ISSN 1823-626X

Journal of Fundamental Sciences

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

Mathematical Modelling of Some Null-Context and Uniform Splicing Systems

S. J. Lim[1], W. H. Fong[2*], N. H. Sarmin[1,2], F. Karimi[1]

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

Received 4 July 2011, Revised 25 August 2011, Accepted 10 September 2011, Available online 30 November 2011

ABSTRACT

The mathematical modelling of splicing system which involves recombination of DNA molecules was first introduced by Head in 1987. Splicing of DNA involves cutting of DNA molecules using the restriction enzymes and re-associating different fragments of DNA molecules using the ligase under some specific chemical conditions. A splicing language, L is generated if there exists a splicing system S for which $L = L(S)$. There are different types of splicing systems which have been discussed by various researchers. Among them are the persistent splicing system, null-context and uniform splicing system. In this paper, some molecular examples on null-context splicing system and uniform splicing systems with different initial strings and combination of restriction enzymes will be discussed. Applications of automata theory on some molecular examples of null-context and uniform splicing languages will also be presented in this paper.

| DNA| Null-Context | Uniform | Splicing System | Automata Theory |

http://dx.doi.org/10.11113/mjfas.v7n2.254

1. INTRODUCTION

Splicing system is a system involving a finite set of initial strings over an alphabet with a finite set of rules. A language is associated with each pair of sets where the first set consists of double-stranded DNA molecules and the second set consists of the recombination behavior allowed by specified classes of enzymatic activities. A new relationship between formal language theory and the study of macromolecules was thus established. A formal language is an abstraction of general characteristics of programming language which consists of a set of all sentences with rules of formation [1]. The set of double-stranded DNA molecules that may arise from an initial set of DNA molecules in the presence of specified restriction enzymes activities is represented as a language over the four-symbol alphabet of deoxyribonucleotide pairs, *a*, *g*, *c*, and *t* which denotes adenine, guanine, cytosine, and thymine respectively [2].

Formal language theory is a division of theoretical computer science and discrete mathematics which is devoted to the study of sets of finite strings from a prescribed finite set as defined in [3,4]. There are different types of splicing languages, including simple splicing languages, semi-simple splicing languages, persistent splicing language, strictly locally testable language, uniform splicing language and null-context splicing languages which have been discussed in [5-9]. In this paper, two types of splicing systems namely null-context and uniform splicing systems will be studied.

2. PRELIMINARIES

In this section, some main definitions used in this research are listed. The formal definitions of splicing system and splicing language are stated below.

Definition 1 [2] **(Splicing system, Splicing language)**
A **splicing system** $S = (A, I, B, C)$ consists of a finite alphabet A, a finite set I of initial strings in A^*, where A^* is denoted by the free monoid over A [3] 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 pattern. For each such triple the string cxd is called a site and the string x is called a crossing. Pattern in B are called left patterns and patterns in C are called right patterns. The language $L = L(S)$ generated by S 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 $L = L(S)$.
The definition of null-context and uniform splicing system is given in the following.

Corresponding author at: Ibnu Sina Institute for Fundamental Science Studies,Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Malaysia
E-mail addresses: : fwh@ibnusina.utm.my (Fong Wan Heng)

Definition 2 [2] (**Null-context splicing system, Null-context splicing language**)
A **null-context splicing system** is a splicing system $S = (A, I, B, C)$ for which each cleavage pattern in B and each in C has the form $(1, x, 1)$. A language L is a **null-context splicing language** if there is a null-context splicing system S for which $L = L(S)$.

Definition 3 [2] (**Uniform splicing system, Uniform splicing language**)
A **uniform splicing system** is a null-context splicing system $S = (A, I, X, X)$ for which there is a positive integer P such that $X = A^p$. A language L is a **uniform splicing language** if there is a uniform splicing system S for which $L = L(S)$.

Below is the concept of crossing disjoint which is also mentioned in the examples for uniform splicing systems.

Definition 4 [2] (**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.

The definitions of deterministic finite accepter (dfa), nondeterministic finite accepter (nfa) and the language L accepted by dfa and nfa are needed for concepts of automata theory in the next section, and are presented in the following.

Definition 5 [2] (Deterministic finite accepter)
A **deterministic finite accepter** or **dfa** is defined by the 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,
$\delta : Q \times \Sigma \rightarrow Q$ is a total function called the transition function,
$q_0 \in Q$ is the initial state,
$F \subseteq Q$ is a set of final states.

Definition 6 [2] (Nondeterministic finite accepter)
A **nondeterministic finite accepter** or **nfa** is defined by the quintuple $M = (Q, \Sigma, \delta, q_0, F)$, where Q, Σ, δ, q_0, F are defined as deterministic finite accepters, but $\delta : Q \times (\Sigma \cup \{\lambda\}) \rightarrow 2^Q$.

Next, the definition of regular is stated below.

Definition 7 [1] (**Regular**)
A language L is called **regular** if and only if there exists some deterministic finite accepter M such that $L = L(M)$.

In the next section, some molecular examples of null-context and uniform splicing systems will be provided.

3. SOME MOLECULAR EXAMPLES OF NULL-CONTEXT AND UNIFORM SPLICING SYSTEMS

Some molecular examples of null-context and uniform splicing systems with different initial strings and different combination of restriction enzymes are discussed in this section. Examples 1 to Example 6 show some molecular examples of null-context and uniform splicing systems. Example 1 and Example 2 show two splicing systems having one initial string each. However, there is only one restriction enzyme involved in Example 1 but two restriction enzymes involved in Example 2.

Example 1.
Let $S = (A, I, B, C)$ be a splicing system where $I = \{aggacatggttccaactc\}$ is the set consisting of one initial string. The set B, $\{(1, catg, 1)\}$, consists of the restriction enzyme *Fat*I with the left cleavage pattern on 5′ overhangs.

Using the initial string that is *aggacatggttccaactc* with the restriction enzyme *Fat*I, the cutting site of restriction enzyme is shown below:

5′-*AGGA*▼*CATGGTTCCAACTC*-3′
3′-*TCCTGTAC*▲*CAAGGTTGAG*-5′,

This cutting sites can also be viewed as

5′-*GAGTTGGAAC*▼*CATGTCCT*-3′
3′-*CTCAACCTTGGTAC*▲*AGGA*-5′

in the opposite direction.

The language resulting from this splicing system is $L = \{aggacatgtcct, aggacatggttccaactc, gagttggaaccatggttccaactc\}$.

Example 2
Let $S = (A, I, B, C)$ be a splicing system where $I = \{aagatcggcgatcttcct\}$ is the set consisting of one initial string. The set B, $\{(1, gatc, 1); (1, gatc, 1)\}$, is the set consisting of the restriction enzymes *Mbo*I and *Dpn*II respectively with the left cleavage pattern on 5′ overhangs.

Considering the initial molecule that is *aagatcggcgatcttcct* with the restriction enzymes *Mbo*I and *Dpn*II, the cutting sites of restriction enzymes are shown below:

5′-*AA*▼*GATCGGC*▼*GATCTTCCT* -3′
3′-*TT CTAG*▲*CCGCTAG*▲*AAGGA*-5′.

The language resulting from this splicing system is $L = \{aa(gatcggc \cup gatcgcc)^* gtacttcct\}$.

Next, Example 3 to Example 6 show four splicing systems having two initial strings each. There is only one restriction enzyme involved in Example 3, but two

restriction enzymes with the same crossings involved in Example 4, which are shown in the following.

Example 3
Let $S = (A, I, B, C)$ be a splicing system where $I = \{ggcaattgctgcagtgcc, acgcgtatgtaattccgga\}$ is the set consisting of two initial strings. The set B, $\{(1, aatt, 1)\}$ consists of the restriction enzyme *Mlu*CI with the left cleavage pattern on 5′ overhangs.

Using the initial strings that are *ggcaattgctgcagtgcc* and *acgcgtatgtaattccgga* with the restriction enzyme *Mlu*CI, the cutting sites of enzyme are shown below:

5′-*GGC*$^{\blacktriangledown}$*AATTGCTGCAGTGCC* -3′ and
3′-*CCGTTAA*$_{\blacktriangle}$*CGACGTCACGG* -5′

5′-*ACGCGTATGT*$^{\blacktriangledown}$*AATTCCGGA* -3′
3′-*TGCGCATACATTAA*$_{\blacktriangle}$*GGCCT* -5′.

Hence, the language resulting from this splicing system is $L = \{ggcaattccgga, ggcaattacatacgcgt, acgcgtatgtaattgctgcagtgcc\}$.

Example 4
Let $S = (A, I, B, C)$ be a splicing system where $I = \{agtgaaattggactccgat, cctaggactgaattcgac\}$ is the set consisting of two initial strings. The set B, $\{(1, aatt, 1); (1, aatt, 1)\}$ is the set consisting of restriction enzymes *Mlu*CI and *Tsp*509I with the left cleavage pattern on 5′ overhangs.

Considering the initial molecules that are *agtgaaattggactccgat* and *cctaggactgaattcgac* with the restriction enzymes *Mlu*CI and *Tsp*509I, the cutting sites of restriction enzymes are shown below:

5′-*AGTG*$^{\blacktriangledown}$*AATTGGACTCCGAT*-3′ and
3′-*TCACTTAA*$_{\blacktriangle}$*CCTGAGGCTA*-5′

5′-*CCTAGGACTG*$^{\blacktriangledown}$*AATTCGAC* -3′
3′-*CCATCCTGACTTAA*$_{\blacktriangle}$*GCTG* -5′.

The language resulting from this splicing system is $L = \{agtgaattcgac, agtgaattcagtcctagg, atcggagtccaattcgac, atcggagtccaattcagtcctagg\}$.

Meanwhile, Example 5 shows a splicing system involving two restriction enzymes of different crossings; while Example 6 shows a splicing system involving two restriction enzymes of same crossings. Example 6 differs from Example 4 in that there are two cutting sites present for each initial string in Example 6.

Example 5
Let $S = (A, I, B, C)$ be a splicing system where $I = \{agtgaaattggactccgat, aaggatcttgtcacaat\}$ is the set consisting of two initial strings. The set B, $\{(1, aatt, 1); (1, gatc, 1)\}$, is the set consisting of restriction enzymes *Mlu*CI and *Mbo*I with the left cleavage pattern on 5′ overhangs.

Considering the initial molecules that are *agtgaaattggactccgat* and *aaggatcttgtcacaat* with the restriction enzymes *Mlu*CI and *Mbo*I, the cutting sites of restriction enzymes are shown below:

5′-*AGTG*$^{\blacktriangledown}$*AATTGGACTCCGAT*-3′ and
3′-*TCACTTAA*$_{\blacktriangle}$*CCTGAGGCTA*-5′

5′-*AAG*$^{\blacktriangledown}$*GATCTTGTCACAAT* -3′
3′-*TTCCTAG*$_{\blacktriangle}$*AACAGTGTTA* -5′.

Hence, the language resulting from this splicing system is $L = \{agtgaattggactccgat, aaggatcttgtcacaat\}$, which are the two initial strings.

Example 6
Let $S = (A, I, B, C)$ be a splicing system where $I = \{ggtcatgcttgacatgaa, cggtccatgtagccatgt\}$ is the set consisting of two initial strings. The set B, $\{(1, catg, 1); (1, catg, 1)\}$, is the set consisting of restriction enzymes *Nla*III and *Hin1*II with the right cleavage pattern on 3′ overhangs.

Considering the initial molecules that are *ggtcatgcttgacatgaa* and *cggtccatgtagccatgt* with the restriction enzymes *Nla*III and *Hin*1II, the cutting sites of restriction enzymes are shown below:

5′-*GGTCATG*$^{\blacktriangledown}$*CTTGACATG*$^{\blacktriangledown}$*AA* -3′ and
3′-*CCA*$_{\blacktriangle}$*GTACGAACT*$_{\blacktriangle}$*GTACTT* -5′

5′-*CGGTCCATG*$^{\blacktriangledown}$*TAGCCATG*$^{\blacktriangledown}$*T* -3′
3′-*GCCAG*$_{\blacktriangle}$*GTAGATCG*$_{\blacktriangle}$*GTACA*-5′.

Hence, the language resulting from this splicing system, $L = \{(ggt \cup cggtc)\ catg\ [(cttga \cup tagc \cup tcaag \cup gcta)\ catg\ (cttga \cup tagc \cup tcaag \cup gcta)]^n\ (gaccg \cup acc \cup t \cup aa)\}$.

Next, Example 7 shows a null-context splicing system that is not uniform.

Example 7
Let $S = (A, I, B, C)$ be a splicing system where $I = \{ggaattcgatcaattgcc, aaccaggttatccaggttg\}$ is the set consisting of two initial strings. The set B, $\{(1, aatt, 1)$;$(1, aatt, 1); (1, ccngg, 1); (1, ccwgg, 1)\}$, where $n = a$ or c or g or t and $w = a$ or t. The set B consists of the enzyme *Mlu*CI, *Tsp*509I, *Bss*KI and *Psp*GI respectively, with the left cleavage pattern on 5′ overhangs.

Using the first initial string that is *ggaattcgatcaattgcc* with the restriction enzymes *Mlu*CI and *Tsp*509I, the cutting sites of restriction enzymes are shown below:

5′-*GG*$^{\blacktriangledown}$*AATTCGATC*$^{\blacktriangledown}$*AATTGCC* -3′ , and
3′-*CCTTAA*$_{\blacktriangle}$*GCTAGTTAA*$_{\blacktriangle}$*CGG* -5′

5′-*GG*▼*AATTCGATC*▼*AATTGCC* -3′
3′-*CCTTAA*▲*GCTAGTTAA*▲*CGG* -5′.

Hence, the language resulting from this splicing system is $L = \{gg(aattgatcg + aattcgatc)^n aattgcc, aaccagg(ttatccagg)^n ttg\}$. This language is recognized as null-context splicing language since it is generated from a null-context splicing system. However, this language is not uniform since the restriction enzymes have crossings of different length.

Null-context and uniform splicing languages can also be applied using automata theory.

4. APLLICATIONS OF AUTOMATA THEORY ON SOME NULL-CONTEXT AND UNIFORM SPLICING LANGUAGE

Automata theory is a mathematical model of computing. An automaton is an abstract model of a digital computer. It is a simple machine which is used for recognizing the languages. An automaton has a finite set of states, one which is designated as the initial state and some of which are designated as final state. The input is a string over a given alphabet [1]. Transition function is the rule for moving from one state to another state. The output of an automaton is called an accepter. Accepter states are sometimes called final states [3]. In this research, the discussion is the finite automata in automata theory. Finite automata are the simplest abstract computational device [4]. There are two types of automata which are deterministic finite accepter (dfa) and non-deterministic finite accepter (nfa).

A language L is called regular if and only if there exists some deterministic finite accepter M such that $L = L(M)$. However, some deterministic and non-deterministic finite accepter can recognize the same class of language, thus a language accepted by some non-deterministic finite accepter is also regular. Hence every regular language can be described by some dfa or some nfa [1]. Finite automaton can recognize languages and thus a finite automaton diagram can be constructed for some given languages.

In the next section, applications of automata theory on some molecular examples of uniform splicing system will be presented.

From Example 1, the language that results from the splicing system is $L = \{aggacatgtcct, aggacatggttccaactc, gagttggaaccatggttccaactc\}$. The regular expression for this language is $(agga + gagttggac)\ catg\ (tcct + gttccactc)$. The non-deterministic automaton diagram for this regular language is shown in Figure 1.

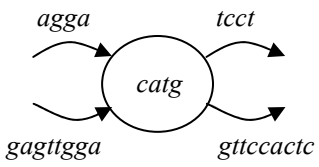

Fig. 1 Automaton diagram for the language in Example 1.

From Example 2, the language that results from the splicing system is $L = \{aa(gatcggc \cup gatcgcc)^n gtacttcct$. The regular expression for this language is $aa(gatcggc + gatcgcc)^* gtacttcct\}$. Thus, the non-deterministic automaton diagram for this regular language is shown in Figure 2.

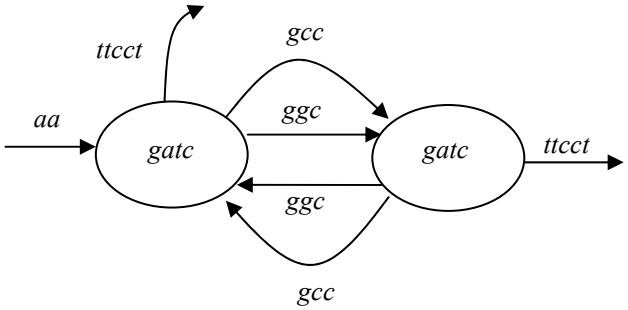

Fig. 2 Automaton diagram for the language in Example 2.

From Example 3, the language that results from the splicing system is $L = \{ggcaattccgga, ggcaattacatacgcgt, acgcgtatgtaattgctgcagtgcc\}$. The regular expression for this language is $\{(ggc + acatacgcgt)\ aatt\ (acatacgcgt + ccgga + gctgcagtgcc)$. Thus, the non-deterministic automata diagram for this regular language is shown in Figure 3.

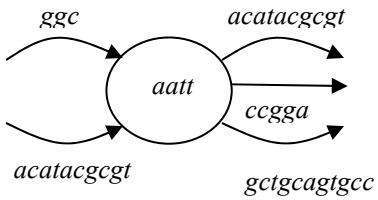

Fig. 3 Automaton diagram for the language in Example 3.

From Example 4, the language that results from the splicing system is $L = \{agtgaattcgac, agtgaattcagtcctagg, atcggagtccaattcgac, atcggagtccaattcagtcctagg\}$. The regular expression for this language is $(agtg + atcggatcc)\ aatt\ (cgac + cagtcctagg)$. Thus, the non-deterministic automaton diagram for this regular language is shown in Figure 4.

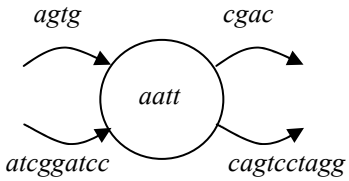

Fig. 4 Automaton diagram for the language in Example 4.

From Example 5, the language that results from the splicing system is $L = \{agtgaattggactccgat, aaggatcttgtcacaat\}$. The regular expression for this language is $\{(agtg(aatt)ggactccgat, aag(gatc)ttgtcacaat)\}$. Thus, the deterministic automata diagram for this regular language is shown in Figure 5.

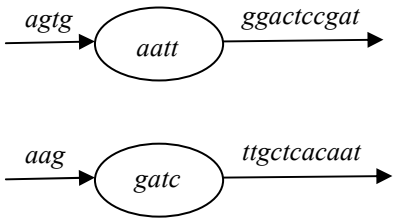

Fig. 5 Automata diagram for the language in Example 5.

From Example 6, the language that results from the splicing system is $L = \{(ggt \cup cggtc)\ catg\ [(cttga \cup tagc \cup tcaag \cup gcta)\ catg\ (cttga \cup tagc \cup tcaag \cup gcta)]^n(gaccg \cup acc \cup t \cup aa)\}$. The regular expression for this language is $\{\{(ggt + cggtc)\ catg\ [(cttga + tagc + tcaag + gcta)\ catg\ (cttga + tagc + tcaag + gcta)]^*(gaccg + acc + t + aa)\}$.Thus, the non-deterministic automaton diagram for this regular language is shown in Figure 6.

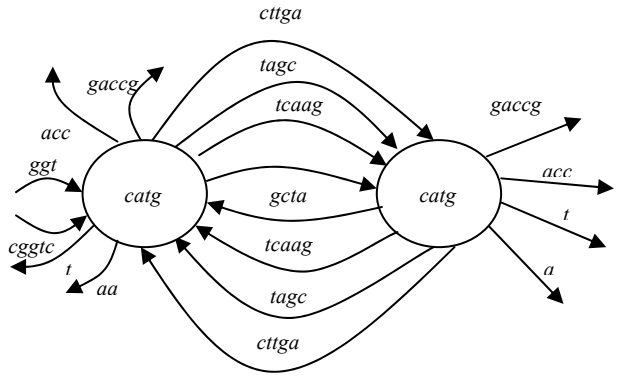

Fig. 6 Automaton diagram for the language in Example 6.

From Example 7, the language that results from the splicing system is $L = \{gg(aattgatcg + aattcgatc)^n aattgcc, aaccagg(ttatccagg)^n ttg\}$. The regular expression for this language is $\{gg(aattgatcg + aattcgatc)^* aattgcc, aaccagg(ttatccagg)^* ttg\}$. Thus, the nondeterministic automata diagram for this regular language is shown in Figure 7.

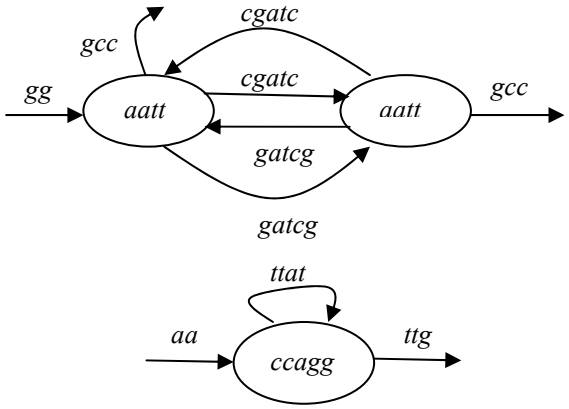

Fig. 7 Automata diagram for regular language in Example 7.

5. CONCLUSION

In this paper, some molecular considerations of null-context and uniform splicing systems are discussed Some molecular examples on null-context and uniform splicing system with different initial strings and different combination of enzymes have been presented. In the last section, applications of automata theory on some molecular examples of null-context and uniform splicing languages are included.

ACKNOWLEDGEMENTS

We would like to thank The Ministry of Education (MOHE) and the Research Management Centre (RMC), UTM for the financial funding through Research University Fund Vote No. 02J65. The fourth author would also like to acknowledge Universiti Teknologi Malaysia for the financial support through the International Fellowship Fund.

REFERENCES

[1] Linz, P. (2006). *An Introduction to Formal Languages and Automata* (4th ed.) USA: Jones and Bartlett Publisher, Inc.
[2] Head, T. Formal Language Theory and DNA, *Bulletin of Mathematical Biology*. 49 (1987) 737-759.
[3] Shallit, J. (2009). *A Second Course in Formal Languages and Automata Theory. New York*: Cambridge University Press.
[4] Hopcroft, J. E., Motwani. R., Ullman, J. D. (2007). *Introduction To Automata Theory, Language and Computation* (3rd ed.). Boston, MA. Pearson Education,Inc.
[5] Fong, W. H., Sarmin. N. H., Yusof. Y. and Karimi. F, *Journal of Fundamental Sciences*, 6 (2010) 142-146.
[6] Fong, W. H. *Modelling of Splicing Systems Using Formal Language Theory*. Ph.D. Thesis. Universiti Teknologi Malaysia; 2008.
[7] Karimi, F., Sarmin, N.H., Fong, W.H., Australian Journal of Basic and Applied Sciences, 5 (2011) 20-24.
[8] Head, T., Discrete Applied Mathematics, 87 (1998) 139-147.
[9] Karimi, F., Sarmin, N.H., Fong, W.H., *The Characterizations of Different Splicing Systems*, The International conference on Mathematical and Computational Biology 2011(ICMCB 2011), Renaissance Melaka Hotel, Melaka, Malaysia, 12-14 April 2011