Malaysian Journal of Fundamental & Applied Sciences

available online at http://mjfas.ibnusina.utm.my

Sticker Systems Over Monoids

N.A. Mohd Sebry[1]*,N. Z. A. Hamzah [2], N.H. Sarmin[1,2], W.H. Fong [2] and S. Turaev[3]

[1]*Department of Mathematical Sciences, Faculty of Science, UTM, 81310 UTM Johor Bahru,Johor, Malaysia,*
[2]*IbnuSina Institute for Fundamental Science Studies, Faculty of Science, UTM, 81310 UTM Johor Bahru,Johor, Malaysia,*
[3]*Faculty of Computer Science and Informational Technology, Universiti Putra Malaysia, Malaysia.*

Received 26 November 2012, Revised 5 January 2012, Accepted 12 April 2012, Available online 28 April 2012

ABSTRACT

Molecular computing has gained many interests among researchers since Head introduced the first theoretical model for DNA based computation using the splicing operation in 1987. Another model for DNA computing was proposed by using the sticker operation which Adlemanused in his successful experiment for the computation of Hamiltonian paths in a graph: a double stranded DNA sequence is composed by prolonging to the left and to the right a sequence of (single or double) symbols by using given single stranded strings or even more complex dominoes with sticky ends, gluing these ends together with the sticky ends of the current sequence according to a complementarity relation. According to this sticker operation, a language generative mechanism, called a sticker system, can be defined: a set of (incomplete) double-stranded sequences (axioms) and a set of pairs of single or double-stranded complementary sequences are given. The initial sequences are prolonged to the left and to the right by using sequences from the latter set, respectively. The iterations of these prolongations produce "computations" of possibly arbitrary length. These processes stop when a complete double stranded sequence is obtained. Sticker systems will generate only regular languages without restrictions. Additional restrictions can be imposed on the matching pairs of strands to obtain more powerful languages. Several types of sticker systems are shown to have the same power as regular grammars; one type is found to represent all linear languages whereas another one is proved to be able to represent any recursively enumerable language. The main aim of this research is to introduce and study sticker systems over monoids in which with each sticker operation, an element of a monoid is associated and a complete double stranded sequence is considered to be valid if the computation of the associated elements of the monoid produces the neutral element. Moreover, the sticker system over monoids is defined in this study.

| Sticker operation| Valence grammar | Valence sticker system | H system |

1. INTRODUCTION

The first DNA computation experiment was successfully performed by Adleman in 1994 [1] and he has proved that DNA computing is possible. The sticker system is introduced in [2] as a formal language model for the self-assembly (annealing and ligation operations)phase of Adleman's experiment [1]. The basic operation of sticker systems isthe sticking operation [2, 3, 4, 5] that forms double stranded DNA sequences fromblocks of arbitrary shape of DNA single strands. The sticker system model uses no enzymes, employs reusable DNA and has a random access memory that requires no strandextension [5, 6]. The interest of this research is to extend the DNA sticker systems over some monoids. Some new concepts of DNA sticker systems over monoids are introduced in this paper.

Corresponding author at:
E-mail address: fide_prestige1206@yahoo.com (Nurul Afidah Mohd Sebry)

2. PRELIMINARIES

2.1 Sticker Operation

A DNA sequence is a double stranded sequence which is composed of adenine (*A*), thymine (*T*), cytosine (*C*) and guanine (*G*). The bonding between the bases follows the restriction of the base pairing proposed by Watson–Crick in 1953 [7] where *A* is paired with *T* and *C* is paired with *G*. The basic formalism of sticker operation is that, two single strands of DNA glued together to form double stranded DNA sequence. This gluing operation will continue as long as more single stranded DNA sequence is added in the solution.

Consider the alphabet V (a finite set of symbols) and a symmetric biological relation ρ, where $\rho \subseteq V \times V$. Let $\#$ be a special symbol not in V, denoting an empty space (the blank symbol). The symbol V^* denotes the setof all strings of symbol over V. It is also considered that the alphabet $V$is ofabstract DNA bases $\{a,\ t,\ c,\ g\}$ and a relation$\rho \subseteq V \times V$ over$V$represent thecomplementary relation(a complements to t and c complements to g). The matching base pairs of double stranded molecules of DNA can be represented as$\binom{x_1}{x_2}$where x_1 is the upper strand and

x_2 is the lower strand. Also, x_1 is complement to x_2. If $x = \binom{x_1}{x_2}$ and $y = \binom{y_1}{y_2}$, then the concatenation of x and y is denoted by $xy = \binom{x_1}{x_2}\binom{y_1}{y_2}$ or $xy = \binom{x_1 y_1}{x_2 y_2}$. Using the elements of $V \cup \#$, the following set of composite symbols is constructed:

$$\binom{V}{V} = \left\{\binom{a}{b} \mid a, b \in V, (a,b) \in \rho\right\},$$
$$\binom{\#}{V} = \left\{\binom{\#}{b} \mid b \in V\right\}, \binom{V}{\#} = \left\{\binom{a}{\#} \mid a \in V\right\}.$$

The Watson-Crick domain is denoted by $WK\rho(V)$, which is associated to the alphabet $V$and the complementary relation ρ. For example, if we have two strings of $w_1 = x_1 x_2 \cdots x_n$and$w_2 = y_1 y_2 \cdots y_n$, then the concatenation of the strings$w_1$and$w_2$is equal to $w_1 w_2 = \binom{x_1}{y_1}\binom{x_2}{y_2}\cdots\binom{x_n}{y_n} \in WK\rho(V)$ and is simplified as$\binom{w_1}{w_2}$. The element$\binom{w_1}{w_2} \in WK\rho(V)$is called a well-formed double stranded sequence (or simply double stranded sequence) or molecules [4, 8]. Here, $w_1$is called the upper strand and w_2 is called the lower strand. Note that, the length of w_1 and w_2 are equal and they are symmetric over the relation ρ, which is $(x_i, y_i) \in \rho$ for $i = 1, 2, \cdots, n$. By means of symmetry, $(y_i, x_i) \in \rho$for$1, 2, \cdots, n$ is also defined.

The concept of incomplete molecules is used in this paper as an abstraction of DNA molecules with sticky ends [4, 6]. These molecules consist of mixed DNA molecules with double and single strands. The incomplete molecule is the element of the set:

$$W\rho(V) = L\rho(V) \cup R\rho\,(V) \cup LR\rho(V)$$

where;

$$L\rho(V) = \left(\binom{\lambda}{V^*} \cup \binom{V^*}{\lambda}\right)\binom{V}{V}_\rho^*,$$

$$R\rho(V) = \binom{V}{V}_\rho^*\left(\binom{\lambda}{V^*} \cup \binom{V^*}{\lambda}\right),$$

$$LR\rho(V) = \left(\binom{\lambda}{V^*} \cup \binom{V^*}{\lambda}\right)\binom{V}{V}_\rho^+\left(\binom{\lambda}{V^*} \cup \binom{V^*}{\lambda}\right).$$

From the given set of $W\rho(V)$, we can illustrate two basic types of possible shapes of elements in $W\rho(V)$as the following:

i) Left single strand and right double strand ($L\rho(V)$),
ii) Left double strand and right single strand ($R\rho(V)$).

The third type of possible shape ($LR\rho(V)$) can be obtained by extending the first two basic types of elements in $W\rho(V)$. Figure 1illustrates the possible shapes of the three types of incomplete molecules [4].

There are eight possible shapes of dominoes used in sticking operation which can lead to different cases when stacked to the other. Notice that, in all cases we have a double stranded sequence of 'x' and an overhang sequence of 'y' or 'y and z'. The sticky ends are placed either in the lower strand or in the upper strand. However, for the third type which is $LR\rho(V)$, its element which is x is in $\binom{V}{V}_\rho^+$and is called well-started double stranded sequence.

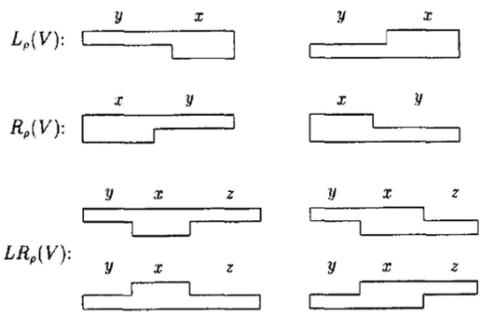

Fig. 1 Possible Shapes of Dominoes

The two strands of dominoes in $W\rho(V)$might have different length and it is useful to indicate the blank space by the symbol #. A partial operation μ stimulates the concatenation and the ligation or annealing operation among the elements of $W\rho(V)$. The operation is known as sticking operation which is the primary implements of the sticker systems [6]. The elements of $W\rho(V)$can be prolonged to the left or to the right with a brick or dominoes providing the complementarity between the corresponding sticky ends of dominoes. The process of prolongation will continue until complete double stranded molecule is obtained.

It is easy to see that there are eight possible cases of sticker operation obtained from the elements of $W\rho(V)$which is illustrated in Figure 2.

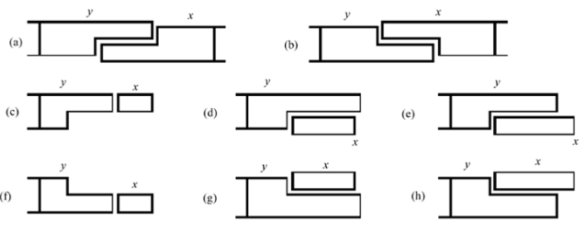

Fig. 2 Eight Possible Cases of Sticker Operation

The sticker operation discussed is a language operation of formal languages that stimulates concatenation operation and annealing operation of the incomplete molecules [6] of $W\rho(V)$to the complete double stranded molecule. The sticker operation is the primary implement of the sticker system as discussed in the next section.

2.2 Sticker System

The experiment done by Adleman [1] in 1994 for solving Hamiltonian Path Problem (HPP) initiated the study of sticker system by Kari et al. [2].

Sticker system is a language generating device which was first introduced by Kari et al. [2] in 1998 as a formal language model for the self-assembly phase of Adleman's experiment. The term regular sticker system was first used in [2] and was then extended to bidirectional sticker systems in [5]. However, both concepts were merged to sticker system as in [4] and the term ‘sticker system’ will be used in this paper.

When forming new molecules, the initial strands called axioms and a well started sequence are utilized and prolonged either to the left or to the right direction by the process of the operation μ [8]. Starting from the axiom and iteratively using the operation of sticking, strands are prolonged by using dominoes or bricks in order to obtain a complete double stranded sequence [9]. In [4], Paun et al. has defined sticker system in a very general form and the formal definition of sticker system in this paper will follow the definition in [4] and will also make use of the definition given by Kari et al. in [2] and [5].

A sticker system is a construct of 4-tuple

$$\gamma = (V, \rho, A, D)$$

where$V$is an alphabet, endowed with the symmetric relation $\rho \subseteq V \times V$ in V, A is a finite subset of axioms ($W\rho(V)$), and $D$is a finite set of pairs(B_d, B_u) where $B_d$and $B_u$are finite subsets of upper and lower stickers of the forms$\binom{\#}{V}^+$and$\binom{V}{\#}^+$, respectively.

To implement the systems, the mechanism of the sticker system starts from the axiom $A$and uses the pairs of $B_d$and $B_u$to prolong the strands to the left or to the right according to sticker operation μ in order to obtain a set of double stranded sequence in $WK\rho(V)$ (a complete molecule). The prolongation process will continue until there is no blank symbol present where a complete double stranded sequence is obtained [2, 3].

For a given sticker system$\gamma = (V, \rho, A, B_d, B_u)$and two sequences $x, y \in W\rho(V)$, we write,

$$x \Longrightarrow^* y \text{ if and only if } y = \mu\big(B_d, \mu(x, B_u)\big).$$

Note that,$\mu\big(B_d, \mu(x, B_u)\big) = \mu(\mu(B_d, x), B_u)$because the prolongation to the left is independent with the prolongation to the right [2]. By$\Longrightarrow^*$, we denote the reflexive and transitive closure of the relation $\Longrightarrow$.

A sequence $x_1 \Longrightarrow x_2 \Longrightarrow \cdots \Longrightarrow x_k$, where $x_1 \in A$, is called a computation inγ with length $k-1$ (notice that, we start from $x_1 \in A$). If $x_k \in WK\rho(V)$, where no sticky end and hence blank symbol is present in the last string of composite symbol, the above computation is considered as complete.

A complete computation of sticker system will produce a complete string, denoted by $w \in WK\rho(V)$. Sticker system will generate the set of all possible languages of the completed strings. The language of all such strings in the sticker operation is the language generated by $\gamma$and is called as sticker language. Sticker language is explained in the next section.

2.3 Sticker Language

The sticker language is a language generating device based on the sticker operation which is an abstract model of the annealing operation occurring in DNA computing [2, 5]. The language generated by a sticker system consist of all strings formed by the set of upper strands of all complete molecule derived from the axioms for which an exactly matching sequence of lower stickers can be found [5, 10].

Sticker system will generate only regular languages (but not all of them) without restrictions [9]. Some restrictions have been introduced in [2, 4, 8, 11] on matching pairs of strands to obtained more powerful languages of sticker systems such as primitive sticker language, balance sticker language, and primitive balance sticker language.

The sticker language is the set of all possible languages (from complete molecules) generated by sticker system. Let $\gamma = (V, \rho, A, D)$be a sticker system, the unrestricted molecular language generated by $\gamma$is defined in [11] as follows:

$$ML(\gamma) = \{(x, y) \in \rho \mid (x, y) \text{is a computation of } \gamma\}.$$

Furthermore, the unrestricted sticker language generated by $\gamma$is the projection onto the first (upper) component of the molecular language, because of the complementarity relation of ρ. The language is denoted as $L(\gamma)$and is defined by:

$$L(\gamma) = \{w \in WK\rho(V) \mid x \Longrightarrow^* w, x \in A\}.$$

In this research,*SL* will denote sticker language. Only complete computation (the set of all upper strands of all complete molecules) as defined above will be considered when defining *SL*. The similarity of this process to Adleman’s experiment can be found in [2].

There are six families of languages generated by sticker systems which are sticker languages, primitive sticker languages, balanced sticker languages, primitive balanced sticker languages, coherent sticker languages, and fair sticker languages denoted by *SL*, *PSL*, *BSL*, *PBSL*, *CSL* and *FSL* respectively. The definitions of these families of languages are well defined in [2].

It is a fact that the families of languages generated by sticker systems are strictly included in the family of regular languages,$SL \subsetneq REG$, $PSL \subsetneq REG$, $BSL \subsetneq REG$, $PBSL \subsetneq REG$, $CSL \subsetneq REG$, and $FSL \subsetneq REG$. The proof ofthese lemmas can be found in [2].

Below is an example to illustrate how sticking operation works on a sticker system.

Example 1 Given a sticker system$\gamma = (V, \rho, A, D)$with $V = \{a, b\}, \rho = \{(a,a), (b,b)\}$, $A = \left\{\binom{b}{b}\binom{a}{\#}\right\}$, and $D = \left\{\binom{a}{\#}\binom{a}{\#}, \binom{a}{\#}\binom{b}{\#}, \binom{\#}{a}\binom{\#}{a}, \binom{\#}{b}\right\}$.

Startingwith $A = \left\{\binom{b}{b}\binom{a}{\#}\right\}$, then the strings in D (of complementarity) are used repeatedly to obtain another new molecule. The computations are depicted in the following:

1. $\binom{b}{b}\binom{a}{\#} \longrightarrow \binom{\#}{a}\binom{\#}{a} \Longrightarrow \binom{b}{b}\binom{a}{a}\binom{\#}{a}$,
2. $\binom{b}{b}\binom{a}{a}\binom{\#}{a} \longrightarrow \binom{a}{\#}\binom{b}{\#} \Longrightarrow \binom{b}{b}\binom{a}{a}^2\binom{b}{\#}$,
3. $\binom{b}{b}\binom{a}{a}^2\binom{b}{\#} \longrightarrow \binom{\#}{b} \Longrightarrow \binom{b}{b}\binom{a}{a}^2\binom{b}{b}$.

From the solution, the language of the sticker system is defined asfollows:

$$ML(\gamma) = \left\{\binom{b}{b}\binom{a}{a}^2\binom{b}{b}\right\} \text{and } L(\gamma) = \{ba^2b\}.$$

The second notation of sticker language which is $L(\gamma)$is used throughout the paper. The following section recalls the definition and properties of monoids.

2.4 Monoids

In abstract algebra, an algebraic structure consists of one or more sets, called underlying sets or carriers or sorts, closed under one or more operations, satisfying some axioms where monoid is grouped in a group-like structure.

Definition 1 [12]**:**Monoid

An algebraic structure, M is said to be a monoid if it satisfies the following properties:

i) *Closure*: M is closed with respect to operation$*$, that is$a, b \in M \Longrightarrow (a, b) \in M$ for every $a, b \in M$.

ii)*Associativity*: The binary operation $*$ is associative in M, that is$a * (b * c) = (a * b) * c$for every $a, b, c \in M$.

iii) *Identity*: There exist an element $e \in M$, such that$a * e = e * a = a$ for every $a \in M$.

The sticker system is first associated with the properties of monoid, M to see if the operation of sticking preserves the operation of monoid. It is obvious that, the sticker system γ satisfies the conditions as a monoid.

2.5 Grammar

To study languages mathematically, a mechanism is needed to describe them. In this section, the notion of grammar is introduced as it is considered a common and powerful tool [13] to describe languages produced by the complete computation of sticker systems over monoids. The basic grammars investigated in formal language theory which is Chomsky grammar is discussed and hence it exhibit the relationship between numbers of language families by the Chomsky hierarchy.

The definition of grammar and Chomsky grammar is listed in the following.

Definition 2 [13]:Grammar

A grammar G is defined as a quadruple $G = (V, T, S, P)$ where

V is a finite set of objects called variables,

$T$is a finite set of objects called terminal symbols,

$S \in V$ isa special symbol called the start variable,

and$P$ is a finite set of production rules.

The sets$V$ and T are nonempty and disjoint.

Definition 3[14] Chomsky Grammar

A Chomsky grammar is a quadruple $G = (N, T, S, P)$ where

N is a finite set of nonterminal symbols,

T is a finite set of terminal symbols,

$S \in V$is a special symbol called the start symbol,

and$P$ is a finite set of production rules.

The sets$N$ and T are nonempty and disjoint. The set P is a finite set of pairs (u, v) where $u, v \in (N \cup T)^*$, such that u contains at least one symbol from N. In this paper, the production rules are written in the form of$u \longrightarrow v$where u is an element of$(N \cup T)^+$, and v is in$(N \cup T)^*$.

The set of languages generated by Chomsky grammar, G, is denoted by$L(G)$and is defined as follows:

$$L(G) = \{w \in T^*; S \Longrightarrow^* w\}.$$

A number of language families are encountered which are recursively enumerable language, context-sensitive language, context-free language and regular languages. The languages are generated by regular grammars, context-free grammars, context-sensitive grammars and unrestricted grammars respectively and the grammarsmake up the Chomsky hierarchy, named after Noam Chomsky, a founder of formal language theory [13]. The families of finite, regular, linear, context-free, context-sensitive and recursively enumerable languages are denoted by *FIN*, *REG*, *LIN*, *CF*, *CS*, and *RE* respectively. They are said to form the Chomsky hierarchy because of the following inclusion: $FIN \subseteq REG \subseteq LIN \subseteq CF \subseteq CS \subseteq RE$.

Among the language families which are of interest are the context sensitive language (*CS*), context-free language (*CF*), and regular language (*REG*). The characterization of the regular language concerning the generative capacity of sticker system will be investigated.

2.6 Valence Grammar and Extended Valence H System

Grammar with regulated rewriting was first introduced by Paun [14], and is termed as valence grammar. In a valence grammar, each production of a Chomsky grammar is associated with an integer value, and computes the value of a derivation by adding the valences, and the valence of the production is only considered if the valences of the productions used add to zero [15].

Definition 4 [14]: Valence Grammar

A valence grammar over monoid is a quintuple

$G = (N, T, S, P, M)$ where N, T, S are specified as in context-free grammar. $M = (M, \diamond)$ is a monoid with neutral element, $e$and binary operation $\diamond$. The set P is a finite set of pairs$r = (P, m)$ with a context-free rule $P$and $m \in M$.

The language $L(G)$generated by G is called valence language and is defined as

$$L(G) = \{w |\, w \in T^*, (s, e) \Longrightarrow^* (w, e)\}.$$

The operation of sticker system is designed by a language generating mechanism so that the valence of the productions used added to zero or one according to additive or multiplicative valence grammar respectively.

Sticker languages produced by the operation of sticking are the possible sets of languages generated by sticker system. In this study, valence sticker language, additive valence sticker language, and multiplicative valence sticker language are introduced with respect to the respective valence grammars.

The set of sticker system with monoid operation is used where for each sticker operation, an element of a monoid is associated and the value of monoid is computed from two used strings to produce a new string.

An extended H system over groups$(\mathbb{Z},+,0)$ and$(\mathbb{Q}_+,\bullet,1)$ have been introduced in [16] and is known as an extended valence H systems. The definition of extended valence H system has been defined in [16] which discusses the computational power of H systems with valences. In the system, a number is associated with initial strings and the numbers are computed with new produced string. Below is the definition of extended valence H system.

Definition 5 [16]: Extended Valence H Systems
Let$(M,\diamond,e)$be a group with operation $\diamond$ and the identity e. An extended valence H system over M is a construct $\gamma=(V,T,A,R)$, where V, T, and R are as in usual extended H system and A is a finite subset of$V^*\times M$.

For $(x,v_1),(y,v_2),(w,v_3)\in V^*\times M$ and $r\in R$, we write $[(x,v_1),(y,v_2)]\vdash_r(w,v_3)\in V^*\times M$ if and only if $(x,y)\vdash_r w$ and $v_3=v_1\diamond v_2$. Then the language generated by the system is defined as

$$L(\gamma)=\{x\in T^*|\,(x,e)\in\sigma^*(A)\}.$$

Using the definition above, Manca and Paun [16] have defined additive H systems and multiplicative H systems to denote the operation with a group of integers and a positive rational numbers respectively.

The valence grammar introduced by Dassow and Paun in 1989 [14] will be used to introduce the concept of sticker system over monoids. The definition of extended valence H system termed by Manca and Paun in [16] is used to define valence sticker system over monoids.

3. VALENCE STICKER SYSTEM

A new definition of sticker system over monoids which is termed as valence sticker systemis introduced in this research and is given in the following:

Definition 6: Valence Sticker System
Let $(M,\diamond,e)$be a monoid with operation $\diamond$ and the identity e. A valence sticker system over a monoid, M is a construct of valence sticker system,$VSS=(V,\rho,A,D,M)$ where $V,\rho,A$are as in the usual definition of sticker system and D is a finite subset of$(B_d\times M,B_u\times M)$.

For$\left[\binom{a}{\#},v_1\right]\in B_u\times M$,$\left[\binom{\#}{b},v_1\right]\in B_d\times Mand\left[\binom{a}{b},v_1\diamond v_2\right]\in WK\rho(V)$,where $v_1,v_2\in M$ and we write $\mu\left\{\left[\binom{a}{\#},v_1\right],\left[\binom{\#}{b},v_1\right]\right\}=\left[\left[\binom{a}{b},v_1\diamond v_2\right]\right]$if and only if$(a,b)\in\rho$.

The set of possible languages generated by valence sticker system is called valence sticker language (VSL) and is defined in the following:

$$VSL(\gamma)=\{w|\,w\in WK\rho(V),(A,e)\Rightarrow^*(w,e)\}.$$

where$w\in WK\rho(V)$ is a complete double strand, produced by iterative operation of sticking on a starting axiom, A with elements in D. The value of the valence assigned to A must be equal to the identity of monoid, which is e and the computation of sticker systems with the associative elements of the monoids produces the identity element, e.

We can define multiplicative valence sticker system ($mVSS$) and additive valence sticker system ($aVSS$) when the binary operation $\diamond$ are multiplicative and additive, respectively, namely $M_1=(\mathbb{Q}_+,\bullet,1)$ and$M_2=(\mathbb{Z},+,0)$. The definitions are in the following:

$$mVSS=(V,\rho,A,D,M_1)\text{where } M_1=(\mathbb{Q}_+,\bullet,1),$$
$$aVSS=(V,\rho,A,D,M_2)\text{where} M_2=(\mathbb{Z},+,0).$$

The set V,ρ,A,D are as defined in the definition of valence sticker system. The set of possible languages generated by valence multiplicative sticker system and valence additive sticker system are termed as multiplicative valence sticker language ($mVSL(\gamma)$) and additive valence sticker language ($aVSL(\gamma)$) respectively and are defined as the following:

$$mVSL(\gamma)=\{w|w\in WK\rho(V),(A,1)\Rightarrow^*(w,1)\},$$
$$aVSL(\gamma)=\{w|w\in WK\rho(V),(A,0)\Rightarrow^*(w,0)\}.$$

The idea behind this computation is with each sticker operation, an element of a monoid is associated to each axiom and the valence of monoid operation of the new string from two used strings is computed. A complete double stranded sequence produced is considered to be valid if the computation of the associated elements of the group produces the identity element of the monoids.

In the following, an example of the computation of sticker system with additive and multiplicative monoidsis given.

Example 2 Given a valence sticker system $VSS=(V,\rho,A,D,M)$with $M=(\mathbb{Z},+,0)$, $\rho=\{(a,a),(b,b)\}$, $A=\left\{\binom{\lambda}{\lambda}\right\}$ and $D\in(B_d\times M,B_u\times M)$, where

$$B_u\times M=\left\{\left[\binom{a}{\#}\binom{a}{\#},-1\right],\left[\binom{b}{\#}\binom{b}{\#},1\right]\right\},\text{and}$$
$$B_u\times M=\left\{\left[\binom{\#}{a}\binom{\#}{a},-1\right],\left[\binom{\#}{b}\binom{\#}{b},1\right]\right\}.$$

We start with $A=\left\{\left[\binom{\lambda}{\lambda},0\right]\right\}$. Then, the string $\left[\binom{a}{\#}\binom{a}{\#},-1\right]$ is joined to the starting axiom A. The computations are as follows:

$$\mu\left\{\left[\binom{\lambda}{\lambda},0\right],\left[\binom{a}{\#}\binom{a}{\#},-1\right]\right\}=\left[\binom{a}{\#}\binom{a}{\#},-1\right],\text{and}$$
$$\mu\left\{\left[\binom{a}{\#}\binom{a}{\#},-1\right],\left[\binom{\#}{a}\binom{\#}{a},-1\right]\right\}=\left[\binom{a}{a}\binom{a}{a},-2\right],$$

Using the sticker operation ofk^{th}iterations with strings$\left[\binom{a}{\#}\binom{a}{\#},-1\right]$followed by $\left[\binom{\#}{a}\binom{\#}{a},-1\right]$ repeatedly, we will have the string of the form$\left[\binom{a}{a}^{2k},-2k\right]$.

The string produced is a complete double stranded molecule but the added valence is not equal to the identity element, which is zero. Hence the computation is continued by using strings$\left[\binom{b}{\#}\binom{b}{\#}, 1\right]$ and followed by $\left[\binom{\#}{b}\binom{\#}{b}, 1\right]$to produce the string of the form$\left[\binom{a}{a}^{2k}\binom{b}{b}^{2}, -2k+2\right]$.

Also, the sticker operation with k^{th}iterations using strings$\left[\binom{b}{\#}\binom{b}{\#}, 1\right]$ and followed by $\left[\binom{\#}{b}\binom{\#}{b}, 1\right]$repeatedly, is used to terminate the $-2k+2$ valence of the produced string of$\left[\binom{a}{a}^{2k}\binom{b}{b}^{2}, -2k+2\right]$. Finally, the sticker operation of the respective strings will produce$\left[\binom{a}{a}^{2k}\binom{b}{b}^{2k}, -2k+2k=0\right]$which is a complete double stranded molecule with valence equal to the identity of the monoid.

Therefore, it can easily be concluded that the language produced by the system is $aVSL(\gamma) = \{a^{2k}b^{2k}; k \geq 1\}$. Hence, a grammar G can be constructed as shown in the following:

$$G = (\{S\}, \{a, b\}, S, P) \text{with productions}$$
$$S \longrightarrow aaSbb$$
$$S \longrightarrow \lambda.$$

From the solution, the language produced is a $CF - REG$language.

The computation of sticker system over monoids must satisfy two conditions which are:

i) The string from the computation is only considered if a complete double stranded molecule is produced,
ii) The valence of the complete double stranded molecule is equal to the identity of the respective monoids.

It is important to know that both conditions must be satisfied when dealing with computation of sticker system over monoids. Furthermore, the choosing of the starting axioms, A and the finite subset of pairs of $B_d \cup B_u$give differences to the language produced from the computation of the system. Besides, if we defineρ of complementarity to other than identity, for example (a, b),a different language will be produced.

4. CONCLUSION

The new definition of sticker system over monoidsdefined as valence sticker system is introduced in this study. An example is given to illustrate the given definition. The language generated by the computation of valence sticker system with groups of monoid of integers and positive rational numbers are also introduced. The language of the respective valence sticker systems are obtained from the complete double stranded molecule produced from the computation.

Hence, a grammar is generated from the produced languageof the computation to classify the grammar as well as the language to the class of Chomsky hierarchy. The computational power of sticker system can be seen through another similar example but not up to recursively enumerable language.

ACKNOWLEDGEMENT

The authors would like to acknowledge Research Management Center (RMC), UTM for the partial financial funding through Vote No. 00H48. The first and the second authors would also like to thank Ministry of Higher Education (MOHE) for MyMaster scholarship programme under the Tenth Malaysia Plan (10MP).

REFERENCES

[1] L.M. Adleman, Molecular Computations of Solutions to Combinatorial Problems. Science, 266 (1994), 1021-1024.

[2] L. Kari, G. Paun, G. Rozenberg, A. Salomaa, and S. Yu, DNA Computing, Sticker Systems and Universality. ActaInformatica. 35 (1998), 401-420.

[3] L. Kari, S. Seki, and P. Sosik, DNA Computing: Foundations and Implications for Computer Science. In G. Rozenberg, T. Back, and J. Kok, Springer-Verlag, 2010.

[4] G. Paun, and G. Rozenberg, Sticker Systems, Theoretical Computer Science, 204 (1998), 183-203.

[5] R. Freund, G. Paun, G. Rozenberg, and A. Salomaa, A Bidirectional Sticker Systems. In R. B. Altman, A. K. Dunker, L. Hunter, and T. E. Klein (Eds.), Pacific Symposium on Biocomputing. World Scientific, Singapore, 1998, 535-546.

[6] J. Xu, Y. Dong, and X. Wei, Sticker DNA Computer Model, Part 1: Theory, Chinese Science Bulletin, 2004, 49(8), 772-780.

[7] T. Audesirk, G. Audesirk, and B. E. Byers, Life on Earth, 4th Edition, United States of America, Pearson Prentice Hall, 2006.

[8] G. Rozenberg, G. Paun, and A. Salomaa, DNA Computing: A New Computing Paradigms, New York, Springer-Verlag, 1998.

[9] A. Alhazov, and M. Ferretti, Computing by Observing Bio-Systems: The Case of Sticker Systems. In C. Ferretti, G. Mauri, and C. Zandron Eds. DNA Computing: 10th International Workshop on DNA Computing, DNA 10, Milan, Italy, Springer-Verlag, 2004, 1-13.

[10] H. J. Hoogeboom, and N. V. Vugt, Fair Sticker Languages, ActaInformatica, 37 (2000), 213-225.

[11] N. V. Vugt, Models of Molecular Computing, University of Leiden, Ph.D. Thesis, 2002.

[12] C. B. Gupta, S. R. Singh, and S. Kumar, Advance Discrete Structure, New Delhi, I. K International Publishing House Pvt. Ltd, 2010.

[13] P. Linz, An Introduction to Formal Languages and Automata, 4th Edition, United States of America, Jones and Bartlett Publishers, 2006.

[14] J. Dassow, and G. Paun, Regulated Rewriting in Formal Language Theory. In EATXS Monograph in Theoretical Computer Science, Springer-Verlag, 18 (1989).

[15] H. Fernau, and R. Stiebe, Regulation by Valences. In B. Rovan (Ed.), Mathematical Foundations of Computer Science 1997 22nd International Symposium, MFCS'97 Proceedings, August 25-29, Springer-Verlag.

[16] V. Manca, and G. Paun, Arithmetically Controlled H Systems, Computer Science of Moldova, 6 (1998), 103-118.