Jurnal Teknologi

Full paper

Probabilistic Semi-Simple Splicing System and Its Characteristics

Mathuri Selvarajoo[a*], Fong Wan Heng[b], Nor Haniza Sarmin[a], 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, Universiti Teknologi Malaysia, 81310 UTM Johor Bharu, Johor, Malaysia
[c]Department of Computer Science, Kulliyyah of Information and Communication Technology, International Islamic University Malaysia, 53100 Kuala Lumpur, Malaysia

*Corresponding author: mathuri87@yahoo.com

Article history

Received :18 March 2013
Received in revised form :
26 April 2013
Accepted :17 May 2013

Graphical abstract

$$\sigma^*(L) = \bigcup_{i\geq 0} \sigma^i(L).$$

Abstract

The concept of splicing system was first introduced by Head in 1987. This model has been introduced to investigate the recombinant behavior of DNA molecules. Splicing systems with finite sets of axioms only generate regular languages. Hence, different restrictions have been considered to increase the computational power up to the recursively enumerable languages. Recently, probabilistic splicing systems have been introduced where probabilities are initially associated with the axioms, and the probability of a generated string is computed by multiplying the probabilities of all occurrences of the initial strings in the computation of the string. In this paper, some properties of probabilistic semi-simple splicing systems, which are special types of probabilistic splicing systems, are investigated. We prove that probabilistic semi-simple splicing systems can also increase the generative power of the generated languages.

Keywords: DNA computing; probabilistic splicing systems; splicing languages; regular languages

Abstrak

Konsep sistem hiris-cantum mula diperkenalkan oleh Head pada tahun 1987. Model ini telah diperkenalkan untuk menyiasat penggabungan semula molekul-molekul DNA. Sistem hiris-cantum dengan set aksiom terhingga hanya menjana bahasa biasa. Oleh itu, batasan yang berbeza telah digunakan untuk meningkatkan kuasa pengkomputeran sehingga ke bahasa rekursif enumerable. Baru-baru ini, sistem hiris-cantum berkebarangkalian telah diperkenalkan di mana kebarangkalian dikaitkan dengan aksiom dan kebarangkalian jujukan yang dihasilkan dikira melalui pendaraban semua kebarangkalian jujukan yang digunakan. Dalam kertas kerja ini, beberapa ciri-ciri sistem hiris-cantum separuh-mudah berkebarangkalian yang merupakan salah satu jenis sistem hiris-cantum berkebarangkalian disiasat. Kami membuktikan bahawa sistem hiris-cantum separuh-mudah berkebarangkalian juga boleh meningkatkan kuasa pengkomputeran bahasa yang dihasilkan

Kata kunci: Pengkomputeran DNA; sistem hiris-cantum berkebarangkalian; bahasa hiris-cantum; bahasa biasa

1.0 INTRODUCTION

Deoxyribonucleic acid (DNA) is the genetic material of organisms in a chain of nucleotides. The nucleotides differ by their chemical bases that are adenine (A), guanine (G), cytosine (C), and thymine (T). DNA bases pair up with each other, A with T and C with G, to form units called base pairs. So, nucleotides can be arranged in two long strands that form a spiral called a double helix. The structure of the double helix is somewhat like a ladder. DNA can be represented as strings over four alphabets, i.e. D = {[A / T], [C / G], [G / C], [T / A]} [1]. Restriction enzymes, found naturally in bacteria, can cut DNA fragments at specific sequences, known as restriction sites; while another enzyme, ligase, can re-join DNA fragments that have complementary ends. This recombination behaviour of restriction enzymes and ligases was modelled in the form of splicing systems and splicing languages.

The concept of splicing system was first introduced by Head in 1987 [2]. This model has been defined to investigate the recombinant behavior of DNA molecules in the presence of restriction enzymes and ligases. In splicing system, a DNA molecule is coded into a string over the alphabets. With some strings over the alphabet as the initial strings (axioms) and some splicing rules, a language can be produced by the splicing system.

Later, various types of splicing languages were defined and studied by different mathematicians. Since splicing systems with finite sets of axioms and rules generate only regular languages [3], several restrictions in the use of rules have been considered, which increase the computational power of the languages generated up to the recursively enumerable languages. This is important from the point of view in DNA computing: splicing systems with restrictions can be considered as theoretical models of universal programmable DNA based computers. Different problems appearing in the area of computer science motivate humans to consider suitable models for the solution of the problems.

In this research, we consider probabilistic splicing systems to introduce a new variant of splicing system [4], called probabilistic semi-simple splicing systems. In such system, 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 operation * defined on the probabilities, i.e. $p(z) = p(x) * p(y)$. Then the language generated by a probabilistic semi-simple splicing system consists of all strings generated by the semi-simple splicing systems whose probabilities are greater than (or smaller than, or equal to) some previously chosen cut-points.

This paper is organized as follows. Section 2 contains some necessary definitions from formal language theory, DNA computing and probabilistic splicing systems. The concept of probabilistic semi-simple splicing systems are introduced in Section 3. In Section 3, we also establish some basic results concerning the generative power of probabilistic semi-simple splicing systems. In Section 4, we indicate some possible topics for future research in this direction.

2.0 PRELIMINARIES

In this section, the main concepts and notations that will be used in this paper are introduced. The theoretical basis of splicing system is under the framework of formal language theory that is mainly the study of finite sets of strings called languages.

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 $\subset$. $\emptyset$ denotes the empty set. The sets of integers, positive rational numbers and real numbers are denoted by $\mathbb{Z}$, $\mathbb{Q}_+$ and $\mathbb{R}$, respectively. The cardinality of a set X is denoted by $|X|$.

Definiton 1. [5]

A finite, nonempty set A of symbols is called an *alphabet*. Any finite sequence of symbols from an alphabet is called a *string*. We use 1 to denote the *empty string* which is a string with no symbols at all.

If A is an alphabet, we use A^* to denote the set of strings obtained by concatenating zero or more symbols from A.

Definition 2. [5]

A formal language L over an alphabet Σ is a subset of Σ^*, that is, a set of words over that alphabet.

The families of languages generated by phrase structure, context-sensitive, context-free, linear and regular grammars are denoted by RE, CS, CF, LIN, and REG respectively. Further we denote the family of finite languages by FIN. The next strict inclusions, named Chomsky hierarchy, holds:

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

Definition 3. [2] Splicing System

Let V be an alphabet, and $\#, \$ \notin V$ two special symbols. A splicing rule over V is a string of the form

$$r = u_1\#u_2\$u_3\#u_4, \text{ where } u_i \in V^*, 1 \le i \le 4.$$

For such a rule r and strings $x, y, z \in V^*$, we write

$$(x, y) \vdash_r z \text{ iff } x = x_1u_1u_2x_2, y = y_1u_3u_4y_2, \text{ and } z = x_1u_1u_4y_2,$$

for some $x_1, x_2, y_1, y_2 \in V^*$.

We say that z is obtained by splicing x and y as indicated by the rule r; u_1u_2 and u_3u_4 are called the *sites* of the splicing. We call x the first term and y the second term of the splicing operation. When understood from the context, we omit the specification of r and we write $\vdash$ instead of $\vdash_r$.

An H scheme is a pair $\sigma = (V, R)$ where V is an alphabet and $R \subseteq V^*\#V^*\$V^*\#V^*$ is a set of splicing rules.

For a given H scheme $\sigma = (V, R)$ and a language $L \subseteq V^*$, we define

$$\sigma(L) = \{z \in V^* | (x, y) \vdash_r z, \text{ for some } x, y \in L, \ r \in R\},$$
$$\sigma^0(L) = L,$$
$$\sigma^{i+1}(L) = \sigma^i(L) \cup \sigma\left(\sigma^i(L)\right), i \ge 0,$$
$$\sigma^*(L) = \bigcup_{i \ge 0} \sigma^i(L).$$

An extended H system is a construct $\gamma = (V, T, A, R)$ where V is an alphabet, $T \subseteq V$ is the terminal alphabet, $A \subseteq V^*$ is the set of axioms, and $R \subseteq V^*\#V^*\$V^*\#V^*$ is the set of splicing rules. When $T = V$, the system is said to be non-extended. The language generated by γ is defined by

$$L(\gamma) = \sigma^*(A) \cap T^*.$$

Here, EH(F_1, F_2) denotes the family of languages generated by extended H systems $\gamma = (V, T, A, R)$ with $A \in F_1$ and $R \in F_2$ where $(F_1, F_2) \in$ {FIN, REG, CF, LIN, CS, RE}.

Theorem 1 [2]

The relations in the following table hold, where at the intersection of the row marked with F_1 with the column marked with F_2 there appear either the family EH(F_1, F_2) or two families F_3, F_4 such that $F_3 \subset$ EH$(F_1, F_2) \subseteq F_4$.

Table 1 The family of languages generated by $\mathbf{F_1}$ and $\mathbf{F_2}$.

F_1 \ F_2	FIN	REG	CF	LIN	CS	RE
FIN	REG	RE	RE	RE	RE	RE
REG	REG	RE	RE	RE	RE	RE
CF	LIN,CF	RE	RE	RE	RE	RE
LIN	CF	RE	RE	RE	RE	RE
CS	RE	RE	RE	RE	RE	RE
RE	RE	RE	RE	RE	RE	RE

Definition 4. [6]

A *semi- simple H (splicing) system* is a triple

$$G = (V, M, A),$$

where V is an alphabet, $M \subseteq V$, and A is a finite language over V. The elements of M are called *markers* and those of A are called *axioms.*

Definition 5. [4]

A *probabilistic H (splicing) system* is a 5-tuple $\gamma = (V, T, A, R, p)$ where V, T, R are defined as for a usual extended H system, $p: V^* \longrightarrow [0,1]$ is a probability function, and A is a finite subset of $V^+ \times [0,1]$ such that

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

Definition 6. [7]

We consider as thresholds (cut-points) sub-segments and discrete subsets of [0, 1] as well as real numbers in [0,1]. We define the following two types of threshold languages with respect to thresholds $\Omega \subseteq [0,1]$ and $\omega \in [0,1]$:

$$L_p(\gamma, * \omega) = \{z \in T^* | (z, p(z)) \in \sigma^*(A) \wedge p(z) * \omega\},$$

$$L_p(\gamma, \star \Omega) = \{z \in T^* | (z, p(z)) \in \sigma^*(A) \wedge p(z) \star \Omega\},$$

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

3.0 RESULTS AND DISCUSSION

In this section we introduce the notion of probabilistic semi-simple splicing systems which is specified with a probability space and operations over probabilities closed in the probability space.

Definition 7.

A *probabilistic semi-simple splicing system* $(pSSEH)$ is a 4-tuple $\gamma = (V, A, R, p)$ where *V* is defined as for a usual extended H system, *R* is the rule in the form $(a, 1; b, 1)$ for $a, b \in A$, *p* is a probabilistic function defined by $p: V^* \rightarrow [0,1]$, and *A* is a subset of $V^* \times [0,1]$ such that

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

Further we define a probabilistic semi-simple splicing operation and the language generated by a probabilistic semi-simple splicing system.

Definition 8.

For strings $(x, p(x)), (y, p(y)), (z, p(z)) \in V^* \times [0,1]$, and $r \in R$, $[(x, p(x)), (y, p(y))] \vdash_r (z, p(z))$, if and only if $(x, y) \vdash_r z$ and $p(z) = p(x) * p(y)$ and $r = (a, 1; b, 1) \in R$.

Definition 9.

The language generated by the probabilistic semi-simple splicing system γ is defined as

$$L(\gamma) = \{ z \in T^* \mid (z, p(z)) \in \sigma^*(A)\}.$$

Remark 1. We should mention that splicing operations may result in the same string with different probabilities. Since in this paper, we focus on strings whose probabilities satisfy some threshold requirements, i.e., the probabilities are merely used for the selection of some strings, this 'ambiguity' does not effect on the selection. When we investigate the properties connected with the probabilities of the strings, we can define another operation together with the multiplication of the strings, for instance, the addition over the probabilities of the same strings, which removes the ambiguity problem.

We denote the family of languages generated by multiplicative probabilistic semi-simple splicing system of type (F_1, F_2) by $pSSEH(F_1, F_2)$ where

$$F_1, F_2 \in \{FIN, REG, CF, LIN, CS, RE\}.$$

Remark 2. In this paper we focus on probabilistic semi-simple splicing systems with finite set of axioms, since we consider a finite initial distribution of probabilities over the set of axioms. Moreover, it is natural in practical point of view: only splicing systems with finite components can be chosen as a theoretical model for DNA based computation devices. Thus, we use the simplified notation $pSSEH(F)$ of the language family generated by probabilistic semi-simple splicing systems with finite set of axioms instead of $pSSEH(F_1, F_2)$ where $F \in \{FIN, REG, CF, LIN, CS, RE\}$ shows the family of languages for splicing rules.

From the definition, the next lemma follows immediately.

Lemma 1

$SSEH(FIN, F) \subseteq pSSEH(F)$ for all families $F \in \{FIN, REG, CF, LIN, CS, RE\}$.

Proof.

Let $G = (V, A, R)$ be a semi-simple splicing system generating the language $L(G) \in SSEH(FIN, F)$ where $F \in \{FIN, REG, CF, LIN, CS, RE\}$.

Let $A = \{x_1, x_2, \dots, x_n\}, n \geq 1$. We define a probabilistic semi-simple splicing system $G' = (V, A', R, p)$ where the set of axioms is defined by $A' = \{(x_i, p(x_i)) | \ x_i \in A, 1 \leq i \leq n\}$ where $p(x_i) = \frac{1}{n}$ for all $1 \leq i \leq n$, then

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

We define the threshold language generated by G' as $L_p(G', > 0)$, then it is not difficult to see that

$$L(G) = L_p(G', > 0).$$

Next, two examples are given to illustrate the application of probability to the semi-simple splicing system.

Example 1 : Consider the semi-simple splicing system

$$G_1 = \left(\{a, b, c\}, \{a, b, c\}, \{aca, aba, baca, caba\}, \left\{\frac{2}{17}, \frac{3}{17}, \frac{5}{17}, \frac{7}{17}\right\}\right).$$

We obtain

$L(G_1, \bar{\eta}) = \left\{ ac^n b^n a ,\ \left(\frac{6}{289}\right)\left(\frac{35}{289}\right)^{n-1} \middle| \ n \geq 1\right\}$, where

$\bar{\eta} = \left(\frac{6}{289}\right)\left(\frac{35}{289}\right)^{n-1}$.

The way to obtain the string is by performing the splicing operation using the markers to the axioms.

Case 1 : Using strings aca & $baca$

i : for the string $aca, p(aca) = \frac{2}{17}$ and using marker *c*,
$[(ac|a, \frac{2}{17})] \vdash_c [(ac), \frac{2}{17}]$,

ii : for the string $baca, p(baca) = \frac{5}{17}$ and using marker *a*,
$[(ba|ca), \frac{5}{17}] \vdash_a [(ca), \frac{5}{17}]$,

iii : for the both strings $aca, (p(aca) = \frac{2}{17})$ and $baca, (p(baca) = \frac{5}{17})$ and using the markers *a* and *c*,

$$\left[ac\middle|a,\left(\frac{2}{17}\right), ba\middle|ca,\left(\frac{5}{17}\right)\right] \vdash_{c,a} \left[(acca),\left(\frac{2}{17}\right)\left(\frac{5}{17}\right)\right],$$

iv : for the strings from (iii) and $baca$ i.e. $acca, (p(acca) = \left(\frac{2}{17}\right)\left(\frac{5}{17}\right))$ and $baca, (p(baca) = \frac{5}{17})$ and using the same markers *a* and *c*,

$$\left[\left(acc|a,\left(\frac{2}{17}\right)\left(\frac{5}{17}\right)\right),\left(ba\,|\,ca,\left(\frac{5}{17}\right)\right)\right] \vdash_{c,a} \left(ac^3a, \left(\frac{2}{17}\right)\left(\frac{5}{17}\right)^2\right),$$

v :for each new string produced $ac^{n-1}a, (p(ac^{n-1}a) = \left(\frac{2}{17}\right)\left(\frac{5}{17}\right)^{n-2})$ and string (ii) $baca, (p(baca) = \frac{5}{17})$ and using the same markers *a* and *c*,

$$\left[\left(ac^{n-1}|a,\left(\frac{2}{17}\right)\left(\frac{5}{17}\right)^{n-2}\right),\left(ba\,|\,ca,\left(\frac{5}{17}\right)\right)\right]$$
$$\vdash_{c,a} \left(ac^na, \left(\frac{2}{17}\right)\left(\frac{5}{17}\right)^{n-1}\right).$$

Case 2 : Using strings aba & $caba$

i : for the string $aba, p(aba) = \frac{3}{17}$ and using marker *b*,
$[(ab|a, \frac{3}{17})] \vdash_b [(ab), \frac{3}{17}]$,

ii : for the string $caba, p(caba) = \frac{7}{17}$ and using marker *a*,
$[(ca|ba), \frac{7}{17}] \vdash_a [(ba), \frac{7}{17}]$,

iii : for the both strings $aba, (p(aba) = \frac{3}{17})$ and $caba, (p(caba) = \frac{7}{17})$ and using the markers *a* and *b*,

$$\left[ab\middle|a,\left(\frac{3}{17}\right), ca\middle|ba,\left(\frac{7}{17}\right)\right] \vdash_{b,a} \left[(abba),\left(\frac{3}{17}\right)\left(\frac{7}{17}\right)\right],$$

iv : for the strings from (iii) and $caba$ i.e., $abba, (p(abba) = \left(\frac{3}{17}\right)\left(\frac{7}{17}\right))$ and $caba, (p(caba) = \frac{7}{17})$ and using the same markers *a* and *c*,

$$\left[\left(abb|a,\left(\frac{3}{17}\right)\left(\frac{7}{17}\right)\right),\left(ca\,|\,ba,\left(\frac{7}{17}\right)\right)\right] \vdash_{b,a} \left(ab^3a, \left(\frac{3}{17}\right)\left(\frac{7}{17}\right)^2\right),$$

v : for each new string produced $ab^{n-1}a, (p(ab^{n-1}a) = \left(\frac{3}{17}\right)\left(\frac{7}{17}\right)^{n-2})$ and string (ii) $caba, (p(caba) = \frac{7}{17})$ and using the same markers *a* and *b*,

$$\left[\left(ab^{n-1}|a,\left(\frac{3}{17}\right)\left(\frac{7}{17}\right)^{n-2}\right),\left(ca\,|\,ba,\left(\frac{7}{17}\right)\right)\right]$$
$$\vdash_{b,a} \left(ab^na, \left(\frac{3}{17}\right)\left(\frac{7}{17}\right)^{n-1}\right).$$

For the strings from Case 1 $\left(ac^na, \left(\frac{2}{17}\right)\left(\frac{5}{17}\right)^{n-1}\right)$ and Case 2 $\left(ab^na, \left(\frac{3}{17}\right)\left(\frac{7}{17}\right)^{n-1}\right)$ using marker *a*,

$$\left[\left(ac^n|a, \left(\frac{2}{17}\right)\left(\frac{5}{17}\right)^{n-1}\right),\left(a\,|\,b^na, \left(\frac{3}{17}\right)\left(\frac{7}{17}\right)^{n-1}\right)\right]$$
$$\vdash_a \left[ac^nb^na, \left(\frac{6}{289}\right)\left(\frac{35}{289}\right)^{n-1}\right].$$

Therefore,

$$L(G_1, p_1) = \left\{\left(ac^kb^ma, \left(\frac{6}{289}\right)\left(\frac{5}{17}\right)^{k-1}\left(\frac{7}{17}\right)^{m-1}\right| k, m \geq 1\right\},$$
$$p_1 = \left(\frac{6}{289}\right)\left(\frac{5}{17}\right)^{k-1}\left(\frac{7}{17}\right)^{m-1}.$$

Using the threshold properties, we can conclude the following:

i : $\eta = 0, \Rightarrow L(G_1, = 0) = \emptyset \in REG$,
ii : $\eta > 0, \Rightarrow L(G_1, > 0) = L(\gamma_1) \in REG$,
iii : $\bar{\eta} = \{\left(\frac{6}{289}\right)\left(\frac{35}{289}\right)^{n-1} | n \geq 1\}, \Rightarrow$
$L(G_1, \bar{\eta}) = \{ac^nb^na \,|n \geq 1\} \in CF - REG$,
iv : $\bar{\bar{\eta}} \neq \{\left(\frac{6}{289}\right)\left(\frac{35}{289}\right)^{n-1} | n \geq 1\}, \Rightarrow$
$L(G_1, \bar{\bar{\eta}}) = \{ac^kb^ma \,|k > m \geq 1\} \cup$
$\{ac^kb^ma \,|m > k \geq 1\} \in CF - REG$.

Example 2: Consider the semi-simple splicing system

$$G_2 = (\{a,b,c,d\},\{a,b,c,d\},\{aba,aca,ada,baca,caba,bada\},$$
$$\{\frac{2}{41},\frac{3}{41},\frac{5}{41},\frac{7}{41},\frac{11}{41},\frac{13}{41}\}).$$

We obtain

$L(G_2, \bar{\eta}) = \left\{ac^nb^nd^na, \left(\frac{2.3.5}{41^3}\right)\left(\frac{7.11.13}{41^3}\right)^{n-1}\right| n \geq 1\}$, where $\bar{\eta} = \left(\frac{2.3.5}{41^3}\right)\left(\frac{7.11.13}{41^3}\right)^{n-1}$.

The way to obtain the string is by performing the splicing operation using the markers to the axioms.

Case 1 : Using strings aca & $baca$

i : for the string $aca, (p(aca) = \frac{3}{41})$ and using marker *c*,
$[(ac|a, \frac{3}{41})] \vdash_c [(ac), \frac{3}{41}]$,

ii : for the string $baca, \left(p(baca) = \frac{7}{41}\right)$ and using marker *a*,
$[(ba|ca), \frac{7}{41}] \vdash_a [(ca), \frac{7}{41}]$,

iii : for the both strings $aca, (p(aca) = \frac{3}{41})$ and $baca, \left(p(baca) = \frac{7}{41}\right)$ and using the markers *a* and *c*,

$$\left[ac\middle|a,\left(\frac{3}{41}\right),ba\middle|ca,\left(\frac{7}{41}\right)\right]\vdash_{c,a}\left[(acca),\left(\frac{3}{41}\right)\left(\frac{7}{41}\right)\right],$$

iv : for the strings from (iii) and $baca$, i.e., $acca, (p(acca) = \left(\frac{3}{41}\right)\left(\frac{7}{41}\right))$ and $baca, \left(p(baca) = \frac{7}{41}\right)$ and using the same markers a and c, $\left[\left(acc|a,\left(\frac{3}{41}\right)\left(\frac{7}{41}\right)\right),\left(ba\,|\,ca,\left(\frac{7}{41}\right)\right)\right]\vdash_{c,a}\left(ac^3a\,,\left(\frac{3}{41}\right)\left(\frac{7}{41}\right)^2\right),$

v : for each new string produced $ac^{n-1}a$, $(p(ac^{n-1}a) = \left(\frac{3}{41}\right)\left(\frac{7}{41}\right)^{n-2})$ and string (ii) $baca, (p(baca) = \frac{7}{41})$ and using the same markers a and c,

$$\left[\left(ac^{n-1}|a,\left(\frac{3}{41}\right)\left(\frac{7}{41}\right)^{n-2}\right),\left(ba\,|\,ca,\left(\frac{7}{41}\right)\right)\right]$$

$$\vdash_{c,a}\left(ac^na\,,\left(\frac{3}{41}\right)\left(\frac{7}{41}\right)^{n-1}\right).$$

Case 2 : Using strings aba & $caba$

i : for the string $aba, (p(aba) = \frac{2}{41})$ and using marker b, $\left[\left(ab|a\,,\,\frac{2}{41}\right)\right]\vdash_b\left[(ab),\frac{2}{41}\right],$

ii : for the string $caba, (\,p(caba) = \frac{11}{41})$ and using marker a, $\left[(ca|ba),\,\frac{11}{41}\right]\vdash_a\left[(ba),\frac{11}{41}\right],$

iii : for the both strings $aba, (p(aba) = \frac{2}{41})$ and $caba, (\,p(caba) = \frac{11}{41})$ and using the markers a and b,

$$\left[ab\middle|a,\left(\frac{2}{41}\right),ca\middle|ba,\left(\frac{11}{41}\right)\right]\vdash_{b,a}\left[(abba),\left(\frac{2}{41}\right)\left(\frac{11}{41}\right)\right],$$

iv : for the string from (iii) and $caba$, i.e. $abba, (p(abba) = \left(\frac{2}{41}\right)\left(\frac{11}{41}\right))$ and $caba, (\,p(caba) = \frac{11}{41})$ and using the same markers a and b,

$$\left[\left(abb|a,\left(\frac{2}{41}\right)\left(\frac{11}{41}\right)\right),\left(ca\,|\,ba,\left(\frac{11}{41}\right)\right)\right]\vdash_{b,a}\left(ab^3a\,,\left(\frac{2}{41}\right)\left(\frac{11}{41}\right)^2\right),$$

v : for each new string produced $ab^{n-1}a, (p(ab^{n-1}a) = \left(\frac{2}{41}\right)\left(\frac{11}{41}\right)^{n-2})$ and string (ii) $caba, (\,p(caba) = \frac{11}{41})$ and using the same markers a and b,

$$\left[\left(ab^{n-1}|a,\left(\frac{2}{41}\right)\left(\frac{11}{41}\right)^{n-2}\right),\left(ca\,|\,ba,\left(\frac{11}{41}\right)\right)\right]$$

$$\vdash_{b,a}\left(ab^na\,,\left(\frac{2}{41}\right)\left(\frac{11}{41}\right)^{n-1}\right).$$

Case 3 : Using strings ada & $bada$

i : for the string $ada, \left(p(ada) = \frac{5}{41}\right)$ and using marker d, $\left[\left(ad|a\,,\,\frac{5}{41}\right)\right]\vdash_d\left[(ad),\frac{5}{41}\right],$

ii : for the string $bada, (\,p(bada) = \frac{13}{41})$ and using marker a, $\left[(ba|da),\,\frac{13}{41}\right]\vdash_a\left[(da),\frac{13}{41}\right],$

iii : for the both strings $ada, \left(p(ada) = \frac{5}{41}\right)$ and $bada, (\,p(bada) = \frac{13}{41})$ and using the markers a and d,

$$\left[ad\middle|a,\left(\frac{5}{41}\right),ba\middle|da,\left(\frac{13}{41}\right)\right]\vdash_{d,a}\left[(adda),\left(\frac{5}{41}\right)\left(\frac{13}{41}\right)\right],$$

iv : for the strings from (iii) and $bada$, i.e., $adda, (p(adda) = \left(\frac{5}{41}\right)\left(\frac{13}{41}\right))$ and $bada, (\,p(bada) = \frac{13}{41})$ and using the same markers a and d, $\left[\left(add|a,\left(\frac{5}{41}\right)\left(\frac{13}{41}\right)\right),\left(ba\,|\,da,\left(\frac{13}{41}\right)\right)\right]\vdash_{d,a}\left[ad^3a\,,\left(\frac{5}{41}\right)\left(\frac{13}{41}\right)^2\right],$

v : for each new string produced $ad^{n-1}a, (p(ad^{n-1}a) = \left(\frac{5}{41}\right)\left(\frac{13}{41}\right)^{n-2})$ and string (ii) $bada, (\,p(bada) = \frac{13}{41})$ and using the same markers a and d,

$$\left[\left(ad^{n-1}|a,\left(\frac{5}{41}\right)\left(\frac{13}{41}\right)^{n-2}\right),\left(ba\,|\,da,\left(\frac{13}{41}\right)\right)\right]$$

$$\vdash_{d,a}\left[ad^na\,,\left(\frac{5}{41}\right)\left(\frac{13}{41}\right)^{n-1}\right].$$

For the strings from Case 1 $\left(ac^na\,,\left(\frac{3}{41}\right)\left(\frac{7}{41}\right)^{n-1}\right)$ and Case 2 $\left(ab^na\,,\left(\frac{2}{41}\right)\left(\frac{11}{41}\right)^{n-1}\right)$ using marker a,

$$\left[\left(ac^n|a\,,\left(\frac{3}{41}\right)\left(\frac{7}{41}\right)^{n-1}\right),\left(a\,|\,b^na\,,\left(\frac{2}{41}\right)\left(\frac{11}{41}\right)^{n-1}\right)\right]$$

$$\vdash_a\left[ac^nb^na\,,\left(\frac{2.3}{41^2}\right)\left(\frac{7.11}{41^2}\right)^{n-1}\right].$$

For the strings resulted from Case 1 and Case 2 $\left[ac^nb^na\,,\left(\frac{2.3}{41^2}\right)\left(\frac{7.11}{41^2}\right)^{n-1}\right]$ and Case 3 $\left[ad^na\,,\left(\frac{5}{41}\right)\left(\frac{13}{41}\right)^{n-1}\right]$ and using marker a,

$$\left[\left(ac^nb^n\,|\,a\,,\left(\frac{2.3}{41^2}\right)\left(\frac{7.11}{41^2}\right)^{n-1}\right),\left(a\,|\,d^na\,,\left(\frac{5}{41}\right)\left(\frac{13}{41}\right)^{n-1}\right)\right]\vdash_a$$

$$\left[ac^nb^nd^na\,,\left(\frac{2.3.5}{41^3}\right)\left(\frac{7.11.13}{41^3}\right)^{n-1}\right].$$

Therefore,

$$L(G_2,p_2) = \left\{\left(ac^kb^md^na\,,\left(\frac{2.3.5}{41^3}\right)\left(\frac{7}{41}\right)^{k-1}\left(\frac{11}{41}\right)^{m-1}\left(\frac{13}{41}\right)^{n-1}\right|k,m\geq 1\right\}.$$

$$p_2 = \left(\frac{2.3.5}{41^3}\right)\left(\frac{7}{41}\right)^{k-1}\left(\frac{11}{41}\right)^{m-1}\left(\frac{13}{41}\right)^{n-1}.$$

Using the threshold properties, we can conclude the following:

i : $\eta = 0, \Rightarrow L(G_2, = 0) = \emptyset \in REG,$
ii : $\eta > 0, \Rightarrow L(G_2, > 0) = L(G_2) \in REG,$
iii: $\bar{\eta} = \{\left(\frac{2.3.5}{41^2}\right)\left(\frac{7.11.13}{41^2}\right)^{n-1} \mid n \geq 1\}, \Rightarrow$
$L(G_2, \bar{\eta}) = \{ac^n b^n d^n a \mid n \geq 1\} \in CS - REG,$
iv : $\bar{\bar{\eta}} \neq \{\left(\frac{2.3.5}{41^2}\right)\left(\frac{7.11.13}{41^2}\right)^{n-1} \mid n \geq 1\}, \Rightarrow$
$L(G_2, \bar{\bar{\eta}}) = \{ac^k b^m d^n a \mid k > m > n \geq 1\} \cup$
$\{ac^k b^m d^n a \mid k > n > m \geq 1\} \cup$
$\{ac^k b^m d^n a \mid m > k > n \geq 1\} \cup$
$\{ac^k b^m d^n a \mid m > n > k \geq 1\} \cup$
$\{ac^k b^m d^n a \mid n > k > m \geq 1\} \cup$
$\{ac^k b^m d^n a \mid n > m > k \geq 1\} \in CS - REG.$

The examples above illustrate that the use of thresholds with probabilistic semi-simple splicing systems increase the generative power of splicing systems with finite components.

We should also mention two simple but interesting facts of probabilistic semi-simple splicing systems. First as Proposition 1 and second as Proposition 2, as stated in the following:

Proposition 1

For any probabilistic semi-simple splicing system (*G*), the threshold language $L(G, = 0)$is the empty set, i.e. $L(G, = 0) = \emptyset$.

Proposition 2

If for each splicing rule r in a probabilistic semi-simple splicing system (*G*), $p(r) < 1$, then every threshold language $L(G, > \eta)$ with $\eta > 0$ is finite.

From Theorem 1, Lemma 1 and Examples 1, 2, we obtain the following two theorems.

Theorem 2

$REG \subset pSSEH(FIN) \subseteq pSSEH(F) = RE$ where
$F \in \{REG, CF, LIN, CS, RE\}.$

Theorem 3

$pSSEH(FIN) - CF \neq \emptyset.$

4.0 CONCLUSION

In this paper we introduced probabilistic semi-simple splicing systems by associating probabilities with strings and also establishing some basic but important facts of probabilistic semi-simple splicing systems. We showed that an extension of semi-simple splicing systems with probabilities increases the generative power of semi-simple splicing systems with finite components. In particular cases, probabilistic semi-simple splicing systems can generate non context-free languages. The problem of strictness of the second inclusion in Theorem 2 and the incomparability of the family of context-free languages with the family of languages generated by probabilistic semi-simple splicing systems with finite components (the inverse inequality of that in Theorem 3) remain open.

Acknowledgement

The first author would like to thank the Malaysian Ministry of Higher Education for the financial funding support through MyBrain15 scholarship. The second and third authors would also like to thank the Malaysian Ministry of Higher Education (MOHE) and Research Management Center (RMC), UTM for their financial funding through Research University Fund Vote No. 07J41.

References

[1] L. Adleman. 1994. *Science*. 266: 1021–1024.
[2] T. Head. 1987. *Bull. Math.Biology*. 49: 737–759
[3] D. Pixton. 1996. *Discrete Applied Mathematics*. 69: 101–124.
[4] Sherzod Turaev, Mathuri Selvarajoo, Fong Wan Heng, Nor Haniza Sarmin. 2013. *Advanced Methods for Computational Collective Intelligence*. New York: Springer. 457: 259–268.
[5] G. Paun, G. Rozenberg, A. Salomaa. 1998. *Handbook of Formal Languages: Vol.1. Word, language, grammar*. *Springer-Verlag*.
[6] E. Goode, D. Pixton. 1996. *Discrete Applied Math*. 72: 96–107.
[7] T. Fowler. 2011. *The Generative Power of Probabilistic and Weighted Context-Free Grammars. Springer-Verlag*. 68: 57–71.