
Multisorted tree algebra

Erick Patrick Zobo, Marcel Fouda Ndjodo

Department of Computer Sciences and Education Technologies (DITE), University of Yaounde I Yaounde, Cameroon

Email address:

epzobo@dite-ens.cm (E. P. Zobo), marcel.fouda@dite-ens.cm (N. M. Fouda)

To cite this article:

Erick Patrick Zobo, Marcel Fouda Ndjodo. Multisorted Tree Algebra. *Applied and Computational Mathematics*. Vol. 3, No. 6, 2014, pp. 295-302. doi: 10.11648/j.acm.20140306.12

Abstract: This paper introduces basic concepts describing a hierarchical algebraic structure called multisorted tree algebra. This structure is constructed by placing multisorted algebra at the bottom of a hierarchy and placing at other intermediate nodes the aggregation of algebras placed at their immediate subordinate nodes. These constructions are different from the one of subalgebras, homomorphic images and product algebras used to characterize varieties in universal algebra theory. The resulting hierarchical algebraic structures cannot be easily classified in common universal algebra varieties. The aggregation method and the fundamental properties of the aggregated algebras have been presented with an illustrative example. Multisorted tree algebras spans multisorted algebra concepts and can be used as modelling framework for building hierarchical abstract data types for information processing in organizations.

Keywords: Multisorted Algebra, Hierarchy, Aggregation, Abstract Data Type

1. Introduction

There are two main approaches to the study of universal algebra as mentioned in [1]. The first approach looks for the constructions on algebras which produce new algebras of the same type. Here, the three main constructions available for producing new algebras from given ones are: the construction of subalgebras, homomorphic images and product algebras. Classes of algebras which are closed to such constructions are called varieties [2]. Secondly, to the study of abstract algebras involves the study of terms and identities. Here, the objective is to define or classify algebras according to the set of identities or axioms they satisfy [3]. A more general approach to study algebraic structures and structural thinking is the category theoretical one as in [4].

Many new algebraic concepts or theories have been proposed in universal algebra literature according to the specificities of their application areas. One can cite for instance, application areas such as multisorted algebras (also called many sorted algebras) [5], hidden algebra [6, 7], ordered sorted algebra [8]. But other algebraic concepts have also been introduced with quite a different description of their structures as the coalgebras [9]. The objectives of these proposals classified under algebraic engineering research have generally been to build algebraic structures for modeling specific computational problems in various scientific domains or to account for the realities of modern

software [10].

However, some complex algebraic computation systems cannot be easily modeled and studied by these various proposed algebraic structures. An example of such complex algebraic structures is the hierarchical computations of data. In such systems as shown in Figure 1, algebraic computations are done at the bottom nodes of a hierarchical system and the results are aggregated and used as inputs for algebraic computations at the highest nodes of the hierarchy till the top. A specific case of such systems is studied by [11].

Among some of the difficulties to study such algebraic structures is the fact that they cannot simply be described as structures consisting of one or more sets of objects with one or more operations on the objects [1, 12]. Hence, strictly speaking they are not universal algebra according to the definition even though they are constructed from them. Input data are given at the bottom nodes to generate a term with a hierarchical shape of outputs data that can be studied per levels, per path or in different ways. Also, the hierarchy studied is different from a diagram in the category of universal algebras with the same signature and cannot be easily described by classic universal algebras.

Therefore, there is a need to describe a framework for the study of such hierarchical algebraic structures constructed by aggregating multisorted universal algebras terms. Furthermore, many information systems processing in decision making systems of hierarchical organizations are

configured in with a structure similar to their hierarchy [13, 14]. Therefore, the availability of a hierarchical algebraic framework will not only expand the algebraic structures types but also contribute to giving an appropriate modeling framework for studying hierarchical computations.

The purpose of this paper is to introduce the basic concepts of multisorted tree algebra and present an example of such structures. We also discuss the used of this framework for modeling problems with hierarchical computational structures.

In section 2, we have reminded of the main concepts of multisorted algebra. The concepts of aggregations of mansorted algebras and fundamentals investigations are presented in section 3. In section 4, we introduce multisorted tree algebra concepts with the restriction on the same signature for algebra on the node of the hierarchy and with the construction of algebra in nodes that are different form the leaves by aggregation of algebras in the leaves. As an illustrative example, a matrix-based multisorted tree algebra is presented. In section 5, we discuss about some mathematical properties of multisorted tree algebras and their questioning in the theoretical framework of universal algebras. Section 6 presents the conclusion and perspectives.

2. Basic Concepts of Classical Algebraic Specification Theory

This section has two subsections. In the first Section we describe the concepts of signature, Algebra and homomorphism. In the second the concepts of term, equation and specification are presented

2.1. Signature, Algebra, Homomorphism

The concepts of the signature of a many sorted algebra as well as the one of many sorted algebra introduced in this section are similar to the definition used by Goguen in [6, 15].

Definition 2.1. (*S*-sorted set) Let S be a nonempty set, whose elements are called sorts, and S -sorted set A is a family of sets $\{A_s, s \in S\}$.

Definition 2.2. (*S*-sorted signature) Let S be a nonempty set, with elements called sorts.

- An S -sorted signature Σ is an $(S^* \times S)$ -sorted set $\{\sum_{w,s}(w, s) \in S^* \times S\}$.
- The elements of $\sum_{w,s}$ are called operation symbols of arity w , and rank (w, s) ; in particular, $\sigma \in \sum_{\square, s}$, s is a constant symbol of sort s .
- Σ is ground signature iff $\sum_{\square, s} \cap \sum_{\square, s'} = \emptyset$ whenever $s \neq s'$.

By convention, $|\Sigma| = \cup_{w,s} \sum_{w,s}$, and $\Sigma' \subset \Sigma$ means $\sum'_{w,s} \subset \sum_{w,s}$ for each w, s . Similarly, union is defined by $(\Sigma' \cup \Sigma)_{w,s} = \sum'_{w,s} \cup \sum_{w,s}$. A common special case is union with a ground signature X , for which we use the notation $\Sigma(X) = \Sigma \cup X$.

Definition 2.3. (Σ -algebra) Let Σ be a S -sorted signature.

A Σ -algebra A is an S -sorted set A with an interpretation of Σ in A , which is a family of arrows $i_{s_1 \dots s_n, s}: \sum_{s_1 \dots s_n, s} \rightarrow [A^{s_1 \dots s_n, s} \rightarrow A_s]$ for each rank $(s_1 \dots s_n, s) \in (S^* \times S)$, which interpret the operation symbols in Σ as actual operations on A . For constant symbols, the interpretation is given by $i_{\square, s}: \sum_{\square, s} \rightarrow A_s$ usually we write just σ for $i_{s_1 \dots s_n, s}(\sigma)$ but if we need to make the dependance on A and the rank (w, s) explicit, we may write $\sigma_{w,s}^A$. The set A_s is called the carrier of A of sort s .

Definition 2.4. (Σ -algebra over commutative semigroups) Let Σ be an S -sorted signature. Let $(E, *) = \{(E_s, *_s)\}_{s \in S}$ be an S -sorted signature family of commutative semigroups. A Σ -algebra over $(E, *)$ is a Σ -algebra A such that $A_s \subseteq E_s$ for each $s \in S$.

Definition 2.5. (Σ -subalgebra) Let Σ be an S -sorted signature and A be a Σ -algebra. A Σ -algebra B is a subalgebra of A when the following conditions are satisfied.

- $B_s \subseteq A_s$ for each $s \in S$;
- For each $\sigma \in \sum_{(w,s)}$ with $(w, s) \in (S^* \times S)$, the graph of $\sigma_{w,s}^B$ is include in the graph of $\sigma_{w,s}^A$.

Definition 2.6. (Σ -homomorphism) Let A and B be Σ -algebras. A Σ -homomorphism h is an S -sorted arrow noted $h: A \rightarrow B$ such that:

$$h_s(\sigma_A(a_1, \dots, a_n)) = \sigma_B(h_s(s_1), \dots, h_s(s_n)),$$

for each $\sigma \in \sum_{(s_1 \dots s_n, s)}$ and $a_i \in A_{s_i}$ for $i = 1, \dots, n$, and $h_s(c_A) = c_B$ for each constant symbol $c \in \sum_{\square, s}$.

2.2. Term, Equation and Specification

Definition 2.7. (Σ -terms) Let Σ be an S -sorted signature.

- The S -sorted set T_Σ of Σ -terms is the smallest set of lists of symbols that contains the constants, $\sum_{\square, s} \subseteq T_{\Sigma, s}$, and such that given $\sigma \in \sum_{(s_1 \dots s_n, s)}$ and $t_i \in T_{\Sigma, s_i}$, then $\sigma(t_1, \dots, t_n) \in T_{\Sigma, s}$.
- We view T_Σ as a Σ -algebra by interpreting $\sigma \in \sum_{(\square, s)}$ as just σ , and $\sigma \in \sum_{(s_1 \dots s_n, s)}$ as the operation sending the n -uplet (t_1, \dots, t_n) to the list $\sigma(t_1, \dots, t_n)$. Then T_Σ is called the Σ -term algebra.

Theorem 2.1. (Initiality) Given a signature Σ with no overloaded constants and a Σ -algebra M , there is a unique Σ -homomorphism $T_\Sigma \rightarrow M$.

Definition 2.8. (Σ -equations) A Σ -equation consists of a ground signature X of variable symbols (disjoints from Σ) plus two $\Sigma(X)$ -terms of the same sorts $\in S$. We may write such an equation abstractly in the form $(\forall X)t = t'$ and concretely in the form $(\forall x, y, z)t = t'$ when $W = \{x, y, z\}$ and the sorts of x, y and z can be inferred from their uses in t and in t' .

Definition 2.9. A specification is a pair (Σ, E) , consisting of a signature Σ and a set E of Σ -equations.

Remark 2.1. The Σ -term algebra T_Σ serves as a standard model for a specification $P = (\Sigma, \emptyset)$. Given Σ and a ground signature X disjoint from Σ , we can form the $\Sigma(X)$ -algebra $T_{\Sigma(X)}$ and then view it as a Σ -algebra by forgetting the names of the new constants in X ; lets denote

this Σ -algebra by $T_\Sigma(X)$.

Proposition 2.1. Let A be a Σ -algebra and an interpretation $a: X \rightarrow A$, there is a unique Σ -homomorphism $\tilde{a}: T_\Sigma(X) \rightarrow A$ extending a , in the sense that $\tilde{a}_s(x) = a_s(x)$ for each $x \in X_s$ and $s \in S$.

Definition 2.10. (satisfaction, variety)

- A Σ -algebra A satisfies a Σ -equation $(\forall X)t = t'$ iff for every $a: X \rightarrow A$ we have $a(t) = a(t')$ in A , written $A \models_\Sigma (\forall X)t = t'$
- A Σ -algebra A satisfies a set E of Σ -equations iff it satisfies each one, written $A \models_\Sigma E$. We may also say that A is a P -algebra, and write $A \models P$ where $P = (\Sigma, E)$.
- The class of all algebras that satisfy P is called variety defined by P .

Theorem 2.2. Given a specification $P = (\Sigma, E)$, any two initial P -algebra are Σ -isomorphic.

Definition 2.11. The abstract data type (abbreviated ADT) defined by a specification P is the class of all initial P -algebras.

3. Multisorted Algebras Aggregations

In this section we introduce a new concept of universal algebras aggregation. By so doing, we first introduce the concept of aggregation operator which is an operator that generate a new algebra supports from the supports of a family of algebras. We then show that when this aggregation operator is compatible with all operations from a universal algebra signature, it generated another universal algebra that satisfies the common properties of the aggregated universal algebras. The first section describes the aggregation operators. Our restriction is on the aggregation of universal algebra with the same signature. The second section focus on the mathematical properties of aggregated universal algebras.

3.1. Aggregation Operators, Aggregation of Algebras and Homomorphisms

Definition 3.1. (θ -Aggregation operator) Let $\langle E, * \rangle$ be a commutative semigroup. Let n be an integer and E^n denote the cartesian product of E defined by n -uplets of elements in E . Let $E^\infty = \bigcup_{n=1}^\infty E^n$. Let S be a set of sorts and $w = s_1 \dots s_n \in S^*$. Let Σ be an S -sorted signature. Let $\{A_{s_1}, \dots, A_{s_n}\}$ be a family of sorted sets and $A_w = A_{s_1} \times \dots \times A_{s_n}$.

A function $\theta: E^\infty \rightarrow E$ is called an aggregator on E .

- A $*$ -aggregator on $\langle E, * \rangle$ in an aggregator $\theta: E^\infty \rightarrow E$ such that for $(x_1, \dots, x_m) \in E^m$ $\theta(x_1, \dots, x_m) = x_1 * \dots * x_m$.
- An $(w, *)$ -aggregator is an aggregator $\theta: A_w^\infty \rightarrow A_w$ defined by a family of $*$ -aggregators $\{\theta_{s_i}: A_{s_i}^\infty \rightarrow A_{s_i}\}_{1 \leq i \leq n}$ such that for each operation symbol $\sigma \in \Sigma_{w,s}$ of arity (w, s) where $w = s_1 \dots s_m$, the following equation is satisfied:

$$\theta_s \left[\sigma_A \left((a_{1j})_j, \dots, \sigma_A \left((a_{nj})_j \right) \right) \right] = \sigma_A \left(\theta_{s_1} [(a_{i1})_i], \dots, \theta_{s_m} [(a_{im})_i] \right)$$

For a finite n -uplet $X = (x_1, x_2, \dots, x_n) \in E^n$ we denote the aggregation $\theta(X)$ of X by $\theta_{i=1}^n x_i$.

In the literature, the symbol Σ generally denotes the aggregation operator for the addition operator of finite or infinite elements. The union operator \cup is an aggregation operator that constructs the union of a family of sets.

Definition 3.2. (Aggregation of Σ -algebras over commutative semigroups) Let $A = \{A_i\}_{1 \leq i \leq n}$ be a family of Σ -algebras where each A_i is a Σ -algebra over commutative semigroups $E_i, 1 \leq i \leq n$. Let θ_Σ be a Σ -aggregator. The aggregation of Σ -algebras in $\{A_i\}_{1 \leq i \leq n}$ by θ_Σ is the S -sorted family of sets noted $\theta_\Sigma(A)_s = \left\{ \theta_s \left(\sigma_{A_s} (a_{ij}) \right), \text{ for all } \sigma \in \Sigma_{w,s}, w = s_1 \dots s_n \in S^*, a_{ij} \in A_{s_j}, 1 \leq i \leq n, 1 \leq j \leq m \right\}$.

Theorem 3.1. (Aggregation of Σ -algebras) Let $A = \{A_i\}_{1 \leq i \leq n}$ be a family of Σ -algebras where each A_i is a Σ -algebra over commutative semigroups $E_i, 1 \leq i \leq n$. Let θ_Σ be a Σ -aggregator. The aggregation of Σ -algebras in $\{A_i\}_{1 \leq i \leq n}$ by θ_Σ is a Σ -algebra.

Proof.

The aggregator of A with θ_Σ is the Σ -algebra $\theta_\Sigma(A)$ defined as follows:

- For each sorts $s \in S$ corresponds the support set
- $\theta_\Sigma(A)_s = \left\{ \theta_s \left(\sigma_{A_s} (a_{ij}) \right), \text{ for all } \sigma \in \Sigma_{w,s}, w = s_1 \dots s_n \in S^*, a_{ij} \in A_{s_j}, 1 \leq i \leq n, 1 \leq j \leq m \right\}$.
- For each symbol $\sigma \in \Sigma_{w,s}$, where $w = s_1 \dots s_n$ corresponds the operation $\sigma_{\theta_\Sigma(A)}: \theta_\Sigma(A)_{s_1} \times \dots \times \theta_\Sigma(A)_{s_m} \rightarrow \theta_\Sigma(A)_s$
- such that: $\sigma_{\theta_\Sigma(A)} \left(\theta_{s_1} [(a_{i1})_i], \dots, \theta_{s_m} [(a_{im})_i] \right) = \theta_s \left[\sigma_{A_1} \left((a_{1j})_j \right), \dots, \sigma_{A_n} \left((a_{nj})_j \right) \right]$

Definition 3.3. (Aggregation of Σ -homomorphisms) Let $h = \{h_i: A_i \rightarrow B_i, 1 \leq i \leq n\}$ be a family of Σ -homomorphisms of Σ -algebras over commutative semigroups. Let θ_Σ be Σ -aggregator. The aggregation of Σ -homomorphisms in $\{h_i: A_i \rightarrow B_i, 1 \leq i \leq n\}$ by θ_Σ is the function defined as follows:

For each

$$\theta_s(\sigma_{A_i}(a_{ij})) \in \theta_\Sigma(A),$$

$$\theta_\Sigma(h)_s(\theta_s(\sigma_A(a_{ij}))) := \theta_s(\sigma_B(h_s(a_{ij})))$$

Theorem 3.2. (Aggregation of Σ -homomorphisms). Let $h = \{h_i: A_i \rightarrow B_i, 1 \leq i \leq n\}$ be a family of Σ -homomorphisms and let θ_Σ be a Σ -aggregator. The aggregation of Σ -homomorphisms in $\{h_i: A_i \rightarrow B_i, 1 \leq i \leq n\}$ by θ_Σ is a Σ -homomorphism.

Proof:

Let $A = \{A_i, 1 \leq i \leq n\}$ and $B = \{B_i, 1 \leq i \leq n\}$ The aggregation of $\{h_i: A_i \rightarrow B_i, 1 \leq i \leq n\}$ with θ_Σ is the Σ -homomorphism $\theta_\Sigma(h): \theta_\Sigma(A) \rightarrow \theta_\Sigma(B)$ defined as follows:

For each $\theta_s(\sigma_{A_i}(a_{ij})) \in \theta_\Sigma(A)$, we have

$$\theta_\Sigma(h)_s(\theta_s(\sigma_{A_i}(a_{ij}))) := \theta_s(\sigma_B(h_s(a_{ij})))$$

To prove that the property of homomorphism is satisfied, let

$$\sigma \in \sum_{s_1 \dots s_n, s}$$

let the terms $a_1 = \theta_{s_1}(\gamma_{A_1}(a_{1i})) \in \theta_{\Sigma(A)}_{s_1}$, ..., and $a_n = \theta_{s_n}(\gamma_{A_n}(a_{ni})) \in \theta_{\Sigma(A)}_{s_n}$

We have the following equations:

$$\begin{aligned} & \theta_\Sigma(h)(\sigma_{\theta_\Sigma(A)}(\theta_{s_1}(\gamma_{A_1}(a_{1i})), \dots, \theta_{s_n}(\gamma_{A_n}(a_{ni})))) \\ &= \theta_\Sigma(h)(\theta_{s_1}(\sigma_{\theta_\Sigma(A)}(\gamma_{A_1}(a_{1i}))), \dots, \theta_{s_n}(\gamma_{A_n}(a_{ni})))) \end{aligned}$$

because θ is a Σ -aggregator.

$$= \theta_s(\sigma_{\theta_\Sigma(B)}(h_{s_1}(\gamma_{A_1}(a_{1i})), \dots, h_{s_n}(\gamma_{A_n}(a_{ni}))))$$

by the definition of $\theta_\Sigma(h)$

$$= \sigma_{\theta_\Sigma(B)}(\theta_{s_1}(h_{s_1}(\gamma_{A_1}(a_{1i}))), \dots, \theta_{s_n}(h_{s_n}(\gamma_{A_n}(a_{ni}))))$$

by de definition of θ

$$= \sigma_{\theta_\Sigma(B)}(\theta_{s_1}(\gamma_{B_1}(h_{s_1}(a_{1i}))), \dots, \theta_{s_n}(\gamma_{B_n}(h_{s_n}(a_{ni}))))$$

because h_{s_i} are homomorphisms

$$= \sigma_{\theta_\Sigma(B)}(\theta_\Sigma(h)_{s_1}(\theta_{s_1}(\gamma_{B_1}(a_{1i}))), \dots, \theta_\Sigma(h)_{s_n}(\theta_{s_n}(\gamma_{B_n}(a_{ni}))))$$

by the definition of $\theta_\Sigma(h)$.

3.2. Some Properties of Aggregation of Algebras

Theorem 3.3. Let $(\forall X)t = t'$ be a Σ -equation. Let $A = \{A_i, 1 \leq i \leq n\}$ be a family of Σ -algebras such that for each $1 \leq i \leq n$ $A_i \models_\Sigma (\forall X)t = t'$ Then $\theta_\Sigma(A) \models (\forall X)t = t'$.

Proof.

Let θ_Σ an aggregator operator. We assume that the sort of t and t' is $s \in S$. Let $a_s: X_s \rightarrow \theta_\Sigma(\{A_i\}_{1 \leq i \leq n})$ be an interpretation of variables from X_s in $\theta_\Sigma(\{A_i\}_{1 \leq i \leq n})$. There is a family of interpretations $a_{i_s}: X_s \rightarrow A_{i_s}$ such that $a_{i_s}(x) = \theta_\Sigma(\{a_{i_s}(x)\}_{1 \leq i \leq n})$

Proposition 2.1 implies that for each $i, 1 \leq i \leq n$ there is an unique homomorphism $\tilde{a}_{i_s}: T_\Sigma(X)_s \rightarrow A_{i_s}$ expanding a_{i_s} . i.e. such that for $x \in X$, $\tilde{a}_{i_s}(x) = a_{i_s}(x)$. Also, there is a unique homomorphism $\tilde{a}_{i_s}: T_\Sigma(X)_s \rightarrow A_{i_s}$

$\tilde{a}_{i_s}: T_\Sigma(X)_s \rightarrow \theta_\Sigma(\{A_{i_s}\}_{1 \leq i \leq n})$ expanding a_s . To verify the satisfaction of equation $(\forall X)t = t'$ we have:

$$\tilde{a}_{i_s}(t) = \theta_\Sigma(\{a_{i_s}(t)\}_{1 \leq i \leq n})$$

$$= \theta_\Sigma(\{a_{i_s}(t')\}_{1 \leq i \leq n}) \text{ because } A_i \models_\Sigma (\forall X)t = t'$$

$$= \tilde{a}_{i_s}(t')$$

Hence, $\theta_\Sigma(\{A_i\}_{1 \leq i \leq n}) \models_\Sigma (\forall X)t = t'$

Corollary 3.1. A Σ -aggregation of Σ -algebras from a

variety defined by (Σ, P) is a Σ -algebra of the same variety.

4. Multisorted Tree Algebra

We propose in this section the foundational concepts of multisorted tree algebra as construction of hierarchy structured algebras by placing universal algebras of the same signature at each leaf of the bottom, and by completing the other nodes till the top by aggregation of universal algebras of their direct subordinate nodes. This section has three subsections. The first presents the concept of valued rooted tree. The second introduces the concept of multisorted tree algebra as a valued rooted tree of algebras, and the third is about basic concepts of algebraic specification in the framework of multisorted tree algebra.

4.1. (θ, Δ) -Valued Rooted Tree

The concept of rooted trees defined below is used to describe hierarchical structure of some organization. The root is the highest authority and the leaves are the executives.

4.1.1. Tree, Rooted Tree, Subtree

1. A tree $\Delta = (\Delta_0, \Delta_1)$ is a connected undirected graph with no simple circuits. The set Δ_0 is the set of nodes or vertices, and the set $\Delta \subseteq \Delta_0 \times \Delta_1$ is the set of edges or arcs.
2. A rooted tree noted $\langle \Delta, R \rangle$ is a tree Δ in which a node R has been designated as the root and every edge is directed away from the root.
3. Let Δ be a tree, if A is a node in Δ_0 other than the root, the parent or the direct hierarchical superior of A is the unique node B in Δ_0 such that there is a directed edge from B to A . when B is the parent of A , then A is called a child or a direct subordinate of B .
4. Let Δ be a tree and $A \in \Delta_0$ a node of Δ . The ancestors or the hierarchical superiors of A are the nodes in the path from the root to A , excluding A . the descendants or the subordinates of a node A are those nodes that have A as an ancestor.
5. A node of a tree Δ is called leaf, or ground node or operating agent or unit if it has no children. Nodes that have children are called intermediate management agent.
6. Let A be a node in a tree Δ , the subtree Δ_A of Δ is a rooted tree with A as its root is the subgraph of the tree consisting of A and all its descendants in Δ .

4.1.2. Tree Levels and Height

1. Let A be a node of a rooted tree $\langle \Delta, R \rangle$, the positive integer l_A is the level of A if it is representing the number of edges of the unique path from the root R to A . The level of the root of Δ is equal to zero. The height of a rooted tree Δ is the maximum of the levels of its nodes.
2. Let A be a node of a tree Δ , let l be an integer such that $l_A \leq l$, the set of nodes of Δ of level l that are children of A is denoted by Δ_A^l .

3. The width m of a tree Δ is the integer corresponding to the number of leaves in Δ .
4. Let m and h be two integers, an (h, m) -rooted tree $\langle \Delta, R \rangle$ is a rooted tree with m leaves and for which the level of each leaf is h . Each integer l , with $0 \leq l \leq h$ is called a hierarchical level of Δ .
5. Let Δ be an (h, m) -rooted tree, let l be an integer, $0 \leq l \leq h$ and let A be a node of Δ . We denote by Δ^l the set of nodes at the hierarchical level l . We denote by Δ^h the set of ground nodes of Δ . We denote by Δ_A^h the ground nodes of Δ that are subordinate of A .

Proposition 4.1. Let $\langle \Delta, R \rangle$ be an (h, m) -rooted tree, let q be two integers and A be a node of Δ

- For each integer l such that $0 \leq l < h$, $\{\Delta_A^q\}_{A \in \Delta^l}$ is a partition of Δ^h
- For each integers l and q such that $0 \leq l < q \leq h$, $\{\Delta_A^q\}_{A \in \Delta^l}$ is a partition of Δ^q .

Definition 4.1. (Valued rooted tree) Let V be a set of values. An (h, m) -rooted tree is a triplet $\langle \Delta, S, f \rangle$ where $\langle \Delta, R \rangle$ is an (h, m) -rooted tree and $f: \Delta_0 \rightarrow V$ a valuation of nodes of Δ .

Definition 4.2. (Θ -extension, extensible operation on tree) Let $\langle \Delta, R \rangle$ be an (h, m) -rooted graph. Let $\langle E, * \rangle$ be a commutative semigroup for which the associated aggregation operator symbol is Θ . Let $f: \Delta^h \rightarrow E$ be a function. The function $\tilde{f}: \Delta_0 \rightarrow E$ is a Θ -extension to Δ_0 of the function f defined on Δ^h when it is defined as follows:

- For each $A \in \Delta^h$, $\tilde{f}(A) = f(A)$;
- For each $A \in \Delta_0$ such that $A \notin \Delta^h$; $\tilde{f}(A) = \Theta(\{f(S)\}_{S \in \Delta_A^h})$

The function f is called a Θ -extensible operation from Δ^h to Δ_0

Definition 4.3. ((Θ, Δ) -valued rooted tree) Let $\langle E, * \rangle$ be a commutative semigroup with Θ the associated aggregation operator symbol. A (Θ, Δ) -valued rooted tree is a triplet $\langle \Delta, R, f \rangle$ where $\langle \Delta, R \rangle$ is an (h, m) -rooted tree and $f: \Delta^h \rightarrow E$ is a Θ -extensible function from Δ^h to Δ_0 .

4.2. (Σ, Θ, Δ) -Multisorted Tree Signature, Algebra and Homomorphism

Definition 4.4. ((Σ, Θ, Δ) -multisorted tree signature) Let $\langle \Delta, R \rangle$ be a rooted tree. A (Δ, S) -multisorted tree signature is a Δ_0 -indexed family of S -multisorted signatures $\{\Sigma^Y\}_{Y \in \Delta_0}$. When $\Sigma^Y = \Sigma$ for all $Y \in \Delta_0$ and Θ is a Σ -aggregator, the (Δ, S) -multisorted tree signature is called a (Σ, Θ, Δ) -multisorted tree signature and denoted $\Sigma_{\Delta, \Theta}$.

Definition 4.5. ((Σ, Θ, Δ) -multisorted tree algebra) Let $\langle E, * \rangle$ be a commutative semigroup with Θ as the associated aggregation operator symbol. Let $Sub(\langle E, * \rangle)$ be the class of commutative sub-semigroup of $\langle E, * \rangle$. Let Σ be a signature and $Alg_E(\Sigma)$ be the class of Σ -algebras with carriers in $Sub(\langle E, * \rangle)$. Let $\langle \Delta, R \rangle$ be a rooted tree. A (Σ, Θ, Δ) -multisorted tree algebra A is a tuple $\langle \Sigma, \Theta, \Delta, R, f \rangle$ such that $\langle \Delta, R, f \rangle$ is a Θ -valued rooted

tree with $f: \Delta_h \rightarrow Alg_E(\Sigma)$ a Θ -extensible function from Δ_h to Δ_0 and such that its Θ -extension \tilde{f} is defined as follows:

- For each $X \in \Delta^h$, $\tilde{f}(X) = f(X)$;
- For each $X \in \Delta_0$ such that $X \notin \Delta^h$; $\tilde{f}(X) = \Theta(\{f(S)\}_{S \in \Delta_X^h})$, is the Σ -algebra obtained by Θ -aggregation of the Σ -algebras from all subordinated leaves of X .

Definition 4.6. ((Σ, Θ, Δ) -multisorted tree homomorphism)

Let A and B be two (Σ, Θ, Δ) -multisorted tree algebras, a (Σ, Θ, Δ) -multisorted tree homomorphism $h: A \rightarrow B$ Let $\langle E, * \rangle$ be a commutative semigroup with Θ as the associated aggregation operator symbol. Let $Sub(\langle E, * \rangle)$ be the class of commutative sub-semigroup of $\langle E, * \rangle$. Let Σ be a signature, $Alg_E(\Sigma)$ be the class of Σ -algebras with carriers in $Sub(\langle E, * \rangle)$ and let $Hom_E(\Sigma)$ the class of Σ -homomorphism between elements of $Alg_E(\Sigma)$. Let $\langle \Delta, R \rangle$ be a rooted tree. A (Σ, Θ, Δ) -multisorted tree homomorphism is a tuple $\langle \Sigma, \Theta, \Delta, R, \alpha \rangle$ such that $\langle \Delta, R, f \rangle$ is a Θ -valued rooted tree with $\alpha: \Delta_h \rightarrow Hom_E(\Sigma)$ a Θ -extensible function from Δ_h to Δ_0 and such that its Θ -extension f is defined as follows:

- For each $X \in \Delta^h$, $\alpha(X) = \alpha(X) : f(X) \rightarrow q(X)$ is a Σ -homomorphism on $Hom_E(\Sigma)$;

For each $X \in \Delta_0$ such that $X \notin \Delta^h$; $\alpha(X) : \Theta(\{f(S)\}_{S \in \Delta_X^h}) \rightarrow$

$\Theta(\{g(S)\}_{S \in \Delta_X^h})$, is the Σ -homomorphism obtained by Θ -aggregation of the Σ -homomorphisms from all subordinated leaves of X .

Hence,

$$\alpha(X)_s (\sigma_A(\Theta(\{a_{1,1}, \dots, a_{m,1}\}), \dots, \Theta(\{a_{1,n}, \dots, a_{m,n}\}))) = \Theta\{\sigma_B(a_{1,1}, \dots, a_{1,n}), \dots, \sigma_B(a_{m,1}, \dots, a_{m,n})\}$$

4.3. (Σ, Θ, Δ) -Multisorted Tree Terms, Equations, Specification

Definition 4.7. ((Σ, Θ, Δ) -multisorted tree terms) Let $A = \langle \Sigma, \Theta, \Delta, R, f \rangle$ be a multisorted tree algebra. The (Σ, Θ, Δ) -set of multisorted ground tree terms is the valued rooted tree $T_{\Sigma, \Theta, \Delta}$ defined by the tuple $\langle \Sigma, \Theta, \Delta, R, f, t \rangle$ such that $\langle \Delta, R, f \rangle$ is a Θ -valued rooted tree with $t: \Delta_h \rightarrow T_{\Sigma}$ a Θ -extensible function from Δ_h to Δ_0 and such that its Θ -extension \tilde{f} is defined as follows:

- $\tilde{f}(X) = T_{\Sigma}$ for $X \in \Delta_h$,
- $\tilde{f}(X) = \Theta\{f(A), A \in \Delta_X^h\}$ for $X \notin \Delta_h$.

$T_{\Sigma, \Theta, \Delta}$ is the (Σ, Θ, Δ) -multisorted tree of the terms.

Theorem 4.1. (Initiality) Given a multisorted signature Σ with no overloaded constants and a (Σ, Θ, Δ) -multisorted tree algebra A , there is a unique (Σ, Θ, Δ) -multisorted tree homomorphism $\text{mea } T_{\Sigma, \Theta, \Delta} \rightarrow A$.

Definition 4.8. ((Σ, Θ, Δ) -multisorted tree specification)

Let $\langle \Sigma, \Theta, \Delta \rangle$ be a multisorted tree signature.

- A $\langle \Sigma, \Theta, \Delta \rangle$ -tree equation consist of a ground $\langle \Sigma, \Theta, \Delta \rangle$ -tree signature X of variable symbols (disjoints from Σ) plus two $\langle \Sigma(X), \Theta, \Delta \rangle$ -tree terms of the same (Δ, S) -sort. We may write such as equation

abstractly in the form $(\forall X)t = t'$.

- A $\langle \Sigma, \Theta, \Delta \rangle$ -specification is a pair $(\langle \Sigma, \Theta, \Delta \rangle, E)$ consisting of a $\langle \Sigma, \Theta, \Delta \rangle$ -tree signature and a set of $\langle \Sigma, \Theta, \Delta \rangle$ -equations.

Proposition 4.2. Given $\langle \Sigma, \Theta, \Delta \rangle$ -tree algebra A and an interpretation $a : X \rightarrow A$, there is a unique (Σ, Θ, Δ) -homomorphism $a : T_{\Sigma, \Theta, \Delta}(X) \rightarrow A$ extending a in the sense that $a_s(x) = a_s(x)$ for each $x \in X_s$ and $s \in S$.

Definition 4.9. (Σ, Θ, Δ) -multisorted tree variety

- A $\langle \Sigma, \Theta, \Delta \rangle$ -tree algebra A satisfies a $\langle \Sigma, \Theta, \Delta \rangle$ -tree equation $(\forall X)t = t'$ iff for every $a : X \rightarrow A$, we have $a(t) = a(t')$ in A , written $A \models_{\Sigma, \Theta, \Delta} (\forall X)t = t'$.
- A $\langle \Sigma, \Theta, \Delta \rangle$ -tree algebra A satisfies a set E of equations iff it satisfies each one, written $A \models_{\Sigma, \Theta, \Delta} E$. We also say that A is a $\langle \Sigma, \Theta, \Delta, E \rangle$ -tree algebra. The class of $\langle \Sigma, \Theta, \Delta, E \rangle$ -tree algebras is called a variety defined by $\langle \Sigma, \Theta, \Delta, E \rangle$.

Definition 4.10.

(Σ, Θ, Δ) -Tree Abstract Data Type) The tree abstract data type (abbreviated TADT) defined by a specification $P = \langle \Sigma, \Theta, \Delta, E \rangle$ is the class of initial P -algebras.

4.4. Matrix-Based Multisorted Tree Algebra

This section presents an example of multisorted tree algebra main components. The sorts are the various types of $(n \times p)$ -matrix. The signature operations are the operations on matrix defined below. We limit our description to the signature and the aggregation operator. The construction of matrix based multisorted algebras can be done for any hierarchy by following the description above.

Definition 4.11. (Matrix and vectors over set) Let E be a non-empty set and n, p two positive integers. An $(n \times p)$ -matrix A over E , also called a matrix of type (E, n, p) , is a function

$$A: \{1, \dots, n\} \times \{1, \dots, p\} \rightarrow E, \\ (i, j) \mapsto a_{ij}$$

We write $A = (a_{ij})_{1 \leq i \leq n, 1 \leq j \leq p}$ and the elements a_{ij} are called the components of A .

- When $n = 1$, the matrix $A = (a_{11}, \dots, a_{1p})$ simply denoted by $A = (a_1, \dots, a_p)$ is a p -horizontal vector.

- When $p = 1$ then $A = \begin{pmatrix} a_{11} \\ \vdots \\ a_{n1} \end{pmatrix}$ simply denoted by

$$A = \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix} \text{ is a } n\text{-vertical vector.}$$

- When $n = p = 1$, then A is an element a of E .
- When the context is clearly defined, the matrix A is denoted by (a_{ij}) and the set of all $n \times p$ -matrix over E is denoted by $\mathcal{M}(E, n, p)$.

Definition 4.12. (Matrix and vectors operators) Let $(E, +, \times)$ be a field and n, p be two positive integers. Let $A = (a_{ij})$ and $B = (b_{ij})$ be two matrices of $\mathcal{M}(E, n, p)$, $U = (u_j)$ an horizontal vector in $\mathcal{M}(E, 1, p)$ and $V = (v_i)$ a vertical vector in $\mathcal{M}(E, n, 1)$. The following operators are

defined:

1. The sum of matrix operator noted $+$ is a function defined as follow :

$$+: \mathcal{M}(E, n, p) \times \mathcal{M}(E, n, p) \rightarrow \mathcal{M}(E, n, p), \\ ((a_{ij}), (b_{ij})) \mapsto (a_{ij} + b_{ij})$$

2. The rows sum of matrix operator noted $\vec{+}$ is a function defined as follows :

$$\vec{+}: \mathcal{M}(E, n, p) \rightarrow \mathcal{M}(E, n, 1), \\ (a_{ij}) \mapsto \begin{pmatrix} \sum_{j=1}^p a_{1j} \\ \vdots \\ \sum_{j=1}^p a_{nj} \end{pmatrix}$$

3. The column sum of matrix operator noted $\downarrow +$ is a function defined as follows :

$$\downarrow +: \mathcal{M}(E, n, p) \rightarrow \mathcal{M}(E, 1, p), \\ (a_{ij}) \mapsto (\sum_{i=1}^n a_{i1}, \dots, \sum_{i=1}^n a_{ip})$$

4. The parallel product of two matrix operator noted $\bar{\times}$ is a function known as Hadamard product and defined as follows :

$$\bar{\times}: \mathcal{M}(E, n, p) \times \mathcal{M}(E, n, p) \rightarrow \mathcal{M}(E, n, p), \\ ((a_{ij}), (b_{ij})) \mapsto (a_{ij} \times b_{ij})$$

5. The horizontal product of matrix and vector operator noted $\vec{\times}$ is a function defined as follows :

$$\vec{\times}: \mathcal{M}(E, n, p) \times \mathcal{M}(E, n, 1) \rightarrow \mathcal{M}(E, n, p), \\ ((a_{ij}), (u_i)) \mapsto (a_{ij} \times u_i)$$

6. The vertical product of matrix and vector operator noted $\downarrow \times$ is a function defined as follows:

$$\downarrow \times: \mathcal{M}(E, n, p) \times \mathcal{M}(E, 1, p) \rightarrow \mathcal{M}(E, n, p), \\ ((a_{ij}), (v_j)) \mapsto (a_{ij} \times v_j)$$

7. The total sum of a matrix or vector operator noted \oplus is a function defined as follows:

$$\oplus: \mathcal{M}(E, n, p) \rightarrow \mathcal{M}(E, 1, 1), \\ (a_{ij}) \mapsto \sum_{i=1}^n \sum_{j=1}^p a_{ij}$$

8. The total product of a matrix or vector operator noted \otimes is a function defined as follows:

$$\otimes: \mathcal{M}(E, n, p) \rightarrow \mathcal{M}(E, 1, 1), \\ (a_{ij}) \mapsto \prod_{i=1}^n \prod_{j=1}^p a_{ij}$$

One may notice that the operator \oplus is the composition of the two operators $\vec{+}$ and $\downarrow +$ defined above.

Remark 4.1. 1. The set $\{\oplus; \otimes; \vec{+}; \downarrow +; \vec{\times}; \bar{\times}; +; \downarrow \times\}$ is a signature of matrix-based multisorted algebra.

2. The addition of matrix is a Σ -aggregator

5. Multisorted Tree Algebras in Universal Algebra Theory

When the hierarchy Δ of a (Θ, Σ, Δ) -multisorted algebra is not reduced to single node, then its Structure is different with the one of any Σ -universal algebra. Comparing the both algebraic structures is as comparing a wall structure with a block used to build that wall. The (Θ, Σ, Δ) -multisorted algebra structure is composed of Σ -algebras networked on a tree. The computations are handled by a network of Σ -algebras that is built by operation from Σ that are combined by Σ aggregations. While in universal algebra the resulting computations are expressed by Σ -terms, in multisorted tree algebras they are expressed as a tree of terms at each node of the hierarchy. Some of these terms are gradually generated from terms at the bottom of the hierarchy. Some similarities can be made with computations systems in the literature as grid computing. Base on their structure as graph of Σ -algebras, one may have the intention to consider multisorted tree algebras as a graphs in the category of Σ -algebra [16, 17]. A graph in $\mathcal{C}(\Sigma)$ has as nodes Σ -algebras and as edges Σ -homomorphisms. Even though some of those graphs may have a shape of tree, they are not multisorted tree algebras because the concept of homomorphism is not contained in the definition of multisorted tree algebra.

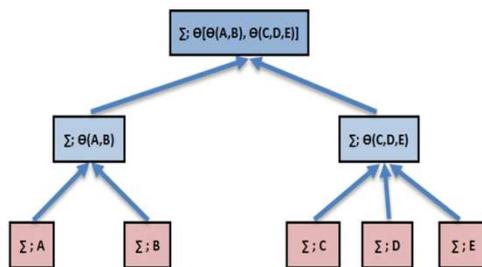


Figure 1. Example of multisorted tree algebra structure.

Multisorted tree algebras are specific algebraic structures constructed by gluing universal algebras through an aggregation operation to account for realities as the bottom up data computations and processing flow to generate different data at each node of a hierarchy. In Figure 1, the leaves contain Σ -algebra with supports noted as A, B, C, D, E , and the intermediate level contains the aggregated Σ -algebras with supports noted as $\Theta[A, B, C]$ and $\Theta[E, F]$. In this example, a list of five terms at the bottom of the hierarchy will generate three terms classified as triangle nodes with the first two at the basis. One can notice that aggregation operators may be used to construct more complex algebraic structures that don't necessarily have the shape of a balanced tree. Multisorted tree algebra therefore offers a mathematical framework for investigating such constructions and their engineering. However, the features and properties of these new mathematical objects need to be studied in detail.

Elsewhere, the foundation of algebraic specification rely on the fact that an Abstract

Data Types is modeled by universal algebra. Therefore, the introduction of multisorted tree algebra induces by analogy

the introduction of Tree Abstract Data Type that could be useful for studying classes programming language in software engineering.

6. Conclusions and Perspectives

In this paper we have introduced a hierarchical algebraic structure called multisorted tree algebra. A brief description of results in multisorted algebra has been given in the first section. Then the concept of aggregation operator has been presented as an operator that generates a Σ -algebra from different other Σ -algebras. It has also been shown that aggregation operators generate a Σ -homomorphism from a given family of Σ -homomorphisms between the aggregated Σ -algebras. Next, multisorted tree algebra has been constructed by placing multisorted algebra at the bottom of a hierarchy, and by placing at other nodes the aggregation of multisorted algebra placed at their immediate subordinate's nodes. Furthermore, an analysis about the features and the place of multisorted tree algebra in the universal algebra theory has been briefly discussed. Among the perspectives generated by this paper, investigations on mathematical features and approaches to the study of multisorted tree algebra are necessary. One of the major reasons to engage in these further studies is the fact that they represent a hierarchy of algebraic structures that cannot be classified in classic varieties of universal algebra theory and also, they are not graphs of categories of universal algebras of given signature. The multisorted tree algebra concept may also be expanded by investigating aggregations of universal algebras of different signatures, or the construction of aggregations on a graph that are not trees. Among the application perspectives, multisorted tree algebra seems to give to computer scientists a different abstract data type described as a hierarchy of data types. It's not pretentious to think that this new concepts could also give an approach to assemble abstract data type architectures and therefore expand results where classic universal algebras has contributed in theoretical computer science domains including for example λ -calculus [18], type theory [19], data structures [20] and algebraic specification [21, 7].

Elsewhere, bottom up data processing in hierarchical organizations are characterized by computations of data at each node of the hierarchy with fact that upper nodes generally compute aggregate data from their direct subordinate nodes. An information system that manages such computations can therefore be modeled by a multisorted tree algebra. If each node is modeled as a data type because of the computations implemented at it level, then the whole system is therefore considered as a hierarchy of data types. Hence, multisorted tree algebras also account for the modeling of such hierarchical computations and information systems [13].

References

- [1] K. D. . S. L. Wismath, Universal Algebra and Applications in Theoretical Computer Science, CRC, 2002.

- [2] S. Burris. . H. P. Shankappanavar, A Course in Universal Algebra, the millenium edition Edition, Springer-Verlag, <http://www.cs.elte.hu/~ewkiss/univ-algebra.pdf>, 1981.
- [3] W. Wechler, Universal Algebra for Computer Scientists, Springer-Verlag, 1992.
- [4] C. Oriat, Etude des speci_cations modulaires: constructions de colimites _nies, diagrammes, isomorphismes, Informatique, Institut National Polytechnique de Grenoble, Laboratoire Logiciels Systmes et Rseaux (LSR-IMAG) (janvier 1996).
- [5] A. Mucka. al., Many-sorted and single-sorted algebras, Algebra Universalis 69 (2013) 171{190.
- [6] J. A. Goguen, Hidden algebraic engineering, in: C. Nahaniv (Ed.), Conference on semi groups and algebraic engineering, University Aisu, 1997.
- [7] J. Goguen, Hidden algebra for software engineering, in: Proc. Conf. Discrete Mathematics and Theoretical Computer Science, Vol. 21 of Australian Computer Science Communications, 1999, pp. 35{59.
- [8] J. Stell, A framework for order-sorted algebra, in: H. Kirchner, C. Ringeissen (Eds.), Algebraic Methodology and Software Technology, Vol. 2422 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2002, pp. 396{411.
- [9] J. J. M. M. Rutten, Universal coalgebra: a theory of systems (2000).
- [10] G. Rosu, Hidden logic, Phd, University of California, San Diego (2000).
- [11] T. V. Zandt, Real-time hierarchical resource allocation, <http://faculty.insead.edu/vanzandt/research-orgs/papers/Resource1.pdf>.
- [12] M. W. A. Knapp, A formal approach to object-oriented software engineering, Theoretical ComputerScience 285 (2002) 519{560.
- [13] T. V. Zandt, Hierarchical computation of the resource allocation problem, European Economic Review39 (1995) 700{708.
- [14] F. H. Trinkl, Hierarchical resource allocation decisions, Policy Sciences 4 (1973) 211{221.
- [15] J. G. . G. Malcom, A hidden agenda, Theoretical Computer science 245 (2000) 55{101.
- [16] M. Barr, C. Wells, Category Theory for Computer Science, Pintice-Hall International, 1990.
- [17] J. L. Fiadeiro, Cathegories for software engineering, Springer, 2005.
- [18] G. Manzonetto, A. Salibra, From -calculus to universal algebra and back, in: MFCS08, volume 5162 of LNCS, 2008, pp. 479{490.
- [19] V. Capretta, Universal algebra in type theory, in: Theorem Proving in Higher Order Logics, 12th International Conference, TPHOLs '99, volume 1690 of LNCS, Springer-Verlag, 1999, pp. 131{148.
- [20] J. V. Guttag, Abstract data types and the development of data structures, Communication of the ACM6 (1977) 396{404.
- [21] J. A. Goguen, G. Malcolm, Software Engineering with OBJ: algebraic specification in action, Vol. Advances in Formal Methods, Kluwer Academic Publishers, 2000.