

HIERARCHY OF CERTAIN TYPES OF DNA SPLICING SYSTEMS

YUHANI YUSOF and NOR HANIZA SARMIN

*Department of Mathematics, Faculty of Science, Universiti Teknologi Malaysia,
81310 UTM Johor Bahru, Johor Darul Takzim, Malaysia
yuhani@ump.edu.my, nhs@utm.my*

T. ELIZABETH GOODE

*Department of Mathematics, Towson University,
7800 York Road, Towson, Maryland, United State of America
egoode@towson.edu*

MAZRI MAHMUD

*Emergency Department, Hospital Tengku Ampuan Afzan, Jalan Tanah Putih,
25100 Kuantan, Pahang Darul Makmur, Malaysia
ppp.mazri@yahoo.com*

FONG WAN HENG

*Ibnu Sina Institute for Fundamental Sciences Studies, Universiti Teknologi Malaysia,
81310 UTM Johor Bahru, Johor Darul Takzim, Malaysia
fwh@ibnusina.utm.my*

A Head splicing system (H -system) consists of a finite set of strings (words) written over a finite alphabet, along with a finite set of rules that acts on the strings by iterated cutting and pasting to create a splicing language. Any interpretation that is aligned with Tom Head's original idea is one in which the strings represent double-stranded deoxyribonucleic acid (dsDNA) and the rules represent the cutting and pasting action of restriction enzymes and ligase, respectively. A new way of writing the rule sets is adopted so as to make the biological interpretation transparent. This approach is used in a formal language-theoretic analysis of the hierarchy of certain classes of splicing systems, namely simple, semi-simple and semi-null splicing systems. The relations between such systems and their associated languages are given as theorems, corollaries and counterexamples.

Keywords: DNA Splicing Systems; formal language theory; DNA computing.

1. Introduction

Watson and Crick firstly described the structure of deoxyribonucleic acid (DNA) in 1953 in form of double-helical structure.¹ The structure of DNA differs from each other by the sequence of their bases namely: *Adenine*, *Guanine*, *Cytosine* and *Thymine*, abbreviated as *A*, *G*, *C* and *T*, respectively. These four bases are tied together by hydrogen bonds using rules of pairing:

$$\begin{array}{l} A, G, C \text{ and } T \\ T, C, G \quad A. \end{array}$$

These rules of pairing can simply be written as *a*, *g*, *c* and *t*, respectively.

The first mathematical modeling of splicing system is illustrated under a branch of Applied Discrete Mathematics and Theoretical Computer Science, namely Formal Language Theory by Head in 1987.² This mathematical model is formally known as the Head's model. This model represents the bases, initial string of DNA and the existing restriction enzymes as a set of an alphabet *A*, a finite initial string *I*, and a left or right pattern *B* or *C*, respectively. This model can simply be written as $S = (A, I, B, C)$.

On making the biological interpretation translucent, a new way of writing the rule sets is adopted, namely Yusof-Goode splicing system. The behind motivation of writing this extension is based on the modification of the existed splicing system rules: Head and Goode-Pixton. This new extension splicing system is presented as splicing system, $S = (A, I, R)$, where *A* is a set of alphabet *a*, *c*, *g* and *t*, an *I* defined as initial dsDNA and *R* is a set of rule either in the form of $(u; x, v; y; x, z)$ for left pattern rule or $(u, x; v; y, x; z)$ for right pattern rule where the *u*, *x*, *v* and *y*, *x*, *z* present the first and second restriction enzymes respectively. Besides, the rule *R* in this splicing system is automatically holds the symmetrical and reflexivity properties. Note that, when the rule is presented without semi-colon i.e. $(u, x, v; y, x, z)$ it means that both left and right patterns are applied.

Restriction enzymes which are presented as a set *R* in this mathematical model are found in bacteria that are chosen from more than 200 types readily enzymes.³ These restriction enzymes can cut the DNA molecules at specific places and yield molecules with blunt or sticky ends based on their crossing sites. For example, the restriction enzyme *BmgBI* produces blunt end, whereas restriction enzyme *BsaHI* produces sticky end during the cutting process. The DNA molecules will paste together with the existence of a ligase.

There are many types of splicing systems, including simple, semi-simple and semi-null splicing systems. The notion of simple splicing system had been introduced by Mateescu et al.⁴ in 1998. A year later, Laun⁵ presented the semi-simple and semi-null splicing system in her PhD thesis. In 2001, the characterization of semi-simple splicing languages is presented in terms of directed graph by Goode and Pixton⁶ while Ceterchi⁷ focused on the algebraic characterization of that splicing system in 2006. In 2008, Fong⁸

introduced some concepts involving simple splicing system using Formal Language Theory. In this paper, the hierarchy of those types splicing systems is presented.

This paper is organized into four sections. The first section is the Introduction, followed by Section 2 which includes some basic definitions that are used in this research. In Section 3, some relations on the hierarchy of different types of splicing systems are presented. Finally, the conclusion is given.

2. Preliminaries

In this section, some formal definitions related to this research are stated. The main definition which is splicing system will firstly be given.

Let A be defined as a fixed finite set to be used as an alphabet and A^* as a free monoid that consists of all strings of symbols in A , including the null string.

Definition 2.1²: (Splicing System)

A **splicing system** $S = (A, I, B, C)$ consists of a finite alphabet A , a finite set I of initial strings in A^* , and finite sets B and C of triples (c, x, d) with c, x and d in A^* . Each such triple in B or C is called a pattern. For each such triple the string $cx d$ is called a site and the string x is called a crossing. Patterns in B are called left patterns and patterns in C are called right patterns.

Next, three types of splicing system discussed in this paper are defined.

Definition 2.2⁵: (Simple Splicing System)

Let $S = (A, I, R)$ be a splicing system in which all rules in R have the form $(a, 1; a, 1)$ where $a \in A$. Then S is called a **simple splicing system**.

Definition 2.3⁵: (Semi-Simple Splicing System)

Let (A, I, R) be a splicing system in which I and R are finite and every rule in R has the form $(a, 1; b, 1)$, where a, b are in A .

Thus $\sigma = (A, R)$ is called a semi-simple splicing scheme and (A, I, R) a **semi-simple splicing system**.

Definition 2.4⁵: (Semi-Null Splicing System)

Let (A, I, R) be a splicing system in which I and R are finite and every rule in R has the form $(u, 1; v, 1)$, where u, v are in A^+ . Thus $\sigma = (A, R)$ is called a semi-null splicing scheme, and (A, I, R) a **semi-null splicing system**.

3. Results and Discussions

Some relations on different types of splicing systems are analyzed and presented as theorems and corollaries in this section. Besides, some counterexamples are presented too. Since Yusof-Goode splicing system model is being used, it will represent the rules R in term of double-triple notation. Hence, the rule of all related splicing systems in Section 2 above can be rewritten as follows:

- Simple splicing system
 $R = (a, 1, 1 : a, 1, 1) / (1, 1, a : 1, 1, a) / (1, a, 1 : 1, a, 1)$, where $a \in A$.
- Semi-simple splicing system, $R = (a, 1, 1 : b, 1, 1)$, where $a, b \in A$.
- Semi-null splicing system, $R = (u, 1, 1 : v, 1, 1)$, where $u, v \in A^+$ or
 $R = (u_1, w, 1 : v_1, w, 1)$, where u_1, v_1 and $w \in A^+$

This first theorem shows how the initial rule of simple splicing system can be translated to the Yusof-Goode form.

Theorem 3.1. *If a simple splicing system with $(a, 1; a, 1)$ rule is replaced with $(a, 1, 1 : a, 1, 1)$ where $a \in A$, then the resulting splicing system will generate the same splicing language given any initial set I .*

Proof. Assume S_1 and S_2 are two arbitrary initial strings in I , r_1 and r_2 are two different rules of the form $(a, 1; a, 1)$ and $(a, 1, 1 : a, 1, 1)$ respectively. Let $S_1 = \alpha u a v \beta$ and $S_2 = \gamma y a z \delta$ and can be presented in either order. Thus, by applying r_1 and r_2 on the same strings of S_1 and S_2 , the following results obtained:

$$\begin{aligned} (\alpha u a v \beta, \gamma y a z \delta) &\xrightarrow{r_1} \alpha u a z \delta, \gamma y a v \beta \\ (\alpha u a v \beta, \gamma y a z \delta) &\xrightarrow{r_2} \alpha u a z \delta, \gamma y a v \beta \end{aligned}$$

Since the resulting strings are same, hence it is proved. \square

In the following theorem, it shows that the rule of simple splicing system $R = (a, 1, 1 : a, 1, 1)$ where $a \in A$ can also be presented in other two different forms in Yusof-Goode splicing system.

Theorem 3.2. *A simple splicing rule $R = (a, 1, 1 : a, 1, 1)$ where $a \in A$ can also be be represented as $(1, 1, a : 1, 1, a)$ or $(1, a, 1 : 1, a, 1)$.*

Proof. Let S_1, S_2 are two arbitrary initial strings in I , r_1, r_2 and r_3 are three different rules of the form $(a, 1, 1 : a, 1, 1)$, $(1, 1, a : 1, 1, a)$ and $(1, a, 1 : 1, a, 1)$. Assume $S_1 = \alpha r a s \beta$ and $S_2 = \gamma t a u \delta$ and can be presented in either order. Hence, by applying r_1, r_2 and r_3 on the same strings of S_1 and S_2 , the following results are obtained:

$$\begin{aligned}
(\alpha ras\beta, \gamma tau\delta) &\xrightarrow{r_1} \alpha rau\delta, \gamma tas\beta \\
(\alpha ras\beta, \gamma tau\delta) &\xrightarrow{r_2} \alpha rau\delta, \gamma tas\beta \\
(\alpha ras\beta, \gamma tau\delta) &\xrightarrow{r_3} \alpha rau\delta, \gamma tas\beta
\end{aligned}$$

Thus it is proved. \square

The conversion of semi-simple splicing system into Yusof-Goode form is described in Theorem 3.3 below.

Theorem 3.3. Every rule of $R = (a, 1; b, 1)$ where $a, b \in A$ can also be represented as $(a, 1, 1 : b, 1, 1)$ without loss of generative power.

Proof. Assume S_1 and S_2 are two arbitrary initial strings in I , r_1 and r_2 are two different rules of the form $(a, 1; b, 1)$ and $(a, 1, 1 : b, 1, 1)$ respectively. Let $S_1 = \alpha uav\beta$ and $S_2 = \gamma ybz\delta$ and can be presented in either order. Thus, by applying r_1 and r_2 on the same strings of S_1 and S_2 , the obtaining strings are as follows:

$$\begin{aligned}
(\alpha uav\beta, \gamma ybz\delta) &\xrightarrow{r_1} \alpha uaz\delta, \gamma ybv\beta \\
(\alpha uav\beta, \gamma ybz\delta) &\xrightarrow{r_2} \alpha uaz\delta, \gamma ybv\beta
\end{aligned}$$

Since the resulting strings are the same, hence it is proved. \square

Note that the same method is applied on proving the conversion of semi-null splicing system into Yusof-Goode notation.

All these rules are the updated corrections for some part in proceeding⁹ and journal¹⁰. In the next Theorem 3.4, a relation between simple and semi-simple splicing system is presented.

Theorem 3.4. Every simple splicing system is semi-simple splicing system of the form (A, I, R) .

Proof. Suppose that t is not an element of a semi-simple splicing system. Hence, there exist a cleavage pattern in R that does not fulfil the form of $(a, 1, 1 : b, 1, 1)$, where a, b are elements of A . Thus, t is not an element of a simple splicing system since each cleavage pattern in R is not in the form of $(a, 1, 1 : a, 1, 1)$ or $(b, 1, 1 : b, 1, 1)$ where a, b are elements of A . \square

Semi-simple and semi-null splicing systems are related as follows:

Theorem 3.5. Every semi-simple splicing system is semi-null splicing system of the form (A, I, R) .

Proof. Suppose that t is not an element of a semi-null splicing system. Thus, there exist a cleavage pattern in R that does not fulfil the form of $(u, 1, 1; v, 1, 1)$, where u, v are elements of A^+ . Hence, t is not an element of a semi-simple splicing system since A is a subset of A^+ . \square

However, there exists a semi-null splicing system that is not semi-simple as given in the following counterexample.

Counterexample 3.1

This example shows that there exists a semi-null splicing system that is not semi-simple. Let $S = (\{a, g, c, t\}, I(\text{unspecified}), \{gagtc(n)_5; 1, 1 : gagtc(n)_5; 1, 1\})$ be a splicing system, where the 5' overhang rule consists of two restriction enzymes namely, *MlyI* and *SchI*.

Thus, S is a semi-null splicing system since left-context of crossing site for both enzymes are disjoint- if the $(n)_5$ are different elements of A^+ . However, S is not semi-simple since $gagtc(n)_5$ is not an elements of A .

Theorems 3.4 and 3.5 lead to Corollary 3.1.

Corollary 3.1. *Every simple splicing system is semi-null.*

4. Conclusions

In this paper, some analysis on different types of splicing systems, namely simple, semi-simple and semi-null splicing systems are done. Their relations are presented as Theorem 3.4, 3.5, Corollaries 3.1 and Counterexample 3.1. It is found that, every simple splicing system is semi-simple and every semi-simple is semi-null. These are all proper inclusions, in other words there exist at least a semi-null which is not a semi-simple and a semi-simple which is not simple. All these relations can be simplified as follows:

simple splicing system \subset semi-simple splicing system \subset semi-null splicing system

Acknowledgments

The first author would like to acknowledge Universiti Malaysia PAHANG (UMP) for her study leave and to MOHE for her SLAI scholarship.

References

1. R. H. Tamarin, *Principle of Genetics*, (The McGraw-Hill Companies, USA, 2001).
2. T. Head, *Bull. Math. Biology* **49**, 737-759 (1987).
3. Research Biolabs Sdn. Bhd., *New England Biolabs 2009-10 Catalog & Technical Reference*, (USA, 2009).
4. A. Mateescu, Gh. Paun, G. Rozenberg and A. Salomaa, *Discrete Applied Mathematics* **84**, 145-163 (1998).

5. E. G. Laun, *Constants and Splicing Systems*, Ph.D Thesis, State University of New York at Binghamton, 1999.
6. E. Goode and D. Pixton, Semi-simple splicing systems, in *Where Mathematics, Computer Science, Linguistics and Biology Meet*, ed. C. Martin-Vide and V. Mitrana (Kluwer Academic Publisher, Dordrecht, 2001), p. 343-352.
7. R. Ceterchi, *Fundamenta Informaticae* **73**(1-2), 19-25 (2006).
8. W. H. Fong, *Modelling of Splicing Systems Using Formal Language Theory*, PhD Thesis, Universiti Teknologi Malaysia, 2008.
9. N.H. Sarmin, Y.Yusof and W.H. Fong, Some Characterizations in Splicing Systems, in *Proc. International Conferences of Mathematical Sciences* (Bolu, Turkey 2010).
10. Y. Yusof, N.H. Sarmin, W.H. Fong and F. Karimi. *Journal of Fundamental Sciences* **6** (2) 142-146 (2010).