

Weighted simple and semi-simple splicing systems

Yee Siang Gan^{1*}, Wan Heng Fong², Nor Haniza Sarmin¹, Sherzod Turaev³

¹Department of Mathematical Sciences, Faculty of Science, UTM, 81310 UTM Skudai, Johor, Malaysia.

²Ibnu Sina Institute for Fundamental Science Studies, Universiti Teknologi Malaysia, 81310, UTM Johor Bahru, Johor, Malaysia.

³Department of Computer Science, Kulliyah of Information and Communication Technology, International Islamic University of Malaysia, 50728 Kuala Lumpur.

*Corresponding Author: ysgn88@gmail.com

Article history :

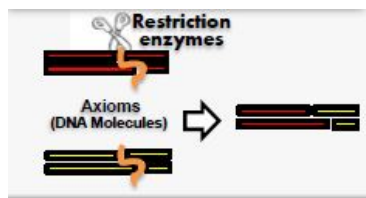
Received 5 July 2014

Revised 20 September 2014

Accepted 12 October 2014

Available online 25 November 2014

GRAPHICAL ABSTRACT



ABSTRACT

The modelling of splicing system has been introduced theoretically by Head in 1987. As time goes on, various splicing systems have been developed, such as one-sided, simple and semi-simple splicing systems. However, in the investigation on the generative power of splicing system, there are limitations on the generative power of splicing system with finite components. In order to overcome the limitation of the usual splicing system, one variant of splicing system has been introduced recently, called the weighted splicing system. In this paper, we associate weights from selected weighting spaces to the axioms of simple and semi-simple splicing systems, thus introducing weighted simple splicing system and weighted semi-simple splicing system. Some examples are presented for weighted simple and semi-simple splicing systems to illustrate their generative power. Lastly, relation of the languages generated by weighted simple and semi-simple splicing systems in the Chomsky hierarchy are also investigated.

Keywords: weights, simple, semi-simple, splicing system, generative power

© 2014 Penerbit UTM Press. All rights reserved
<http://dx.doi.org/10.11113/mjfas.v10n4.324>

1. INTRODUCTION

A *splicing system*, one of the earliest mathematical models for DNA computation [1], uses a *splicing operation* which operates as follows: starting from two well-formed double stranded sequences as axioms in the splicing system, the splicing operation cuts the strings at specific subsequences. Then, first part of one sequence is connected to the second part of the other sequence, and vice versa. Hence, new well-formed double stranded strings can be obtained from the iterative process of splicing [2].

It was shown that splicing systems with finite sets of axioms and rules generate only the regular languages [3]. To increase the generative power of the usual splicing systems, many restrictions have been introduced. Several researches show that splicing systems with finite components and additional control are computationally complete, i.e., Turing equivalent (see [2, 4, 5]). In our recent work [6], we introduced weighted splicing systems – a new variant of splicing systems imposing weights to strings generated by a splicing system and selecting with respect to thresholds. Our research is motivated from the idea of using weights as restrictions in formal languages theory such as weighted grammars and automata (see [7–12]), which have broad applications in computer science related areas.

In a weighted splicing system, weights from weighting spaces (sets of numbers, matrices and other discrete structures) are initially associated to the axioms of the splicing system. Then the weight of a generated string is computed by the weights of the generating strings by the weight operation. Some subsets of thresholds (cut-points) are considered in the languages generated by weighted splicing systems to select some splicing results successful. The results show that threshold languages of weighted splicing systems have a higher generative power than the usual splicing systems [6]. In this paper, we study the generative power of weighted simple and semi-simple splicing systems.

This paper is organised as follows: Section 1 gives the background and introduction of the paper. Section 2 presents some necessary definitions and notations from the theories of formal languages and splicing systems. In addition, the concepts of weighted splicing systems and the threshold languages generated by weighted splicing systems are also given in this section. The definitions of weighted simple splicing systems and semi-simple splicing systems as well as their threshold languages are given in Section 3. In Section 4, some examples are provided to illustrate the generative power of weighted simple and semi-simple splicing systems. Also in this section, some new facts on the generative power of weighted simple and semi-simple splicing system are presented. Lastly, the conclusion is given in Section 5.

In the next section, some preliminaries of formal languages and splicing system are presented.

2. PRELIMINARIES

Readers are assumed to be familiar with the basic concepts in formal language theory and DNA computing. For further information of formal language theory, please refer to [13–15]. In this paper, the membership of an element to a set is denoted by \in while the negation of the set membership is denoted by \notin . The symbol \subseteq denotes the inclusion while \subset denotes the strict (proper) inclusion. The empty set is denoted by the symbol \emptyset . The set of integers, positive rational numbers, n -dimensional vector space over integers and the set of matrices with integer entries are denoted by \mathbb{Z} , \mathbb{Q}^+ , \mathbb{Z}^n and M , respectively. The zero matrix is denoted by \mathbf{O} . The symbols $+$ and \times denote the usual addition and multiplication operations, 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 [13]. For these language families, the next strict inclusions, named *Chomsky hierarchy*, holds:

Theorem 1 [13]

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

Next, we give some basic definitions of iterative splicing systems [2].

Let V be an alphabet, and $\#, \$ \notin V$ two special symbols. A *splicing rule* over V is a string of the form $r = u_1 \# u_2 \$ u_3 \# u_4$, where $u_1, u_2, u_3, u_4 \in V^*$.

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

$$(x, y) \quad_r \quad z$$

if and only if

$$x = x_1 u_1 u_2 x_2, \quad y = y_1 u_3 u_4 y_2, \quad z = x_1 u_1 u_4 y_2,$$

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

The string z is said to be obtained by splicing x, y , as indicated by the rule r ; $u_1 u_2$ and $u_3 u_4$ are called the *site* of the splicing. We call x the *first term* and y the *second term* of the splicing operation.

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 write

$$\sigma(L) = \{z \in V^* \mid (x, y) \quad_r \quad z, \text{ for some } x, y \in L, r \in R\}.$$

Then, for $L \subseteq V^*$, we define

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

where

$$\sigma^0(L) = L,$$

$$\sigma^{i+1}(L) = \sigma^i(L) \cup \sigma(\sigma^i(L)), i \geq 0.$$

Next, we give the formal definition for simple splicing systems and semi-simple splicing systems in [16, 17].

A *simple splicing system* is a triple $\gamma = (V, A, R)$, where V is an alphabet, A is a finite language over V , $R \subseteq V$ is a set of splicing rules. The set A is also called the *axiom* of simple splicing system and the elements of R are also called *markers* in which all rules have the form $(a, 1; a, 1)$ for some $a \in A$.

For $x, y, z \in V^*$ and $a \in M$, we write

$$(x, y) \quad^a \quad z$$

if and only if

$$x = x_1 a x_2, \quad y = y_1 a y_2 \quad \text{and} \quad z = x_1 a y_2, \text{ for } x_1, x_2, y_1, y_2 \in V^*.$$

A *semi-simple splicing system* is a triple $\gamma = (V, A, R)$ where V is an alphabet, A is a finite language over V , and $R \subseteq V$ is a set of splicing rules. The set A is also called the *axiom* and the elements of R are the rules in which all rules have the form $(a, 1; b, 1)$ for $a, b \in A$. For $x, y, z \in V^*$ and $(a, 1; b, 1) \in M$, we write

$$(x, y) \quad^{(a,1;b,1)} \quad z$$

iff $x = x_1 a x_2, y = y_1 b y_2$ and $z = x_1 a y_2$ for $x_1, x_2, y_1, y_2 \in V^*$.

The language generated by a splicing system $\gamma = (V, A, R)$ is defined by $L(\gamma) = \sigma_M^*(A)$.

The family of languages generated by simple and semi-simple splicing systems are denoted by *SH* and *SSH* respectively such that $\gamma = (V, A, R)$.

For *SH*, we have the following theorem.

Theorem 2 [2]

$$\text{FIN} \subseteq \text{SH} \subseteq \text{REG}.$$

In the next section, new variants of weighted splicing systems called weighted simple splicing systems and weighted semi-simple splicing systems are introduced.

3. PROPERTIES OF WEIGHTED SIMPLE AND SEMI-SIMPLE SPLICING SYSTEMS

Weighted simple and semi-simple splicing systems are the respectively usual simple and semi-simple splicing systems which are specified with a weighting space (i.e., any discrete structure such as sets of numbers, matrices, etc.) and operation over weights from the selected

weighting spaces. The formal definitions for weighted simple and semi-simple splicing system are given next.

Definition 1. A weighted simple splicing system is a 6-tuple

$$\gamma = (V, A, R, \omega, M, \square)$$

where

V, R are defined as the usual simple splicing system,

A is a subset of $V^* \times M$,

$\omega: V^* \rightarrow M$ is a weight function,

M is a weighting space, and

\square is the operation over the weights, $\omega(x), x \in V^*$.

Definition 2. A weighted semi-simple splicing system is a 6-tuple

$$\gamma = (V, A, R, \omega, M, \square)$$

where

V, R are defined as the usual semi-simple splicing system,

A is a subset of $V^* \times M$.

$\omega: V^* \rightarrow M$ is a weight function,

M is a weighting space, and

\square is the operation over the weights $\omega(x), x \in V^*$.

Next, the weighted simple and semi-simple splicing operation and the languages generated by them are given.

Definition 3. For a weighted simple splicing system $\gamma = (V, A, R, \omega, M, \square)$ where $(x, \omega(x)), (y, \omega(y)), (z, \omega(z)) \in V^* \times M$ and $a \in R$,

$$[(x, \omega(x)), (y, \omega(y))] \stackrel{a}{\square} (z, \omega(z))$$

iff $x = x_1ax_2, y = y_1ay_2, z = x_1ay_2$, for $x_1, x_2, y_1, y_2 \in V^*$ and $\omega(z) = \omega(x) \square \omega(y)$.

Definition 4. For a weighted semi-simple splicing system $\gamma = (V, A, R, \omega, M, \square)$ where $(x, \omega(x)), (y, \omega(y)), (z, \omega(z))$ and $(a, 1; b, 1) \in R$, such that $a, b \in V$.

$$[(x, \omega(x)), (y, \omega(y))] \stackrel{(a,1;b,1)}{\square} (z, \omega(z))$$

iff $x = x_1ax_2, y = y_1by_2, z = x_1ay_2$ for $x_1, x_2, y_1, y_2 \in V^*$ and $\omega(z) = \omega(x) \square \omega(y)$.

For a weighted simple (or semi-simple) splicing system $\gamma = (V, A, R, \omega, M, \square)$, the set of weighted strings obtained by splicing strings in A according to splicing rules in R and the weight operation \square are defined as the following.

Definition 5. Let $\gamma = (V, A, R, \omega, M, \square)$ be a weighted simple splicing system. Then

$$\sigma_\omega(A) = \{(z, \omega(z)) : (x, y) \stackrel{a}{\square} z \wedge \omega(z) = \omega(x) \square \omega(y)$$

for some $(x, \omega(x)), (y, \omega(y)) \in A$ and $a \in R\}$.

Definition 6. Let $\gamma = (V, A, R, \omega, M, \square)$ be a weighted semi-simple splicing system. Then

$$\sigma_\omega(A) = \{(z, \omega(z)) : (x, y) \stackrel{(a,1;b,1)}{\square} z \wedge \omega(z) = \omega(x) \square \omega(y)$$

for some $(x, \omega(x)), (y, \omega(y)) \in A$ and $a, b \in V, (a, 1; b, 1) \in R\}$.

Further, for a weighted simple (or semi-simple) splicing system $\gamma = (V, A, R, \omega, M, \square)$, we define the closure of A under splicing with respect to rules in R and the weight operation \square .

Definition 7. Let $\gamma = (V, A, R, \omega, M, \square)$ be a weighted simple (or semi-simple) splicing system. Then

$$\sigma_\omega^*(A) = \bigcup_{i \geq 0} \sigma_\omega^i(A)$$

where

$$\sigma_\omega^i(A) = \sigma_\omega^{i-1}(A) \cup \sigma_\omega(\sigma_\omega^{i-1}(A)) \text{ for } i = 1, 2, \dots, \sigma_\omega^0(A) = A.$$

Definition 8. The weighted language generated by a weighted simple (or semi-simple) splicing system $\gamma = (V, A, R, \omega, M, \square)$ is defined as

$$L_\omega(\gamma) = \sigma_\omega^*(A).$$

Remark 1. As a weighting space, one can consider different sets and (algebraic) structures, for instance, the sets of integers, rational numbers, real numbers, the sets of Cartesian products of the sets of numbers, the set of matrices with integer entries, groups, etc. Then, the operations over weights of strings are defined with respect to the chosen weighting space. In this paper, we consider as weighting spaces the sets of integers, positive rational numbers, the set of Cartesian products of integers and the set of matrices with integer entries.

In order to increase the generative power of the respectively weighted splicing systems, the languages generated by weighted simple and semi-simple splicing systems have been considered with some subsets of the weighting space called the thresholds (cut-points). In the following, the formal definitions for threshold languages for weighted simple and semi-simple splicing systems are given.

Definition 9. Let $L_\omega(\gamma)$ be the language generated by a weighted simple (or semi-simple) splicing system $\gamma = (V, A, R, \omega, M, \square)$. A threshold language $L_\omega(\gamma, \hat{a}\tau)$ with respect to a threshold (cut-point) $\tau \in \mathcal{M}$ is a subset of $L_\omega(\gamma)$ defined by

$$L_\omega(\gamma, \hat{a}\tau) = \{z \mid (z, \omega(z)) \in \sigma_\omega^*(A) \text{ and } \omega(z) \hat{a} \tau\}$$

where $\hat{a} \in \{=, >, <\}$ is called the *mode* of $L_\omega(\gamma, \hat{a}\tau)$.

Remark 2. We can also consider a threshold as a subset of M . Then, the mode for such a threshold is defined as a membership to the threshold set, i.e., for a threshold set $A \subseteq M$, the modes are \in and \notin .

The family of threshold languages generated by weighted simple and semi-simple splicing systems are denoted by ωSH and ωSSH respectively such that $\gamma = (V, A, R)$.

In the next section, the generative power for weighted simple and semi-simple splicing systems is discussed.

4. THE GENERATIVE POWER OF WEIGHTED ONE-SIDED SPLICING SYSTEMS

Here, the generative power of weighted simple and semi-simple splicing systems is investigated with respect to different weighting spaces and cut-points. First, we state the proposition which is immediately obtained from the definitions in Section 3.

Proposition 1. For $X \in \{S, SS\}$ and weighting spaces $(M, \square) \in \{(\square, +), (\square^k, \oplus), (\square^+, \times), (M, +)\}$,
 $X \subseteq \omega X(M, \square)$.

In the following examples, the generative powers of weighted simple and semi-simple splicing system are illustrated. We show that by using different weighting spaces, the generative power of weighted simple and semi-simple splicing systems with the same finite sets of axioms and splicing rules are higher than their respective usual splicing systems. In Examples 1 and 2, the generative power of weighted simple splicing systems with two and three axioms respectively, are considered; while in Examples 3 and 4, the weighted semi-simple splicing system with two and three axioms respectively, are considered.

Example 1. Consider a weighted simple splicing system $\gamma_1 = (\{a, b\}, \{(bba, \tau_1), (baab, \tau_2)\}, \{a, b\}, \omega, M, \square)$.

With the application of marker a iteratively to the axioms bba and $baab$, the generated string is $bba^m b$ for all $m \geq 1$, while the application of the marker a iteratively to the axioms $baab$ and bba , the generated string is $b^n a$ for all $n \geq 1$. Furthermore, $(bba^m b, b^n a) \stackrel{b}{\rightarrow} (bba^m b^n a)$.

Thus

$$L_\omega(\gamma_1) = \{bba^m b^n a \mid (bba^m b^n a, \omega(bba^m b^n a)) \in \sigma^*(A), n, m \geq 1\},$$

where $A = \{(abca, \tau_1), (acba, \tau_2)\}$.

First, we choose the set \square of all integers as the weighting space M , the addition $+$ of integers as the operation \square , $\tau_1 = 1$ and $\tau_2 = -1$. Then, it is clear that

$$L_\omega(\gamma_1) = \{bba^m b^n a \mid (bba^m b^n a, n - m) \in \sigma^*(A), n, m \geq 1\},$$

and

$$L_\omega(\gamma_1, = 0) = \{bba^m b^n a \mid n \geq 1\} \in \mathbf{CF-REG},$$

$$L_\omega(\gamma_1, > 0) = \{bba^m b^n a \mid n > m \geq 1\} \in \mathbf{CF-REG},$$

$$L_\omega(\gamma_1, < 0) = \{bba^m b^n a \mid n > k \geq 1\} \in \mathbf{CF-REG}.$$

These three languages are context-free and not regular.

Second, we choose the set of Cartesian product of all integers $\square \times \square$ as the weighting space. Let the operation \square be the componentwise addition of pairs from $\square \times \square$, i.e., for any two $(x_1, x_2), (y_1, y_2) \in \square \times \square$, the sum is defined by $(x_1 + y_1, x_2 + y_2)$ and the ordering relation is also defined componentwise, i.e., $(x_1, x_2) > (y_1, y_2)$ if and only if $x_1 > y_1$ and $x_2 > y_2$. Let $\tau_1 = (1, 1)$, $\tau_2 = (-1, -1)$. Then with the cut-point $(0, 0)$, the same languages as above are generated.

Third, the same languages as above are also generated when the set of all 2×2 matrices of integers are considered as the weighting space M , the componentwise addition as the operation \square , the componentwise ordering as the ordering relation, $\tau_1 = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$ and $\tau_2 = \begin{pmatrix} -1 & -1 \\ -1 & -1 \end{pmatrix}$ as the weights, and $\tau = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$ as the cut-point.

Example 2. Consider a weighted simple splicing system

$$\gamma_2 = (\{a, b, c, x, y, z\}, \{(xaay, \tau_1), (ybbz, \tau_2), (zccx, \tau_3)\}, \{a, b, c\}, \omega, M, \square).$$

With the application of the marker a iteratively to the axiom $xaay$, the generated string is $xa^n y$ for $n \geq 1$; the application of the marker b iteratively to the axiom $ybbz$, the generated string is $yb^m z$ for $m \geq 1$; and the application of the marker c iteratively to the axiom $zccx$, the generated string is $zc^k x$. Furthermore,

$$(xa^n y, yb^m z) \stackrel{a}{\rightarrow} (xa^n b^m z) \text{ for } n, m \geq 1,$$

$$(xa^n b^m z, zc^k x) \stackrel{a}{\rightarrow} (xa^n b^m c^k x) \text{ for } n, m, k \geq 1.$$

Then, the language generated by the weighted splicing system γ_2 is

$$L_\omega(\gamma_2) = \{xa^n b^m c^k x \mid (xa^n b^m c^k x, \omega(xa^n b^m c^k x)) \in \sigma^*(A), n, m, k \geq 1\},$$

where $A = \{(xaay, \tau_1), (ybbz, \tau_2), (zccx, \tau_3)\}$.

Next, we define different threshold languages with different weighting spaces and operations.

First, let $M = \square^+$, the operation \square be the usual multiplication, and $\tau_1 = 3^{-1}$, $\tau_2 = 5^{-1}$, $\tau_3 = 15$. Then,

$$L_{\omega}(\gamma_2) = \{xa^n b^m c^k x \mid (xa^n b^m c^k x, 3^{k-n} 5^{k-m}) \in \sigma^*(A), k, m, n \geq 1\}.$$

We choose $\tau = 1$ as a cut-point, and define the following threshold languages:

$$\begin{aligned} L_{\omega}(\gamma_2, = 1) &= \{xa^n b^n c^n x \mid n \geq 1\} \in \mathbf{CS} - \mathbf{CF}, \\ L_{\omega}(\gamma_2, > 1) &= \{xa^n b^m c^k x \mid k > n, m \geq 1\} \in \mathbf{CF} - \mathbf{REG}, \\ L_{\omega}(\gamma_2, < 1) &= \{xa^n b^m c^k x \mid n, m > k \geq 1\} \in \mathbf{CF} - \mathbf{REG}. \end{aligned}$$

Second, let $M = \square \times \square$ and \square as the componentwise addition of pairs from $\square \times \square$, $\tau_1 = (1, 0)$, $\tau_2 = (-1, 1)$ and $\tau_3 = (0, -1)$. Then,

$$L_{\omega}(\gamma_2) = \{xa^n b^m c^k x \mid (xa^n b^m c^k x, (n-m, m-k)) \in \sigma^*(A), k, m, n \geq 1\}.$$

Consequently,

$$\begin{aligned} L_{\omega}(\gamma_2, = (0, 0)) &= \{xa^n b^n c^n x \mid n \geq 1\} \in \mathbf{CS} - \mathbf{CF}, \\ L_{\omega}(\gamma_2, > (0, 0)) &= \{xa^n b^m c^k x \mid n > m > k \geq 1\} \in \mathbf{CS} - \mathbf{CF}, \\ L_{\omega}(\gamma_2, < (0, 0)) &= \{xa^n b^m c^k x \mid k > m > n \geq 1\} \in \mathbf{CS} - \mathbf{CF}. \end{aligned}$$

Third, the same languages as above can be generated by γ_2 if we choose the set of all 2×2 matrices of integers as the weighting space, the componentwise addition as the operation \square , $\tau_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, $\tau_2 = \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix}$, $\tau_3 = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}$,

and $\tau = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$ as the cut-point for the threshold languages.

Next, we consider two examples of weighted semi-simple splicing system.

Example 3. Consider a weighted semi-simple splicing system,

$$\gamma_3 = (\{a, b\}, \{(abca, \tau_1), (acba, \tau_2)\}, \{r_1 = (b, 1; c, 1), r_2 = (c, 1; b, 1), r_3 = (b, 1; a, 1)\}, \omega, M, \square).$$

With the application of rule r_1 iteratively to the axioms $abca$ and $acba$, the generated string is $ab^n a$ for all $n \geq 1$; the application of rule r_2 iteratively to the axioms $acba$ and $abca$, the generated string is $ac^m a$ for all $m \geq 1$; applying rule r_3 to the strings $ab^n a$, $n \geq 1$, and $ac^m a$, $m \geq 1$, the string $ab^n c^m a$ is obtained. Thus,

$$L_{\omega}(\gamma_3) = \{ab^n c^m a \mid (ab^n c^m a, \omega(ab^n c^m a)) \in \sigma^*(A), n, m \geq 1\},$$

where $A = \{(abca, \tau_1), (acba, \tau_2)\}$.

First, we choose the set \square of all integers as the weighting space M , the addition $+$ of integers as the operation \square , $\tau_1 = -1$ and $\tau_2 = 1$. Then, it is clear that

$$L_{\omega}(\gamma_3) = \{ab^n c^m a \mid (ab^n c^m a, n-m) \in \sigma^*(A), n, m \geq 1\},$$

Then for $\tau = 0$,

$$\begin{aligned} L_{\omega}(\gamma_3, = 0) &= \{ab^n c^n a \mid n \geq 1\} \in \mathbf{CF} - \mathbf{REG}, \\ L_{\omega}(\gamma_3, > 0) &= \{ab^n c^m a \mid n > m \geq 1\} \in \mathbf{CF} - \mathbf{REG}, \end{aligned}$$

$$L_{\omega}(\gamma_3, < 0) = \{ab^n c^m a \mid n > k \geq 1\} \in \mathbf{CF} - \mathbf{REG}.$$

These three languages are context-free and not regular.

Second, we choose the Cartesian product of the set of all integers $\square \times \square$ as the weighting space. Let the operation \square be the componentwise addition of pairs from $\square \times \square$, i.e., for any two $(x_1, x_2), (y_1, y_2) \in \square \times \square$, the sum is defined by $(x_1 + y_1, x_2 + y_2)$ and the ordering relation is also defined by componentwise, i.e., $(x_1, x_2) > (y_1, y_2)$ if and only if $x_1 > y_1$ and $x_2 > y_2$. Let $\tau_1 = (-1, -1)$, $\tau_2 = (1, 1)$. Then with cut-point $(0, 0)$, the same languages as above are generated.

Third, the same languages as above are also generated when the set of all 2×2 matrices of integers are considered as the weighting space M , the componentwise addition as the operation \square , the componentwise ordering

as the ordering relation, $\tau_1 = \begin{pmatrix} -1 & -1 \\ -1 & -1 \end{pmatrix}$ and $\tau_2 = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$ as

the weights, and $\tau = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$ as the cut-point.

Example 4. Consider a weighted semi-simple splicing system

$$\gamma_4 = (\{a, b, c, d\}, \{(aba, \tau_1), (aca, \tau_2), (ada, \tau_3)\},$$

$$\{r_1 = (b, 1; a, 1), r_2 = (c, 1; a, 1), r_3 = (d, 1; a, 1)\}, \omega, M, \square)$$

One can see that for all $k, m, n \geq 1$, the application of rule r_1 iteratively to the axiom aba , the generated string is $ab^n a$; the application of rule r_2 iteratively to the axiom aca , the generated string is $ac^m a$; the application of rule r_3 iteratively to the axiom ada , the generated string is $ad^k a$. Furthermore, string $ab^n c^m a$ is generated by iteratively applying rule r_3 to the strings $ab^n a$ and $ac^m a$. Lastly, by iteratively applying rule r_2 to $ab^n c^m a$ and $ad^k a$, the string $ab^n c^m d^k a$ is obtained. Thus, the language generated by the weighted splicing system γ_4 is

$$L_{\omega}(\gamma_4) = \{ab^n c^m d^k a \mid (ab^n c^m d^k a, \omega(ab^n c^m d^k a)) \in \sigma^*(A), n, m, k \geq 1\},$$

where $A = \{(aba, \tau_1), (aca, \tau_2), (ada, \tau_3)\}$.

Next, we define different threshold languages with different weighting spaces and operations by firstly let $M = \square^+$, the operation \square be the usual multiplication, and $\tau_1 = 3^{-1}$, $\tau_2 = 5^{-1}$, $\tau_3 = 15$.

Then,

$$L_{\omega}(\gamma_4) = \{ab^n c^m d^k a \mid (ab^n c^m d^k a, 3^{k-n} 5^{k-m}) \in \sigma^*(A), k, m, n \geq 1\}.$$

We choose $\tau = 1$ as a cut-point, and define the following threshold languages:

$$L_{\omega}(\gamma_4, = 1) = \{ab^n c^n d^n a \mid n \geq 1\} \in \mathbf{CS} - \mathbf{CF},$$

$$L_{\omega}(\gamma_{4, > 1}) = \{ab^n c^m d^k a \mid k > n, m \geq 1\} \in \mathbf{CF-REG},$$

$$L_{\omega}(\gamma_{4, < 1}) = \{ab^n c^m d^k a \mid n, m > k \geq 1\} \in \mathbf{CF-REG}.$$

Second, let $M = \square \times \square$, \square as the componentwise addition of pairs from $\square \times \square$, and $\tau_1 = (1, 0)$, $\tau_2 = (-1, 1)$, $\tau_3 = (0, -1)$. Then,

$$L_{\omega}(\gamma_4) = \{ab^n c^m d^k a \mid (ab^n c^m d^k a, (n-m, m-k)) \in \sigma^*(A), k, m, n \geq 1\}.$$

Consequently, for $\tau = (0, 0)$,

$$L_{\omega}(\gamma_{4, = (0,0)}) = \{ab^n c^n d^k a \mid n \geq 1\} \in \mathbf{CS-CF},$$

$$L_{\omega}(\gamma_{4, > (0,0)}) = \{ab^n c^m d^k a \mid n > m > k \geq 1\} \in \mathbf{CS-CF},$$

$$L_{\omega}(\gamma_{4, < (0,0)}) = \{ab^n c^m d^k a \mid k > m > n \geq 1\} \in \mathbf{CS-CF}.$$

Third, the same languages as above can be generated by γ_4 if we choose the set of all 2×2 matrices of integers as the weighting space, the componentwise addition as the operation

$$\square, \quad \tau_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \tau_2 = \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix},$$

$$\tau_3 = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}, \text{ and } \tau = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \text{ as the cut-point for the threshold languages.}$$

From Examples 1 to 4, Theorem 1 is obtained.

Theorem 1 For $X \in \{S, SS\}$ and $F_1 \in \{\mathbf{FIN}, \mathbf{REG}, \mathbf{CF}, \mathbf{LIN}\}$,

$$\omega X(M, \square) - F_1 \neq \emptyset$$

where $(M, \square) \in \{(\square, +), (\square^k, \oplus), (\square^+, \times), (M, +)\}$.

From Lemma 1 and 2, Proposition 1, Example 1 and 2, the following theorem is obtained.

Theorem 2 For simple splicing system,

$$(a) \quad \mathbf{FIN} \subset \omega SH(M, \square) \subseteq \mathbf{RE}$$

$$(b) \quad SH \subset SH(M, \square) \subseteq \mathbf{RE}$$

where $(M, \square) \in \{(\square, +), (\square^k, \oplus), (\square^+, \times), (M, +)\}$.

5. CONCLUSION

In this paper, two new variants of weighted splicing systems called weighted simple splicing systems and semi-simple splicing systems have been introduced. Some new facts on the weighted simple and semi-simple splicing system are also given. Furthermore, we show that simple and semi-simple splicing systems with finite components of weights have a higher generative power as compared to the usual simple and semi-simple splicing systems. Moreover, some examples which illustrate the generative power of weighted simple and semi-simple splicing systems are also presented. These examples show that some non-regular and non-context-free languages can be generated by weighted simple and semi-simple splicing

systems. Lastly, the generative power of weighted simple and semi-simple splicing systems are generalised by relating the threshold languages generated by weighted simple and semi-simple splicing systems to the Chomsky hierarchy.

ACKNOWLEDGEMENTS

The first author would like to thank Ministry of Education Malaysia for the MyBrain15 Scholarship. The second and third authors are supported by Ministry of Education Malaysia and Research Management Center UTM through Research University Grant Vote No. 08H45.

REFERENCES

- [1] T. Head, Bull. Math. Biol. 49(6) (1987) 737.
- [2] G. Paun, G. Rozenberg, A. Salomaa, DNA Computing: New Computing Paradigms, Springer-Verlag, Berlin, 1998.
- [3] D. Pixton, Discrete Appl. Math. 69 (1996) 101.
- [4] R. Freund, L. Kari, Gh. Paun, Theory Comput. Syst. 32 (1999) 69.
- [5] Gh. Paun, J. Autom. Lang. Comb. 1(1) (1996) 27.
- [6] S. Turaev, Y.S. Gan, M. Othman, N.H. Sarmin, W.H. Fong, in: Z. Li, X. Li, Y. Liu, Z. Cai (Eds.), Computational Intelligence and Intelligent Systems, Springer, China, 2012, pp. 416 – 424.
- [7] A. Salomaa, Inform. Control. 15 (1969) 529.
- [8] K.S. Fu, T. Li, Inform. Sci. 1(4) (1969) 403.
- [9] M. Mizumoto, J. Toyoda, K. Tanaka, Inform. Process. Lett. 2(3) (1973) 74.
- [10] M. Mizumoto, J. Toyoda, K. Tanaka, J. Comput. Syst. Sci. Int. 10(2) (1975) 219.
- [11] A. Alexandrakis, S. Bozapalidis, Inform. Process. Lett. 24(1) (1987) 1.
- [12] M. Droste, W. Kuich, H. Vogler, Handbook of Weighted Automata, Springer-Verlag, Berlin, 2009.
- [13] G. Rozenberg, A. Salomaa, Handbook of Formal Languages, Springer-Verlag, Berlin, 1997.
- [14] J. Dassow, G. Paun, Regulated Rewriting in Formal Language Theory, Springer-Verlag, Berlin, 1989.
- [15] P. Linz, An Introduction to Formal Languages and Automata, Jones and Bartlett Publishers, USA, 2006.
- [16] [16] A. Mateescu, Gh. Paun, G. Rozenberg, A. Salomaa, Discrete Appl. Math. 84 (1998) 145.
- [17] E. Goode, D. Pixton, in: C. Martine-Vide, V. Mitrana (Eds.), Where Mathematics, Computer Science, Linguistics and Biology Meet, Kluwer Academic Publ, Dordrecht, 2001, pp. 343 – 357.