Malaysian Journal of Fundamental and Applied Sciences Vol.9, No.3 (2013) 150-155

ISSN 1823-626X

Malaysian Journal of Fundamental and Applied Sciences

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

Probabilistic Sticker Systems

Mathuri Selvarajoo[1]*, Fong Wan Heng[2], Nor Haniza Sarmin[1] and Sherzod Turaev[3]

[1]*Department of Mathematical Sciences, Faculty of Science, Universiti Teknologi Malayisa, 81310 UTM Johor Bahru, Johor, Malaysia*
[2]*Ibnu Sina Institute for Fundamental Science Studies, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor, Malaysia*
[3]*Department of Computer Science, Kulliyyah of Information and Communication Technology, International Islamic University Malaysia, 53100 Kuala Lumpur*

Received 10 December 2012, Revised 25 March 2013, Accepted 20 April 2013, Available online 5 May 2013

ABSTRACT

A model for DNA computing using the recombination behaviour of DNA molecules known as a *sticker system* has been introduced by Adleman in 1994. A sticker model is an abstract computational model which uses the Watson-Crick complementary principle of DNA molecules. Starting from the axioms – incomplete double stranded sequences, and iteratively using sticking operations, complete double stranded sequences are obtained. It is known that sticker systems with finite sets of axioms and sticker rules generate only regular languages. Hence, different types of restrictions have been considered to increase the computational power of sticker systems. In this paper, we introduce *probabilistic sticker systems* in which probabilities are initially associated with the axioms, and the probability of the generated string is computed by multiplying the probabilities of all occurrences of the initial strings used in the computation of the string.

| DNA Computing | Sticker system |Probabilistic | Regular languages | Computationalpower |

http://dx.doi.org/10.11113/mjfas.v9n3.101

1. INTRODUCTION

The first formal tool for the generation of languages from DNA recombination was introduced by Head in 1987, known as *splicing systems*. Another language generating model which uses the sequential recombinant behaviour ("sticker operation") of DNA molecules is a *sticker system*. The sticker operation is used by Adleman in his experiment of computing a Hamiltonian path in a graph by using DNA molecules [1]. DNA or deoxyribonucleic acid is a genetic material in humans and all organisms. The structure of DNA is a double helix (helicoidal) which is composed of four nucleotides: **A** (adenine), **C** (cytosine), **G** (guanine), and **T** (thymine), which is paired as **A**-**T**, **C**-**G** according to *Watson-Crick complementary* [2].

The concept of sticker system as a language generating model based on sticker operations was first proposed by Kari[2]. The axioms and strings generated bya sticker system are considered as encoded modelsof single and double stranded DNA molecules. Moreover, the sticker operations have the advantages over splicing operations used in splicing systemsbecause the sticker operations require no strands extension and use no enzymes [3]. The initial sequences of DNA are prolonged to the left and right, producing computations of possible arbitrary length and the process stop when a complete double stranded sequence is obtained and no sticky ends exist [2].

2. STICKER SYSTEMS

2.1 Introduction

The sticker systems are language generating devices based on the sticker operation introduced by Adleman which has been turned into a model of the techniques used in his successful experiment of computing a Hamiltonian path in a graph by using DNA [1].

The conceptof regular sticker systems (one-sided sticker systems) was first considered in [2] and extended to bidirectional sticker systems in [4]. When forming new complete double stranded sequences, 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 sticker operation μ [5]. Starting from the axiom and iteratively using the operation of sticking, strands are prolonged in order to obtain a complete double stranded sequence.

A sticker system is a construct of 4-tuple

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

where$V$ is an alphabet, $\rho \subseteq V \times V$ is the symmetric relation 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 lower and upper stickers of the forms $\binom{V}{\#}^+$ and $\binom{\#}{V}^+$, respectively. In order to generate a complete double stranded string in $W_\rho(V)$, the sticker system starts from the axioms of$A$ and uses the pairs of B_d and $B_u$to prolong the

*Corresponding author. E-mail: mathuri87@yahoo.com strand to the left or to the right according to the sticker operation μ in order[2, 3, 4].

For a given sticker system $\gamma = (V, \rho, A, B_d, B_u)$ and two sequences $x, y \in (W_\rho(V))$, we write, $x \Rightarrow^* y$ if and only if $y = \mu(B_d, \mu(x, B_u))$. Note that, sequence $\mu(B_d, \mu(x, B_u)) = \mu(\mu(B_d, x), B_u)$ because the prolongation to the left is independent with the prolongation to the right [2]. By $\Rightarrow^*$, we denote the reflexive and transitive closure of the relation $\Rightarrow$. A sequence $x_1 \Rightarrow x_2 \Rightarrow \cdots \Rightarrow x_k$, where $x_1 \in A$, is called a computation in γ 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 [2].

A complete computation of a sticker system will produce a complete string, denoted by $w \in WK_\rho(V)$. The 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 γ and is called a *sticker language* [5, 6, 7]. Sticker operations and languages will be explained in the next section.

2.2 Sticker Operations

Let V be an alphabet (a finite set of abstract symbols) endowed with a symmetric relation ρ (of complementarity), $\rho \subseteq V \times V$. Let # be a special symbol not in V, denoting an empty space (the blank symbol).

Using the elements of $V \cup \{\#\}$, we construct the composite symbols of the following sets:

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

We denote

$$W_\rho(V) = \binom{V}{V}_\rho^* S(V),$$

where

$$S(V) = \binom{\#}{V}^* \cup \binom{V}{\#}^*.$$

$W_\rho(V)$is the set of well-started strings and $S(V)$ is the set ofthe combinations of upper and lower single strings. We call the elements of these set well-started sequences. In general, V^* is the set of all strings, including the empty string denotedby λ, composed of elements of V, and $V^+$is the set $V^* - \{\lambda\}$.

The sticker operation, denoted by μ, is a partially defined mapping from $W_\rho(V) \times S(V)$ to $W_\rho(V)$, defined as follows:

For $x \in W_\rho(V), y \in S(V)$and $z \in W_\rho(V)$, we have $\mu(x, y) = z$ if and only if one of the following cases holds [2]:

1. $x = \binom{a_1}{b_1} \cdots \binom{a_k}{b_k} \binom{a_{k+1}}{\#} \cdots \binom{a_{k+r}}{\#} \binom{a_{k+r+1}}{\#} \cdots \binom{a_{k+r+p}}{\#}$,

$$y = \binom{\#}{c_1} \cdots \binom{\#}{c_r},$$
$$z = \binom{a_1}{b_1} \cdots \binom{a_k}{b_k} \binom{a_{k+1}}{c_1} \cdots \binom{a_{k+r}}{c_r} \binom{a_{k+r+1}}{\#} \cdot \binom{a_{k+r+p}}{\#},$$

for$k \geq 0, r \geq 1, p \geq 1, a_i \in V, 1 \leq i \leq k+r+p, b_i \in V, 1 \leq i \leq k, c_i \in V, 1 \leq i \leq r$, and$(a_{k+i}, c_i) \in \rho, 1 \leq i \leq r$;

2. $x = \binom{a_1}{b_1} \cdots \binom{a_k}{b_k} \binom{a_{k+1}}{\#} \cdots \binom{a_{k+r}}{\#}$,

$$y = \binom{\#}{c_1} \cdots \binom{\#}{c_r} \binom{\#}{c_{r+1}} \cdots \binom{\#}{c_{r+p}},$$
$$z = \binom{a_1}{b_1} \cdots \binom{a_k}{b_k} \binom{a_{k+1}}{c_1} \cdots \binom{a_{k+r}}{c_r} \binom{\#}{c_{r+1}} \cdots \binom{\#}{c_{r+p}},$$

for$k \geq 0, r \geq 0, p \geq 0, r+p \geq 1$
$a_i \in V, 1 \leq i \leq k+r, b_i \in V, 1 \leq i \leq k,$
$c_i \in V, 1 \leq i \leq r+p$,and$(a_{k+i}, c_i) \in \rho, 1 \leq i \leq r$;

3. $x = \binom{a_1}{b_1} \cdots \binom{a_k}{b_k} \binom{\#}{b_{k+1}} \cdots \binom{\#}{b_{k+r}} \binom{\#}{b_{k+r+1}} \cdot \binom{\#}{b_{k+r+p}}$,

$$y = \binom{c_1}{\#} \cdots \binom{c_r}{\#},$$
$$z = \binom{a_1}{b_1} \cdots \binom{a_k}{b_k} \binom{c_1}{b_{k+1}} \cdots \binom{c_{k+r}}{b_{k+r}} \binom{\#}{b_{k+r+1}} \cdot \binom{\#}{b_{k+r+p}},$$

for$k \geq 0, r \geq 1, p \geq 1, a_i \in V, 1 \leq i \leq k, b_i \in V, 1 \leq i \leq k+r+p, c_i \in V, 1 \leq i \leq r$, and $(c_i, b_{k+i}) \in \rho, 1 \leq i \leq r$;

4. $x = \binom{a_1}{b_1} \cdots \binom{a_k}{b_k} \binom{\#}{b_{k+1}} \cdots \binom{\#}{b_{k+r}}$,

$$y = \binom{c_1}{\#} \cdots \binom{c_r}{\#} \binom{c_{r+1}}{\#} \cdots \binom{c_{r+p}}{\#},$$
$$z = \binom{a_1}{b_1} \cdots \binom{a_k}{b_k} \binom{c_1}{b_{k+1}} \cdots \binom{c_r}{b_{k+r}} \binom{c_{r+1}}{\#} \cdots \binom{c_{r+p}}{\#},$$

for$k \geq 0, r \geq 0, p \geq 0, r+p \geq 1, a_i \in V, 1 \leq i \leq k, b_i \in V, 1 \leq i \leq k+r, c_i \in V, 1 \leq i \leq r+p$, and $(c_i, b_{k+i}) \in \rho, 1 \leq i \leq r$.

2.3 Sticker Languages

A sticker language (SL) is the language generated by a sticker system which consists of all strings formed by the set of upper strands of all complete strings derived from the axioms for which an exactly matching sequence of lower stickers can be found [5, 6]. Sticker system will normally generate only regular languages without restrictions [8].

Definition 1[9]

A *deterministic finite accepter* or *dfa* is defined by the quintuple

$$M = (Q, \Sigma, \delta, q_0, F),$$

where

- Q is a finite set of internal states,
- Σ is a finite set of symbols called the input alphabet,

- $\delta : Q \times \Sigma \to Q$ is a total function called the transition function,
- $q_0 \in Q$ is the initial state,
- $F \subseteq Q$is a set of final states.

Definition 2[9]

A language L is called *regular* if and only if there exists some deterministic finite accepter$M$ such that $L = L(M)$.

Some restrictions have been introduced in [2, 4, 6, 7, 9] on matching pairs of strands to obtain more powerful languages of sticker systems such as simple sticker languages, primitive sticker languages, and balance sticker languages.

Let $\gamma = (V, \rho, A, D)$ be a sticker system, the unrestricted molecular language generated by $\gamma$is defined in [9] as follows: $ML(\gamma) = \{(x, y) \in \rho | (x, y)$is a computation of $\gamma\}$. Furthermore, the unrestricted sticker language generated by γ is the projection onto the first (upper) component of the molecular language, because of the complementarity relation of ρ. The language denoted as $L(\gamma)$ is defined by:

$$L(\gamma) = \{w \in WK_\rho(V) | x \Rightarrow^* w, x \in A\}.$$

The similarity of this process to Adleman's experiment can be found in [2].

There are few types of languages generated by sticker systems which are the one-sided sticker languages, regular sticker languages, simple sticker languages, simple one-sided sticker languages and simple regular sticker languages denoted by *OSSL*, *RSL*, *SSL*, *SOSSL* and *SRSL*,respectively. The definitions of these families of languages are listed in the following.

Definition 3 [4]

A language resulting from a sticker system is called a *one-sided sticker language* (*OSSL*) if for each pair $(u, v) \in D$, we have either $u = \lambda$or $v = \lambda$.

Definition 4 [4]

A language resulting from a sticker system is called a *regular sticker language* (*RSL*) if for each pair $(u, v) \in D$, we have $u = \lambda$.

Definition 5 [4]:

A language resulting from sticker system is called *simple sticker language* (*SSL*) if for all pairs $(u, v) \in D$, we have either $u, v \in \binom{V^*}{\lambda}$or $u, v \in \binom{\lambda}{V^*}$.

Definition 6 [4]:

A language resulting from sticker system is called *simple one-sided sticker language* (*SOSSL*) if for all pairs $(u, v) \in D$, we have either $u, v \in \binom{V^*}{\lambda}$ or $u, v \in \binom{\lambda}{V^*}$ and for each pair $(u, v) \in D$, we have either $u = \lambda$or $v = \lambda$.

Definition 7 [4]

A language resulting from a sticker system is called a *simple regular sticker language* (*SRSL*) if for all pairs $(u, v) \in D$, we have either $u, v \in \binom{V^*}{\lambda}$ or $u, v \in \binom{\lambda}{V^*}$ and for each pair $(u, v) \in D$, we have $u = \lambda$.

3. PROBABILISTIC STICKER SYSTEM, OPERATION AND LANGUAGE

3.1 Probabilistic Sticker Systems

In this section, some notations of probabilistic sticker systems, operations and languages over probability are introduced.

Definition 8

A *probabilistic sticker system* (*pSS*) is a 5-tuple

$$\gamma = (V, \rho, A_p, D_p, p),$$

where$V$ is an alphabet, $\rho \subseteq V \times V$ is the symmetric relation in V, $A_p$is a finite subset of axioms $(W_\rho(V) \times p)$,D_p is a finite set of pairs $[(B_d \times B_u) \times p]$ where B_d and $B_u$are finite subsets of lower and upper stickers of the forms $\binom{\#}{V}^+$ and $\binom{V}{\#}^+$respectively and $p : V^* \to [0, 1]$is a probability function such that

$$\sum_{(x, p(x)) \in A_p, D_p} p(x) = 1.$$

3.2 Probabilistic Sticker Operations

Definition 9

For $(x, p(x)) \in A_p$ and $(u, p(u)), (v, p(v)) \in D_p$,

$$[x, p(x)] \overset{*}{\Rightarrow} [y, p(y)]$$

if and only if

i)

$(y, p(y)) = \mu\left[(u, p(u)), \mu\left((x, p(x)), (v, p(v))\right)\right]andp(y) = p(u) \cdot [p(x) \cdot p(v)]$.

ii)

$(y, p(y)) = \mu\left[\mu\left((x, p(x)), (u, p(u))\right), (v, p(v))\right]$and $p(y) = [p(x) \cdot p(u)] \cdot p(v)$.

3.3 Probabilistic Sticker Languages

Definition 10

The language generated by a probabilistic sticker system is defined as

$$pSL(\gamma) = \{y \in WK_p(V) | (x, p(x)) \overset{*}{\Rightarrow} (y, p(y)) \text{for } (x, p(x)) \in A_p\}.$$

In order to increase the generative power of probabilistic sticker systems, we consider a threshold (cut-point) sub-segments and discrete subset of [0, 1] as well as real numbers in [0, 1]. We define the following two types of threshold languages with respect to thresholds $\alpha \subseteq [0,1]$ and $\beta \in [0,1]$:

$$pSL(\gamma, * \alpha) = \left\{y \in WK_p(V) \middle| (x, p(x)) \overset{*}{\Rightarrow} (y, p(y))\right\} \text{for } (y, p(y)) \in WK_p(V) \wedge p(y) * \alpha.$$

$$pSL(\gamma, \#\beta) = \left\{y \in WK_p(V) \middle| (x, p(x)) \overset{*}{\Rightarrow} (y, p(y))\right\} \text{for } (y, p(y)) \in WK_p(V) \wedge p(y) \# \beta$$

where $* \in \{=, \neq, \geq, >, \leq, <\}$ and $\# \in \{\in, \notin\}$ are threshold modes.

Lemma 1

The relation $SL \subseteq pSL$ holds where pSL indicates the set of all languages generated by probabilistic sticker systems.

Proof.

Consider a sticker system $\gamma = (V, \rho, A, D)$. Then the language generated by the sticker system γ is

$$SL(\gamma) = \left\{z \in WK(V) \middle| x \overset{*}{\Rightarrow} z, x \in A\right\}.$$

Let $\gamma_1 = (V, \rho, A_p, D_p, p)$ be a probabilistic sticker system where

$$A_p = \{(x_i, p(x_i)) | x_i \in A, 1 \leq i \leq n\},$$

$$D_p = \{(u_i, p(u_i)), (v_i, p(v_i)) | u_i, v_i \in D, 1 \leq i \leq n\}$$

and $p(\theta_i) = 1/n$ for all $1 \leq i \leq n, \theta \in \{x, u, v\}$, then

$$\sum_{i=1}^{n} p(\theta_i) = 1.$$

The language generated by the probabilistic sticker system γ_1:

$$pSL(\gamma_1, * \alpha) = \{z \in WK_p(V) \mid [x, p(x)] \overset{*}{\Rightarrow} [z, p(z)] \text{for } [x, p(x)] \in A_p \text{where } p(z) = p(x) \cdot p(\tau_1) \cdot p(\tau_2) \cdots p(\tau_n) \text{ for } \tau_1, \tau_2, \ldots, \tau_n \in D_p\}.$$

We define the threshold language generated by γ_1 as $pSL(\gamma_1, > 0)$, then it is not difficult to see that

$$SL(\gamma) = pSL(\gamma_1, > 0).$$

□

Example 1

Given a probabilistic sticker system $\gamma = \{V, \rho, A, D, p\}$ where

$V = \{a, b\}$,

$\rho = \{(a,a), (b,b)\}$,

$$A = \left\{\binom{a}{\lambda}\binom{\lambda}{b}, \left(\frac{2}{28}\right)\right\},$$

$$D = \left\{\left(\binom{\lambda}{a}\binom{\lambda}{a}, \frac{3}{28}\right), \left(\binom{b}{\lambda}\binom{b}{\lambda}, \frac{5}{28}\right),\right.$$

$$\left.\left(\binom{a}{\lambda}, \frac{7}{28}\right), \left(\binom{\lambda}{b}, \frac{11}{28}\right)\right\}.$$

For the given sticker system, we start the computation with the axiom in A being attached to its complementary axiom from D. The first step of computation starts with prolongation to the right side and it is shown below:

$$\left[\left\{\binom{a}{\lambda}\binom{\lambda}{b}, \frac{2}{28}\right\}, \left\{\binom{b}{\lambda}\binom{b}{\lambda}, \frac{5}{28}\right\}\right] \Rightarrow$$

$$\left\{\binom{a}{\lambda}\binom{b}{b}\binom{b}{\lambda}, \left(\frac{2}{28}\right) \cdot \left(\frac{5}{28}\right)\right\}.$$

The computation is complete when a complete double strand sequence is obtained, that is when no sticky end exists in the string. Here, sticky end still exists from the first step. Therefore, the computation has to be continued until the double strand sequence is obtained.

For this example, the second step of the computation is shown in the following:

$$\left[\left\{\binom{a}{\lambda}\binom{b}{b}\binom{b}{\lambda}, \left(\frac{2}{28}\right) \cdot \left(\frac{5}{28}\right)\right\}, \left\{\binom{\lambda}{b}, \frac{11}{28}\right\}\right] \Rightarrow$$

$$\left\{\binom{a}{\lambda}\binom{b}{b}\binom{b}{b}, \left(\frac{2}{28}\right) \cdot \left(\frac{5}{28}\right) \cdot \left(\frac{11}{28}\right)\right\}.$$

To obtain a complete computation on the right side, the sticking operations have to follow the above steps. Now, to obtain a complete computation on the left side, the computation follows the steps below:

$$\left[\left\{\binom{a}{\lambda}\binom{\lambda}{b}, \frac{2}{28}\right\}, \left\{\binom{\lambda}{a}\binom{\lambda}{a}, \frac{3}{28}\right\}\right] \Rightarrow$$

$$\left\{\binom{\lambda}{a}\binom{a}{a}\binom{\lambda}{b}, \left(\frac{2}{28}\right) \cdot \left(\frac{3}{28}\right)\right\},$$

$$\left[\left\{\binom{\lambda}{a}\binom{a}{a}\binom{\lambda}{b}, \left(\frac{2}{28}\right) \cdot \left(\frac{3}{28}\right)\right\}, \left\{\binom{a}{\lambda}, \left(\frac{7}{28}\right)\right\}\right] \Rightarrow ,$$

$$\left\{\binom{a}{a}\binom{a}{a}\binom{\lambda}{b}, \left(\frac{2}{28}\right) \cdot \left(\frac{3}{28}\right) \cdot \left(\frac{7}{28}\right)\right\}.$$

By joining both the stickers, the complete double strand string with the probability $\left\{\left(\frac{2}{28}\right)\cdot\left(\frac{3\cdot 7\cdot 5\cdot 11}{28}\right)^{n}\right\}$ is obtained. Then the language produced is

$$L(\gamma, p)=\{(a^{2k}b^{2m},\left(\frac{2}{28}\right)\cdot\left(\frac{3\cdot 7}{28^2}\right)^{k}\left(\frac{5.11}{28^2}\right)^{m})\mid k,m\geq 1\},$$

where $p=\left(\frac{2}{28}\right)\cdot\left(\frac{3\cdot 7}{28^2}\right)^{k}\left(\frac{5.11}{28^2}\right)^{m}$. The computation is shown below:

$$\left[\begin{matrix}\left\{\begin{pmatrix}a\\a\end{pmatrix}\begin{pmatrix}\lambda\\a\end{pmatrix}\begin{pmatrix}a\\\lambda\end{pmatrix}\begin{pmatrix}\lambda\\b\end{pmatrix},\left(\frac{2}{28}\right)\cdot\left(\frac{3}{28}\right)\cdot\left(\frac{7}{28}\right)\right\},\\ \left\{\begin{pmatrix}a\\\lambda\end{pmatrix}\begin{pmatrix}\lambda\\b\end{pmatrix}\begin{pmatrix}b\\\lambda\end{pmatrix}\begin{pmatrix}b\\b\end{pmatrix},\left(\frac{2}{28}\right)\cdot\left(\frac{5}{28}\right)\cdot\left(\frac{11}{28}\right)\right\}\end{matrix}\right]\Rightarrow$$

$$\left\{\begin{pmatrix}a\\a\end{pmatrix}\begin{pmatrix}a\\a\end{pmatrix}\begin{pmatrix}b\\b\end{pmatrix}\begin{pmatrix}b\\b\end{pmatrix},\left(\frac{2}{28}\right)\cdot\left(\frac{3}{28}\right)\cdot\left(\frac{7}{28}\right)\cdot\left(\frac{5}{28}\right)\cdot\left(\frac{11}{28}\right)\right\}.$$

Hence, a complete computation is obtained. We can iteratively prolonged a complete double strand DNA sequences to generate the general form of the language produced using the same sequences as done earlier.

We can produce a language of sticker system as $L(\gamma, p)=\{a^{2n}b^{2n}\mid n\geq 1\}$ when the probability p is equal to $\left\{\left(\frac{2}{28}\right)\cdot\left(\frac{3\cdot 7\cdot 5\cdot 11}{28}\right)^{n}\right\}$.

Therefore,

$$L(\gamma, p)=\{(a^{2k}b^{2m},\left(\frac{2}{28}\right)\cdot\left(\frac{3\cdot 7}{28^2}\right)^{k}\left(\frac{5.11}{28^2}\right)^{m})\mid k,m\geq 1\},$$

where $p=\left(\frac{2}{28}\right)\cdot\left(\frac{3\cdot 7}{28^2}\right)^{k}\left(\frac{5.11}{28^2}\right)^{m}$.

Using the threshold properties, we can conclude the following:

i : $\eta=0,\Rightarrow L(\gamma,=0)=\varnothing\in REG,$

ii : $\eta>0,\Rightarrow L(\gamma,>0)=L(\gamma)\in REG,$

iii :

$$\bar{\eta}=\{\left(\frac{2}{28}\right)\left(\frac{3.7.5.11}{28^4}\right)^{n}\mid n\geq 1\},\Rightarrow$$

$$L(\gamma,\bar{\eta})=\left\{a^{2n}b^{2n}\mid n\geq 1\right\}\in CF-REG,$$

iv :

$$\dot{\eta}\neq\{\left(\frac{2}{28}\right)\left(\frac{3.7.5.11}{28^4}\right)^{n}\mid n\geq 1\},\Rightarrow$$

$$L(\gamma,\dot{\eta})=\left\{a^{k}b^{m}\mid k>m\geq 1\right\}\cup$$

$$\left\{a^{k}b^{m}\mid m>k\geq 1\right\}\in CF-REG.$$

Proposition 1

For any probabilistic sticker system (G), the threshold language $L(G,=0)$is the empty set, i.e. $L(G,=0)=\emptyset$.

Proposition 2

If for each prolongation in a probabilistic sticker system (G), $p(r)<1$,then every threshold language $L(G,>\eta)$ with $\eta>0$ is finite.

4. RESTRICTED VARIANTS OF PROBABILISTIC STICKER SYSTEMS

In this section, we define probabilistic variants of some restrictions of sticker systems.

Definition 11

A probabilistic sticker system $\gamma=(V,\rho,A_p,D_p,p)$is said to be a *probabilistic one-sided sticker system* (*pOSL*) if for each pair $((u,p(u),(v,p(v))\in D_p$ either $(u,p(u))\rightarrow\left(\lambda,\frac{1}{n}\right)$or $\left(v,p(v)\right)\rightarrow\left(\lambda,\frac{1}{n}\right)$ where $\frac{1}{n}$is the probability of each axiom.

Definition 12

A probabilistic sticker system $\gamma=\left(V,\rho,A_p,D_p,p\right)$is said to be a *probabilistic regular sticker system*(*pRSL*) if for each pair $((u,p(u),(v,p(v))\in D_p,(u,p(u))\rightarrow\left(\lambda,\frac{1}{n}\right)$ where $\frac{1}{n}$is the probability of each axiom.

Definition 13

A probabilistic sticker system $\gamma=\left(V,\rho,A_p,D_p,p\right)$is said to be a *probabilistic simple sticker system*(*pSSL*) if for all pairs $((u,p(u),(v,p(v))\in D_p$,either$((u,p(u),(v,p(v))$ $\in(\frac{\lambda}{V^*})\times p$or $((u,p(u),(v,p(v))\in(\frac{V^*}{\lambda})\times p$.

Definition 14

A probabilistic sticker system$\gamma=\left(V,\rho,A_p,D_p,p\right)$is said to be a *probabilistic simple one-sided sticker system*(*pSOSL*) if for all pairs $((u,p(u),(v,p(v))\in D_p$, either

$((u,p(u),(v,p(v)) \in (\frac{\lambda}{V^*}) \times p$ or $((u,p(u),(v,p(v)) \in (\frac{V^*}{\lambda}) \times p$ and for each pair $((u,p(u),(v,p(v)) \in D_p$,either $(u,p(u)) \to \left(\lambda,\frac{1}{n}\right)$or $\left(v,p(v)\right) \to \left(\lambda,\frac{1}{n}\right)$ where $\frac{1}{n}$is the probability of each axiom.

Definition 15

A probabilistic sticker system $\gamma = \left(V,\rho,A_p,D_p,p\right)$is said to be a *probabilistic simple regular sticker system*($pSRSL$) if for all pairs $((u,p(u),(v,p(v)) \in D_p$,either $((u,p(u),(v,p(v)) \in (\frac{\lambda}{V^*}) \times p$ or $((u,p(u),(v,p(v)) \in (\frac{V^*}{\lambda}) \times p$ and for each pair $((u,p(u),(v,p(v)) \in D_p$, and for each pair $((u,p(u),(v,p(v)) \in D_p,(u,p(u)) \to \left(\lambda,\frac{1}{n}\right)$where $\frac{1}{n}$is the probability of each axiom.

5. CONCLUSION

In this paper, the definition of a new restriction of sticker system namely the probabilistic sticker system has been introduced. In addition, some results show that the generative power of sticker systems can be increased with the presence of probability. In addition, the definitions of some new restrictions of variants for sticker system, namely, *probabilisticone-sided sticker systems*, *probabilisticregular sticker systems, probabilisticsimple sticker systems*, *probabilistic simple one-sided sticker systems and probabilistic simple regular sticker systems*are also introduced**.**

ACKNOWLEDGEMENT

The first author would like to thank the Ministry of Higher Education Malaysia for the financial funding through MyBrain15 scholarship. The second and third authors would also like to thank the Ministry of Higher Education (MOHE) and Research Management Centre (RMC), UTM for the financial funding through Research University Fund Vote No. 05J15.

REFERENCES

[1] L.M. Adleman, Science, 266 (1994) 1021-1024.
[2] L. Kari, G. Paun, G. Rozenberg, A. Salomaa, and S. Yu, ActaInformatica, 35 (1998) 401-420.
[3] L. Kari, S. Seki, and P. Sosik, DNA Computing: Foundations and Implications for Computer Science, Springer-Verlag, Berlin, 2010.
[4] R. Freund, G. Paun, G. Rozenberg, and A. Salomaa, Pacific Symposium on Biocomputing, (1998) 535-546.
[5] G. Paun, and G. Rozenberg, Theoretical Computer Science, 204 (1998) 183-203.
[6] J. Xu, Y. Dong, and X. Wei, Chinese Science Bulletin, 49(8) (2004) 772-780.
[7] G. Rozenberg, G. Paun, and A. Salomaa, DNA Computing: A New Computing Paradigms,Springer-Verlag,New York, 1998.
[8] A. Alhazov, and M. Ferretti, Computing by Observing Bio-Systems: The Case of Sticker Systems, DNA Computing, Springer-Verlag,Italy, 2004, 1-13.
[9] P. Linz, An Introduction to Formal Languages and Automata. 4th Edition,Jones and Bartlett Publishers,United States of America, 2006.