Journal of Theoretical and Applied Information Technology
30th November 2013. Vol. 57 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A STUDY ON VARIOUS BIO-INSPIRED COMPUTING MODELS

ANAND MAHENDRAN [#1], RAMYA GOVINDARAJ [*1]

[#]Associate Professor,School of Computing Sciences and Engineering,
VIT University,Vellore-632014, Tamilnadu, India.
*Assistant Professor(Senior) ,School of Information Technology and Engineering,
VIT University,Vellore-632014, Tamilnadu, India.
E-mail: [#]manand@vit.ac.in , [*]engr_ramyaa@yahoo.com

ABSTRACT

Natural computing is the field of research that investigates various computing models and computational techniques which are inspired from nature. It is an interdisciplinary field that nudges the natural sciences with computing science and attempts to understand the world around us in terms of information processing. In the last few decades, natural computing which includes biologically inspired computing is an area being pursued with a great deal of interest. In this paper, we make an attempt to explore the working nature, applications, variants and generative power of various bio-inspired computing models such as membrane computing, insertion-deletion systems, Watson-Crick automata, splicing systems and sticker systems.

Keywords: *Bio-Inspired Computing, Membrane Computing, Watson-Crick Automata, Splicing System, Sticker System.*

1. INTRODUCTION

Bio-inspired computing is the field which relates mathematics, computer science and biology. The mathematical models that are created for living things using computer science concepts are called as bio-inspired computing models. Formal language theory concepts are used in the development of these computation models. Here, first we start explaining with membrane computing system, which deals with the abstraction of computing ideas and the functioning of living cells. It defines the way the cells are organized in tissues or in higher order structures, and how it is separated from its environment by a membrane. Insertion and deletion operations are observed to be common in the field of biology. In particular, DNA processing along with the RNA editing can be implemented and formulated in formal languages [9, 10, 11]. Based on the Insertion-deletion system and matrix grammar, matrix insertion deletion system has been developed [13, 14]. Watson-Crick automata, initiated in [16], represents best instance in mathematical model which abstracts biological properties for a number of computational purposes. Some variants and limitations of Watson-Crick automata were identified and investigated. For instance, initial stateless-Watson-Crick finite automata, reverse-Watson-Crick automata, two-way finite Watson-Crick automata, Watson-Crick memory with Watson-Crick transducers were measured in [17]. The operation of the splicing system relates with the behavior of DNA molecules under the restriction of ligases and enzymes. In 1994, Leonard M.Adleman puts his theory of DNA computing to the test on a problem called the *Traveling salesman problem* (formally known as directed Hamiltonian Path-HP problem) which is about finding the shortest path through a series of points [4]. Sticker system is one of the computational model sticker systems which uses Watson-crick method in DNA computing. This paper is organized as follows: section-II deals with biological computing, section-III deals with membrane computing, section-IV deals with matrix insertion deletion system, section-V deals with Watson-Crick automata, section-VI deals with splicing system, section-VII deals with sticker system and finally conclusion.

2. BIOLOGICAL COMPUTING

Biological systems have many advantages over computer systems, as they use far less energy, can survive faults and are even able to get a good result. The domain of biocomputation has a couple of classification [1, 2, 3]: the use of biology or biological processes as inspiration, metaphor or enabler in increasing new computing technologies. In addition to that, latest areas of computer science field and the information science tools have contributed models to investigate biology from a

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

special theoretical perspective. For instance, DNA computation, storage devices, nanofabrication and sensing, health care, biocomputation also has implications for necessary scientific exploration. It can offer biologists, an IT-oriented standard to perform computation of cells or else process information of it, or help computer scientists to construct the algorithms supported on natural systems. The best example for this computation is evolutionary genetic algorithms. Bio computing comprises the potential extremely controlling tool.

3. MEMBRANE COMPUTING

Membrane computing is about the functioning of living cells and its cooperation. Also it defines the way the cells are organized in tissues or in higher order structures. Membranes enclose the protected reactors compartments where specific biochemical processes take place in inside the cell. It encloses the nucleus where the genetic material is placed. Vesicles (small bubble within a cell) enclosed by membranes can transport packages of molecules from a part of the cell to other parts of the cell. S.Marcus puts it in an equational form: Life = DNA software + membrane hardware. There are cells living alone named as unicellular organisms, such as ciliates, bacteria, etc. But in nature the cells are organized in tissues, organs, organisms and its communities [5, 6].

3.1 P Systems with Active Membranes

It represents a distributed and parallel computing model. Basic data structures are multi-sets, strings or numerical variables. There are two main categories of P system: *hierarchical* and *tissue P systems* [7, 8]. Parallel molecular computing models are mainly based on processing of cell-like membrane structures in multisets of objects. Various variants have been applied for creating, dividing, merging, or dissolving membranes, duplicating the membranes and these variants were already shown to be computationally universal, equal in power to *Turing machines*. Rules are able to perform the operation for modifying the membrane structure: membrane creation:$[_i\ a\ ]_i \rightarrow [_j\ b\ ]_j$,membrane division: $[_i\ a\ ]_i \rightarrow [_k\ b\ ]_k [_j\ c\ ]_j$, membrane dissolution: $[_i\ a\ ]_i \rightarrow a$, membrane duplication: $[_i\ a\ ]_i \rightarrow [_k\ b\ [_j\ c\ ]_j\ ]_k$ where a, b are objects and i, j, k are labels of possible membranes.

3.2 Transition P System

In the P systems, each region contains a *multiset* of symbol-objects, which correspond to the chemicals swimming in a solution in a cell compartment. A system is based on multiset-rewriting rules is called as transition P systems [8]. There are three main types of rules:

(1) Multiset rewriting rules (one uses to call them, simply, evolution rules).

(2) Communication rules.

(3) Rules for handling membranes.

3.3 Multiset-based Tree Model for Membrane Computing

A multiset is a collection of objects in which objects are allowed to repeat finitely in most of the application areas. The use of square brackets is to represent a multiset ,which is quasi-general. Thus, a multiset containing one occurrence of *a*, two occurrences of *b*, and three occurrences of *c* is notationally written as [a,b,b,c,c,c] or it can be written as $[a, b, c]_{1,2,3}$.

It can also be represented in the following ways:[[*a*, *b*, *b*, *c*, *c*, *c*]] ,[*a*, *b*, *b*, *c*, *c*, *c*],[*a; b; c*]1;2;3,[*a*, 2*b*, 3*c*] ,[*a*.1, *b*.2, *c*.3] ,[1/*a*, 2/*b*, 3/*c*] ,[*a*1, *b*2, *c*3,[*a*1*b*2*c*3].

A multiset (α) is a mapping from generic set of objects to some set of numbers [5, 6].

$\alpha = [x, y, z]_{1, 2, 3}$ (a mapping from a ground set *D* to N)

There are two main properties:

(i) The multiset of rules is applicable to the multiset of objects available in the respective region,

(ii) The multiset is maximal; no further rule can be added to it.

$\Pi = (V, \mu, w_1, \ldots, w_n, (R_1, \rho_1), \ldots, (R_n, \rho_n), i_0), n \geq 1$

- V - alphabet of objects.
- μ - set of catalysts.
- M - is a membrane structure labeled with 1, 2, . . . ,m.
- $w_1, w_2, \ldots, w_m$ are strings over V representing the multisets of objects present in the regions 1, 2, . . . ,m of the membrane structure.
- $R_1, R_2, \ldots,$ Rn are finite sets of evolution rules associated with the regions 1, 2, . . . ,m of the membrane structure.
- i_0 - is either one of the labels 1, 2, …, m, and then the respective region is the output region of the system, or it is 0, and then the result of a computation is collected in the environment of the system.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The rules are of the form u → v or u →vδ.δ is special symbol not in V , length of u is called as radius of the rule u → v. [Computing with membranes-Gheorghe paun]

4. MATRIX INSERTION-DELETION SYSTEM

Insertion and deletion are the common operations in biology. It can be used in DNA processing and RNA editing. Based on the transformations, an insertion-deletion system has been formulated [9,11]. In general, the insertion and deletion operations can be defined as follows, the operations of insertion and deletion were initially considered with a linguistic approach. The insertion procedures, along with its iterated variants are generalized versions of Kleene's operations of concatenation and closure, while the deletion method simplifies the quotient process [10, 12]. A study of properties of the insertion deletion operations also has more interesting biological motivations. It corresponds to a mismatched annealing operation [14, 15] in DNA sequences; these kinds of operations are also in the evolution processes in the type of position as well as in RNA editing. This biological intention of insertion-deletion operations are in the framework of molecular computing. In general case, an insertion operation is defined as "adding a substring to a specified string" in a particular (both left and right) context, and on the other side deletion operation is defined as "removing a substring of a specified string" from a particular (both left and right) context.

4.1 DNA and RNA molecules

DNA molecules contains the strings consisting of four symbols namely 1) *A-adenine, 2) C-cytosine, 3) G-guanine and 4) T-thymine.* In the same way RNA molecules contains the strings consisting of symbols namely A, *U (uracil)*, G and C. Since the bio-molecular structures can be defined in terms of sequence of symbols (i.e., strings) there exists a correlation between formal grammars and bio molecular structures. The following example shares a common point between formal grammar and molecular strings. Let us take the gene sequence S = CTATCGCGATAG. As A = T, T = A, G = C and C = G, the above gene sequence resembles the context-free language {w w^R | w ∈ {a, b}*} [12, 15].

4.2 Definition

The insertion-deletion system is given as γ = (V, T, A, R) [13]
Where,

- V is an alphabet, T $\subseteq$ V ,
- A is a finite language over V called as axioms,
- R is a finite set insertion deletion rule.

An insertion rule is in the form of $(u, \lambda/\alpha, v)$ where u, v ∈ V* and $\alpha \in V^+$ which offers the rewriting rule $uv \rightarrow u\alpha v$. A deletion rule is of the form: $(u, \beta/\lambda, v)$ where u, v ∈V* and β∈ V$^+$ which offers the rewriting rule $u\beta v \rightarrow uv$.

4.3 Representing Bio-Molecular Structures

Based on the complementary pairs that occur in biomolecules, the strings form some structures. If such structures are formed within gene sequence then it is called as intramolecular structures. The intramolecular structures that are commonly noticed in DNA, are *stem and loop, hairpin, pseudoknot, cloverleaf, dumbbell* and *attenuator*. If the structures are formed across the gene sequences then it is called intermolecular structures such as *nick language, holliday, double stranded language, replication fork*. The above mentioned intramolecular and intermolecular structures are modelled by matrix insertion-deletion systems [13].

5. WATSON-CRICK AUTOMATA

Watson-Crick automata are running on double side tapes in finite state automata to explore the computation in DNA molecules. Finite languages can be accepted with two states in Watson-Crick system. Unary regular language can be accepted by a non-deterministic Watson-Crick automaton with three states [16]. It can be measured as a special case in Watson-Crick automata where the two reading heads are necessary to move jointly. Nano machines can be developed using nano-engineering field and it is going to parse the DNA molecules. In the same way, artificial and smart drug designs were developed. Theoretical notions for these nano machines can be observed by Watson-Crick automata and it is based on the design of finite automata operating on molecules of DNA. These types of machines have two independent read-heads in finite automata and it is working on two strands sequences. It can be read

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

from left to right by "read only heads" and it is controlled by common state. Two strands of the input are related by a complementarity relation, which is same as DNA nucleotides in Watson-Crick systems [18].

It can read more than one letter in blocks and in these systems, among the number of states (two or three) one state can be defined already in indefinite distinct automata. Watson-Crick automata, randomly read long finite series of characters at a time. A weaker-operational description of determinism, known as "weak determinism", which is undecidable. Non-deterministic Watson-Crick automaton is a weakly deterministic [16, 18] and it has the complementarity relation exist in identity. Thus, the classification of the notion of 'strong determinism', set into both the deterministic element and the information that the complementarity relation exist in identity [8][9]. The concept of parallel communicating Watson-Crick automata systems consists of several Watson-Crick finite automata which parses independently the similar input and replace the information on request by communicating states to each other [18].

5.1 Definition

Let V be a finite set of alphabet V^* be a finite set of words in V, λ be a empty word. A word $u \in V^*$ is prefix of V and $|u|$ be a length of u. We can say $u \in V^*$ is a prefix of a word $\rho \subseteq V \times V$ be a Watson-Crick complementarity relation (symmetric relation) on V. This symmetric relation is biologically inspired in the double-stranded DNA by the Watson-Crick automata complementarity of nucleotides molecule. According to the DNA molecule representation, DNA molecules are viewed as two strings written in one on bottom of the other.

$$\begin{bmatrix} v \\ v \end{bmatrix} \rho = \{ \begin{bmatrix} a \\ b \end{bmatrix} \mid a, b \in V, (a, b) \in \rho \},$$

$$WC\,\rho(V) = \begin{bmatrix} v \\ v \end{bmatrix}^* \rho$$

Watson-Crick automata domain associated to ρ and V. A non-deterministic Watson-Crick finite automaton contains 6-tuple, $M = (V, \rho, Q, q_0, F, P)$, where V is a input alphabet, $\rho \subseteq V \times V$ is said to be a complementary relation, Q is a finite set of states, $q_0 \in Q$ (set of initial states), $F \subseteq Q$(set of final states), P is the set of transition rules of the form $q\begin{pmatrix} w1 \\ w2 \end{pmatrix} \rightarrow q$ stated that if the Watson-Crick automaton is in a state q and parses $w_1 \in V^*$ in the position of upper strand and $w_2 \in V^*$ in the position of lower strand, then it can go into the state q_0 [17,18].

6. SPLICING SYSTEM

Splicing systems were initiated as a formal language model of the recombinant performance of DNA sequences [21]. Consider an alphabet V, two special symbols, #; $ $\notin$ V. V* is the set of all strings over V, including the empty string, denoted as λ. |x| we denote the length of x $\in$V*. A splicing rule over V is a string $s_1\#s_2\$s_3\#s_4$, where $s_1, s_2, s_3, s_4 \in V^*$. The maximum of $|u_i|$, $1 \leq i \leq 4$, is called the radius of the splicing rule. For a splicing rule = s1#s2$s3#s4 and four strings x, y, w $\in$V* we write,

$(x; y) \vdash r\ (w,z)$ iff $x = x_1 s_1 s_2 x_2$, $y = y_1 s_3 s_4 y_2$,
$w = x_1 s_1 s_4 y_2$, $z = y_1 s_3 s_2 x_2$, for some $x_1, x_2, y_1, y_2 \in V^*$.

Splice the strings x,y at the sites s_1,s_2,s_3,s_4 respectively. A pair σ=(V,R), where V is an alphabet,R is the set of splicing rules over V. Splicing scheme σ=(V,R) and its language $L \subseteq V^*$ we define, $\sigma(L)$={w $\in$V* | (x,y) $\vdash_r$ (w,z), for some x; y $\in$ L,r $\in$ R}[Gheorghe Paun,2000].
Then an extended H system is defined as γ = (V, T, A, R)
Where,

- V is an alphabet,
- $A \subseteq V^*$,
- $R \subseteq V^*\#V^*\$V^*\#V^*$. (T is the terminal alphabet, A is the set of axioms, and R is the set of splicing rules).

When T =V, the system is called as non-extended system.
The pair σ=(V,R) is the underlying H scheme of γ. The language accepted by γ is defined by $L(\gamma) = \sigma^*(A) \cup T^*$ [19,20].

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

6.1 Types of splicing system

Gheorghe.Paun initiated flat splicing systems which are novel systems. Flat splicing systems are of importance for verifying language-theoretic results. It is used to separate the procedures on formal languages, grammars from the operation of circular closure (circularization). It emerges that proofs for linear words are at times very simple because they involved directly on standard conditions on formal languages [20, 21, 22].

The splicing operations are considered in both simple (regular) and iterated splicing [18]. A simple splicing scheme is a pair $\sigma = (A, R)$, where A is an alphabet is a regular language classified in A*#A*$A*#A* for #,$ are the two special symbols($\# \notin V, \$ \notin V$).

7. STICKER SYSTEM

Sticker system, a computability model started by Karietal.L-1998 is a language producing device based on the sticker operation. Sticker system is one of the computational model in which the Watson-Crick complementarity is used in DNA computing. This system which has a general form, along with the blocks of arbitrary shapes has been annealed to the currently built sequences [26, 27]. The generative power of several variants of sticker systems are compared with families of Chomsky hierarchy [29].Characterizations of regular, linear, and recursively enumerable languages are obtained in this framework.

7.1 The sticking operation

DNA sequences are double stranded construction composed of four nucleotides [28] A, C, G, T and pairing of A-T; C-G is called Watson-Crick complementarity. The sticker system is working on more complex structures (higher order structure) of DNA molecules. The advantage behind this complex sticker system is that the characterization of recursively enumerable language [27].

8. CONCLUSION

In this paper we presented an importance of biological computing along with its various models. The important aspects of membrane computing, Matrix insertion deletion system, Watson-Crick automata, splicing system and sticker system were discussed. Membrane computing gives computational models that abstracts from the living cells structure and functioning. Such models are proved to be computationally powerful and efficient to solve NP-complete problems. Watson crick automata recognize several regular languages in an efficient manner. An insertion-deletion system has been extended by adding some additional controls. We have discussed here the adaptation of the idea of matrix grammars for insertion-deletion systems. Splicing system is used to illustrate the generative power of several recombination procedures, which is one of the developing areas in molecular computing and also a part of formal language theory. Finally, we have discussed the theoretical study on sticker operation, which is generally identified as dry molecular computing.

REFERENCES:

[1] Biologically inspired computing, Josh Bongard, University of Vermont. (vol. 42 no. 4) pp. 95-98,(2009).

[2] Biologically inspired computing,Nancy, Forbes Computing in Science & Engineering (Volume:2 , Issue: 6)

[3] Lakshmanan, K., M. Anand and Eric Villemonte De la Clergerie, Modelling intermolecular structures and defining ambiguity in gene sequences using matrix insertion-deletion systems, (2011).

[4] Leonard M.Adleman, Molecular computation of solutions combinatorial problems, ,Science,new series,volume 266,Issue5187,1021-1024.

[5] Gheorghe Paun, Membrane Computing: An Introduction ,ACM computing classification,ISBN 3-540-43601-4 Springer verlog berlin Heidelberg ,New York (1998).

[6] Membrane Computing:New Results, New Problems Francesco Bernardini VT Meeting,(2003).

[7] Gheorghe Paun, Mario J. Perez-Jimenezb, Membrane computing: Brief introduction, recent results and applications. BioSystems 11–22, (2006).

[8] Gheorghe Paun, Computing with membranes, ,Institute of mathematics of the Romanian academy.November (1998)

[9] Kari L,On insertion and deletion in formal languages,*Ph.D Thesis*, University of Turku.

[10] Kari L. and G.Thierrin (1996). Contextual insertions/deletions and computability, *Information and Computation*, Vol. 131/1, pp. 47-61.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[11] Haussler, D. Insertion and iterated insertion as operations on formal languages, *Ph.D. Thesis*, University of Colorado, (1982)
[12] Haussler, D. Insertion Languages, *Information Sciences*, Vol. 131(1), pp. 77—89, (1983). Boulder.
[13] Lakshmanan Kuppusamy, Anand Mahendran and Shankara NarayananKrishna, Matrix Insertion-Deletion Systems for Bio-Molecular Structures,Pages301-312 Springer-Verlag Berlin,Heidelberg,(2011).
[14] Ion Petre,Sergey Verlan, Matrix insertion–deletion systems, Formal Languages and Automata Theory,Dec(2010).
[15] Shravan Kumar, Moniphal Say, Ranjith Kumar, Insertion Deletion Systems, IIT Bombay,April (2006).
[16] Elena Czeizler, Eugen Czeizler, Parallel Communicating Watson-Crick automata Systems, , Theoretical Computer Science Volume 358, Issue 1, 31 Pages 142–147, (2006),
[17] Gh. Paun, G. Rozenberg, A. Salomaa ,*DNA Computing. New Computing Paradigms* ,Springer-Verlag, Berlin, (1998).
[18] Elena Czeizler, Eugen Czeizler, Lila Kari,Kai Salomaa ,On the descriptional complexity of Watson_Crick automata Elena. Theoretical Computer Science 410, 3250_3260, (2009).
[19] Arto salomaa,The formal languages theory column,Turku Centre for Computer Science and, Department of Mathematics, University of Turku, Finland.
[20] Gheorghe Paun, On the splicing operation, Discrete Applied Mathematics,(1995).
[21] Gheorghe Paun, Grzegorz Rozenberg ,Arto Salomaa, Computing by splicing, Theoretical Computer Science, 321-336, (1996).
[22] Gheorghe Paun, DNA computing based on splicing: universality results, Institute of Mathematics of the Romanian Academy P.O. Box 1-764, Bucuresti, Romania,(1996)
[23] Rosalba Zizza Rosalba Zizza, Splicing systems, Dipartimento di Informatica, University of Salerno, Italy,(2010).
[24] Jean Berstela, Luc Boassonb, Splicing systems and the Chomsky hierarchy, Isabelle Fagnota.c ,(2011)
[25] Rosalba Zizza,Splicing systems, Dipartimento di Informatica, University of Salerno, Italy Rosalba Zizza, Scholarpedia, 5(7):9397, (2010).
[26] Gheorghe Paun, Sticker systems , Grzegorz Rozenberg ,(1998).
[27] Masilamani, V. ; Venkatesan, A.S.P. ; Thomas,P Systems Based on Sticker Operations, D.G. Bio-Inspired Computing: Theories and Applications (BIC-TA), 2011 Sixth International Conference: 27-29,(2011).
[28] Artiom Alhazov, Matteo Cavaliere, Computing by Observing Bio-systems: The Case of Sticker Systems, DNA Computing Lecture Notes in Computer Science Volume 3384, 2005, pp 1-13,(2005).
[29] L. Kari, Gh. Paun, G. Rozenberg ,DNA computing, sticker systems, and universality, Volume 35, Issue 5, pp 401-42,(1998).