An Introduction to *n*-th Order Limit Language

Siti Hajar Mohd Khairuddin, Muhammad Azrin Ahmad, and Noraziah Adzhar

***Abstract*—The study of the splicing system swiftly grew after Tom Head investigated the biochemical process modelling involving the DNA in 1987. The process of the splicing system consists of a cut and paste of the DNA molecules. Splicing language produced by the splicing system can be classified into inert, transient, and limit language. Previously, second-order limit language was described as a new set of language from the previous splicing language. In this research, we would like to extend the study to *n*-th order limit language by investigating the effect of the number of rules involved in the splicing system. Following from here, its properties are explored using the formal language theory.**

***Index Terms*—DNA, *n*-th order limit language, splicing system, splicing language,**

I. Introduction

THE deoxyribonucleic acid (DNA) is a natural compound spotted in all prokaryotic and eukaryotic cells with a complex atomic structure [1]. The DNA is a particle in charge of conveying and transmitting the inherited materials or the hereditary guidelines from the guardians to the offspring and for the generation of proteins. Hence, DNA is found in every living life form and is principally in charge of hereditary data's legacy in every living being. The cells in the human body have a similar DNA placed in the core of cells known as the nucleus, which is sometimes located in the mitochondrion [2]. The DNA structure is reflected in Figure 1 as follows.

Manuscript received January 18, 2021; revised September 10, 2021. This work was supported in part by the Minister of Higher Education (MOHE) Malaysia for the financial funding through Fundamental Research Grant Scheme (FRGS), FGRS/1/2018/STG06/UMP/03/1.

Siti Hajar Mohd Khairuddin is a postgraduate student at the Centre for Mathematical Sciences, Universiti Malaysia Pahang, 26300 Lebuhraya Tun Razak, UMP Gambang, Pahang, Malaysia (corresponding author to provide phone: +601112914899; e-mail: sitihajarmohdkhairuddin96@gmail.com).

Muhammad Azrin Ahmad is a senior lecturer at the Centre for Mathematical Sciences, Universiti Malaysia Pahang, 26300 Lebuhraya Tun Razak, UMP Gambang, Pahang, Malaysia (e-mail: azrin@ump.edu.my).

Noraziah Adzhar is a senior lecturer at the Centre for Mathematical Sciences, Universiti Malaysia Pahang, 26300 Lebuhraya Tun Razak, UMP Gambang, Pahang, Malaysia (e-mail: noraziahadzhar@ump.edu.my).

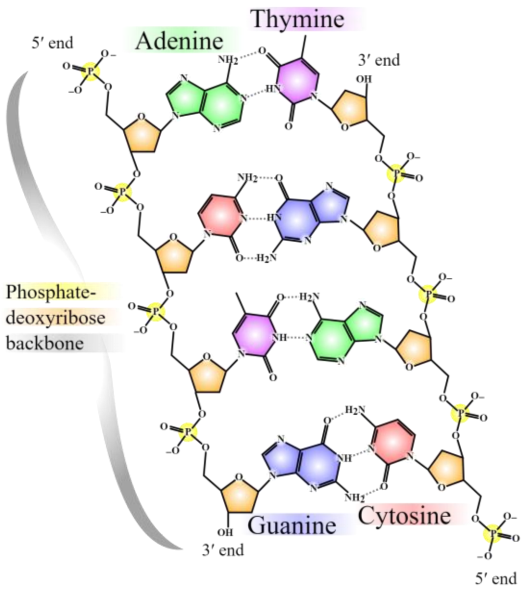

Figure 1. The Structure of the DNA

The DNA structure is a long, twofold helix that takes after a stepping stool, turned at both ends, as portrayed in Figure 1. It is known as nucleic acids, comprising nucleotides. Nucleotides contain three essential substances: a phosphate group, a sugar group, and nitrogenous bases [3], as shown in Figure 2. The sugar element essentially has five carbon atoms numbered between 5' and 3'. The phosphate is attached to the 5' carbon binding the base to the 1' carbon, while the 3' carbon is attached to a hydroxyl group (OH) [4].

The DNA mechanism follows the Watson-Crick complementary, where Adenine (A) is combined with Thymine (T) and Guanine (G) coupled with Cytosine (C), and vice versa (Watson & Crick, 1953). Therefore, it can be paired and represented as *a*, *g*, *c*, *t*, as illustrated below.

A	**G**	**C**	**T**
T	**C**	**G**	**A**

Moreover, the enzyme that divides DNA into fragments at the recognition site is called restriction enzyme or restriction endonuclease or restrictase. Restriction enzymes distinguish a specific sequence, where every restriction enzyme(s) has its targeted site, facilitating it to bind to the molecules and then splice them [5].

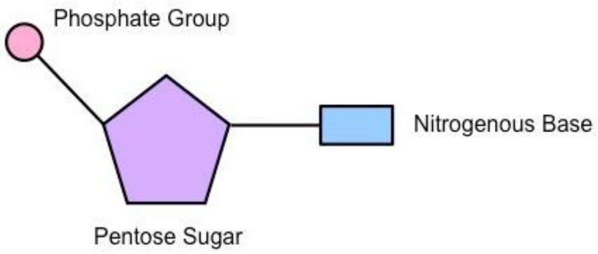

Figure 2. The Structure of Nucleotides

However, the knowledge of DNA can be applied in such a way that incorporates formal language theory with molecular biology. Formal language theory is one of the branches of computational science defining the analysis of finite string sets of symbols (called languages) selected from a defined finite set (called alphabets). Besides, the molecular biology relation in this field is the study of informational macromolecules. Other than that, Head implemented a formal introduction that incorporates knowledge of the formal language theory, which then contributed to the mathematical modelling of the splicing system [6]. There are five splicing systems known as Head [6], Paun (Paun *et al.*, 1996), Pixton [8], Goode-Pixton (Goode, 2004), Yusof-Goode [10] and fuzzy splicing systems [11]. A splicing system is defined as the process modelled from a biological perspective that is also a recreation of the new hybrid DNA (Goode, 1991).

Splicing language is a product resulting from splicing the DNA molecules analyses using the formal language theory. In 2004, Goode proposed that splicing language can be classified into inert/adult, transient, and limit language [9]. This paper focuses on the limit language, initially introduced by Goode [9] after considering the DNA molecules behaviour at the final stage. Following from here, the wet splicing procedure, which is a laboratory process of splicing the DNAs, was conducted. After the reaction completes or has reached a limit, the DNA molecules were examined and analysed, known as limit language. Moreover, this language is also used to develop the *n*-th order limit language.

There are a variety of splicing languages that have been previously introduced. For example, in 2012, Yusof introduced the splicing language called active persistence language [10]. Other than that, in 2014, a new language was introduced from the fuzzy splicing system [11], where the language is fuzzy and crisp. Besides that, in 2017, another splicing language called two stages splicing language was introduced in (Mudaber et al., 2017). Furthermore, the splicing language had been discussed in many ways, such as its subclass of splicing language in [14], the predictor of splicing language by using C++ in [15], computing the splicing language from DNA splicing systems (Ismail et al., 2018) and others. Therefore, the idea of previous researchers regarding languages of the splicing system enhances many more researchers to continue the research of splicing languages, such as the order of limit language, the hierarchy of splicing language, etc.

Besides, Ahmad proposed a second-order limit language based on the first-order limit language [17], which motivates the discussion on the *n*-th order limit language further explored in this paper.

The basic definitions used in this paper are provided in the next section.

II. Preliminaries

In this section, the ideas related to splicing languages are described.

Alphabets, *A* [18]
An alphabet, A is a finite non-empty set of symbols.

Strings [18]
A string is a finite sequence of symbols from the alphabet *A*.

Language, *L* [18]
A set of strings, all of which are chosen from A^*, where A is a particular alphabet called language.

Head Splicing System 4 [6]
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, while the string x is called a crossing. Patterns in B are called left patterns, while patterns in C are called right patterns. The string cxd shows that the alphabets c and d are called left context and the right context, respectively. The Head splicing system comprises four different sets of elements A,I,B and C, elaborated as follows.

A is a set of alphabet,
I is a set of initial strings,
B is a set of rules, representing a $5'$-overhang or blunt end,
C is a set of rules, representing a $3'$-overhang.

Next, the definition of limit, second-order limit language and *n*-th order limit language is presented.

Limit Language [9]
Limit language, better known as first-order limit language, is defined as a splicing language that results from the remaining molecules after the splicing system has reached its equilibrium state or is completed.

Second Order Limit Language [19]
Let $L(S)$ be the splicing language of a splicing system S and $L_1(S)$ be the first order limit language. A splicing language is called a second-order limit language, denoted by $L_2(S)$, if the set of string produced in $L_2(S)$ is distinct from the set of strings of $L(S)$. In other words, $L_2(S)\cap L(S)=\varnothing$, $L_2(S)\cap L_1(S)=\varnothing$, and $L_2(S)\not\subset L(S)$.

The derivation of the *n*-th order limit language is obtained from the the combination of the idea of the second-order limit language and *n*-th order limit language from Goode [9] and Ahmad [19]. Furthermore, the definition of *n*-th order limit language focuses on the crossing part segment. The following is the definition of *n*-th order limit language.

***n*-th order Limit Language [9]**

Let L_{n-1} be the set of second-order limit words of L, the set L_n of *n*-order limit words of L to be the set of the first-order limit of L_{n-1}. We obtain L_n from L_{n-1} by deleting the words that are transient in L_{n-1}.

Then, the original definition is improvised as follows.

Let $L(S)$ be the splicing language of the splicing system S. We then define $L_n(S)$ such that n represents the order of the limit language. Initial strings of the splicing system S consist of *cxd*, where c and d are the left and right content, respectively and x is the crossing site. The *n*-th order limit language is defined by the number of rules that act on each crossing sites, x in which the set of rules is different from each other. Note that the rules must have the same length of crossing sites. A splicing language is called *n*-th order limit language, denoted by $L_n(S)$, if the set of string produce in $L_n(S)$ is distinct from the set of strings of $L_1(S), L_2(S), \ldots, L_n(S)$ such that $\bigcap_{n=1}^{n} L_n(S) = \varnothing$ and $L_1(S) \not\subset L_2(S) \not\subset \ldots \not\subset L_n(S)$. [20]

III. Main Result

In this section, a theorem on the formation of the *n*-th order limit language is presented, where each rule has the same length of the crossing site. The proof of the splicing model uses only one initial string. The same method can be used for *m* number of initial strings, which does not affect the outcome of the combination of string.

Theorem 1: If a splicing system contains *m* number of initial strings and *p* number of rules, then the splicing system generates the *n*-th order limit language.

Proof Suppose that *n*-th order limit language is produced. Next, assume that there is *m* number of initial strings and *p* number of rules involved. The following six cases need to be considered; namely Case 1: one initial string and a rule; Case 2: one initial string and two rules; Case 3: one initial string and three rules; Case 4: one initial string and *p* rule; Case 5: two initial strings and *p* rules and Case 6: *m* initial strings and *p* rules.

Case 1 : An initial string and a rule

Let $S = \left(\{a,g,c,t\},\{\mu vwyy\gamma\},\{v,wy,y\},\varnothing\right)$, where v and y, and w and x, are the complementary to one another and $\mu,\gamma,v,w,x,y \in A^*$. The initial string is presented below.

$$\begin{array}{c} 5'-\mu v^{\blacktriangledown} wyy\gamma - 3' \\ 3'-\mu' yxv_{\blacktriangle} v\gamma' - 5' \end{array}.$$

The following is the splicing language that is generated after the splicing process:

$$L(S) = \{\mu vwyy\gamma, \mu vvxy\mu', \gamma' vwyy\gamma\},$$

where $\mu',\gamma' \in A^*$.

Since the order is defined by the number of rules, the first order limit language or the limit language is presented below.

$$L_1(S) = \{\mu vvxy\mu', \gamma' vwyy\gamma\}.$$

Case 2: An initial string and two rules

Let

$S = \left(\{a,c,g,t\},\{\mu vwyyvwwx\gamma\},\{(v,wy,y),(v,ww,x)\},\varnothing\right)$,

where v and y, and w and x, are complementary to one another and $\mu,\gamma,v,w,x,y \in A^*$. The initial string is presented below.

$$\begin{array}{c} 5'-\mu v^{\blacktriangledown} wyyv^{\blacktriangledown} wwx\gamma - 3' \\ 3'-\mu' yxv_{\blacktriangle} vyxx_{\blacktriangle} w\gamma' - 5' \end{array}.$$

The following is the splicing language that is generated after the splicing process.

$$L(S) = \left\{\begin{array}{l} \mu vwwx\gamma, \mu vvxy\mu', \gamma' wwwx\gamma, \mu vxxyvvxy\mu', \\ \mu vwyyvwwx\gamma, \ldots \end{array}\right\},$$

where $\mu',\gamma' \in A^*$.

Since the order is defined by the number of rules, the second-order limit language is presented below.

$$L_2(S) = \left\{\begin{array}{l} \mu v(wyyv \cup xxyv)^* wwx\gamma, \\ \mu v(wyv \cup xxyv)^* vxy\mu', \\ \gamma' w(wyyv \cup xxyv)^* wwx\gamma \end{array}\right\},$$

where the symbol $(*)$ and $(\cup)$ in regular expression denotes that the string $wyyv$ and $xxyv$ can occur recursively alternately, respectively.

Case 3: **An initial string and three rules**

Let $S = \left(\begin{array}{l} \{a,c,g,t\},\{\mu vwyyvwwxvwxy\gamma\}, \\ \{(v,wy,y),(v,ww,x),(v,wx,y)\},\varnothing \end{array}\right)$, where v and y, and w and x, are complementary to one another and $\mu,\gamma,v,w,x,y \in A^*$. The initial string is presented below.

$$\begin{array}{c} 5'-\mu v^{\blacktriangledown} wyyv^{\blacktriangledown} wwxv^{\blacktriangledown} wxy\gamma - 3' \\ 3'-\mu' yxv_{\blacktriangle} vyzz_{\blacktriangle} wyxw_{\blacktriangle} v\gamma' - 5' \end{array}.$$

The following is the splicing language that is generated after the splicing process.

$$L(S) = \left\{\begin{array}{l} \mu vwxy\gamma, \mu vvxy\mu', \gamma' vwxy\gamma, \mu vwyyvwxy\gamma, \\ \mu vwyyvvxy\mu', \gamma' vwyyvwxy\gamma, \mu vwyyvwwxvwxy\gamma, \\ \mu vwyyvwwxwwyz\mu', \gamma' vwyyvwwxvwxy\gamma, \\ \mu vwyyvwwxvwxywwxy\gamma, \mu vwyyvwwxvwxywvxy\mu', \\ \gamma' vwyyvwwxvwxywwxy\gamma, \\ \mu vwyyvwwxvwxywxxywvxy\gamma, \\ \mu vwyyvwwxvwxywxxywwxy\mu', \\ \gamma' vwyyvwwxvwxywxxywvxy\gamma, \ldots \end{array}\right\},$$

where $\mu',\gamma' \in A^*$.

Since the order is defined by the number of rules, the third-order limit language is presented below.

$$L_3(S)=\begin{Bmatrix}\mu v(wyyv\cup wwxv\cup wxyw\cup xxyv)^*wxy\gamma,\\ \mu v(wyyv\cup wwxv\cup wxyw\cup xxyv)^*vxy\mu',\\ \gamma'v(wyyv\cup wwxv\cup wxyw\cup xxyv)^*wxy\gamma\end{Bmatrix}.$$

Case 4: An initial string and *p* number of rules

Let

$$S=\begin{pmatrix}\{a,c,g,t\},\{\mu vwyyvwwxvwxy\ldots\gamma\},\\ \{(v,wy,y),(v,ww,x),(v,wx,y),\ldots,(a_p,x_p,b_p)\},\varnothing\end{pmatrix},$$

where v and y, and w and x, are complementary to one another and $\mu,\gamma,v,w,x,y\in A^*$. The initial string is presented below.

$$\begin{matrix}5'-\mu v^{\blacktriangledown}wyyv^{\blacktriangledown}wwxv^{\blacktriangledown}wxy\ldots\gamma-3'\\ 3'-\mu'yxv_{\blacktriangle}vyzz_{\blacktriangle}wyxw_{\blacktriangle}v\ldots\gamma'-5'\end{matrix}.$$

The following is the splicing language that is generated after the splicing process:

$$L(S)=\begin{Bmatrix}\mu vwxy\gamma,\mu vvxy\mu',\gamma'vwxy\gamma,\mu vwyyvwxy\gamma,\\ \mu vwyyvvxy\mu',\gamma'vwyyvwxy\gamma,\mu vwyyvwwxvwxy\gamma,\\ \mu vwyyvwwxwwyz\mu',\gamma'vwyyvwwxvwxy\gamma,\\ \mu vwyyvwwxvwxywwxy\gamma,\mu vwyyvwwxvwxywvxy\mu',\\ \gamma'vwyyvwwxvwxywwxy\gamma,\\ \mu vwyyvwwxvwxywxxywvxy\gamma,\\ \mu vwyyvwwxvwxywxxywwxy\mu',\\ \gamma'vwyyvwwxvwxywxxywvxy\gamma,\ldots\end{Bmatrix},$$

where $\mu',\gamma'\in A^*$.

Since the order is defined by the number of rules, the *n*-th order limit language is presented below.

$$L_n(S)=\begin{Bmatrix}\mu v(wyyv\cup wwxv\cup wxyw\cup xxy\ldots)^*\gamma,\\ \mu v(wyyv\cup wwxv\cup wxyw\cup xxy\ldots)^*\mu',\\ \gamma'v(wyyv\cup wwxv\cup wxyw\cup xxy\ldots)^*\gamma\end{Bmatrix}.$$

Thus, by increasing the number of rules, the only affected part is the combination of strings inside $(\ )$. The overall pattern of strings produced is the same as the previous cases because only one initial string is used in this splicing system. Then, the order of limit language is determined by *p* number of rules used.

Two initial string and *p* rules are used in the splicing system. The Head splicing system and the limit language is presented as follows.

Case 5 Two initial strings and *p* number of rules

Let $S=\begin{pmatrix}\{a,g,c,t\},(\{\mu vwyyvwwx\ldots\gamma\},\{\varepsilon vwyyvwwx\ldots\phi\}),\\ \{(v,wy,y),(v,ww,x),\ldots\}\varnothing\end{pmatrix},$

where v and y, and w and x, are complementary to one another and $\mu,\gamma,\varepsilon,\phi,v,w,x,y\in A^*$. The initial string is presented below.

$$\begin{matrix}5'-\mu v^{\blacktriangledown}wyyv^{\blacktriangledown}wwx\ldots\gamma-3'\\ 3'-\mu'yxv_{\blacktriangle}vyxx_{\blacktriangle}w\ldots\gamma'-5'\end{matrix}$$

$$\begin{matrix}5'-\varepsilon v^{\blacktriangledown}wyyv^{\blacktriangledown}wwx\ldots\phi-3'\\ 3'-\varepsilon'v^{\blacktriangledown}wyyv^{\blacktriangledown}wwx\ldots\phi'-5'\end{matrix}.$$

The following is the splicing language that is generated after the splicing process:

$$L(S)=\begin{Bmatrix}\mu vwxy\gamma,\mu vvxy\mu',\gamma'vwxy\gamma,\varepsilon vwxy\phi,\varepsilon vvxy\varepsilon',\\ \phi'vwxy\phi,\mu vvxy\varepsilon',\gamma'vwxy\phi,\mu wxyz\phi,\\ \mu vwyyvwxy\gamma,\mu vwyyvvxy\mu',\gamma'vwyyvwxy\gamma,\\ \varepsilon vwyyvwxy\phi,\varepsilon vwyyvvxy\varepsilon',\phi'vwyyvwxy\phi,\\ \mu vwyyvvxy\varepsilon',\gamma'vwyyvwxy\phi,\mu vwyyvwxy\phi,\\ \mu vwyywxxyvwxy\gamma,\mu vwyywxxyvvxy\mu',\\ \gamma'vwyywxxyvwxy\gamma,\mu vwyywxxyvwxywwxy\gamma,\\ \mu vwyywxxyvwxyxwyz\mu',\\ \gamma'vwyywxxyvwxywwxy\gamma,\\ \mu vwyywxxyvwxywxxyvwxy\gamma,\\ \mu vwyywxxyvwxywxxywwxy\mu',\\ \gamma'wxzzwxxyvwxyxyyzvwxy\gamma,\ldots\end{Bmatrix},$$

where $\mu',\gamma',\varepsilon,\phi\in A^*$.

Since the order is defined by the number of rules, the *n*-th order limit language is presented below.

$$L_n(S)=\begin{Bmatrix}\mu v(wyyv\cup xxyv\cup vxy\ldots)^*\gamma,\\ \mu v(wyyv\cup xxyv\cup vxy\ldots)^*\mu',\\ \gamma'v(wyyv\cup xxyv\cup vxy\ldots)^*\gamma,\\ \varepsilon v(wyyv\cup xxyv\cup vxy\ldots)^*\phi,\\ \varepsilon v(wyyv\cup xxyv\cup vxy\ldots)^*\varepsilon',\\ \phi'v(wyyv\cup xxyv\cup vxy\ldots)^*\phi,\\ \mu v(wyyv\cup xxyv\cup vxy\ldots)^*\varepsilon',\\ \gamma'v(wyyv\cup xxyv\cup vxy\ldots)^*\phi,\\ \mu v(wyyv\cup xxyv\cup vxy\ldots)^*\phi,\\ \varepsilon v(wyyv\cup xxyv\cup vxy\ldots)^*\gamma\end{Bmatrix}.$$

Case 6: *m* initial string and *p* number of rules

Let $S=\begin{pmatrix}\{a,g,t,c\},\{(a_1x_1b_1),(a_2x_2b_2),\ldots,(a_mx_mb_m)\},\\ \{(a_1,x_1,b_1),(a_2,x_2,b_2),\ldots,(a_p,x_p,b_p)\},\{\varnothing\}\end{pmatrix}$ be the Head splicing system. The initial string axb can represent many crossing sites, and the rules can also apply to all initial strings if the rules follow Watson-Cricks complementary. So, after the splicing process occurs, the *n*-th order limit language is presented below.

$$L_n=\begin{Bmatrix}a_1(x_1\cup x_2\cup\ldots\cup x_p)^*b_1,a_1(x_1\cup x_2\cup\ldots\cup x_p)^*a'_1,\\ b'_1(x_1\cup x_2\cup\ldots\cup x_p)^*b_1,\ldots,\\ a_2(x_1\cup x_2\cup\ldots\cup x_p)^*b_m,a_2(x_1\cup x_2\cup\ldots\cup x_p)^*a'_m,\\ b'_2(x_1\cup x_2\cup\ldots\cup x_p)^*b_m,\ldots,\\ a_m(x_1\cup x_2\cup\ldots\cup x_p)^*b_m,a_m(x_1\cup x_2\cup\ldots\cup x_p)^*a'_m,\\ b'_m(x_1\cup x_2\cup\ldots\cup x_p)^*b_m.\end{Bmatrix},$$

where $x_1\cup x_2\cup\ldots\cup x_n$ in the above generalisation represents the combination of the string of the language. All the cases are summarised in the following table.

TABLE I
SUMMARISATION OF CASES

Case	Number of Initial Strings, m	Number of Rules, p	Order of Limit Language, n
1	1	1	First
2	1	2	Second
3	1	3	Third
4	1	p	n-th
5	2	p	n-th
6	m	p	n-th

The above cases are sufficient for proving the theorem. ■

Next, the following Lemma is established.

Lemma 1
If $p \geq 1$ rule is used in the splicing system, then $2p-2$ different combination of the string in the language will be produced.

Proof
In this Lemma, nine cases are presented which are three cases are obtained in theorem above which are Case 1, Case 2 and Case 3 and another six cases are developed and presented below.

Case 4: Two initial strings and a rule

Let $S=\left(\{a,g,c,t\},\left(\{\mu vwyy\gamma\}\{\varepsilon vwyy\phi\}\right),\{(v,wy,y)\}\varnothing\right)$, where v and y, and w and x, are complementary to one another and $\mu,\gamma,v,w,x,y\in A^*$. The initial string is presented below.

$$\begin{matrix} 5'-\mu v^{\blacktriangledown}wyy\gamma-3' & 5'-\varepsilon v^{\blacktriangledown}wyy\phi-3' \\ 3'-\mu' yxv_{\blacktriangle}v\gamma'-5' & 3'-\varepsilon' yxv_{\blacktriangle}v\phi'-5' \end{matrix}.$$

The following is the splicing language that is generated after the splicing process:

$$L(S)=\left\{\begin{matrix} \mu vwyy\gamma, \mu vvxy\mu', \gamma' vwyy\gamma \\ \varepsilon vwyy\phi, \varepsilon vvxy\varepsilon', \phi' vwyy\phi, \\ \mu vvxy\varepsilon', \gamma' vwyy\phi, \mu vwyy\phi, \varepsilon vwyy\gamma \end{matrix}\right\},$$

where $\mu',\gamma',\varepsilon',\phi'\in A^*$.

From there, two initial strings and a rule will produce ten different sets of strings. Since the order is defined by the number of rules, the first-order limit language is presented below.

$$L(S)=\left\{\begin{matrix} \mu vvxy\mu', \gamma' vwyy\gamma, \varepsilon vvxy\varepsilon', \phi' vwyy\phi, \\ \mu vvxy\varepsilon', \gamma' vwyy\phi, \mu vwyy\phi, \varepsilon vwyy\gamma \end{matrix}\right\}.$$

Case 5: Two initial strings and two rules

Let $S=\left(\begin{matrix} \{a,g,c,t\},\left(\{\mu vwyyvwwx\gamma\}\{\varepsilon vwyyvwwx\phi\}\right), \\ \{(v,wy,y),(v,ww,x)\},\varnothing \end{matrix}\right)$,

where v and y, and w and x, are complementary to one another and $\mu,\gamma,v,w,x,y\in A^*$ The initial string is presented below.

$$\begin{matrix} 5'-\mu v^{\blacktriangledown}wyyv^{\blacktriangledown}wwx\gamma-3' & 5'-\varepsilon v^{\blacktriangledown}wyyv^{\blacktriangledown}wwx\phi-3' \\ 3'-\mu' yxv_{\blacktriangle}vyxx_{\blacktriangle}w\gamma'-5' & 3'-\varepsilon' yxv_{\blacktriangle}vyxx_{\blacktriangle}w\phi'-5' \end{matrix}.$$

The following is the splicing language that is generated after the splicing process:

$$L(S)=\left\{\begin{matrix} \mu vwwx\gamma, \mu vvxy\mu', \gamma' vwwx\gamma, \\ \varepsilon vwwx\phi, \varepsilon vvxy\varepsilon', \phi' vwwx\phi, \\ \mu vvxy\varepsilon', \gamma' vwwx\phi, \mu vwwx\phi, \\ \varepsilon vwwx\gamma, \mu wyyzwwyz\mu', \mu vwyyvwwx\gamma, \\ \mu vxxyvvxy\mu', \ldots \end{matrix}\right\},$$

where $\mu',\gamma',\varepsilon,\phi\in A^*$.

Since the order is defined by the number of rules, the second-order limit language is presented below.

$$L_2(S)=\left\{\begin{matrix} \mu v(wyyv\cup xxyv)^* wwx\gamma, \\ \mu v(wyv\cup xxyv)^* vxy\mu', \\ \gamma' w(wyyv\cup xxyv)^* wwx\gamma, \\ \varepsilon v(wyyv\cup xxyv)^* wwx\phi, \\ \varepsilon v(wyv\cup xxyv)^* vxy\varepsilon', \\ \phi' w(wyyv\cup xxyv)^* wwx\phi, \\ \mu w(xzzw\cup yyzw)^* wyz\varepsilon', \\ \gamma' w(wyyv\cup xxyv)^* wwx\phi, \\ \mu v(wyyv\cup xxyv)^* wwx\phi, \\ \varepsilon v(wyyv\cup xxyv)^* wwx\gamma \end{matrix}\right\}.$$

Case 6: **Two initial and strings of three rules**

Let

$$S=\left(\begin{matrix} \{a,g,c,t\},\left(\{\mu vwyyvwwxvwxy\gamma\}\{\varepsilon vwyyvwwxvwxy\phi\}\right), \\ \{(v,wy,y),(v,ww,x),(v,wx,y)\}\varnothing \end{matrix}\right),$$

where v and y, and w and x, are complementary to one another and $\mu,\gamma,\varepsilon,\phi,v,w,x,y\in A^*$. The initial string is presented below.

$$\begin{matrix} 5'-\mu v^{\blacktriangledown}wyyv^{\blacktriangledown}wwxv^{\blacktriangledown}wxy\gamma-3' \\ 3'-\mu' yxv_{\blacktriangle}vyzz_{\blacktriangle}wyxw_{\blacktriangle}v\gamma'-5' \end{matrix}$$

$$\begin{matrix} 5'-\varepsilon v^{\blacktriangledown}wyyv^{\blacktriangledown}wwxv^{\blacktriangledown}wxy\phi-3' \\ 3'-\varepsilon' yxv_{\blacktriangle}vyzz_{\blacktriangle}wyxw_{\blacktriangle}v\phi'-5' \end{matrix}.$$

The following is the splicing language that is generated after the splicing process:

$$L(S)=\left\{\begin{matrix} \mu vwxy\gamma, \mu vvxy\mu', \gamma' vwxy\gamma, \varepsilon vwxy\phi, \varepsilon vvxy\varepsilon', \\ \phi' vwxy\phi, \mu vvxy\varepsilon', \gamma' vwxy\phi, \mu vwxy\phi, \mu vwyyvwxy\gamma, \\ \mu vwyyvvxy\mu', \gamma' vwyyvwxy\gamma, \varepsilon vwyyvwxy\phi, \\ \varepsilon vwyyvvxy\varepsilon', \phi' vwyyvwxy\phi, \mu vwyyvvxy\varepsilon', \\ \gamma' vwyyvwxy\phi, \mu vwyyvwxy\phi, \mu vwyyvwwxvwxy\gamma, \\ \mu vwyyvwwxvvxy\mu', \gamma' vwyyvwwxvwxy\gamma, \\ \mu vwyyvwwxvwxywwxy\gamma, \mu vwyyvwwxvwxywvxy\mu', \\ \gamma' vwyyvwwxvwxywwxy\gamma, \mu vwyyvwwxvwxywwxyvwxy\gamma, \\ \mu vwyyvwwxvwxywwxyvvxy\mu', \\ \gamma' vwyyvwwxvwxywwxyvwxy\gamma, \ldots \end{matrix}\right\},$$

where $\mu',\gamma',\varepsilon,\phi\in A^*$.

Since the order is defined by the number of rules, the third-order limit language is presented below.

$$L_3(S)=\begin{Bmatrix} \mu v(wyyv\cup wwxv\cup wxyw\cup xxyv)^*wxy\gamma, \\ \mu v(wyyv\cup wwxv\cup wxyw\cup xxyv)^*vxy\mu', \\ \gamma' v(wyyv\cup wwxv\cup wxyw\cup xxyv)^*wxy\gamma, \\ \varepsilon v(wyyv\cup wwxv\cup wxyw\cup xxyv)^*wxy\phi, \\ \varepsilon v(wyyv\cup wwxv\cup wxyw\cup xxyv)^*vxy\varepsilon', \\ \phi' v(wyyv\cup wwxv\cup wxyw\cup xxyv)^*wxy\phi, \\ \mu v(wyyv\cup wwxv\cup wxyw\cup xxyv)^*vxy\varepsilon', \\ \gamma' v(wyyv\cup wwxv\cup wxyw\cup xxyv)^*wxy\phi, \\ \mu v(wyyv\cup wwxv\cup wxyw\cup xxyv)^*wxy\phi, \\ \varepsilon v(wyyv\cup wwxv\cup wxyw\cup xxyv)^*wxy\gamma \end{Bmatrix}.$$

Case 7: Three numbers of initial strings and a rule

Let $S=\begin{pmatrix}\{a,c,g,t\},\{(\mu vwyy\gamma),(\varepsilon vwyy\phi),(\eta vwyy\omega)\}, \\ \{(v,wy,y)\},\varnothing\end{pmatrix},$

where v and y, and w and x, is complementary to one another and $\mu,\gamma,\varepsilon,\phi,\eta,\omega,v,w,x,y\in A^*$. The initial string is presented below.

$$\begin{matrix}5'-\mu v^{\blacktriangledown}wyy\gamma-3' \\ 3'-\mu' yxv_{\blacktriangle}v\gamma'-5'\end{matrix}\qquad\begin{matrix}5'-\varepsilon v^{\blacktriangledown}wyy\phi-3' \\ 3'-\varepsilon' yxv_{\blacktriangle}v\phi'-5'\end{matrix}$$

$$\begin{matrix}5'-\eta v^{\blacktriangledown}wyy\omega-3' \\ 3'-\eta' yxv_{\blacktriangle}v\omega'-5'\end{matrix}.$$

The splicing language produced after the splicing process is as follows:

$$L(S)=\begin{Bmatrix}\mu vwyy\gamma,\mu vvwy\mu',\gamma' vwyy\gamma,\varepsilon vwyy\phi,\varepsilon vvwy\varepsilon', \\ \phi' vwyy\phi,\eta vwyy\omega,\eta wwyz\eta',\omega' vwyy\omega,\mu vvwy\varepsilon', \\ \gamma' vwyy\phi,\mu vwyy\phi,\varepsilon vwyy\gamma,\mu vvwy\eta',\gamma' vwyy\omega, \\ \mu vwyy\omega,\eta vwyy\gamma,\varepsilon vwyy\mu,\phi', vwyy\omega,\varepsilon vwyy\omega,\eta vwyy\phi\end{Bmatrix},$$

where $\mu',\gamma',\varepsilon',\phi',\eta',\omega'\ldots\in A^*$.

From there, three numbers of initial string and a rule will produce twenty-one different set of strings.

Since the order is defined by the number of rules, the first-order limit language is presented below.

$$L_1(S)=\begin{Bmatrix}\mu vvwy\mu',\gamma' vwyy\gamma,\varepsilon vvwy\varepsilon',\phi' vwyy\phi,\eta wwyz\eta', \\ \omega' vwyy\omega,\mu vvwy\varepsilon',\gamma' vwyy\phi,\mu vwyy\phi,\varepsilon vwyy\gamma, \\ \mu vvwy\eta',\gamma' vwyy\omega,\mu vwyy\omega,\eta vwyy\gamma,\varepsilon vwyy\mu,\phi', \\ vwyy\omega,\varepsilon vwyy\omega,\eta vwyy\phi\end{Bmatrix},$$

Case 8: Three initial strings and two rules

Let $S=\begin{pmatrix}\{a,c,g,t\},\begin{Bmatrix}(\mu vwyyvwwx\gamma),(\varepsilon vwyyvwwx\phi), \\ (\eta vwyyvwwx\omega)\end{Bmatrix}, \\ \{(v,wy,y),(v,ww,x)\},\varnothing\end{pmatrix},$

where w and z, and x and y, are complementary to one another and $\mu,\gamma,\varepsilon,\phi,\eta,\omega,\ldots,w,x,y,z\in A^*$. The initial string is presented below.

$$\begin{matrix}5'-\mu v^{\blacktriangledown}wyyv^{\blacktriangledown}wwx\gamma-3' \\ 3'-\mu' yxv_{\blacktriangle}vyxx_{\blacktriangle}w\gamma'-5'\end{matrix}\qquad\begin{matrix}5'-\varepsilon v^{\blacktriangledown}wyyv^{\blacktriangledown}wwx\phi-3' \\ 3'-\varepsilon' yxv_{\blacktriangle}vyxx_{\blacktriangle}w\phi'-5'\end{matrix}$$

$$\begin{matrix}5'-\eta v^{\blacktriangledown}wyyv^{\blacktriangledown}wwx\omega-3' \\ 3'-\eta' yxv_{\blacktriangle}vyxx_{\blacktriangle}w\omega'-5'\end{matrix}.$$

The following is the splicing language that is generated after the splicing process:

$$L(S)=\begin{Bmatrix}\mu vwyy\gamma,\mu vvxy\mu',\gamma' vwyy\gamma,\mu vxyyvwyy\gamma, \\ \mu vxyywwyz\mu',\gamma' vxyyvwyy\gamma,\varepsilon vwyy\phi,\varepsilon vvxy\varepsilon', \\ \phi' vwyy\phi,\varepsilon vxyyvwyy\phi,\varepsilon vxyywwyz\varepsilon', \\ \phi' vxyyvwyy\phi,\eta vwyy\omega,\eta vvxy\eta',\omega' vwyy\omega, \\ \eta vxyyvwyy\omega,\eta vxyyvvxy\eta',\omega' vxyyvwyy\omega,\mu vvxy\varepsilon', \\ \gamma' vwyy\phi,\mu vwyy\phi,\mu vxyyvvxy\varepsilon',\gamma' vxyyvwyy\phi, \\ \mu vxyyvwyy\phi,\varepsilon vwyy\gamma,\mu vvxy\eta',\gamma' vwyy\omega,\mu vwyy\omega, \\ \eta vwyy\gamma,\varepsilon vwyy\mu,\phi', vwyy\omega,\varepsilon vwyy\omega,\eta vwyy\phi,\ldots\end{Bmatrix},$$

where $\mu',\gamma',\varepsilon',\phi',\eta',\omega'\ldots\in A^*$.

Since the order is defined by the number of rules, the second-order limit language is presented below.

$$L_2(S)=\begin{Bmatrix}\mu v(wyyv\cup xxyv)^*wwx\gamma,\mu v(wyyv\cup xxyv)^*vxy\mu', \\ \gamma' v(wyyv\cup xxyv)^*wwx\gamma,\varepsilon v(wyyv\cup xxyv)^*wwx\phi, \\ \varepsilon v(wyyv\cup xxyv)^*vxy\varepsilon',\phi' v(wyyv\cup xxyv)^*wwx\phi, \\ \eta v(wyyv\cup xxyv)^*wwx\omega,\eta v(wyyv\cup xxyv)^*vxy\eta', \\ \omega' v(wyyv\cup xxyv)^*wwx\omega,\mu v(wyyv\cup xxyv)^*vxy\varepsilon', \\ \gamma' v(wyyv\cup xxyv)^*wwx\phi,\mu v(wyyv\cup xxyv)^*wwx\phi, \\ \varepsilon v(wyyv\cup xxyv)^*wwx\gamma,\mu v(wyyv\cup xxyv)^*vxy\eta', \\ \gamma' v(wyyv\cup xxyv)^*wwx\omega,\mu v(wyyv\cup xxyv)^*wwx\omega, \\ \eta v(wyyv\cup xxyv)^*wwx\gamma,\varepsilon v(wyyv\cup xxyv)^*wwx\mu, \\ \phi' v(wyyv\cup xxyv)^*wwx\omega,\varepsilon v(wyyv\cup xxyv)^*wwx\omega, \\ \eta v(wyyv\cup xxyv)^*wwx\phi\end{Bmatrix}.$$

Case 9: Three initial strings and three rules

Let

$S=\begin{pmatrix}\{a,g,c,t\}, \\ \begin{pmatrix}\{\mu vwyyyvwwxvwxy\gamma\}\{\varepsilon vwyyyvwwxvwxy\phi\} \\ (\eta vwyyyvwwxvwxy\omega)\end{pmatrix}, \\ \{(v,wy,y),(v,ww,x),(v,wx,y)\}\varnothing\end{pmatrix}$, where

w and z, and x and y, are complementary to one another and $\mu,\gamma,\varepsilon,\phi,\eta,\omega,\ldots,v,w,x,y\in A^*$. The initial string is presented below.

$$\begin{matrix}5'-\mu v^{\blacktriangledown}wyyv^{\blacktriangledown}wwxv^{\blacktriangledown}wxy\gamma-3' \\ 3'-\mu' yxv_{\blacktriangle}vyzz_{\blacktriangle}wyxw_{\blacktriangle}v\gamma'-5'\end{matrix}$$

$$\begin{matrix}5'-\varepsilon v^{\blacktriangledown}wyyv^{\blacktriangledown}wwxv^{\blacktriangledown}wxy\phi-3' \\ 3'-\varepsilon' yxv_{\blacktriangle}vyzz_{\blacktriangle}wyxw_{\blacktriangle}v\phi'-5'\end{matrix}$$

$$\begin{matrix}5'-\eta v^{\blacktriangledown}wyyv^{\blacktriangledown}wwxv^{\blacktriangledown}wxy\omega-3' \\ 3'-\eta' yxv_{\blacktriangle}vyzz_{\blacktriangle}wyxw_{\blacktriangle}v\omega'-5'\end{matrix}.$$

The following is the splicing language that is generated after the splicing process.

$$L(S)=\left\{\begin{array}{l} \mu vwxy\gamma,\mu vvxy\mu',\gamma' vwxy\gamma,\varepsilon vwxy\phi,\varepsilon vvxy\varepsilon', \\ \phi' vwxy\phi,\mu vvxy\varepsilon',\gamma' vwxy\phi,\mu vwxy\phi,\mu vwyyvwxy\gamma, \\ \mu vwyyvvxy\mu',\gamma' vwyyvwxy\gamma,\varepsilon vwyyvwxy\phi, \\ \varepsilon vwyyvvxy\varepsilon',\phi' vwyyvwxy\phi,\mu vwyyvvxy\varepsilon', \\ \gamma' vwyyvwxy\phi,\mu vwyyvwxy\phi,\mu vwyyvwwxvwxy\gamma, \\ \eta vwyy\omega,\eta vvxy\eta',\omega' vwyy\omega,\eta vxyyvwyy\omega,\eta vxyyvvxy\eta', \\ \omega' vxyyvwyy\omega,\mu vvxy\varepsilon',\gamma' vwyy\phi,\mu vwyy\phi,\mu vxyyvvxy\varepsilon', \\ \gamma' vxyyvwyy\phi,\mu vxyyvwyy\phi,\varepsilon vwyy\gamma,\mu vvxy\eta',\gamma' vwyy\omega, \\ \mu vwyy\omega,\eta vwyy\gamma,\varepsilon vwyy\mu,\phi', vwyy\omega,\varepsilon vwyy\omega,\eta vwyy\phi,\ldots \end{array}\right\},$$

where $\mu',\gamma',\varepsilon',\phi',\eta',\omega'\ldots\in A^*$.

Since the order is defined by the number of rules, the third-order limit language is presented below.

$$L_3(S)=\left\{\begin{array}{l} \mu v(wyyv\cup wwxv\cup wxyw\cup xxyv)^*wwx\gamma, \\ \mu v(wyyv\cup wwxv\cup wxyw\cup xxyv)^*vxy\mu', \\ \gamma' v(wyyv\cup wwxv\cup wxyw\cup xxyv)^*wwx\gamma, \\ \varepsilon v(wyyv\cup wwxv\cup wxyw\cup xxyv)^*wwx\phi, \\ \varepsilon v(wyyv\cup wwxv\cup wxyw\cup xxyv)^*vxy\varepsilon', \\ \phi' v(wyyv\cup wwxv\cup wxyw\cup xxyv)^*wwx\phi, \\ \eta v(wyyv\cup wwxv\cup wxyw\cup xxyv)^*wwx\omega, \\ \eta v(wyyv\cup wwxv\cup wxyw\cup xxyv)^*vxy\eta', \\ \omega' v(wyyv\cup wwxv\cup wxyw\cup xxyv)^*wwx\omega, \\ \mu v(wyyv\cup wwxv\cup wxyw\cup xxyv)^*vxy\varepsilon', \\ \gamma' v(wyyv\cup wwxv\cup wxyw\cup xxyv)^*wwx\phi, \\ \mu v(wyyv\cup wwxv\cup wxyw\cup xxyv)^*wwx\phi, \\ \varepsilon v(wyyv\cup wwxv\cup wxyw\cup xxyv)^*wwx\gamma, \\ \mu v(wyyv\cup wwxv\cup wxyw\cup xxyv)^*vxy\eta', \\ \gamma' v(wyyv\cup wwxv\cup wxyw\cup xxyv)^*wwx\omega, \\ \mu v(wyyv\cup wwxv\cup wxyw\cup xxyv)^*wwx\omega, \\ \eta v(wyyv\cup wwxv\cup wxyw\cup xxyv)^*wwx\gamma, \\ \varepsilon v(wyyv\cup wwxv\cup wxyw\cup xxyv)^*wwx\mu, \\ \phi' v(wyyv\cup wwxv\cup wxyw\cup xxyv)^*wwx\omega, \\ \varepsilon v(wyyv\cup wwxv\cup wxyw\cup xxyv)^*wwx\omega, \\ \eta v(wyyv\cup wwxv\cup wxyw\cup xxyv)^*wwx\phi \end{array}\right\}.$$

Let p be the number of rules in the Head splicing system. The limit language produced from Case 1 until Case 9 and the combinations of the strings of the language are observed. The Lemma is then proved by principle of double induction.

TABLE 2
COMBINATION OF THE STRING OF THE LANGUAGE

Cases	Number of Initial Strings, m	Number of Rules, p	Number of Combination of the String of the language
1	1	1	0
2	1	2	2
3	1	3	4
4	2	1	0
5	2	2	2
6	2	3	4
7	3	1	0

Based on the result which has been tabulated in the above table, it can be seen the combination of language, produce $2p-2$ for $p\geq 1$.

The result in Table 2 is rearranged by classifying them into the number of rules. By fixing the number of rules accordingly, we can observe a pattern from one iteration to another is 2.

TABLE 3
THE NUMBER OF THE COMBINATION OF THE STRING OF THE LANGUAGE AND THE PATTERN PRODUCE BASED ON THE NUMBER OF THE COMBINATION OF THE STRING

Case m	Case p	The Number of the Combination of the String	The Pattern Produce Based on the Number of the Combination of the String
1	1		
2	1	0	0
3	1		
1	2		
2	2	2	2
3	2		
1	3		
2	3	4	2
3	3		

Let $f(m,p)$ be a function that represents the number of combinations of the strings that consist of m number of initial strings and p number of rules of the splicing system. Since there are two variables in the function given, therefore, proof by induction must be shown for both variables. By the principle of double induction, we must show the following.

i. $f(m,p)$ is true for $m=1$ or $p=1$.

ii. For all $p\geq 1$, if $f(m,k)$ is true, then $f(m,k+1)$ is true.

iii. For all $m\geq 1$, if $f(k,p)$ is true for all $p\geq 1$, then $f(k+1,p)$ is true for all $p\geq 1$.

Then, $f(m,p)$ is true for all $m\geq 1$ and $p\geq 1$.

By double induction, the first part is we need to prove $f(1,1)=0$.

For $p=m=1$,

$$\begin{aligned} f(1,1) &= 2k-2 \\ &= 2(1)-2 \\ &= 0 \end{aligned}$$

As we can see in the table above, the number of combinations of the string of the language for $p=1$ is zero. Then, $f(m,p)$ for $m=1$ and $p=1$ is true.

Then, for all $p \geq 1$, assume $f(m,k)$ is true for all $k \geq 1$, then assume $f(m,k+1)$ is true for all $k \geq 1$.

Then, assume $p=k$.

$f(m,k)=2k-2$ is true

Then, we need to show for $p=k+1$ is also true.

$$f(m,k+1)=2(k+2)-2$$
$$=2k$$

Then, we need to prove for $(k+1)$-th iteration. In Table 3, the pattern produces based on the number of the combination of the strings for the next iteration is 2. Thus,

$$(k+1)\text{-}th = k\text{-th} + \text{ next iteration}$$
$$f(m,k+1) = f(m,k)+2$$
$$=2k-2+2$$
$$=2k$$

Therefore, $f(m,p)$ for all $p \geq 1$ is true.

The third part of the double induction is to prove the second variable of the function which is the number of the initial string in the splicing system.

For all $m \geq 1$, assume $f(k,p)$ is true for all $p \geq 1$, then we need to prove $f(k+1,p)$ is true for all $p \geq 1$.

$f(k,p)=2p-2$ is true.

In Table 3, the pattern produces based on the number of the combinations of the strings for the next iteration is 0 when the number of rules is fixed at a value. Thus,

$$(k+1)\text{-th} = k\text{-th} + \text{ next iteration}$$
$$f(k+1,p) = f(k,p)+0$$
$$=2p-2+0$$
$$=2p-2$$

Therefore, $f(m,p)$ for all $m \geq 1$ is true. Then, $f(m,p)=2p-2$ is true for $m \geq 1$ and $p \geq 1$. ■

IV. CONCLUSION

In this paper, the definition of the *n*-th order limit language is improvised and introduced. Based on the definition given by Goode (Goode, 2004), the order of the limit language is determined by comparing the language produced. Based on the improved definition, the *n*-th order limit language is defined by the number of rules involved in the splicing system. Given that the rules must have the same length, it ensures the combination of strings follows the pattern presented in the main result. Thus, the number of rules, which results from the combination of strings produced, defines the *n*-th order limit language. From the theorem, we can conclude that the number of rules with a specific constant length of the rules defines the *n*-th order limit language.

ACKNOWLEDGMENT

Siti Hajar Mohd Khairuddin would like to offer a heartfelt appreciation to her supervisor, Dr. Muhammad Azrin bin Ahmad, and her co-supervisor, Dr. Noraziah binti Adzhar, for providing her with the amazing chance to do this study, and she also would like to thank everyone who helped her complete this project.

REFERENCES

[1] R. H. Tamarin, *Principle of Genetics*, 7th. ed. USA: The McGraw-Hill Companies, 2001, pp. 6-7.

[2] E. Picardi, *RNA Bioinformatics*. NY: Springer, 2016, pp. 63-100.

[3] W. P. Alberts B., A. Johnson, J . Lewis, M. Raff and K. Roberts, "Molecular biology of the cell." *Scand. J. Rheumatol.*, vol. 32, no. 2, pp. 125-125, 2003.

[4] A. R. Lebeck and L. Dwyer, *Introducton to DNA Self Assembled Computer Design*. Boston, London: Artech House, 2008, pp. 293-303.

[5] L. Kari, "DNA Computing: Arrival of Biological Mathematics," *Springer*, vol. 19 , no. 2, pp. 2-9, 1997.

[6] T. Head, "Formal Language Theory and DNA: An Analysis of the Generative Capacity of Specific Recombinant Behaviors," *Bull. Math. Biol.*, vol. 49, no. 6, pp. 737–759, 1987.

[7] G. Păun, G. Rozenberg, and A. Salomaa, "Computing by splicing," *Theor. Comput. Sci.*, vol. 168, no. 2, pp. 321–336, 1996.

[8] D. Pixton, "Regularity of Splicing Languages," *Discret. Appl. Math.*, vol. 69, pp. 101–12, 1996.

[9] T. E. Goode, *Aspects of Molecular Computing*. New York: Springer-Verlag, 2004, ch. Splicing to the Limit.

[10] Y. Yusof, "DNA Splicing System Inspired by Bio Molecular Operation," Ph.D. thesis, Universiti Teknologi Malaysia, 2012.

[11] F. Karimi and S. Turaev, "Fuzzy Splicing System," in *6th Int. Conf., ICCCI,* Seoul, Korea, 2014, pp 20-29.

[12] T. E. Goode, "Wet Splicing System," *DIMACS Ser. Discret. Math. Theor. Comput. Sci.*, vol. 48, pp. 73-83, 1991.

[13] M. H. Mudaber, Y. Yusof, and M. S. Mohamad, "Differentiating the persistency and permanency of some two stages DNA splicing language via Yusof-Goode (Y-G) approach," in *1st Int. Conf. on Appl. & Industrial Math. and Stat.*, Kuantan, Pahang, Malaysia, 2017, pp. 1-7.

[14] P. Bonizzoni and G. Mauri, "Regular splicing languages and subclasses," *Theor. Comput. Sci.*, vol. 340, no. 2, pp. 349–363, 2005.

[15] W. L. Lim, Y. Yusof, and N. Rosli, DNA "Splicing Language Predictor," *Creat. Innov. Technol. Res. Expo.*, vol. 49, pp. 131-404, 2014.

[16] N. I. Ismail, W. H. Fong, and N. H. Sarmin, "Computation of splicing languages from DNA splicing system with one palindromic restriction enzyme," *Mal. J. Fund. Appl. Sci.*, vol. 14, no. 2, pp. 188–192, 2018.

[17] M. A. Ahmad, "An extension of first order limit language," in *AIP Conf. Proc.*, Kuala Lumpur, Malaysia, 2014, pp. 627-631.

[18] L. Peter, *An Introduction to Formal Languages and Automata,* 5th. ed. Sudbury, MA: Jones and Bartlett, LLC, 2012, pp. 12-177.

[19] M. A. Ahmad, "Second Order Limit Language and its Properties in Yusof-Goode Splicing System," Ph.D. thesis, Universiti Teknologi Malaysia, 2016.

[20] S. H. M. Khairuddin, M. A. Ahmad, and N. Adzhar, "Effect of m Number of Initial String on the n -th Order Limit Language,"*Technol. Reports Kansai Univ.*, vol. 62, no. 7, pp. 3681–3689, 2020.

S. H. M. Khairuddin is a Master of Science (MSc) student by research at Universiti Malaysia Pahang. My area of research is on formal language theory and automata. It also includes DNA, splicing system and splicing languages. My research is under the supervision of Dr. Muhammad Azrin Ahmad and Dr. Noraziah Adzhar. I have been appointed as a research grant assistant under the Fundamental Research Grant Scheme (FRGS) by the Ministry of Higher Education (MOHE) Malaysia, FGRS/1/2018/STG06/UMP/03/1.