ISSN 1823-626X

Journal of Fundamental Sciences

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

Some relations on different types of splicing systems

Yuhani Yusof[1]*, Nor Haniza Sarmin[1], Fong Wan Heng[2] and Fariba Karimi[1]

[1] *Department of Mathematics, Faculty of Science, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor, Malaysia.*
[2] *Ibnu Sina Institute for Fundamental Science Studies, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor, Malaysia.*

Received 21 April 2010, Revised 8 October 2010, Accepted 15 October 2010, Available online 11 November 2010

ABSTRACT

Splicing system is a formal characterization of the generative capacity of specified enzymatic activities acting on initial DNA molecules that was first initiated by Head in 1987. This splicing system is formally illustrated under the framework of Formal Language Theory which is a branch of Theoretical Computer Science and Applied Discrete Mathematics. There are many types of splicing systems including null-context, simple, semi-simple and semi-null. In this paper, some relations for those types of splicing systems are presented.

| Splicing systems | Formal Language Theory |

® 2010 Ibnu Sina Institute. All rights reserved.

1. INTRODUCTION

Deoxyribonucleic Acid (DNA) is the genetic material in an organism. These materials are made by joining nucleotides in a repetitive way into long and chain-like polymers. Nucleotides consist of three components, namely phosphate, sugar and a nitrogeneous base. The carbons of the sugars are numbered $1'$ to $5'$. The structure of DNA was firstly described by Watson and Crick in 1953 [1]. In fact, they found that a possible structure for DNA was one in which two helices coiled around one another, called a double helix structure, with the sugar phosphate backbones on the outside and the bases on the inside.

Nucleotides in DNA differ by their bases namely: *Adenine* (*A*), *Guanine* (*G*), *Cytosine* (*C*) and *Thymine* (*T*). Two single strands of DNA molecules can anneal together to form a double-stranded DNA (dsDNA) molecule. The bases hold together by hydrogen bonds in standard complementary, where:

A hydrogen bonds to *T*,
G hydrogen bonds to *C*,
C hydrogen bonds to *G* and
T hydrogen bonds to *A*.

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

For example, a sequence of DNA can be represented as *ccaacatg*, [*C*/*G*][*C*/*G*][*A*/*T*][*A*/*T*][*C*/*G*][*A*/*T*][*T*/*A*][*G*/*C*], or

5'… *C C A A C A T G*…3'
3'… *G G T T G T A C*…5',

where 5'… *C C A A C A T G*…3' and 3'… *G G T T G T A C*…5' denotes single strands of DNA.

Nowadays, there exist more than 200 types of restriction enzymes [2]. These restriction enzymes are found in bacteria which can cut the DNA molecules at specific places, resulting in molecules with sticky or blunt ends based on their cleavage pattern. The place where restriction enzyme can cut a molecule is called a cutting site, which is denoted as ▼ and ▲. For example, the restriction site for enzyme *EcoRI* is denoted as

5'… *G* ▼*A A T T* *C*…3'
3'… *C* *T T A A* ▲*G*…5',

while the restriction site for enzyme *AluI* is denoted as

5'… *A G* ▼ *C* *T*…3'
3'… *T C* ▲ *G A* …5'.

Therefore, the restriction enzyme *Aci*I is said to produce sticky end whereas the restriction enzyme *Alu*I is said to produce blunt end during cutting. New hybrid molecules then arise when the DNA cut by restriction enzymes are pasted together by a ligase. This operation is called the ligating operation and the result is a molecule of recombinant DNA.

Corresponding author at: Department of Mathematics, Faculty of Science
Universiti Teknologi Malaysia, 81310 UTM Skudai, Johor, Malaysia.
E-mail address: yuhani@ump.edu.my (Yuhani Yusof)

The next section discusses the mathematical modeling of splicing system in DNA and the researches that have been done by other mathematicians.

2. MATHEMATICAL MODELLING

The mathematical modeling of splicing system that was developed by Head [3] is formally illustrated under the framework of Formal Language Theory. This modeling initiates the new relationship between formal language theory and the study of macromolecules. This mathematical model (*A*, *I*, *B*, *C*) consists of:

A – the four bases of *a*, *g*, *c* and *t*,
I – a finite set of initial strings of DNA molecules,
B – the set rules consisting 5' overhangs and blunt end of restriction enzymes, and
C – the set of rules consisting 3' overhangs of restriction enzymes.

There are many types of splicing systems, for instance, null-context, simple, semi-simple and semi-null splicing systems. Head introduces the notion of persistent splicing system and null-context splicing system in [3]. Then, each null-context splicing system is shown to be persistent. Besides, the definition of constant for a string in a splicing system is also given.

In 1998, Mateescu *et al.* [4] introduced the notion of simple splicing systems, and that for every simple splicing system, the language generated is regular. Besides, several characteristics of simple splicing systems are mentioned. A decade after, Fong [5] introduced some concepts involving simple splicing system using Formal Language Theory. The relation between splicing system and automaton is also shown. Since splicing languages are regular, thus they can be recognized by automata diagrams.

In 1999, Laun [6] studied on some characterization of simple splicing languages and null-context splicing system. Besides, the relationships between semi-simple and semi-null splicing languages are extensively researched. Later, in 2001, Goode and Pixton [7] focused on the characterization of semi-simple splicing languages in terms of directed graphs. The relationship between semi-simple splicing language and constants are also studied. Ceterchi [8] focused on the algebraic characterization of that splicing system in 2006.

In this paper, the relations of four types of splicing systems namely, null-context, simple, semi-simple and semi-null, are presented.

3. PRELIMINARIES

This section includes some formal definitions used in this research. The first two definitions are on splicing system and splicing language.

Let *A* be defined as a fixed finite set to be used as an alphabet, *A** as a free monoid that consists of all strings of symbols in *A*, including the null string, and the symbol A^+ that denotes *A** but with exception of the null string. The definitions of splicing system and splicing language are given below.

Definition 1: [3] **(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 *cxd* 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. □

Definition 2: [3] **(Splicing Language)**

The language *L*(*S*) is the language generated by a splicing system *S* which consists of the strings in *I* and all strings that can be obtained by adjoining the words *ucxfq* and *pexdv* to *L* whenever *ucxdv* and *pexfq* are in *L*, and (*c*, *x*, *d*) and (*e*, *x*, *f*) are patterns of the same hand. A language *L* is a **splicing language** if there exists a splicing system *S* for which *L* = *L*(*S*). □

For a triple in a splicing system, some crossings are disjoint as mentioned in the next definition.

Definition 3: [9] **(Crossing Disjoint)**

A splicing system *S* = (*A, I, B, C*) is **crossing disjoint** if there do not exist patterns (*a*, *x*, *b*) in *B* and (*c*, *x*, *d*) in *C* with the same crossing *x*. Sometimes, a crossing is defined as a constant as given in the following definition.

Definition 4: [3] **(Constant)**

With respect to a language over *A,* a string *c* in *A** is a **constant** if, whenever *ucv* and *pcq* are in the language, *ucq* and *pcv* are also in the language. Next, the four types of splicing systems discussed in this paper are defined.

Definition 5: [6] **(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 6: [6] **(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 σ = (*A*, *R*) is called a semi-simple splicing scheme and (*A, I, R*) a **semi-simple splicing system**. □

Definition 7: [6] **(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 σ = (*A*, *R*) is called a semi-null splicing scheme, and (*A, I, R*) a **semi-null splicing system**. □

Definition 8: [3] **(Null-Context Splicing System)**

A **null-context splicing system** is a splicing system *S* = (*A*, *I*, *B*, *C*) for which each cleavage pattern in *B* and *C* has the form $(1, x, 1)$. □

As mentioned by Mateescu in [4] that, for every simple splicing system, the language generated is regular, the meaning of regular is defined below.

Definition 9: [10] **(Regular)**

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

In the next section, some relations of four types of splicing systems are presented.

4. RESULTS AND DISCUSSIONS

In this section, some relations on the four mentioned splicing systems in the previous section are presented as propositions and corollaries. The rule *R*, as in Head splicing system, will be represented in term of triples. Hence, based on the concept stated by Matesscu *et al.* in [4], the rule of simple, semi-simple and semi-null splicing system in Definitions 5, 6 and 7 can be presented as follows:

- Simple splicing system, $R = \{(1,a,1;1,a,1)\}$, where $a \in A$.
- Semi-simple splicing system, $S = \{(1,a,1;1,b,1)\}$, where $a,b \in A$.
- Semi-null splicing system, $S = \{(1,u,1;1,v,1)\}$, where $\mathrm{u}, v \in A^+$.

Simple and semi-simple splicing systems are related as follows:

Proposition 1

Every simple splicing system is semi-simple splicing system of the form (*A, I, S*). □

Proof

Suppose that *t* is not an element of a semi-simple splicing system. Hence, there exists a cleavage pattern in *S* that does not fulfill the form of $(1,a,1;1,b,1)$, where *a*, *b* are elements of *A*. Thus, *t* is not an element of a simple splicing system since there exist a cleavage pattern in *S* which is not in the form of $(1,a,1;1,a,1)$ or $(1,b,1;1,b,1)$, where *a, b* are elements of *A*. ■

In Proposition 2, semi-simple and semi-null splicing systems will be analyzed.

Proposition 2

Every semi-simple splicing system is semi-null splicing system of the form (*A, I, S*). □

Proof

Suppose that *t* is not an element of a semi-null splicing system. Thus, there exists a cleavage pattern in *S* that does not fulfill the form of $(1,u,1;1,v,1)$, for any *u*, *v* elements of A^+. Hence, *t* is not an element of a semi-simple splicing system since *A* is a subset of A^+. ■

However, the converse of Proposition 2 is not true as presented in Examples 1 and 2 below. Example 1 is a splicing system which has restriction enzymes *BssK*I and *Tsp*509I with 5′ overhangs; while Example 2 is a splicing system which has restriction enzymes *Hpy*99I and *Nla*III with 3′ overhangs. These two examples show that there exists a semi-null splicing system that is not semi-simple.

Example 1

Let $S = (\{a,g,c,t\}, I(\text{unspecified}), \{1,ccngg,1;1,ttaa,1\}, \varnothing)$ be a splicing system where *n* = *a* or *c* or *g* or *t*. The rule *B* consists of two restriction enzymes namely, *BssK*I and *Tsp*509I with the cleavage patterns as follows:

Cleavage pattern for the enzyme *BssK*I:

5'…▼*CCNGG* …3'
3'… *GGNCC*▲ …5' .

Cleavage pattern for the enzyme *Tsp*509I:

5'…▼*AATT* …3'
3'… *TTAA*▲…5' .

Thus, *S* is a semi-null splicing system since both disjoint crossings (*ccngg* and *aatt*) are elements of A^+. However, *S* is not semi-simple since *ccngg* and *ttaa* are not elements of *A*. ■

Example 2

Let $S = (\{a,g,c,t\}, I(\text{unspecified}), \varnothing, \{1,cgwcg,1;1,catg,1\})$ be a splicing system where *w* = *a* or *t*. The rule *C* consists of two restriction enzymes namely, *Hpy*99I and *Nla*III with the cleavage patterns as follows:

Cleavage pattern for the enzyme *Hpy*99I:

5'… *CGWCG*▼…3'
3'…▲*GCWGC* …5' .

Cleavage pattern for the enzyme *Nla*III:

5'… *CATG*▼…3'

$3'\ldots{}_{\blacktriangle}GTAC\ \ldots 5'$.

Thus, S is a semi-null splicing system since both disjoint crossings (*cgwcg* and *catg*) are elements of A^{+}. However, S is not semi-simple since *cgwcg* and *catg* are not elements of A. ■

Propositions 1 and 2 lead to Corollary 1.

Corollary 1

Every simple splicing system is semi-null. ■

In the next proposition, the relation between semi-null and null-context splicing systems is presented.

Proposition 3

Every semi-null splicing system is null-context splicing system of the form $S = (A, I, B, C)$. □

Proof

Suppose that t is not an element of a null-context splicing system. Hence, there exists a cleavage pattern in B or C that does not fulfill the form of $(1, x, 1)$. Thus, t is not an element of a semi-null splicing system by its form of pattern. ■

However, the converse of Proposition 3 is not true as presented in Examples 3 and 4 in the following. Example 3 is a splicing system which has restriction enzymes *Dpn*II and *Mbo*I with $5'$ overhangs; while Example 4 is a splicing system which has restriction enzymes *Nla*III and *Hin*1II with $3'$ overhangs. These two examples show that there exists a null-context splicing system that is not semi-null.

Example 3

In this example, it shows that there exists a null-context splicing system that is not semi-null. Let $S = (\{a, g, c, t\}, I(\text{unspecified}), \{1, gatc, 1; 1, gatc, 1\}, \varnothing)$ be a splicing system. The rule B consists of two restriction enzymes namely, *Dpn*II and *Mbo*I with the cleavage patterns as follows:

Cleavage pattern for the enzyme *Dpn*II:

$5'\ldots{}^{\blacktriangledown}GATC\ \ldots 3'$
$3'\ldots\ CTAG_{\blacktriangle}\ldots 5'$.

Cleavage pattern for the enzyme *Mbo*I:

$5'\ldots{}^{\blacktriangledown}GATC\ \ldots 3'$
$3'\ldots\ CTAG_{\blacktriangle}\ldots 5'$.

Thus, S is a null-context splicing system since cleavage pattern in B has the form $(1, x, 1)$. However, S is not semi-null since both restriction enzymes have the same crossing *gatc*. ■

Example 4

Let $S = (\{a, g, c, t\}, I(\text{unspecified}), \varnothing, \{1, catg, 1; 1, catg, 1\})$ be a splicing system. The rule C consists of two restriction enzymes namely, *Nla*III and *Hin*1II with the cleavage patterns as follows:

Cleavage pattern for the enzyme *Nla*III:

$5'\ldots\ CATG^{\blacktriangledown}\ldots 3'$
$3'\ldots{}_{\blacktriangle}GTAC\ \ldots 5'$.

Cleavage pattern for the enzyme *Hin*1II:

$5'\ldots\ CATG^{\blacktriangledown}\ldots 3'$
$3'\ldots{}_{\blacktriangle}GTAC\ \ldots 5'$.

Thus, S is a null-context splicing system since cleavage pattern in C has the form $(1, x, 1)$. However, S is not semi-null since both restriction enzymes have the same crossing *catg*. ■

Corollary 1 and Proposition 3 lead to Corollary 2.

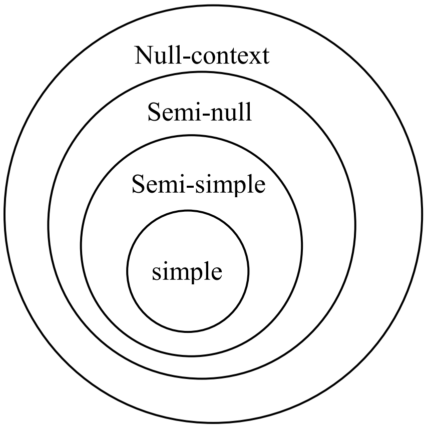
Corollary 2

Every simple splicing system is null-context. ■

5. CONCLUSION

In this paper, some relations on four types of splicing systems, namely null-context, simple, semi-simple and semi-null splicing systems are discussed and presented as Propositions 1, 2 and 3, Corollaries 1 and 2, and Examples 1 and 2. These relations can be simplified as follows:

simple splicing system $\subseteq$ semi-simple splicing system $\subseteq$ semi-null splicing system $\subseteq$ null-context splicing system,
or in the diagram below,



ACKNOWLEDGEMENT

We would like to thank the Ministry of Higher Education (MOHE) for the FRGS Vote No. 78482. The first author would also 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, Seventh Edition (2001).
[2] Research Biolabs Sdn. Bhd., New England Biolabs 2007-08 Catalog & Technical Reference, (2007).
[3] T. Head, Bull. Math. Biology, 49 (1987) 737-759.
[4] A. Mateescu, Gh. Paun, G. Rozenberg and A. Salomaa, Discrete Applied Mathematics, 84 (1998) 145-163.
[5] W. H. Fong, Ph.D. Thesis, Universiti Teknologi Malaysia, (2008).
[6] E. G. Laun, Ph.D. Thesis, State University of New York at Binghamton, (1999).
[7] E. Goode and D. Pixton, In: Martin-Vide, C. and Mitrana, V. eds. Where Mathematics, Computer Science, Linguistics and Biology Meet, (2001) 343-352.
[8] R. Ceterchi, Fundamenta Informaticae, 73(1-2) (2006) 19-25.
[9] R. W. Gatterdam, International Journal of Computer Math., 31(1989) 63-67.
[10] P. Linz, An Introduction to Formal Languages and Automata, Third Edition (2001)