}r_{13}]M_2[aacdr_{21}]M_4[r_{41}]\}].$$

The rule $ac \to \delta$ can be used in membrane M_2, dissolving it and releasing the remaining objects ad. Thus, membrane M_1 will contain the string $abbdd$.

$$C_3 : \mu = [\{M_1[abbddr^1_{11}r^1_{12}r_{13}]M_4[r_{41}]\}].$$

It is now possible for the first time to use the rule $dd \to (a, in_4)$ from membrane M_1. It consumes the two copies of d and sends a copy of a to membrane M_4. No further rule can be applied, and the "life" of the super-cell stops here.

$$C_4 : \mu = [\{M_1[abbr^1_{11}r^1_{12}r_{13}]M_4[ar_{41}]\}].$$

11 Conclusion

The paper is an attempt to indicate that a multiset-based tree model may prove useful in membrane computing and by extension to other computing devices, especially of biological orientation.

Moreover, the application of the saw-like permutation can be exploited in describing various algebraic properties of a *tree structure*, and hence, that of a *membrane structure*.

ACKNOWLEDGEMENT

We wish to acknowledge that most of the basic concepts are borrowed from the works of Dershowitz and Manna and that of Gh. Păun.

References

[1] D. A. Bailey, *Java Structure,* McGraw-Hill, 1999.

[2] W. Blizard, *Multiset theory*, Notre Dame Journal of formal logic, 1989, Vol. 30, pp. 36–66.

[3] W. Blizard, The development of multiset theory, *Modern logic*, 1991, Vol 1, pp. 319–352.

[4] N. Dershowitz, Z. Manna, *Proving Termination with Multiset Ordering*, Comm. ACM, 1975, Vol. 22, pp. 465–476.

[5] L. L. Dornhoff, F. E. Hohn, *Applied Modern Algebra*, Macmillan Publishing Co, 1978.

[6] J. Eyisi, *A Grammar of English, the Student's Companion,* Rex Charles publications, 2006.

[7] R. W. Hilton, *Managerial Accounting*, MacGraw-Hill/Irwin, 1996.

[8] J. G. Kemeny, *Introduction to finite Mathematics,* Prentice-Hall, 1957.

[9] Gh. Păun, *Computing with Membranes*, TUCS Report 208, Turku Centre for Computer Science, 1998.

[10] Gh. Păun, *Introduction to Membrane Computing*, Springer-Verlag, 2002.

[11] D. Singh, A. M. Ibrahim, Y. Tella, J. N. Singh, *Fundamentals of Multisets and their Applications*, Novi Sad J. Math, 2007, Vol. 37, pp. 73–92.

[12] D. J. Tailor, *Biological Science*, Cambridge, 1996.

[13] The P System web page: http://ppage.psystems.eu/.

D. Singh, C.M. Peter

Received October 30, 2010
Revised February 4, 2011

Department of Mathematics,
Ahmadu Bello University, Zaria, Nigeria

E–mails:
D. Singh: *mathdss@yahoo.com*,
C. M. Peter: *macpee3@yahoo.com*