A formalism of the *object compounds* viewed as information processing support

Luciana Morogan
Military Technical Academy
Bucharest, Romania
morogan.lucian@gmail.com

Ajith Abraham
IT For Innovations, VSB - Technical University of Ostrava
Ostrava, Czech Republic
**Machine Intelligence Research Labs (MIR Labs)*
Washington, USA
ajith.abraham@ieee.org

***Abstract*—This paper aims to create a background for information processing support. We introduce formal classes called *object compounds*. By object compounds, we refer to the formalism needed for the construction of (biological inspired equivalent) classes of molecules, compounds or complex - like objects.**

***Keywords*-NaturalComputing; Soft Computing; formal languages; information processing**

I. Introduction

We observe an increasing interest and convergence among mathematics, computing and biology. By means of mathematical concepts and computing devices, the analyzed quantity of biological data is in continuous expansion. The concept of Natural Computing was introduced by several researchers ([5], [6], [14], [15]) in order to design computing systems mimicking nature inspiration by observing computational processes in nature or analyzing complex natural phenomena. We only refer to works like [1], [8-13] , [16-22] as a few examples of the variety of areas that can be touched by the nature inspiration methods. It is certain that at both levels of information technology and fundamental research, this highly interdisciplinary field connects all natural sciences with computing and mathematics sciences. An entire convolution of areas and topics are defining it by including, on one hand, as much pure theoretical research as algorithms and software applications (see [2] or [4]). On the other hand, areas of biology, chemistry and physics experimental research are also included. A meaningful interpretation of the meaning of Natural Computing was introduced in [6] as being ”the field of research that investigates models and computational techniques inspired by nature and, dually, attempts to understand the world around us in terms of information processing”.

Inspired by nature and in a natural computing spirit, we propose a theoretical design (for the moment) of object compounds. The purpose is to create a background for information processing support as by object compounds we understand the formalism needed in the construction of (biological inspired equivalent) classes of molecules, compounds or complex - like objects.

In the next Sections, we introduce the formalism imposed by the theoretical design of object compounds. We start with some preliminaries and then introduce the construction of the formal classes of object compounds.

II. Preliminaries

We assume the reader is familiar with the fundamental concepts from formal languages. Those readers who are not familiar may please refer to [7] and [15].

Let us choose O to be a nonempty finite alphabet of abstract symbols (called *objects*) representing chemicals in an aqueous solution. If we denote by O^* the set of all strings of objects in O and by $\lambda \in O^*$ the empty string, then O^* is the free monoid generated by O under the operation of *concatenation*. For $x \in O^*$ we may say that the string of objects x *may* represent the chemicals in solution needed for the construction of a well formed molecule (the order is not important). The notation $|x|$ represents the number of objects occurrences in x (the total number of the chemicals needed in the formation of a molecule represented by x) and we called it *the length* of x. For $a \in O$, the number of occurrences of object a in x is denoted $|x|_a$ and it is called the *multiplicity* of a in x (or simply the *number of copies* of object a in x), $|x|_a \in \mathbf{N}$. If we denote with $alph(x)$ ($alph(x) \subseteq O$) the set of symbol objects forming x, then

$$|x| = \sum_{a \in alph(x)} |x|_a.$$

The length of the string of objects obtained by the deletion from x of all symbols not in $U, U \subseteq O$ is denoted $|x|_U$ and

$$|x|_U = \sum_{a \in U} |x|_a.$$

The last working definition given here is that of the Parikh vector of x associated with O for $O = \{a_1, a_2, \ldots, a_m\}, m \in \mathbf{N}^+$ (the order of $a_1, a_2, \ldots, a_m$ is arbitrary but fixed) is defined as the function $\psi_O : O^* \to \mathbf{N}^m$,

$$\psi_O(x) = (|x|_{a_1}, |x|_{a_2}, \ldots, |x|_{a_m})$$

(for a language $L \subseteq O^*$ we have $\psi(L) = \{\psi_O(x) | x \in L\}$). We make the observation that all words in O^* have Parikh vectors of the same size.

We choose to denote the number of symbol objects occurrences in an aqueous solution by $|O|$ (not being confused with the number of objects in alphabet O). Symbol objects are considered from the alphabet O. For any $a \in O$, the number of occurrences of the object a in solution is denoted as $|O|_a$ and it is called the *multiplicity* of a in solution. The number $|O|_a \in \mathbf{N}$ represents the number of copies of a in solution ($a \in O$). If $|O|_a \overset{notation}{=} *$, then we say that the object a is found in an arbitrary number of copies.

As O is the alphabet *objects* representing chemicals in an aqueous solution, what has a major importance is the concentration, the number of copies of each object. In order to illustrate this, we consider a multiset as the data structure we will work with. Formally, a multiset over the alphabet O of objects is a mapping $M : O \to \mathbf{N} \cup \{*\}$, defined by

$$M(a) = \begin{cases} |M|_a, & \text{if } a \in O \text{ and it represents the number} \\ & \text{of copies of } a \text{ into the multiset M} \\ *, & \text{if there is an arbitrary finite number} \\ & \text{of copies of } a \text{ into the multiset M} \end{cases}$$

We make the observation that the mapping M, describing the multiset, must not be confused with the notation $|M|_a$ representing the number of copies of a into the multiset M. We denote by $alph(M) = \{a \in O | M(a) > 0\}$ the finite alphabet of the multiset M (one may say the support of M). Generally, we can consider that $S \subseteq O$ is interpreted as the multiset defined by

$$\begin{cases} \forall a \in S, & S(a) = 1 \\ \forall a \notin S, & S(a) = 0 \end{cases}$$

(any subset of O is a multiset in which each object element is found in a unique copy).

For any two multisets $M_1, M_2 : O \to \mathbf{N} \cup \{*\}$ a series of properties can be defined:

- For $M_1(a) \neq *$ and $M_2(a) \neq *$, we say that M_2 is *included* in M_1 if and only if $M_1(a) \geq M_2(a)$, for all $a \in O$.
- The *union* of M_1 with M_2 is the multiset $M_1 \cup M_2 : O \to \mathbf{N} \cup \{*\}$, $(M_1 \cup M_2)(a) = M_1(a) + M_2(a)$, for all $a \in O$. (If $M_1(a) = *$ or $M_2(a) = *$ then $(M_1 \cup M_2)(a) = *$.)
- The *difference* between M_1 and M_2 (defined only if M_2 is included in M_1) is the multiset $M_1 - M_2 : O \overset{\circ}{\longrightarrow} \mathbf{N} \cup \{*\}$, $(M_1 - M_2)(a) = M_1(a) - M_2(a)$, for all $a \in O$, $M_1(a) \geq M_2(a)$, $M_1(a), M_2(a)$ finite. (If $M_1(a) = *$, then $(M_1 - M_2)(a) = *$)

From these properties above, a multiset M over a finite support can be written as

$$M = \{(a, M(a)) | a \in alph(M)\}.$$

Example. For $alph(M_1) = alph(M_2) = \{a, b, c \in O | M_i(a) > 0, M_i(b) > 0, M_i(c) > 0, i \in \{1, 2\}\}$, if $M_1 = \{(a, 3), (b, 2), (c, *)\}$ then M_1 is the multiset over O with the support $alph(M_1)$ and consists in three objects a, b, c such as object a appears in three copies, b is found in two copies and c in an arbitrary finite number of copies in M_1. Similarly, $M_2 = \{(a, 1), (b, 2), (c, 3)\}$. It can be easily seen that M_2 is included in M_1.

- As $(M_1 \cup M_2)(a) = M_1(a) + M_2(a) = 3 + 1 = 4$, $(M_1 \cup M_2)(b) = M_1(b) + M_2(b) = 4$ and $(M_1 \cup M_2)(c) = M_1(c) + M_2(c) = *$, we obtain the multiset $M_1 \cup M_2 = \{(a, 4), (b, 4), (c, *)\}$.
- As $(M_1 - M_2)(a) = M_1(a) - M_2(a) = 3 - 1 = 2$, $(M_1 - M_2)(b) = M_1(b) - M_2(b) = 0$ and $(M_1 - M_2)(c) = M_1(c) - M_2(c) = *$, we obtain the multiset $M_1 - M_2 = \{(a, 2), (c, *)\}$.

Obviously, every string $x \in O^*$ describes a multiset M_x over O and we write $M_x = \{(a, |x|_a) | a \in alph(x)\}$. We recall that the alphabet O is defined as $O = \{a_1, a_2, \ldots, a_m\}, m \in \mathbf{N}^+$. Let us consider the string $x = a_{i_1} a_{i_2} \ldots a_{i_n} \in O^*, n \in \mathbf{N}^+$ being finite, $1 \leq i_j \leq m$, $1 \leq j \leq n$ (the order of $a_{i_1}, a_{i_2}, \ldots, a_{i_n}$ is arbitrary in the alphabet O, but fixed) and S_n the set of all permutations of n elements. If we choose any permutation $\pi \in S_n$, then we define a mapping $f_\pi : O^n \to O^n$ such as $f_\pi(a_{i_1} \ldots a_{i_n}) = a_{\pi(i_1)} \ldots a_{\pi(i_n)} \overset{notation}{=} y, y \in O^*$ and $\psi_O(y) = (|y|_{a_1}, |y|_{a_2}, \ldots, |y|_{a_m})$. We defined y, a permutation of x.
From

- $alph(x) = alph(y)$;
- $\psi_O(y) = (|y|_{a_1}, |y|_{a_2}, \ldots, |y|_{a_m})$;
- because y is a permutation of x,

$$(|x|_{a_1}, |x|_{a_2}, \ldots, |x|_{a_m}) = (|y|_{a_1}, |y|_{a_2}, \ldots, |y|_{a_m});$$

we have that $\psi_O(x) = \psi_O(y)$.
So, if

- $M_x = \{(a_{i_1}, |x|_{a_{i_1}}), (a_{i_2}, |x|_{a_{i_2}}), \ldots (a_{i_n}, |x|_{a_{i_n}})\}$;
- $M_y = \left\{(a_{\pi(i_1)}, |y|_{a_{\pi(i_1)}}), (a_{\pi(i_2)}, |y|_{a_{\pi(i_2)}}), \ldots (a_{\pi(i_n)}, |y|_{a_{\pi(i_n)}})\right\}$;
- y is obtained by a permutation of x;

then $M_x = M_y$.
We may say that permutations of the same string $x \in O^*$ are describing exactly the objects in the support of M and their multiplicities (permutations of the same string are describing exactly the same multiset over O).

We will represent the multiset $M = \{(a_{i_1}, M(a_{i_1})), (a_{i_2}, M(a_{i_2})), \ldots, (a_{i_n}, M(a_{i_n}))\}$, with $alph(M) = \{a_{i_1}, \ldots, a_{i_n}\}$, as the string $a_{i_1}^{M(a_{i_1})} a_{i_2}^{M(a_{i_2})} \ldots a_{i_n}^{M(a_{i_n})}$.

Example. Let us consider the string $x \in O^*, x = abcaadff$ with $alph(x) = \{a, b, c, d, f\}$. The Parikh vector associated with x over O is

$$\psi_O(x) = (|x|_a, |x|_b, |x|_c, |x|_d, |x|_f) = (3, 1, 1, 1, 2).$$

If we consider the permutation $\pi \in S_8$,

$$\pi = \left(\begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 3 & 1 & 2 & 5 & 4 & 8 & 7 & 6 \end{array} \right)$$

such as there is a mapping $f_\pi : O^8 \to O^8$ with $f_\pi(a_1 \dots a_8) = a_{\pi(1)} \dots a_{\pi(8)} = y$, then $f_\pi(abcaadff) = cabaaffd = y$ and the Parikh vector associated with y is $\psi_O(y) = (|y|_a, |y|_b, |y|_c, |y|_d, |y|_f) = (3, 1, 1, 1, 2)$. As $alph(y) = alph(x)$, we have that $\psi_O(x) = \psi_O(y)$ when y is a permuted string obtained from x.

The notion of a multiset may be extended for strings. A multiset over O^* is a mapping $M : O^* \to \mathbf{N} \cup \{*\}$, defined by

$$M(x) = \begin{cases} |M|_x, & \text{if } x \in O^* \text{ and it represents the number} \\ & \text{of copies of the string } x \text{ into the} \\ & \text{multiset M} \\ *, & \text{if there is an arbitrary finite number} \\ & \text{of copies of the string } x \text{ into the} \\ & \text{multiset M} \end{cases}$$

Once again, we make the observation that the mapping M, describing the multiset, must not be confused with the notation $|M|_x$ representing the number of copies of x into the multiset M.
We denote by

$$alph(M) = \{x \in O^* | \exists a_{i_j} \in alph(x), i_j = \overline{1, m}, j = \overline{1, n},$$

$n \in \mathbf{N}, alph(x) \subseteq O$, such as $M(a_{i_j}) > 0$ and $\psi_O(x) = (|x|_{a_1}, \dots, |x|_{a_m})\}$

the finite alphabet of multiset M (one may say the support of M). Generally, we can consider that $S \subseteq O^*$ is interpreted as the multiset defined by

$$\begin{cases} \forall x \in S, & S(x) = 1 \\ \forall x \notin S, & S(x) = 0 \end{cases}$$

(any subset of O^* is a multiset in which each string object is found in a unique copy).

The strings $x \in alph(M)$ are legally strings of chemicals in solution (corresponding to the biologically micro-molecules) needed in the formation of well formed macro-molecules.

The multiset M of a finite support can be represented either by a set $M = \{(x, M(x)) | x \in alph(M)\}$, either as a string of the form $x_1^{M(x_1)} x_2^{M(x_2)} \dots x_n^{M(x_n)}$, for $alph(M) = \{x_1, \dots, x_n\}$ and the multiplicities of the objects in the support $M(x_1), M(x_2), \dots, M(x_n)$. All permutations of this string identify objects in the support (alphabet) of M and their multiplicities. For two multisets M_1 and M_2 the inclusion, union and difference between them are defined as above.

Example. The string $(ab)^3(abb)^2(aa)^*$ describes the multiset of string objects containing three copies of ab, two copies of abb and an arbitrary number of copies of the string aa. Here, $alph(M) = \{ab, abb, aa\} \cup \{(,)\}$, where the parentheses are used for a better representation of string molecules in M.

III. Formal classes of object compounds

In this section we formalize an hierarchical construction comprising the three nested languages that describe the primary, secondary and tertiary structures of object compounds. Objects are the elementary data we will work with. By object compounds we understand the formalism needed in the construction of classes of molecules, compounds or complex - like objects (for biological inspired equivalent see [3]).

A. Formalization of primary structures of object compounds

We construct here a finite set of rules for object compounds formation. We will refer to the formed object compound with the term of *molecular formula*. The construction is inspired from biological molecular compounds formation (molecules, macro-molecules, organic compounds or organic complexes).

We consider the sets of string objects over O^* recurrently defined as follows: if for all $k, i \in \{1, ..., n\}$ we denote by $M_k(a_i)$ the multiplicity of a_i in M_k for all objects $a_i \in alph(M_k)$ with $alph(M_k) \subseteq O^*$, then

- $M_1 = \{a_i^{M_1(a_i)} | \forall i \in \{1, ..., n\}, a_i \in alph(M_1), alph(M_1) = \{a_1, ..., a_n\}\}$
- $M_2 = \{a_{i_1}^{M_2(a_{i_1})} a_{i_2}^{M_2(a_{i_2})} | \forall i_1, i_2 \in \{1, ..., n\}, a_{i_1}, a_{i_2} \in alph(M_2), alph(M_2) = \{a_1 a_2, a_1 a_3, ..., a_n a_{n-1}\} \cup alph(M_1)\}$

...

- $M_n = \{a_1^{M_n(a_1)} a_2^{M_n(a_2)} ... a_n^{M_n(a_n)} | \forall i \in \{1, ..., n\}, a_i \in alph(M_n), alph(M_n) = \{a_1 a_2 ... a_n\} \cup alph(M_{n-1}) \cup \dots \cup alph(M_1)\}$

For all $k \in \{1, ..., n\}$, we define

$M_k = \{a_{i_1}^{M_k(a_{i_1})} a_{i_2}^{M_k(a_{i_2})} ... a_{i_k}^{M_k(a_{i_k})} | \forall i_1, ..., i_k \in \{1, ..., n\},$ $a_{i_1}, ..., a_{i_k} \in alph(M_k), alph(M_k) = \{a_1 a_2 ... a_k, \dots,$ $a_{n-k+1} a_{n-k+2} ... a_n\} \cup alph(M_{k-1}) \cup \dots \cup alph(M_1)$ and for all $r_1, r_2 \in \{1, ..., n\}$ with $r_1 \neq r_2$ we have $i_{r_1} \neq i_{r_2}\}$.

If we denote $[M] = \{M_k | k \in \{1, .., n\}, n \in \mathbf{N}, n \geq 1$ $(finite), M_k$ defined as above$\}$, then we have $[M]^*$ the set of all string objects formed with elements of M_k for

$k \in \{1, ..., n\}$ under the operation of concatenation.

Over $[M]^*$ we define a *rewriting rule*, denoted "⇒" ($\Rightarrow \subseteq [M]^* \times [M]^*$), such as for any $k_1, k_2 \in \{1, ..., n\}$ with $k_1 \geq k_2$ and $M_{k_1}, M_{k_2} \subseteq [M]^*$ the pair $(x_1, x_2) \in M_{k_1} \times M_{k_2}$. We say that the string x_1 is *rewrited* into the string x_2 and we write $x_1 \Rightarrow x_2$ if and only if the string $x_1 \in M_{k_1}$ has the form

$$x_1 = a_{i_1}^{M_{k_1}(a_{i_1})} a_{i_2}^{M_{k_1}(a_{i_2})} ... a_{i_{r_1}}^{M_{k_1}(a_{i_{r_1}})} ... a_{i_{r_p}}^{M_{k_1}(a_{i_{r_p}})} ... a_{i_{k_1}}^{M_{k_1}(a_{i_{k_1}})}$$

such as

- there is $r_1, r_2, ..., r_p \in \{1, ..., k_1\}$, $1 \leq p \leq k_1$ with $i_{r_1} = i_{r_2} = ... = i_{r_p} \overset{notation}{=} i_r$
- for all $j \in \{1, ..., k_1\}$ with $j \neq r_l$ for all $l \in \{1, ..., p\}$ we have $i_j \neq i_{r_l}$

then the string $x_2 \in M_{k_2}$ has the form

$$x_2 = a_{i_1}^{M_{k_2}(a_{i_1})} ... a_{i_r}^{M_{k_2}(a_{i_r})} ... a_{i_{k_2}}^{M_{k_2}(a_{i_{k_2}})}$$

where

- $M_{k_2}(a_{i_r}) = M_{k_1}(a_{i_{r_1}}) + M_{k_1}(a_{i_{r_2}}) + ... + M_{k_1}(a_{i_{r_p}})$
- if for all $j_1 \in \{1, ..., k_1\}$ with $j_1 \neq r_l$, for all $l \in \{1, ..., p\}$ and for all $j_2 \in \{1, ..., k_2\}$ with $j_2 \neq r$, we have $i_{j_1} = i_{j_2}$, then $M_{k_1}(a_{i_{j_1}}) = M_{k_2}(a_{i_{j_2}})$.

We make the observation that this definition must not be confusing regarding one possible condition found in the writing of x_1 consisting in $i_{r_1} = i_{r_2} = ... = i_{r_p} \overset{notation}{=} i_r$ with the definition of terms of M_k above which must be distinct. The reason of this potential confusion can be easily understand, but the reader can observe the distinction between M_k belonging to $[M]$ and x_1 belonging to M_{k_1} which in turn is included in $[M]^*$ ($[M]$ is not the same with $[M]^*$).

We will denote the set $[M]^*_{\Rightarrow} = \{x_2 \in M_{k_2} |$ there is $(x_1, x_2) \in M_{k_1} \times M_{k_2}$ for $k_1 \geq k_2$ and $M_{k_1}, M_{k_2} \in [M]^*$ such as for all $x_1 \in M_{k_1}, x_1 \Rightarrow x_2\}$. The set $[M]^*_{\Rightarrow}$ is called the *compounds domain* and, to simplify the writing, from now on we will refer to it by the notation CD.

Example. Let us take two strings from M_1, let say $a_1^{M_1(a_1)}$ and $a_2^{M_1(a_2)}$. Under the concatenation, a new string $a_1^{M_1(a_1)} a_2^{M_1(a_2)}$ is formed. It belongs to M_2. *Example.* Considering the string objects $a_1^{M_1(a_1)} \in M_1$ and $a_1^{M_2(a_1)} a_2^{M_2(a_2)} \in M_2$, under the concatenation operation, it results the new string $a_1^{M_1(a_1)} a_1^{M_2(a_1)} a_2^{M_2(a_2)}$. Applying the rewriting rule above, in M_2 is obtained a new string of the form $a_1^{M_1(a_1)+M_2(a_1)} a_2^{M_2(a_2)}$.

In the following we will consider P_1 a finite set of rules in $M_1 \times ... \times M_1 \times CD$.

1) Molecular formula production rules: Production rules of molecular formulas are elements of the form $(x, y) \in P_1$ with $x \in M_1 \times ... \times M_1$. We consider $M_1 \times ... \times M_1$ of m times if we suppose we want to form an object compound formula in which composition we need m elementary objects (each of them representing the biological equivalent of chemicals swimming in an aqueous solution). We have $y \in CD$. The rules have the form:

$$M_1/x \rightarrow M_1^{new}/y$$

where if

- $M_1 = \Big\{a_1^{M(a_1)}, ..., a_{i_1}^{M(a_{i_1})}, ..., a_{i_2}^{M(a_{i_2})}, ..., a_{i_m}^{M(a_{i_m})}, ... a_n^{M(a_n)} | \forall k \in \{1, ..., n\}, a_k \in alph(M_1), alph(M_1) = \{a_1, ..., a_n\}\}$ and
- $x = (a_{i_1}^{j_1}, a_{i_2}^{j_2}, ..., a_{i_m}^{j_m})$ for $j_k \in \mathbf{N}, 0 \leq j_k \leq M(a_{i_k})$

then

- $y = a_{i_1}^{j_1} a_{i_2}^{j_2} ... a_{i_m}^{j_m}$ and
- $M_1^{new} = \Big\{a_1^{M(a_1)}, ..., a_{i_1}^{M(a_{i_1})-j_1}, ..., a_{i_2}^{M(a_{i_2})-j_2}, ..., a_{i_m}^{M(a_{i_m})-j_m}, ... a_n^{M(a_n)} | \forall k \in \{1, ..., n\}, a_k \in alph(M_1), alph(M_1) = \{a_1, ..., a_n\}\}$.

If there is $a_k \in M_1$ such as $M_1(a_k) = *$, then $M_1(a_k) - j_k = *$.

We call such elements $y \in CD$ well-formed *molecular formulas*.

2) Molecular formula decomposition rules: Decomposition rules of molecular formulas are elements of the form $(y, x) \in P_1$ with $x \in M_1 \times ... \times M_1$. We consider $M_1 \times ... \times M_1$ of m times if we suppose that, by applying this rule, the molecular formula is decomposed into m elementary objects. We have $y \in CD$. The rules have the form:

$$M_1/y \rightarrow M_1^{new}/x$$

where if

- $M_1 = \Big\{a_1^{M(a_1)}, ..., a_{i_1}^{M(a_{i_1})}, ..., a_{i_2}^{M(a_{i_2})}, ..., a_{i_m}^{M(a_{i_m}}, ... a_n^{M(a_n)} | \forall k \in \{1, ..., n\}, a_k \in alph(M_1), alph(M_1) = \{a_1, ..., a_n\}\}$ and
- $y = a_{i_1}^{j_1} a_{i_2}^{j_2} ... a_{i_m}^{j_m}$

then

- $x = (a_{i_1}^{j_1}, a_{i_2}^{j_2}, ..., a_{i_m}^{j_m})$ for $j_k \in \mathbf{N}, 0 \leq j_k \leq M(a_{i_k})$ and
- $M_1^{new} = \Big\{a_1^{M(a_1)}, ..., a_{i_1}^{M(a_{i_1})+j_1}, ..., a_{i_2}^{M(a_{i_2})+j_2}, ..., a_{i_m}^{M(a_{i_m})+j_m}, ... a_n^{M(a_n)} | \forall k \in \{1, ..., n\}, a_k \in alph(M_1), alph(M_1) = \{a_1, ..., a_n\}\}$.

If there is $a_k \in M_1$ such as $M_1(a_k) = *$, then $M_1(a_k) + j_k = *$.

Example. If we consider $alph(M_1) = \{a, b, c\}$ and $M_1 = \left\{a^3, b^2, c^*\right\}$, then the molecule a^3c^5 is obtained by applying the rule $M_1/(a^3, c^5) \rightarrow M_1^{new}/a^3c^5$ with $M_1^{new} = \left\{b^2, c^*\right\}$.
Example. At the decomposition of the molecular formula a^3c^5, if we suppose that the initial content of M_1 from the previous example, then the rule $M_1/a^3c^5 \rightarrow M_1^{new}/(a^3, c^5)$ is applied and the content of M_1^{new} is $M_1^{new} = \left\{a^6, b^2, c^*\right\}$.

3) Primary structures of object compounds: If instead of $x \in M_1 \times ... \times M_1$ we use elements $x \in M_{k_1} \times ... \times M_{k_n}$, for any arbitrary or any specific choice of $k_1, ..., k_n \in \{1, ..., n\}$, then we get well-formed molecular formulas for any object compounds (thus reflecting the biological inspired molecules, macro-molecules, organic compounds or organic complexes). The set of all compounds obtained in ways described above is called the *set (language) of primary structures* of object compounds. In conclusion, we illustrated, so far, that *the primary structure construction of an object compound (one may say object complex) is represented by its molecular formula*. The set (the language) of all primary structures of object compounds (object complexes) is presented below:

$S_I = \{y \in CD|$ there is the set of rules P_1 over $M_{k_1} \times ... \times M_{k_n} \times CD$ and there is $x \in M_{k_1} \times ... \times M_{k_n}$, for any arbitrary order or any specific choice of $k_1, ..., k_n \in \{1, ..., n\}$, such as $(x, y) \in P_1\}$.

(The rule $(x, y) \in P_1$ has the form $M_{k_1} \times ... \times M_{k_n}/x \rightarrow M_{k_1}^{new} \times ... \times M_{k_n}^{new}/y$ defined as above.)

B. Formalization of secondary structures of object compounds

We consider P_2 a set of rules attaching a *structural arrangement* to the primary structure of an object compound, we call it the *secondary structure*. The secondary structure construction of an object compound is represented by its *structural formula*. This structural arrangement is inspired by the way that objects bind together in molecules formation, in a mathematical sense, without having the same complexity as in a real biology way. The idea of defining such a structural formula comes from some physical and chemical properties of a real organic compound: the way atoms (different groups of atoms) bind together in forming organic compounds or complexes.

In our model, we consider two more symbols over CD: ”|” and ”—”. The symbol ”|” is used to illustrate the attachment of the structural arrangement to the primary structure of a compound, while ”—” represents the bounds between atom-like objects (different groups of atom-like objects) in compounds (complexes) formation. We make once again the observation that although in biology and bio-chemistry there are different types of chemical bounds, it is not our goal to represent an accurate reality, but to construct a mathematical model of a molecule, compound or complex. This is the reason why the symbol ”—” will represent only one bounding type (or all bounding types if one prefers) between atoms-like objects (between different groups of atoms-like objects).

The rule of structural arrangement of an object molecule/compound/complex in P_2, has the form $(y, w) \in P_2$ with $P_2 \subseteq S_I \times CD \cup \{|, -\}$ and we denote it by the symbol $\vdash_{S_I}$. It associates, to each molecular formula, a distribution of atoms-like objects (different groups of atoms-like objects) in the formation of a structural formula.

For $y \in S_I$ of the form $y = a_1^{j_1}...a_m^{j_m}$ we have $y \vdash_{S_I} w$ if and only if

- $z = a_1^{j_{11}}...a_m^{j_{m1}} - a_1^{j_{12}}...a_m^{j_{m2}} - ..... - a_1^{j_{1p}}...a_m^{j_{mp}}$, for $1 \leq p \leq m, p \in \mathbf{N}$ and for all $l \in \{1, ..., m\}$ we have $\sum_{k=1}^{p} j_{lk} = j_l$ with $0 \leq j_{lk} \leq j_l$ where
  - p represents the number of atoms-like (groups/complex - like) objects y is formed of;
  - If $j_{lk} = 0$ then the proper group will no contain the element $a_l^{j_{lk}}$ for $k \in \{1, ..., p\}$ and $l \in \{1, ..., m\}, k, l, \in \mathbf{N}$;
  - Each $a_1^{j_{1k}}...a_m^{j_{mk}}$ with $k \in \{1, ..., p\}$ is an element from CD;
- We denote by Nom a complete set of all legally existing objects, string objects or string object compounds coming from a domain where the applicability of this construction is needed. We need this set to validate the well-formed structural formulas. Let it be a mapping $ver : CD \cup \{-\} \rightarrow \{0, 1\}$, such as

  $$ver(z) = \begin{cases} 1, & \text{if } z \in Nom \\ 0, & \text{otherwise} \end{cases}$$

  If $ver(z) = 1$ then $w \overset{notation}{=} z$ and w is called a well-formed *structural formula* and it will be denoted by $\overline{z}$.

The attachment of the structural formula to its molecular formula defines *the set (the language) of all secondary structures* of object compounds (object complexes) and it is denoted by:

$S_{II} = \{(y|\overline{z})|$ for all $y \in S_I$, there is $\overline{z} \in CD \cup \{|, -\}$ such as $y \vdash_{S_I} \overline{z}\}$.

C. Formalization of tertiary structures of object compounds

We consider the set P_3 of rules attaching a spatial orientation (of atoms/groups of atoms-like objects in its constitution) to a compound from S_{II}. The *universal alphabet* of all possible spatial configurations of objects is Γ. As our work has its inspiration in biology where the spatial arrangement of atoms/groups of atoms in a real molecule is finite (for example, there are only a few possibilities of different protein spatial conformations), the term *universal alphabet* is used here to refer to all three-dimensional structures of all object compound types. In other words, Γ is the set of all three-dimensional structures, the spatial arrangements of the secondary structures in S_{II} and we will refer to it as *the set of all spatial configurations*.

Let it be $\Gamma = \{\Gamma_1, \Gamma_2, ..., \Gamma_q\}$, where $q \in \mathbf{N}, q \geq 1, q$ fixed. Γ^* is the set of all possible strings of configurations under the operation of concatenation. A rule in P_3 has the form $(u, v) \in P_3$ with $P_3 \subseteq S_{II} \times \Gamma^*$ and we denote it by the symbol $\models_{S_{II}}$. We write that

$$\text{for } u \in S_{II} \text{ there is } v \in \Gamma^* \text{ such as } u \models_{S_{II}} (u|v).$$

So, we have

$$\text{for } u \in S_{II}, u = (y|\overline{z}) \text{ there is } v \in \Gamma^* \text{ such as } (y|\overline{z}) \models_{S_{II}} (y|\overline{z}|v).$$

We define the *set (language) of all tertiary structures* of object compounds (object complexes) as

$S_{III} = \{(y|\overline{z}|v)|$ for $y \in S_I, y \vdash_{S_I} \overline{z}$ there is $v \in \Gamma^*$ such as $(y|\overline{z}) \models_{S_{II}} (y|\overline{z}|v)\}$.

This hierarchical construction of the complex cascading language of objects molecules (compounds or complexes), comprising the three nested languages describing their primary, secondary and tertiary structures, can be graphically represented as it can be seen in **Figure 1**.

IV. Conclusions

Still, found in its early stages, a generally accepted common language to describe the biological inspired ideas already introduced within the pages of this article is still to be developed and commonly accepted. We introduced the original construction of the formal classes in defining the object compounds (a formalism needed in the construction of biological inspired equivalent classes of molecules, compound or complex - like objects). The purpose was found into creating a formal background for this kind of information processing support.

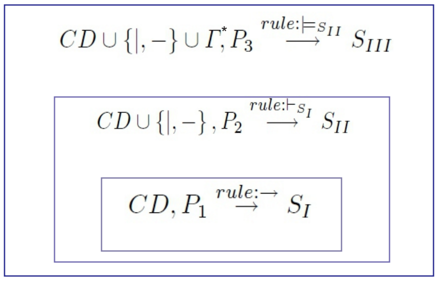
$CD \cup \{|, -\} \cup \Gamma^*, P_3 \overset{rule:\models_{S_{II}}}{\longrightarrow} S_{III}$
$CD \cup \{|, -\}, P_2 \overset{rule:\vdash_{S_I}}{\longrightarrow} S_{II}$
$CD, P_1 \overset{rule:\rightarrow}{\rightarrow} S_I$

Figure 1. The hierarchically construction of the complex cascading language of objects compounds. It comprisis three nested languages describing their primary, secondary and tertiary structures.

References

[1] A. Abraham, *Neuro-Fuzzy Systems: State-of-the-Art Modeling Techniques* in ”Connectionist Models of Neurons, Learning Processes, and Artificial Intelligence”, Lecture Notes in Computer Science, Volume. 2084, J. Mira and A. Prieto (Eds.), Springer-Verlag Germany, 2001, pp. 269-276.

[2] A. Abraham, E. Corchado, J. M. Corchado, *Hybrid learning machines* in ”Elsevier Science”, Volume 72, Issues 13-15, August 2009, pp. 2729-2730.

[3] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts and P. Walter, *Molecular Biology of the Cell*, 4th ed. New York, USA: Garland Science, 2002.

[4] E. Corchado, A. Abraham, A. de Carvalho, *Editorial: Hybrid intelligent algorithms and applications* in ”Information Sciences: an International Journal”, Volume 180, Issue 14, Elsevier Science Inc., July 2010, pp. 2633-2634.

[5] L. N. de Castro, *Fundamentals of Natural Computing: An Overview* in ”Physics of Life Reviews”, 4, March, 2007, pp. 1-36. Available: http://linkinghub.elsevier.com/retrieve/pii/S1571064506000315.

[6] L. Kari and G. Rozenberg, *The Many Facets of Natural Computing* in ”Communications of the ACM”, 51, October, 2008, pp. 72-83. Available: http://portal.acm.org/citation.cfm?id=1400200.

[7] M. Lothaire, *Combinatorics on words. Cambridge Mathematical Library, Encyclopedia of mathematics and its applications*, 2nd ed., Cambridge University Press, 1997.

[8] G. Michaels, *Cryptography and linguistic of macromolecules, Studying the codes and language of life* in ”EMSL Lecture notes”, April, 2004.

[9] L. Morogan, *A Model of Synaptic Formation Between Neurons in a Network* in ”International Journal of Computer Information Systems and Industrial Management Applications”, 2011, pp. 88-95.

[10] L. Morogan, *View from inside one neuron* in "Proceedings of The International Conference on Digital Information Processing and Communications" (ICDIPC2011), "Communications in Computer and Information Science" (CCIS) Series of Springer LNCS, 2011.

[11] S. Mukkamala, A. H. Sung and A. Abraham, *Intrusion Detection Using Ensemble of Soft Computing Paradigms*, Third International Conference on Intelligent Systems Design and Applications, Intelligent Systems Design and Applications, Advances in Soft Computing, Germany: Springer-Verlag, 2003, pp. 239-248.

[12] Gh. Paun and M.J. Perez-Jimenez, *Recent computing models inspired from biology: DNA and membrane computing* in "Theoria", No. 18, 2003, pp. 72-84.

[13] J. Reif and T. LaBean, *Autonomous programmable biomolecular devices using self-assembled DNA nanostructures* in "CACM", 50(9), 2007, pp. 46-53.

[14] G. Rozenberg, *Computer science, informatics and natural computing personal reflections* in "New Computational Paradigms: Changing Conceptions of What Is Computable", Berlin Heidelberg: Springer-Verlag, 2008, pp. 373-379.

[15] G. Rozenberg and A. Salomaa, *Handbook of Formal Languages: Word, language, grammar*, Berlin Heidelberg: Springer-Verlag, 1997.

[16] A. Abraham, Artificial Neural Networks, Handbook for Measurement Systems Design, Peter Sydenham and Richard Thorn (Eds.), John Wiley and Sons Ltd., London, ISBN 0-470-02143-8, pp. 901-908, 2005.

[17] A. Abraham, Intelligent Systems: Architectures and Perspectives, Recent Advances in Intelligent Paradigms and Applications, Abraham A., Jain L. and Kacprzyk J. (Eds.), Studies in Fuzziness and Soft Computing, Springer Verlag Germany, ISBN 3790815381, Chapter 1, pp. 1-35, 2002.

[18] J. Corchado, J. Bajo, D. Tapia and A. Abraham, Using Heterogeneous Wireless Sensor Networks in a Telemonitoring System for Healthcare, IEEE Transactions on Information Technology in Biomedicine, IEEE, USA, Volume 14, Issue 2, pp. 234-240, 2010.

[19] F. Xhafa, E. Alba, B. Dorronsoro, B. Duran and A. Abraham, Efficient Batch Job Scheduling in Grids Using Cellular Memetic Algorithms, Studies in Computational Intelligence, Springer Verlag, Germany, ISBN: 978-3-540-69260-7, pp. 273-299, 2008.

[20] A.Abraham, EvoNF: A Framework for Optimization of Fuzzy Inference Systems Using Neural Network Learning and Evolutionary Computation, The 17th IEEE International Symposium on Intelligent Control, ISIC02, IEEE Press, ISBN 0780376218, pp 327-332, 2002.

[21] H. Liu, A. Abraham and A. Ella Hassanien, Scheduling Jobs on Computational Grids Using Fuzzy Particle Swarm Algorithm, Future Generation Computing Systems, Elsevier Science, Netherlands, Volume 26, pp. 1336-1343, 2010.

[22] H. Liu, A. Abraham and M. Clerc, Chaotic Dynamic Characteristics in Swarm Intelligence, Applied Soft Computing Journal, Elsevier Science, Volume 7, Issue 3, pp. 1019-1026, 2007.