Computer Science Journal of Moldova, vol.26, no.3(78), 2018

# Imbrication algebras – algebraic structures of nesting order

Ioachim M. Drugus, Volodymyr G. Skobelev

**Abstract**

This paper is about "imbrication algebras", universal algebras with one binary operator in their signature, the operator for formation of ordered pairs, called here "pairing operator", and with the "characteristic property of ordered pairs" as their sole axiom. These algebras have been earlier introduced by the first author as reducts of "aggregate algebras", universal algebras proposed as models for a set theory convenient for formalization of data structures. The term "aggregate" is used to generalize three fundamental notions of set theory: set, atom and ordered pair. Thus, this paper initiates the research of aggregate algebras by narrowing the focus to one type of their main reducts – the reduct which deals with ordered pairs.

**Keywords:** cancellative magma, Catalan number, Merkle tree, ordered pair, quasi-variety.

**MSC 2010:** 08A70, 68P05, 05C05.

## 1 Introduction

The full name of the algebras, which are the subject matter of current paper, is intended to be "*imbrication order algebras*". In [1], (p. 87), where this kind of algebraic structures was introduced, these algebras were referenced shorter, as "order algebras". In this paper, we will prefer for them another short term – "imbrication algebras". The main reason why different short forms of the same term are preferred in different situations is that the term denotes a very general notion, which manifests as different phenomena in more concrete settings.

The term “imbrication” is often used in linguistics and computer science interchangeably with the term “nesting”. However, this term is also extensively used with a meaning different from “nesting” – a meaning conveyed by the words: “overlapping”, “interlacing”, “interweaving”. The second meaning is more relevant to the topic of the paper [1], and the first meaning of “imbrication”, that of “nesting”, is perfectly relevant to the topic of current paper.

Whereas nesting is a phenomenon which might have a large number of manifestations, we will focus on one kind of such manifestations – the nesting of ordered pairs, and we will put this in precise terms in the next section.

## 2 The imbrication algebras

It is a wide practice to refer to a symbol of an operation as “operator”, and this practice is convenient for universal algebra, since the signature consists of operation symbols, i.e. “operators”. There is also a practice to refer as “operator” to a mapping from a space to another space (e.g. “linear operator”), and this practice is convenient because it uses one term – “operator”, rather than two terms – “operation” and “operator”. In this second case, the expression “symbol of the operator” stands for what is called “operator” in the first case.

An ordered pair $(a,\ b)$ can be treated as the result of application of a binary operation, which we will prefer to call “operator”, both because it can be treated as an action upon symbols, a “syntactic operator” (see subsection 2.2), and because the ordered pairs make up a “space”, to be more precise – a plane. We will call this operator “pairing operator”, because a function which encodes an ordered pair of natural numbers into one natural number is usually called “pairing function” (Cantor defined one of the first pairing functions – the “Cantor’s pairing function”). We will treat $(a,\ b)$ as the result of application of the pairing operator to the objects $a$ and $b$, and since in the notation “$(a,\ b)$” a symbol of operation or operator (like here “$f(a,\ b)$”) is missing, we will consider the “empty symbol” as the symbol of the pairing operator.

**Definition 1.** *An imbrication algebra is a universal algebra with a sole operator's symbol in its signature, the empty symbol, and with the universal closure of the formula below as its single axiom:*

$$(x,\ y) = (x',\ y') \to x = x' \ \&\ y = y'.$$

The axiom of the imbrication algebras is the property owned by ordered pairs defined in one set theory or another. No matter how these are defined, this property is called "characteristic property of the ordered pairs". Whereas in set theory this is treated as a property of ordered pairs, in algebra this should be treated as a property of the pairing operator. Going forward, the universal closure of the formula above will be referenced as "pairing axiom".

Even though in ZF set theory, the ordered pair is a notion defined through the notion of set, there are also set theories, like Bourbaki's set theory, where the pairing operator is in the signature and it has the pairing axiom.

The first example of an imbrication algebra given in this paper is the universe (of discourse) of ZF set theory (usually denoted as $V$), which needs to be equipped with a pairing operator (like the one defined by Kuratowski, see next paragraph), to form a universal algebra. This imbrication algebra has a proper class as its support, and thus, this is a "large algebra". There are many large imbrication algebras – such are the universes (of discourse) of various set theories equipped with a pairing operator, which, by definition of the notion of ordered pair, must satisfy the pairing axiom.

There are many pairing operators in set theory, among which the best known is the Kuratowski's paring operator, used to define an ordered pair $(x, y)$ as $\{\{x\}, \{\{x, y\}\}$. It is easy to check that any two pairing operators which equip $V$ define the same imbrication algebra.

## 2.1 The class of imbrication algebras

According to one of several equivalent definitions (see e.g. [4], p. 219), a class of universal algebras of same signature ("similar algebras") is a *quasivariety*, if it contains a one-element algebra and is closed under isomorphisms, subalgebras and reduced products. A quasivariety is also closed under products, subdirect products, and ultrafilter products.

The following property of quasivarieties is important for any constructive approach used in algebra: if a class of universal algebras is a quasivariety, then proceeding from a subset of this class, regarded as a "generating basis", one can construct other algebras by applying the operations over algebras mentioned in previous sentence.

The next proposition could be named "theorem" because of its high importance to this domain of research.

**Proposition.** *The class of imbrication algebras is a quasi-variety.*

**Proof.** The pairing axiom is equivalent to the conjunction of the universal closures of the following two formulas:

$$(x, y) = (x', y) \to x = x',$$
$$(x, y) = (x, y') \to y = y'.$$

Unlike the pairing axiom, these two formulas are quasi-identities, i.e. each of them has the form of an implication, the antecedent of which is a conjunction of equations of two terms (here, there is only one equation in conjunction), and the consequent of which is one such equation. According to the Theorem 2.25 of [4], since any imbrication algebra satisfies a set of quasi-identities, this class is a quasi-variety.

Q.E.D.

## 2.2 Application domains of imbrication algebras

We have mentioned above important examples of implication algebras – the universes of set theories equipped with a pairing operator. In this section, we will discuss about two domains, which can be considered as "native land" of the imbrication algebras – domains, where imbrication algebras can be used as an apparatus. One of these domains is the syntax of languages, natural or artificial, and the other is "brain informatics" – a discipline preoccupied by modeling mental phenomena, in particular, mental structures. One can say that the mental structures are constructed also according to a certain kind of "syntax". Thus, it sounds appropriate to say that imbrication algebras relate to syntax, and pairing operator is a "syntactic operator".

Imbrication algebras reflect a special kind of order – the order, which appears as a result of using *balanced* distribution of brackets (we prefer round brackets, parentheses) for the *complete* disambiguation of an expression. Notice, that in previous sentence, two words are emphasized – "balanced" and "complete". The word "balanced" is emphasized because the nesting order appears for balanced, and only for balanced, brackets – i.e. the balance is essential for nesting. The word "complete" is emphasized because a distribution of balanced brackets enclosing *two*, and only *two*, subterms is essential for the nesting order called "imbrication".

To *completely* disambiguate an expression of form "$a_1 * ... * a_n$", where "*" is an operator, one needs to apply, in steps, a process of enclosing a *pair* of adjacent terms – subexpressions processed in this manner at a previous step, a process which can be called "pairing". The imbrication algebras can be described as algebras reflecting *complete* disambiguation by using pairing, and not partial disambiguation done by the use of an arbitrary distribution of balanced parentheses.

Whereas an application domain of imbrication algebras is the syntax of languages (natural or artificial), these algebras are most useful for the practice of grouping the subexpressions of an expression. The process of grouping (in particular, of grouping done by pairing) is sometimes referenced as "association" like in the term "left (right) association rule". For generality sake, we will refer to a balanced distribution of brackets as "association pattern". Thus, *left association* or *right association* are *association patterns*.

In [2], an approach to data called "Atomification-Aggregation-Association approach" (see also [3]) denoted as "A3" was introduced to serve as an alternative to currently widely used "Entity-Relationship" approach denoted as "ER". The A3 approach was proposed as a data model for "mental content" – a concept which makes sense in brain informatics. The A3 approach presupposes that all data structures are built by iterative application of three mental operations, one of which associates one entity *to* another entity (the order is important) and, in this manner, builds an "association pair" (an *association pair* is an *association pattern* – the simplest association pattern).

The imbrication algebras explicate the algebraic aspect of the data structures built by iterative application of the "association operation" of A3 approach. We will refer to such structures simply, as "associations". The fact that an entity is associated with another entity can be imaged graphically by representing the entities as small cirles connected by arrows. In such a representation, when an entity is associated with itself, the arrow has a source coinciding with the target, and the direction of the arrow does not matter, so that one can drop the arrowhead and just use a non-oriented loop.

However, associations differ from those data structures which can be represented by directed graphs, since an association can, in turn, be associated with another association, like in the diagram in Figure 1, where the loop in $b$ is associated with the node $c$ and this association pair is, in turn, the target of another association. Since the kind of graphs which allow such "imbrication" differs from "directed graphs", they require a name and we will refer to them as "association graphs".

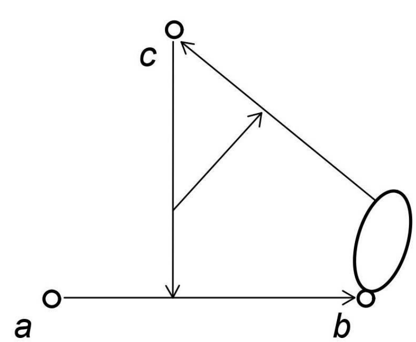


Figure 1. The diagram of the association $((c,(a,b)),((b,b),c))$

## 3 Imbrication algebras and the order theory

It is common to treat the preorder (also called quasi-order), as a binary relation, and this treatment is considered as the widest explication of the conception of order. However the notion of "cyclic order", a kind of order known to humanity since the expression "clockwise" and "counter-clockwise" were coined, cannot be explicated in terms of preorder. Therefore, the algebraists use a ternary relation to explicate the conception of cyclic order. Since there might exist also other kinds of order which cannot be expressed like preorder, the term "order theory" was coined to cover all phenomena called "order". However, there does not seem to exist a discipline "under this name" with a unified approach to the conception of order.

As earlier mentioned, the imbrication algebras were introduced in [1], where these were referenced as "order algebras". The reason for using the word "order" within this compound term was that in [1] these algebras were treated as reducts of "aggregate algebras" – a kind of algebraic structures intended for algebraization of a special set theory. The term "order algebra" was used in that paper for two reasons: (a) the term refers to an algebra with one operator – that of formation of the *ordered* pair and (b) the notion of ordered pair is used in all definitions of order within the aggregate algebras, where you can also use the concept of set.

In the "aggregate theory" of [1], alongside the paring operator, there are also the operator of formation of a singleton $\{x\}$, and the operation of union of two sets. The aggregate theory can be extended by introducing the operation of union for a family of sets, so that the theory treats arbitrary (i.e. also infinite) aggregates. Thus, the notion of preoder can be easily expressed in terms of aggregate theory, and thus all the theories about the "classical" explication of order in terms of a binary relation can have the aggregate theory as a foundation.

The pairing operator adds a new type of order to aggregate theory – the imbrication order. This is the order of nesting in expressions and the order of the ordered binary trees. There can be defined also other orders – orders, which "mix up" the classic order and the imbrication order. This raises hopes that the aggregate theory can serve as an appropriate foundation for the discipline called "order theory".

As an exercise, one can try to explicate in terms of imbrication algebra, without involving set theoretic operations, the conception of "clockwise order". An explication of this is given by the association ((...((1,2),3),...), 12), but one should expect that this kind of order can also be defined by any other of 11 associations obtained from this one by a circular substitution. There can be used also other association patterns for explication of clockwise order. The choice of one explication or another is necessarily *a matter of convention*, similar to how the Kuratowski's definition of ordered pair is a matter of convention.

## 4 Imbrication algebras and Jónsson-Tarski algebras

For an ordered pair $a = (u, v)$, denote $u$ as $a^+$ and $v$ as $a^\times$ (these are the original notations used in [5]). One can temporarily call "projections" the two maps, $x \mapsto x^+$ and $x \mapsto x^\times$, even though the expressions like "left (right) projection" or "first (second) projection" make little sense. The "characteristic property of ordered pairs" guarantees that the projections are univocal maps, where they are defined, but nothing in the definition of imbrication algebra guarantees that the projections are defined for all elements of the algebra.

**Definition 2.** *An imbrication algebra with both projections defined for all its elements is called Jónsson-Tarski algebra.*

The algebras introduced in [5] and called here "Jónsson-Tarski algebras" are examples of a proper subclass of the class of imbrication algebras which is known to mathematical community.

## 5 Algebraic closure of pairing operator

Let $A$ be any fixed non-empty set. We consider the binary operator, the value of which for any elements $x, y \in A$ (taken in this order) is the object denoted $(x, y)$ (it is supposed that the symbols "(", "," and ")" are not elements of the set $A$). It is evident that the object $(x, y)$ $(x, y \in A)$ can be treated as an ordered pair, if we accept the following axiom postulating the properties of such objects:

$$(\forall x_1, x_2, y_1, y_2 \in A)((x_1, y_1) = (x_2, y_2) \Leftrightarrow x_1 = x_2 \& y_1 = y_2). \tag{1}$$

We define

$$A^{(0)} = A, \tag{2}$$

$$A^{(n)} = \bigcup_{i=0}^{n-1} \{(x, y) | x \in A^{(i)}, y \in A^{(n-1-i)}\} \ (n = 1, 2, \dots). \tag{3}$$

For each non-empty set $A$ formulae (2) and (3) define inductively the sequence

$$A^{(0)}, A^{(1)}, \dots, A^{(n)}, \dots \tag{4}$$

of non-empty sets. Due to this, the axiom (2) can be extended from the set $A = A^{(0)}$ onto the set

$$\mathcal{A} = \bigcup_{n=0}^{\infty} A^{(n)}, \tag{5}$$

i.e. the following axiom can be accepted:

$$(\forall x_1, x_2, y_1, y_2 \in \mathcal{A})((x_1, y_1) = (x_2, y_2) \Leftrightarrow x_1 = x_2 \& y_1 = y_2). \tag{6}$$

Due to formulae (2)-(6) the following three propositions are true.

**Proposition 1.** *For each non-empty set $A$, if $(x_1, y_1) = (x_2, y_2)$ $(x_1, x_2, y_1, y_2 \in \mathcal{A})$, then there exist the single non-negative integers $i$ and $j$, such that $x_1, x_2 \in A^{(i)}$ and $y_1, y_2 \in A^{(j)}$.*

**Proposition 2.** *For each non-empty set $A$ the sequence (4) consists of non-empty pair-wise non-intersecting sets.*

**Proposition 3.** *For each non-empty set $A$ the set $\mathcal{A}$ is an infinite set.*

Proceeding from formulae (2)-(6), we can define for each non-empty set $A$ the $A$-associated magma

$$\mathcal{M}_A = (\mathcal{A}, \circ), \tag{7}$$

such that

$$x \circ y = (x, y) \tag{8}$$

for all $x, y \in \mathcal{A}$.

Now we establish the basic characteristics of the $A$-associated magma $\mathcal{M}_A = (\mathcal{A}, \circ)$, i.e. those ones, that are true for each non-empty set $A$.

**Theorem 1.** *For each non-empty set $A$ the binary operation in the $A$-associated magma $\mathcal{M}_A = (\mathcal{A}, \circ)$ is a surjection $\circ : \mathcal{A} \times \mathcal{A} \to \bigcup\limits_{n=1}^{\infty} A^{(n)}$.*

**Proof.** Let $A$ be any non-empty set.

Due to formula (5), for any elements $x, y \in \mathcal{A}$ there exist the single non-negative integers $i$ and $j$, such that $x \in A^{(i)}$ and $y \in A^{(j)}$.

Due to formulae (3) and (8), we get that $x \circ y = (x, y) \in A^{(i+j+1)}$.

Since $i$ and $j$ are non-negative integers, then $i + j + 1$ is a positive integer. Thus, $x \circ y \in \bigcup\limits_{n=1}^{\infty} A^{(n)}$ for all $x, y \in \mathcal{A}$, i.e. the inclusion $\mathrm{Val}\circ \subseteq \bigcup\limits_{n=1}^{\infty} A^{(n)}$ holds.

Let $z$ be any element of the set $\bigcup\limits_{n=1}^{\infty} A^{(n)}$. Then (see Proposition 2) there exists the single positive integer $n$, such that $z \in A^{(n)}$.

Due to Proposition 2, and formulae (3) and (8), there exists the single non-negative integer $i \leq n - 1$, such that $z = (x, y) = x \circ y$, where $x \in A^{(i)}$ and $y \in A^{(n-1-i)}$. Thus, $z \in \mathrm{Val}\circ$ for any $z \in \bigcup\limits_{n=1}^{\infty} A^{(n)}$, i.e. the inclusion $\bigcup\limits_{n=1}^{\infty} A^{(n)} \subseteq \mathrm{Val}\circ$ holds.

Inclusions $\mathrm{Val}\circ \subseteq \bigcup\limits_{n=1}^{\infty} A^{(n)}$ and $\bigcup\limits_{n=1}^{\infty} A^{(n)} \subseteq \mathrm{Val}\circ$ imply that the identity $\mathrm{Val}\circ = \bigcup\limits_{n=1}^{\infty} A^{(n)}$ holds, i.e. the mapping $\circ : \mathcal{A} \times \mathcal{A} \to \bigcup\limits_{n=1}^{\infty} A^{(n)}$ is some surjection.

Q.E.D.

**Theorem 2.** *For each non-empty set $A$ the $A$-associated magma $\mathcal{M}_A = (\mathcal{A}, \circ)$ is a cancellative magma.*

**Proof.** Let $A$ be any non-empty set.

Formulae (6) and (8) imply that for any $x, y, z \in \mathcal{A}$

$$x \circ y = x \circ z \Leftrightarrow (x, y) = (x, z) \Leftrightarrow x = x \& y = z \Leftrightarrow y = z,$$

i.e. $\mathcal{M}_A = (\mathcal{A}, \circ)$ is a left-cancellative magma.

Similarly, formulae (6) and (8) imply that for any $x, y, z \in \mathcal{A}$

$$x \circ y = z \circ y \Leftrightarrow (x, y) = (z, y) \Leftrightarrow x = z \& y = y \Leftrightarrow x = z,$$

i.e. $\mathcal{M}_A = (\mathcal{A}, \circ)$ is a right-cancellative magma.

Thus, the $A$-associated magma $\mathcal{M}_A = (\mathcal{A}, \circ)$ is a left-cancellative and a right-cancellative, both. Due to this factor, the $A$-associated magma $\mathcal{M}_A = (\mathcal{A}, \circ)$ is a cancellative magma.

Q.E.D.

Theorems 1 and 2 imply that the following proposition is true.

**Proposition 4.** *For each non-empty set $A$ the $A$-associated magma $\mathcal{M}_A = (\mathcal{A}, \circ)$ is not a quasigroup.*

**Remark 1.** In proof of Theorem 1 it has been pointed that for any $i, j = 0, 1, \dots$, if $x \in A^{(i)}$ and $y \in A^{(j)}$, then $x \circ y = (x, y) \in A^{(i+j+1)}$. This factor implies that for any fixed non-negative integers $i$ and $n$, such that $n \leq i$, if $a \in A^{(i)}$ and $b \in A^{(n)}$, then each of the equations $a \circ x = b$ and $y \circ a = b$ has no solutions in the $A$-associated magma $\mathcal{M}_A = (\mathcal{A}, \circ)$.

## 6 The interrelation between elements of the set $\mathcal{A}$ and finite binary trees

Let $A$ be any fixed non-empty set.

Each element $w \in \mathcal{A}$ can be uniquely presented by the rooted labeled finite binary tree $\mathcal{D}_w$, designed by the following procedure ($V$ is the set of vertices, $v^{(R)}$ is the root, $E$ is the set of arcs, $f : V \to \mathcal{A}$ is the labeling mapping):

`L-TREE`
**Input**: a string $w \in \mathcal{A}$.
**Output**: the tree $\mathcal{D}_w$.

**begin**;
  $V := \{v^{(R)}\}$;
  $f(v^{(R)}) := w$;
  $E := \emptyset$;
  $M_1$: **if** $V = \emptyset$
    **then** HALT,
     **else go to** $M_2$;
  $M_2$: **select** $v \in V$;
  $V := V \setminus \{v\}$;
  **if** $f(v) \in A$
    **then go to** $M_1$,
     **else** (in this case $f(v) = (u_L, u_R)$, where $u_L, u_R \in \mathcal{A}$)
      $V := V \cup \{v_L, v_R\}$;
      $E := E \cup \{(v, v_L), (v, v_R)\}$;
      $f(v_L) := u_L$;
      $f(v_R) := u_R$;
      **go to** $M_1$;
**end**;

It is evident that the procedure `L-TREE` is some variant of a top-down parser. Due to formulae (2)-(5), it terminates for any element $w \in \mathcal{A}$, and its output is the tree $\mathcal{D}_w$.

**Remark 2.** In any tree $\mathcal{D}_w$ ($w \in \mathcal{A}$) each vertex either has two sons (if it is an internal vertex), or has no sons (if it is a leaf).

We set

$$\mathfrak{D}_{\mathcal{A}} = \{\mathcal{D}_w | w \in \mathcal{A}\}. \tag{9}$$

Due to formulae (2)-(6), the following two propositions are true.

**Proposition 5.** *For each non-empty set $A$ an element $\mathcal{D}_w \in \mathfrak{D}_{\mathcal{A}}$ is a rooted labeled binary tree with $n$ ($n = 2, 3, \dots$) leafs if and only if there are $n$ appearances of elements of the set $A$ in the string $w$.*

**Proposition 6.** *For each non-empty set $A$ the following formula holds*

$$(\forall w_1, w_2 \in \mathcal{A})(w_1 \neq w_2 \Leftrightarrow \mathcal{D}_{w_1} \neq \mathcal{D}_{w_2}). \tag{10}$$

**Remark 3.** Formulae (9) and (10) imply that for each non-empty set $A$, the sets $\mathcal{A}$ and $\mathfrak{D}_{\mathcal{A}}$ have the same cardinality.

The more subtle characteristics of the structure of the set $\mathcal{A}$ can be established as follows.

Let $\tilde{\mathcal{D}}_w$ ($w \in \mathcal{A}$) be the rooted unlabeled binary tree that can be obtained by erasing all labels of vertices in the tree $\mathcal{D}_w$. It is evident that for each non-empty set $A$, the unlabeled rooted binary tree $\tilde{\mathcal{D}}_w$ ($w \in \mathcal{A}$) can be treated as the structure of the string $w \in \mathcal{A}$.

We set

$$\tilde{\mathfrak{D}}_{\mathcal{A}} = \{\tilde{\mathcal{D}}_w | w \in \mathcal{A}\}. \tag{11}$$

and define the mapping $\psi_{\mathcal{A}} : \mathfrak{D}_{\mathcal{A}} \to \tilde{\mathfrak{D}}_{\mathcal{A}}$ by the identity:

$$\psi_{\mathcal{A}}(\mathcal{D}_w) = \tilde{\mathcal{D}}_w \ \ (w \in \mathcal{A}). \tag{12}$$

Formulae (11), (12), and Proposition 6 imply that for each non-empty set $A$, the elements of the factor-set $\mathfrak{D}_{\mathcal{A}}/\ker \psi_{\mathcal{A}}$ define the sets of all strings $w \in \mathcal{A}$ with the same structure $\tilde{\mathcal{D}}_w$.

**Remark 4.** For each non-empty set $A$, the set $\tilde{\mathfrak{D}}_{\mathcal{A}}$ is a countable set, since it is the set of all non-empty finite rooted binary trees, such that each vertex either has two sons, or has no sons. Due to this factor, in what follows, we will omit the subscript $\mathcal{A}$, i.e. we will write $\tilde{\mathfrak{D}}$, since for any non-empty sets $A_1$ and $A_2$ the identity $\tilde{\mathfrak{D}}_{\mathcal{A}_1} = \tilde{\mathfrak{D}}_{\mathcal{A}_2}$ holds.

Due to Proposition 5, we can define on the set $\tilde{\mathfrak{D}}$ the partition $\pi = \{B_n | n = 2, 3, \dots\}$ as follows: $B_n$ ($n = 2, 3, \dots$) consists of all elements of the set $\tilde{\mathfrak{D}}$ that are unlabeled rooted binary trees with $n$ leafs. It is well known that the following proposition is true.

**Proposition 7.** *For any* $n = 2, 3, \ldots$

$$|B_n| = C_{n-1},$$

*where* $C_{n-1}$ *is the Catalan number.*

**Remark 5.** The Catalan numbers can be computed by formula:

$$C_n = \frac{1}{n+1}\binom{2n}{n} \quad (n = 1, 2, \ldots).$$

The structure of the strings $w \in \mathcal{A}$ has been identified above with the finite rooted unlabeled binary trees $\tilde{\mathcal{D}}_w \in \tilde{\mathfrak{D}}$. Another (equivalent) model for the structure of the strings $w \in \mathcal{A}$ can be designed as follows.

Let $B = A \cup \{\ ,\}$, and $del_B(w)$ $(w \in \mathcal{A})$ be the string obtained from the string $w$ by deleting all letters $b \in B$. We set

$$del_B(\mathcal{A}) = \{del_B(w) | w \in \mathcal{A}\}.$$

The language $del_B(\mathcal{A})$ can be characterized as follows.

**Proposition 8.** *For any non-empty set* $A$ *the language* $del_B(\mathcal{A})$ *is some proper non-empty sub-language of the Dyck language* $L_{D(2)}$ *over the* 2*-letters alphabet.*

**Proof.** The Dyck language $L_{D(2)}$ over the 2-letters alphabet $\{\alpha, \beta\}$ is the context-free language that can be generated, for example, by the following two production rules ($S$ is the single non-terminal symbol, and $\lambda$ is the empty symbol): $S \to \lambda$ and $S \to \alpha S \beta S$.

Identifying the symbol $\alpha$ with the opening parentheses (, and the symbol $\beta$ with the closing parentheses ), we get that $del_B(\mathcal{A}) \subseteq L_{D(2)}$.

Since $() \in del_B(\mathcal{A})$, we get that $del_B(\mathcal{A}) \neq \emptyset$.

Since $()() \in L_{D(2)}$ and $()() \notin del_B(\mathcal{A})$, we get that $del_B(\mathcal{A}) \subset L_{D(2)}$.

Q.E.D.

## 7 Applications of the $A$-associated magma

Let $A$ be any non-empty set.

We illustrate briefly, how the $A$-associated magma $\mathcal{M}_A = (\mathcal{A}, \circ)$ can be used as some conceptual model in mathematics and its applications.

*Example 1.* The operation $\circ$ can be naturally extended on the power set $\mathcal{B}(\mathcal{A})$ as follows:

$$X \circ Y = \{(x, y) | x \in X \& y \in Y\} \tag{13}$$

for any subsets $X, Y \in \mathcal{B}(\mathcal{A})$.

It is evident that formula (13) defines the operation of the Cartesian product on the set $\mathcal{A}$.

Thus, the operation of the Cartesian product can be treated as some binary operation over some magma that satisfies Theorems 1 and 2, and Proposition 4.

*Example 2.* Let $\diamond$ be any binary operation defined on the set $A$. i.e. $\mathcal{M}_\diamond = (A, \diamond)$ is any magma with the carrier $A$. The $A$-associated magma $\mathcal{M}_A = (\mathcal{A}, \circ)$ can be treated as the formal presentation for the magma $\mathcal{M}_\diamond = (A, \diamond)$ of all possible results for the operation $\diamond$ over the strings of elements of the set $A$ as follows.

Let $a_1, \ldots, a_n \in A$ $(n \geq 2)$. We can select any finite rooted unlabeled binary tree $\tilde{\mathcal{D}} \in \tilde{\mathfrak{D}}$ with $n$ leafs, and label the leafs, from left to right, by the elements $a_1, \ldots, a_n$. Thus, we have defined the order for the execution of operation $\diamond$ over the string $a_1 \diamond \cdots \diamond a_n$.

Using down-top parsing, we can label all internal vertices of the selected tree due to the following rule: if for the internal vertex $v$ its left son $v_L$ is labeled by the element $b_1 \in A$ and its right son $v_R$ is labeled by the element $b_2 \in A$, then the vertex $v$ is labeled by the element $b_1 \diamond b_2 \in A$.

It is evident that the label of the root of the selected tree is the result of the operation $\diamond$ over the string $a_1 \diamond \cdots \diamond a_n$, when the order for execution of this operation is defined by the selected finite rooted unlabeled binary tree $\tilde{\mathcal{D}} \in \tilde{\mathfrak{D}}$ with $n$ leafs, when the leafs are labeled, from left to right, by the elements $a_1, \ldots, a_n$.

*Example 3.* Any Merkle tree [7] is a rooted complete finite labeled binary tree, such that each vertex either has two sons, or has no sons.

The leafs of a Merkle tree are labeled by some data. Any internal vertex of a Merkle tree is labeled due to the following rule: if for the internal vertex $v$ its left son $v_L$ is labeled by the element $b_1$ and its right son $v_R$ is labeled by the element $b_2$, then the vertex $v$ is labeled by the element $HASH(b_1, b_2)$. Thus, the root of a Merkle tree is labeled by the hash of all the data in the tree.

The Merkle trees are applied for resolving various Problems in Cryptography, such as consistency and audit proofs, data synchronization, etc. Currently an incremental construction of the Merkle trees, i.e. uncomplete and growing Merkle trees, are applied for time-stamps based on a public-key cryptosystems [8].

Let $A$ be the set of all strings that are either data, or hash values for some fixed hash-function $HASH$, and the binary operation $\diamond$ on the set $A$ is computing the hash-value for concatenation of two strings. Then we get the situation, that has been considered in Example 2.

## 8 Conclusions

Same as above, in this final section, "$A$" is the denotation of an arbitrary "fixed" set, and "$\mathcal{A}$" denotes both a superset of "$A$" and the magma with this superset as its support – a magma, which is referenced as "$A$-associated magma". These denotations introduced in section 5 will be used with the same meaning here.

This paper presents the results of a first attempt to characterize the imbrication algebras in terms of universal algebra. Among these results, the most noteworthy is elucidation of a number of deep internal links between any imbrication algebra and the infinite magma $A$.

The importance of the $A$-associated magma is that it can serve as a theoretical basis for the research of various algebraic systems with a single binary operation – algebraic systems, which are also referenced as "(algebraic) binary structures". Also, taking into account the established here interrelationship between the elements of the superset $\mathcal{A}$ of the set $A$ and the finite binary trees, one can expect that imbrication algebras can serve as a theoretical foundation for the implementation of provers and solvers, intended to deal with various binary structures.

The development of the structure of provers and solvers can be envisioned as one of the main directions of future research. In this regard, many problems arise naturally – problems, which have an importance on their own. Some of such problems are listed immediately below.

Firstly, this is the problem of software implementation of tokenization and parsing. The set $A$, which is "fixed" in this paper, can be either finite or countable, and it is natural to assume that the elements of the set $A$ are always presented as consecutive positive binary integers. Therefore, the problem of tokenization can be formulated as follows: *Is it true that, for any finite or countable set $A$, and a given string $w$, the string $w$ consists only of symbols "0", "1", "(", "," and ")"?* In case of an affirmative answer, the problem of parsing can be formulated like this: *for any finite or countable set $A$, and any given string $w$, is it true that $w \in \mathcal{A}$?* Possibly, the most effective is such an interaction of the two modules, when the parsing module calls, when necessary, the tokenization module.

Secondly, this is the problem of software implementation of checking of the parsing equivalence for the strings $w_1, w_2 \in \mathcal{A}$. This problem can be formulated like this: *given an arbitrary finite or countable set $A$, and two strings $w_1, w_2 \in \mathcal{A}$, is it true that $\tilde{\mathcal{D}}_{w_1} = \tilde{\mathcal{D}}_{w_1}$?*

Thirdly, this is the problem of software implementation for generation of all solutions of an equation in a given magma.

Fourthly, this is the problem of software implementation for generating all generalized Merkle trees (for the fixed hash function) of the given height, with the same label of the root.

## References


[1] I. Drugus, "Towards an Algebraic Set Theory for the Formalization of Data Structures," in *Proc. Conf. on Mathematical Foundations of Informatics MFOI2018, July 2-6, 2018*, (Chisinau, Moldova), 2018, pp. 73–89.

[2] I. Drugus, "A Wholebrain approach to the Web," in *Proc. of Web Intelligence – Intelligent Agent Technology Conference*, (Silicon Valley), 2007, pp. 68–71.

[3] I. Drugus, "Universics: a Common Formalization Framework for Brain Informatics and Semantic Web," in *Web Intelligence and Intelligent Agents*, Vucovar: InTech Publishers, 2010, pp. 55–78.

[4] S. Burris and H. P. Sankappanavar, *A course in universal algebra* (Graduate Texts in Mathematics, vol. 78), 1st ed., New-York, Heidelberg, Berlin: Springer-Verlag, 1981, 276 p. ISBN: 0-387-90578-2.

[5] B. Jónsson and A. Tarski, "On two properties of free algebras," *Mathematica Scandinavica*, vol. 9, 1961, pp. 95–101. https://doi.org/10.7146/math.scand.a-10627.

[6] V.A. Shcherbacov, *Elements of quasigroup theory and applications*, Boca Raton, Florida, USA: CRC Press, 2017, 576 p.

[7] A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*, Princeton, NJ, USA: Princeton University Press, 2016, 336 p.

[8] M. Ogava, E. Horita, and S. Ono, "Proving properties of incremental Merkel trees," *LNAI*, vol. 3632, 2005, pp. 424–440.

Ioachim M. Drugus, Volodymyr G. Skobelev Received September 30, 2018

Ioachim M. Drugus
Institute of Mathematics and Computer Science
5 Academiei str., MD-2028, Chisinau, Republic of Moldova
Phone: +373 699 79 938
E–mail: `ioachim.drugus@math.md`

Volodymyr G. Skobelev
V.M. Glushkov Institute of Cybernetics of NAS of Ukraine
40 Glushkova ave., Kyiv, Ukraine, 03187
Phone: +38 063 431 86 05
E–mail: `skobelevvg@gmail.com`