AIP Conference Proceedings

The properties of probabilistic simple regular sticker system

Mathuri Selvarajoo, Wan Heng Fong, Nor Haniza Sarmin, and Sherzod Turaev

Citation: AIP Conference Proceedings **1682**, 020047 (2015); doi: 10.1063/1.4932456
View online: http://dx.doi.org/10.1063/1.4932456
View Table of Contents: http://scitation.aip.org/content/aip/proceeding/aipcp/1682?ver=pdfcov
Published by the AIP Publishing

The Properties of Probabilistic Simple Regular Sticker System

Mathuri Selvarajoo[a], Wan Heng Fong[b], Nor Haniza Sarmin[a] and Sherzod Turaev[c]

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

Abstract. A mathematical model for DNA computing using the recombination behavior of DNA molecules, known as a *sticker system*, has been introduced in 1998. In sticker system, the sticker operation is based on the Watson-Crick complementary feature of DNA molecules. The computation of sticker system starts from an incomplete double-stranded sequence. Then by iterative sticking operations, a complete double-stranded sequence is obtained. It is known that sticker systems with finite sets of axioms and sticker rule (including the simple regular sticker system) generate only regular languages. Hence, different types of restrictions have been considered to increase the computational power of the languages generated by the sticker systems. In this paper, we study the properties of *probabilistic simple regular sticker systems*. In this variant of sticker system, probabilities are associated with the axioms, and the probability of a generated string is computed by multiplying the probabilities of all occurrences of the initial strings. The language are selected according to some probabilistic requirements. We prove that the probabilistic enhancement increases the computational power of simple regular sticker systems.

Keywords: DNA computing; sticker system; probability; regular language; computational power
PACS: 02.20-a, 02.50.Cw, 87.14.gk

INTRODUCTION

A formal tool for the generation of languages from DNA recombination was introduced by Head in 1987, known as the mathematical modeling of splicing system [1]. There is in fact another method of DNA computing, known as the sticker operation. Sticker operation is a model of the techniques used by Adleman in his experiment of computing a Hamiltonian path in a graph by using DNA [2]. 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 complementarity [1].

The concept of sticker system was proposed by Kari in 1998 based on the sticker operation [3]. Sticker operation starts from a given set of incomplete double-stranded sequences encoded with DNA molecules. 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, that is when no "sticky end" exists [4]. When a complete double-stranded sequence is obtained, a sticker language is generated. The unrestricted case corresponds to the Adleman's experiment in generating only regular languages (a weak coding) [5]. Hence, additional restrictions are imposed to generate higher level of languages according to the Chomsky hierarchy.

In [6] we introduced a new variant of sticker systems, called *probabilistic sticker systems*, where probabilities (real numbers in the range [0, 1]) are associated with the axioms, and the probability $p(z)$ of the string z generated from two strings x and y is calculated from the probability $p(x)$ and $p(y)$ according to the multiplication operation $*$ defined on the probabilities, i.e., $p(z) = p(x) * p(y)$. Then the language generated by a probabilistic sticker system consists of all strings generated by the sticker systems whose probabilities are greater than (or smaller than, or equal to) some previously chosen cut-points. In this paper we study the inclusion properties of *probabilistic simple regular sticker systems*. This paper is organized as follows: first, the necessary definitions and results from formal language theory and DNA computing are recalled. Next, the concept of a probabilistic simple regular sticker system is introduced. Moreover, some preliminary results concerning the computational power of probabilistic

The 22nd National Symposium on Mathematical Sciences (SKSM22)
AIP Conf. Proc. 1682, 020047-1–020047-9; doi: 10.1063/1.4932456

simple regular sticker systems are also discussed. Finally, this paper ends with the conclusion of the research and possible directions for future study.

PRELIMINARIES

In this section we recall some prerequisites by giving basic notions of the theories of formal languages, sticker systems and probabilistic sticker systems which are used in this paper. The reader is referred to [3, 5-7] for more detailed information. Throughout the paper we use the following general notations. The symbol $\in$ denotes the membership of an element to a set while the negation of set membership is denoted by $\notin$. The inclusion is denoted by $\subseteq$ and the strict (proper) inclusion is denoted by $\subset$. The empty set is denoted by $\varnothing$. The sets of integers, positive rational numbers and real numbers are denoted by $\mathbb{Z}$, $\mathbb{Q}^+$ and $\mathbb{R}$ respectively.

The families of recursively enumerable, context-sensitive, context-free, linear, regular and finite languages are denoted by **RE**, **CS**, **CF**, **LIN**, **REG**, and **FIN** respectively. For these language families, the next strict inclusions, named Chomsky hierarchy, hold

$$\mathbf{FIN} \subset \mathbf{REG} \subset \mathbf{LIN} \subset \mathbf{CF} \subset \mathbf{CS} \subset \mathbf{RE}.$$

Definition 1. Deterministic Finite Acceptor [7]

A *deterministic finite accepter* or *DFA* is defined by a 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 transition function, $q_0 \in Q$ is the initial state, and $F \subseteq Q$ is a set of final states.

Definition 2. Nondeterministic Finite Acceptor [7]

A *nondeterministic finite accepter* or *NFA* is defined by a 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, $\delta : Q \times \left(\Sigma \cup \{\lambda\}\right) \to 2^Q$ is a transition function, $q_0 \in Q$ is the initial state, and $F \subseteq Q$ is a set of final states.

Definition 3. Regular Language [7]

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

The concept of *sticker system* was first considered in [3] and extended to *bidirectional sticker system* in [8]. 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 by the process of the sticker operation μ (see [9]). Starting from the axiom and by iteratively using the sticker operation, the strands are prolonged in order to obtain a complete double-stranded sequence.

Let V be an alphabet (a finite set of abstract symbols) endowed with a symmetric relation ρ (of complementarity), i.e., $\rho \subseteq V \times V$. The symbol V^* is the set of all strings, including the empty string denoted by $\#$, composed of elements of V, and $V^+ = V^* - \{\#\}$. We write $\begin{pmatrix} V^* \\ V^* \end{pmatrix}$ instead of $V^* \times V^*$. We denote $\begin{bmatrix} V \\ V \end{bmatrix}_\rho = \left\{ \begin{pmatrix} a \\ b \end{pmatrix} \middle| a, b \in V, (a,b) \in \rho \right\}$ and $WK_\rho(V) = \begin{bmatrix} V \\ V \end{bmatrix}_\rho^*$.

The set $WK_\rho(V)$ is called the *Watson-Crick domain* associated to the alphabet V and complementarity relation ρ. We also denote

$$W_\rho(V) = L_\rho \cup R_\rho \cup LR_\rho,$$

where

$$L_\rho(V) = \left(\begin{pmatrix} \# \\ V^* \end{pmatrix} \cup \begin{pmatrix} V^* \\ \# \end{pmatrix} \right) \begin{bmatrix} V \\ V \end{bmatrix}_\rho^*,$$

$$R_\rho(V) = \begin{bmatrix} V \\ V \end{bmatrix}_\rho^* \left(\begin{pmatrix} \# \\ V^* \end{pmatrix} \cup \begin{pmatrix} V^* \\ \# \end{pmatrix} \right),$$

$$LR_\rho(V) = \left(\begin{pmatrix} \# \\ V^* \end{pmatrix} \cup \begin{pmatrix} V^* \\ \# \end{pmatrix} \right) \begin{bmatrix} V \\ V \end{bmatrix}_\rho^+ \left(\begin{pmatrix} \# \\ V^* \end{pmatrix} \cup \begin{pmatrix} V^* \\ \# \end{pmatrix} \right).$$

Any element of $W_\rho(V)$ which contains at least a position $\begin{bmatrix} a \\ b \end{bmatrix}, a \neq \#, b \neq \#,$ is called a *well-started* double-stranded sequence.

Let $x = x_1 x_2 x_3 \in W_\rho(V)$ where $x_2 \in WK_\rho(V) - \left\{ \begin{bmatrix} \# \\ \# \end{bmatrix} \right\}$ is a well-started double-stranded sequence and $y \in W_\rho(V)$ is a double-stranded sequence. The sticker operation, denoted by $\mu(x, y)$, is defined as follows:

1. $x_3 = \begin{pmatrix} u \\ \# \end{pmatrix}, y = \begin{pmatrix} \# \\ v \end{pmatrix} y',$ for $u, v \in V^*$ such that $\begin{bmatrix} u \\ v \end{bmatrix} \in WK_\rho(V)$ and $y' \in R_\rho(V)$ then $\mu(x, y) = x_1 x_2 \begin{bmatrix} u \\ v \end{bmatrix} y'$;
2. $x_3 = \begin{pmatrix} \# \\ u \end{pmatrix}, y = \begin{pmatrix} v \\ \# \end{pmatrix} y',$ for $u, v \in V^*$ such that $\begin{bmatrix} u \\ v \end{bmatrix} \in WK_\rho(V)$ and $y' \in R_\rho(V)$ then $\mu(x, y) = x_1 x_2 \begin{bmatrix} u \\ v \end{bmatrix} y'$;
3. $x_3 = \begin{pmatrix} u_1 \\ \# \end{pmatrix}, y = \begin{pmatrix} u_2 \\ \# \end{pmatrix},$ for $u_1, u_2 \in V^*$ then $\mu(x, y) = x_1 x_2 \begin{pmatrix} u_1 u_2 \\ \# \end{pmatrix}$;
4. $x_3 = \begin{pmatrix} u_1 u_2 \\ \# \end{pmatrix}, y = \begin{pmatrix} \# \\ v \end{pmatrix},$ for $u_1, u_2, v \in V^*$ such that $\begin{bmatrix} u_1 \\ v \end{bmatrix} \in WK_\rho(V)$ then $\mu(x, y) = x_1 x_2 \begin{bmatrix} u_1 \\ v \end{bmatrix} \begin{pmatrix} u_2 \\ \# \end{pmatrix}$;
5. $x_3 = \begin{pmatrix} u \\ \# \end{pmatrix}, y = \begin{pmatrix} \# \\ v_1 v_2 \end{pmatrix},$ for $u, v_1, v_2 \in V^*$ such that $\begin{bmatrix} u \\ v_1 \end{bmatrix} \in WK_\rho(V)$ then $\mu(x, y) = x_1 x_2 \begin{bmatrix} u \\ v_1 \end{bmatrix} \begin{pmatrix} \# \\ v_2 \end{pmatrix}$;
6. $x_3 = \begin{pmatrix} \# \\ v_1 \end{pmatrix}, y = \begin{pmatrix} \# \\ v_2 \end{pmatrix},$ for $v_1, v_2 \in V^*$ then $\mu(x, y) = x_1 x_2 \begin{pmatrix} \# \\ v_1 v_2 \end{pmatrix}$;

7. $x_3 = \begin{pmatrix} \# \\ v_1 v_2 \end{pmatrix}$, $y = \begin{pmatrix} u \\ \# \end{pmatrix}$, for $u, v_1, v_2 \in V^*$ such that $\begin{bmatrix} u \\ v_1 \end{bmatrix} \in WK_\rho(V)$ then $\mu(x, y) = x_1 x_2 \begin{bmatrix} u \\ v_1 \end{bmatrix}\begin{pmatrix} \# \\ v_2 \end{pmatrix}$;

8. $x_3 = \begin{pmatrix} \# \\ v \end{pmatrix}$, $y = \begin{pmatrix} u_1 u_2 \\ \# \end{pmatrix}$, for $u_1, u_2, v \in V^*$ such that $\begin{bmatrix} u_1 \\ v \end{bmatrix} \in WK_\rho(V)$ then $\mu(x, y) = x_1 x_2 \begin{bmatrix} u_1 \\ v \end{bmatrix}\begin{pmatrix} u_2 \\ \# \end{pmatrix}$.

$\mu(y, x)$ can also be defined in the symmetric way.

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, $A \in LR_\rho(V)$ is a finite set of axioms, and D is a finite subset of $W_\rho(V) \times W_\rho(V)$.

For a given sticker system $\gamma = (V, \rho, A, D)$ and two sequences $x, y \in LR_\rho(V)$, we write, $x \Rightarrow y$ if and only if $\mu(u, \mu(x, v))$ for some $(u, v) \in D$. Note that, $\mu(u, \mu(x, v)) = \mu(\mu(u, x), v)$. 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 $k-1$. If $x_k \in WK_\rho(V)$, the above computation is considered as *complete*.

The "*language of molecules*", denoted by $LM(\gamma)$, is the set of all double-stranded strings over V produced at the end of complete computations in γ i.e.,

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

The *sticker language* generated by γ is defined by

$$L(\gamma) = \left\{ w \in V^* \mid \begin{bmatrix} w \\ w' \end{bmatrix} \in LM_n(\gamma) \text{ for some } w' \in V^* \right\}.$$

A sticker system $\gamma = (V, \rho, A, D)$ is called *simple-regular* if for all pairs $(u, v) \in D$ we have either $u, v \in \begin{pmatrix} \# \\ V \end{pmatrix}^*$ or $u, v \in \begin{pmatrix} V \\ \# \end{pmatrix}^*$, and for each $(u, v) \in D$ we have $u, v \in \begin{pmatrix} \# \\ V \end{pmatrix}^*$. The languages generated by simple-regular sticker system are known as *simple-regular sticker language* and denoted as *SRSL*.

Next, some important definitions of probabilistic sticker system are explained.

Definition 4 [6]: Probabilistic Sticker System

A *probabilistic sticker system* (*pSS*) is a 4-tuple $\gamma = (V, \rho, A_p, D_p)$, where V is an alphabet, $\rho \subseteq V \times V$ is a symmetric relation, $A_p \in LR_\rho(V) \times [0,1]$ is a finite set of axioms, D_p is a finite subset of $(W_\rho(V) \times [0,1]) \times (W_\rho(V) \times [0,1])$, and $p : V^* \to [0,1]$ is a probability function such that

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

Definition 5 [6]: Probabilistic Sticker Operation

A probabilistic sticker operation is defined as follows:

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

$$\left[\left(x, p\left(x\right)\right)\right] \Rightarrow^{*} \left[\left(y, p\left(y\right)\right)\right]$$

if and only if

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

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

Definition 6 [6]: Probabilistic Sticker Languages

1. The *probabilistic molecular language* (*pSLM*) generated by a probabilistic sticker system γ is defined as

$$pSLM\left(\gamma\right) = \left\{\left(y, p(y)\right) \in WK_p(V) \times \left[0,1\right] \mid \left(x, p\left(x\right)\right) \Rightarrow^{*} \left(y, p\left(y\right)\right) \text{ for } \left(x, p(x)\right) \in A_p\right\}.$$

2. The *probabilistic sticker language* (*pSL*) generated by a probabilistic sticker system γ is defined by

$$pSL\left(\gamma\right) = \left\{\left(w, p\left(y\right)\right) \mid \left(y, p\left(y\right)\right) \in pSLM(\gamma) \text{ and } y = \begin{bmatrix} w \\ w' \end{bmatrix} \text{ where } w, w' \in V^{*}\right\}.$$

RESULTS ON PROBABILISTIC SIMPLE REGULAR STICKER SYSTEM

In this section we introduce the notion of probabilistic simple regular sticker system which is specified with a probability space and operations over probabilities closed in the probability space. We construct several examples of probabilistic simple regular sticker systems and prove some lemmas and theorems in order to show the computational power of probabilistic simple regular sticker systems.

Definition 7: Probabilistic Simple Regular Sticker System

A probabilistic sticker system $\gamma = \left(V, \rho, A_p, D_p\right)$ is called *simple regular* (*pSRSS*) if for all pairs $\left[(u, p(u)), (v, p(v))\right] \in D_p$, we have $\left(u, p(u)\right), \left(v, p(v)\right) \in \begin{pmatrix} \# \\ V \end{pmatrix}^{*} \times \left[0,1\right]$.

Definition 8: Probabilistic Simple Regular Sticker Language

The languages generated by a probabilistic simple regular sticker system γ are known as *probabilistic simple regular sticker language* (*pSRSL*) and is defined as

$$pSRSL\left(\gamma\right) = \left\{\left(y, p(y)\right) \in WK_p(V) \times \left[0,1\right] \mid \left(x, p\left(x\right)\right) \Rightarrow^{*} \left(y, p\left(y\right)\right) \text{ for } \left(x, p(x)\right) \in A_p\right\}.$$

The probability extension for regular sticker systems can be used as selection of special subsets of the languages generated by the regular sticker systems according to some thresholds (cut-points) which are sub-segments, discrete subsets of $[0,1]$ and real numbers in $[0,1]$. We define the following two types of *threshold languages* with respect to thresholds $\alpha \in [0,1]$ and $\beta \subseteq [0,1]$:

$$pSRSL(\gamma, *\alpha) = \{ w \mid (w, p(y)) \in pSRSL(\gamma) \text{ and } p(y) * \alpha \},$$

$$pSRSL(\gamma, \bullet\beta) = \{ w \mid (w, p(y)) \in pSRSL(\gamma) \text{ and } p(y) \bullet \beta \},$$

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

We denote the family of all languages generated by simple regular sticker systems and the family of all threshold languages generated by probabilistic simple regular sticker systems by $SRSL$ and $pSRSL$ respectively. The following lemma immediately follows from the definitions above.

Lemma 1 $SRSL \subseteq pSRSL$.

Proof

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

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

Let $\gamma' = (V, \rho, A_p, D_p)$ be a probabilistic simple regular 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 $p(\theta_i) = \frac{1}{m}$ 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 simple regular sticker system γ':

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

We define the threshold language generated by γ' as $pSRSL(\gamma', > 0)$, thus we obtain that

$$SRSL(\gamma) = pSRSL(\gamma', > 0).$$

□

Next, two examples are illustrated to explain the computation of probabilistic simple regular sticker system.

Example 1 Given a probabilistic simple regular sticker system $\gamma_1 = \{V, \rho, A_p, D_p\}$ where

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

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

$$A_p = \left\{ \left\{ \begin{pmatrix} aabb \\ \# \end{pmatrix}, \left(\frac{2}{10} \right) \right\} \right\},$$

$$D_p = \left\{ \left(\begin{pmatrix} \# \\ aa \end{pmatrix}, \frac{3}{10} \right), \left(\begin{pmatrix} \# \\ bb \end{pmatrix}, \frac{5}{10} \right) \right\}.$$

Then,

$$pSRSLM(\gamma) = \left\{ \left(\begin{pmatrix} aa \\ aa \end{pmatrix}^n \begin{pmatrix} bb \\ bb \end{pmatrix}^n, \left(\frac{2}{10} \right) \left(\frac{3}{10} \right)^{2n-1} \left(\frac{5}{10} \right)^{2n-1} \right) \middle| n \geq 1 \right\}.$$

Using the threshold properties, we can conclude that

i. $pSRSL(\gamma_1, = 0) = \varnothing \in \mathbf{REG}$;

ii. $pSRSL(\gamma_1, > 0) = SRSL(\gamma) \in \mathbf{REG}$;

iii. $pSRSL\left(\gamma_1, = \left(\frac{2}{10} \right) \left(\frac{3}{10} \right)^{2n-1} \left(\frac{5}{10} \right)^{2n-1} \right) = \left\{ a^{2n} b^{2n} \middle| n \geq 1 \right\} \in \mathbf{CF} - \mathbf{REG}$.

Example 2 Given a probabilistic simple regular sticker system $\gamma_2 = \left\{ V, \rho, A_p, D_p \right\}$ where

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

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

$$A_p = \left\{ \left\{ \begin{pmatrix} aab \\ \# \end{pmatrix}, \left(\frac{2}{28} \right) \right\}, \left\{ \begin{pmatrix} bcc \\ \# \end{pmatrix}, \left(\frac{3}{28} \right) \right\} \right\},$$

$$D_p = \left[\left(\begin{pmatrix} \# \\ aa \end{pmatrix}, \frac{5}{28} \right), \left(\begin{pmatrix} \# \\ b \end{pmatrix}, \frac{7}{28} \right), \left(\begin{pmatrix} \# \\ cc \end{pmatrix}, \frac{11}{28} \right) \right].$$

Case 1: For the string $\left\{ \begin{pmatrix} aab \\ \# \end{pmatrix}, \left(\frac{2}{28} \right) \right\}$,

It is not difficult to see that

$$pSRSLM(\gamma) = \left\{ \left(\left(\begin{pmatrix} aa \\ aa \end{pmatrix} \begin{pmatrix} aa \\ aa \end{pmatrix} \right)^{2n} \left(\begin{pmatrix} b \\ b \end{pmatrix} \begin{pmatrix} b \\ b \end{pmatrix} \right)^n, \left(\frac{2}{28} \right) \cdot \left(\frac{5}{28} \right)^{2n-1} \cdot \left(\frac{7}{41} \right)^{2n-1} \middle| n \geq 1 \right\}.$$

Case 2: For the string $\left\{\begin{pmatrix} bcc \\ \# \end{pmatrix}, \left(\frac{3}{28}\right)\right\}$,

It is not difficult to see that

$$pSRSLM(\gamma) = \left\{ \left(\begin{pmatrix} b \\ b \end{pmatrix} \begin{pmatrix} b \\ b \end{pmatrix} \right)^{n} \left(\begin{pmatrix} cc \\ cc \end{pmatrix} \begin{pmatrix} cc \\ cc \end{pmatrix} \right)^{2n}, \left(\frac{3}{28}\right) \cdot \left(\frac{7}{28}\right)^{2n-1} \cdot \left(\frac{11}{28}\right)^{2n-1} \middle| n \geq 1 \right\}.$$

By joining both strings from Case 1 & Case 2, the molecular language produced is

$$pSRSLM(\gamma_2) = \left\{ \left(\begin{pmatrix} aa \\ aa \end{pmatrix} \begin{pmatrix} aa \\ aa \end{pmatrix} \right)^{2n} \left(\begin{pmatrix} bb \\ bb \end{pmatrix} \begin{pmatrix} bb \\ bb \end{pmatrix} \right)^{2n} \left(\begin{pmatrix} cc \\ cc \end{pmatrix} \begin{pmatrix} cc \\ cc \end{pmatrix} \right)^{2n}, \left(\frac{2 \cdot 3}{28^2}\right) \cdot \left(\frac{5}{28}\right)^{2n-1} \cdot \left(\frac{7}{28}\right)^{4n-2} \cdot \left(\frac{11}{28}\right)^{2n-1} \right\}.$$

Using the threshold properties, we can conclude that

i. $pSRSL(\gamma_2, = 0) = \varnothing \in \mathbf{REG}$;
ii. $pSRSL(\gamma_2, = 0) = \varnothing \in \mathbf{REG}$;
iii. $pSRSL\left(\gamma_2, = \left(\frac{2 \cdot 3}{28^2}\right) \cdot \left(\frac{5}{28}\right)^{2n-1} \cdot \left(\frac{7}{28}\right)^{4n-2} \cdot \left(\frac{11}{28}\right)^{2n-1} \middle| n \geq 1 \right\}\right) = \left\{ a^{2n} b^{2n} c^{2n} \middle| n \geq 1 \right\} \in \mathbf{CS} - \mathbf{CF}$.

Proposition 1
For any probabilistic simple regular sticker system γ, the threshold language $L_p(\gamma = 0)$ is the empty set, i.e. $L_p(\gamma = 0) = \varnothing$.

Proposition 2
For each prolongation in a probabilistic simple regular sticker system γ, every threshold language $L_p(\gamma > \eta)$ with $\eta > 0$ is finite.

Since the erasing operation does not exist in sticker systems, the theorem below follows immediately.

Theorem 1 $pSRSL \subseteq \mathbf{CS}$.

From Lemma 1 and Example 2, we obtain the following theorem.

Theorem 2 $pSRSL - \mathbf{CF} \neq \varnothing$.

From Lemma 1 and Theorem 2, we have Theorem 3.

Theorem 3 $SRSL \subset pSRSL$

CONCLUSIONS

In this paper, we introduce probabilistic simple regular sticker system by associating probabilities to the axioms and dominoes of the simple regular sticker system. The computational power of the languages generated by probabilistic simple regular sticker systems are explored through lemma and theorems. The higher the computational power of a sticker system, the higher the capability of modeling a DNA based computer. DNA based computers are important as it runs with high speed and the capability of memory information are enormous. Besides, this research shows the strictness of the inclusion $SRSL \subset pSRSL$. However, the studies of restrictions such as primitive and bounded delay variants of probabilistic simple regular sticker systems remain open. Furthermore, the probabilistic modification in simple regular sticker system is very useful in the study of stochastic and uncertainty processes.

ACKNOWLEDGMENTS

The first author would like to thank the Ministry of Education (MOE) Malaysia for the financial funding through MyBrain15 scholarship. The second and third authors would also like to thank MOE and Research Management Center (RMC), UTM for the financial funding through Research University Fund Vote No. 08H45. The fourth author acknowledges the financial support by MOE Fundamental Research Grant Scheme FRGS13-066-0307.

REFERENCES

1. L. Adleman, *Sci. Amer* **279**(2), 54-61 (1998).
2. T. Head, *Bull. Math. Biol* **49,** 737 – 759 (1987).
3. L. Kari, G. Paun, G. Rozenberg, A. Salomaa, and S. Yu, *Acta Inform* **35**, 401-420 (1998).
4. L. Adleman, *Science* **266**, 1021-1024 (1994).
5. G. Paun, and G. Rozenberg, *Theoret. Comput. Sci* **204**, 183-203 (1998).
6. M. Selvarajoo, W.H. Fong, N.H. Sarmin, and S. Turaev, *Mal. J. Fund. Appl. Sci* **9**(3), 150-155 (2013).
7. P. Linz, *An Introduction to Formal Languages and Automata*, United States: Jones and Bartlett Publishers, 2012.
8. R. Freund, G. Paun, G. Rozenberg, and A. Salomaa, " Bidirectional Sticker System", Proceeding of Pacific Symposium on Biocomputing. Hawaii, United States, 1998, pp. 536-546.
9. A. Alhazov, and M. Ferretti, "Computing by Observing Bio-Systems" in *The Case of Sticker Systems*, Proceeding of The 10th International Workshop on DNA Computing. Milan, Italy, 2004, pp. 1-13.
10. J. Xu, Y. Dong, and X. Wei, *Chinese Sci. Bull* **49**(8), 772-780 (2004).