Redefining *n*-th Order Limit Languages in Extended-H Splicing System

SITI HAJAR MOHD KHAIRUDDIN[1], MUHAMMAD AZRIN AHMAD[1],
MOHDSHAM MOHAMAD[1], RAJALINGAM SOKKALINGAM[2]

[1]Centre For Mathematical Sciences, Universiti Malaysia Pahang Al-Sultan Abdullah, Lebuh Persiaran Tun Khalil Yaakob, 26300 Kuantan, Pahang, , MALAYSIA
[2]Department of Fundamental and Applied Sciences, Universiti Teknologi PETRONAS, Bandar Seri Iskandar, Perak, MALAYSIA

Abstract: This research delves into the fascinating intersection of formal language theory and molecular biology by examining DNA splicing systems. DNA splicing is a process of rearranging genetic material by cutting and rejoining DNA strands. Researchers have developed computational models inspired by these mechanisms, which are splicing systems that allow for simulation and analysis of the process of cutting and pasting DNA to produce new strands. This study builds upon prior research that defined the *n*-th order limit language, a concept initially introduced by Goode and Pixton and subsequently refined by defining based on rules perspectives. While originally rooted in the biological characteristics of DNA splicing, this concept is now being re-examined within the framework of the Păun splicing system. This shift in focus moves away from purely biological models and embraces language generation processes, aligning with the well-established Chomsky hierarchy. By redefining the *n*-th order limit language in this context, the research seeks to strengthen the theoretical foundation and expand the practical applications of DNA splicing systems in formal language theory.

Key-Words: DNA splicing system, formal language theory, splicing system, splicing language, *n*-th order limit language, Chomsky hierarchy.

Received: March 19, 2024. Revised: August 23, 2024. Accepted: October 9, 2024. Available online: November 11, 2024.

1 Introduction

In the formal language theory field and informational macromolecules, the DNA splicing system stands as a computational model motivated by the biological process with regards to DNA recombination or splicing, [1]. In molecular biology, DNA splicing refers to the rearrangement of genetic material by cutting and rejoining DNA strands, which is a fundamental mechanism for generating genetic diversity and facilitating various cellular processes, [2]. In the field of computer science and linguistics, Formal language theory stands as a foundational pillar that delves into the systematic study of languages, their structures, and the mechanisms through which they can be recognized and generated, [3]. This theory provides a framework for understanding the intricacies of language, both natural and artificial, by utilizing mathematical models and abstract concepts.

Besides, an intriguing application of formal language theory can be found in DNA splicing systems, which bridges the gap between computer science and molecular biology. DNA splicing involves the rearrangement of genetic information within DNA sequences, [4]. By viewing DNA sequences as strings and applying formal grammar, researchers can model the splicing process as language, [5]. This unique intersection showcases the versatility of Formal language theory beyond traditional linguistic applications, contributing to advancements in biocomputing and DNA manipulation techniques.

In the past studies that have been conducted, the *n*-th order limit language definition, which was introduced by [6], has been improvised in [7]. Initially, *n*-th order limit language, L_n is defined as deleting words that are transient in L_{n-1}. The number of rules, as well as the number of initial strings contained in the splicing system, are highlighted in the extension of the research done by [7]. The characteristics, the relation with finite state automata and validation of the model in laboratory experiments have been studied, explored, and performed, respectively, [7]. However, the results are derived from a model of a splicing system which is aimed at preserving the biological characteristics with respect to the splicing process.

Therefore, in this research, the direction is shifted and explored to a model based on language generation according to Chomsky's hierarchy. In addition, some modifications, such as the variable of the splicing system applied when forming the definition from another type of splicing system, which is widely used when it comes to research that focuses on the generation of language which is Păun splicing system, [8].

This research addresses critical inquiries concerning Păun splicing systems, with the overarching goal of advancing our understanding and applications in formal language theory. Previously, the *n*-th order limit language was defined using the Head splicing system, which is the forerunner splicing system that only focuses on the biological perspectives. In this research, the investigation focuses on the reframing of the *n*-th order limit language in the Păun splicing system framework, which is related to language-based, with an emphasis on aligning its definition with a model rooted in language generation processes. The goal is to enhance the precision and applicability of the existing definition, providing a more robust theoretical foundation for Păun splicing systems.

In the following section, the foundational aspects of this research are introduced.

2 Preliminaries

This section provides several fundamental definitions pertaining to formal language theory as well as the *n*-th order limit language.

Definition 1: Alphabets, [9]
The alphabet, represented by the symbol, Σ, refers to a fundamental concept in formal language theory, serving as the primary set of symbols utilised to create strings in a language. It establishes the basis for defining the grammar rules as well as the syntax pertaining to a language.

Definition 2: Strings, [10]
A string represents a finite sequence of characters or symbols selected from a particular alphabet, expressed as $|w|$.

Definition 3: Language, [11]
L is expressed as a set of possibly finite sets of strings pertaining to some finite alphabet. Here, Σ^* denotes the set of all possible strings of a finite alphabet Σ. Note that, $L \subseteq \Sigma^*$.

Definition 4: Limit Language, [2]
Limit language, often known as a first-order limit language, is a splicing language generated by the molecules that persist after the splicing system has finished or accomplished equilibrium.

Then, the Head splicing systems used in this study are presented below.

Definition 5: Head Splicing System, [4]
The spicing system is comprised of four unique groups of elements, for instance, A, I, B as well as C, which are explained below:
A represents an alphabet set,
I denotes a set of initial strings,
B symbolises a set of rules, which resembles 5'−overhang or blunt end,
C denotes a set of rules, which resembles 3'−overhang.

Then, an example of language production by the Head splicing system is presented.

Example 1
Let $S = (\{w, x, y, z\}, \{\alpha wxzzzwxxy\beta\}, \{(w, xz, z), (w, xx, y)\}$ $\emptyset)$, in which w and z, as well as x and y, complement each other and $\alpha, \beta, w, x, y, z \in A^*$.

Then, the splicing language formed by the splicing system is referred to as $L(S) = \{\alpha wxxt\beta, \alpha wwyz\alpha', \beta xxxy\beta', \alpha wyyzwwyz\alpha',$ $\beta, \alpha wxzzzwyyzwwyz\alpha', \ldots\}$, where $\alpha', \beta' \in A^*$.

Based on the number of rules, the splicing system's language limit at the second order is as follows:

$$L_2(S) = \left\{ \begin{array}{l} \alpha w(xzzw \cup yyzw)^* xxy\beta, \\ \alpha(xzzw \cup yyzw)^* wyz\alpha', \\ \beta'(xzzw \cup yyzw)^* xxy\beta \end{array} \right\}$$

Subsequently, the *n*-th order limit language is displayed as below.

Definition 6: *n*-th Order Limit Language, [6]
Assume L_{n-1} is the set of second-order limit language pertaining to L, whereas the set of L_n pertaining to *n*-th order limit language of L to represent the set of the first-order limit language of L_{n-1}. Note that L_n may be attained from L_{n-1} by differentiating the strings transient in L_{n-1}.

Khairuddin et al. subsequently altered the innovative concept, observing that while the rules governing the splicing system determine the sequence of the limit language, they must entail certain limitations, [7]. The polished iteration of the definition can be seen as follows.

Definition 7: *n*-th Order Limit Language from the Rules Perspective, [7]
Finite rules utilised in the splicing system signify the limit language order generated by the splicing system. Here, the rules must be non-identical with each other to satisfy the requirement of the *n*-th order limit language. Moreover, the rules must also maintain consistent crossing site lengths. This criteria defines a splicing language. *n*-th order limit language, expressed by $L_n(S)$ provided that the set of string produced in $L_n(S)$ is distinct from the set of strings of

$L_1(S), L_2(S), ..., L_n(S)$. In other words, $\bigcap_{i=1}^{n} L_i = \emptyset$ and $L_1(S) \not\subset L_2(S) \not\subset ... \not\subset L_n(S)$.

The *n*-th order limit language generalization is presented by the following cases in the table as follows. Table 1 presents the definition of the *n*-th order limit language based on the view of rules utilised in the splicing system. Six cases are consider a scenario with *c* initial strings and *b* rules. The following six cases must be addressed: Case 1: one initial string and one rule; Case 2: one initial string and two rules; Case 3: one initial string and three rules; Case 4: one initial string and *b* rules; Case 5: two initial strings and *b* rules; and Case 6: *c* initial strings and *b* rules.

Table 1. The Cases of Generalization of the *n*-th Order Limit Language

No of Initial Strings	Number of Rules	Order of Limit Language
1	1	First-order Limit Language
1	2	Second-order Limit Language
1	3	Third-order Limit Language
1	*b*	*n*-th order Limit Language
2	*b*	*n*-th order Limit Language
c	*b*	*n*-th order Limit Language

The *n*-th order definition originally derived from the Head splicing system, which is built upon the actual splicing process of DNA in a laboratory lab. Since in this research, the exploration is on a language-based splicing system, which is an extended H splicing system developed by Păun, the definition above will be improved from time to time to encounter the requirement of the production of the language based on language perspectives.

Thus, the *n*-th order limit language generalization by using is given as follows.

Example 2 Let $S = (\{a, c, g, t\}, \{(\mu wxzzwxxy \ldots \gamma), \ldots,\}, \{(w, xz, z), (w, xx, y), \ldots, (e_b, x_b, f_b)\}, \emptyset$ in which w and z, as well as x and y, accomplish one another whereas $\mu, \gamma, w, x, y, z \in A^*$. Additionally, the initial string is elaborated below.

$$5' - \mu wxzzwxxywxyz \ldots \gamma - 3'$$
$$3' - \mu' zywwzywxzzyxw \ldots \gamma' - 5'$$

After the splicing process, the splicing language is formed as below:

$$L(S) = \left\{ \begin{array}{l} \mu wxyz\gamma, \mu wwyz\mu', \gamma' wxyz\gamma, \\ \mu wxzzwxyz\gamma, \mu wxzzwwyz\mu', \\ \gamma' wxzzwxyz\gamma, \\ \mu wxzzwxxywxyz\gamma, \\ \mu wxzzwxxywwyz\mu', \\ \gamma' wxzzwxxywxyz\gamma, \\ \mu wxzzwxxywxyzxxyz\gamma, \\ \mu wxzzwxxywxyzxwyz\mu', \\ \gamma' wxzzwxxywxyzxxyz\gamma, \\ \mu wxzzwxxywxyzxyyzwxyz\gamma, \\ \mu wxzzwxxywxyzxyyzwwyz\mu', \\ \gamma` wxzzwxxywxyzxyyzwxyz\gamma, \ldots \end{array} \right\},$$

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

Provided that the order is determined via the rules used in the splicing system is shown below.

$$L_n(S) = \left\{ \begin{array}{l} \mu w\,(xzzw \cup xxyw \cup xyzx \cup yyzw \\ \cup xyz \ldots \cup \ldots)^* \gamma, \\ \mu w\,(xzzw \cup xxyw \cup xyzx \cup yyzw \\ \cup wyz \ldots \cup \ldots)^* \mu', \\ \gamma' w\,(xzzw \cup xxyw \cup xyzx \cup \\ yyzw \cup xyz \ldots \cup \ldots)^* \gamma \end{array} \right\}$$

Next, the grammar definition is presented to enhance the understanding of language production, as the splicing system used in this research is based on language production within the framework of the Chomsky hierarchy.

Definition 8: Grammar, [12]
Grammar is used to typify a language L. Based on Chomsky, it is a term employed in listing the strings of Σ^* in L. Grammar has 4 kinds of tuples, which are V, T, S and P represented as $G = (V, T, S, P)$, in which V represents non-terminal, T denotes the terminal symbol, S represents the start symbol, and P refers to the production. Chomsky categorizes grammar into four types: type 0, type 1, type 2, and type 3, representing unrestricted grammar, context-sensitive grammar, context-free grammar as well as regular grammar, accordingly. The diagram below illustrates the grammar hierarchy.

Since the n-th order limit language only generates a regular language, as portrayed in the n-th order limit language study in the Head splicing system, [7], this investigation seeks strategies to enhance the computational power of the n-th order limit language by emphasizing on language generation. Consequently, the extended-H splicing system is chosen for its theoretical advantages. The EH splicing system definition is given below.

Definition 9: Extended-H splicing system, [13] Suppose extended Head splicing system is denoted by $\gamma = (V, T, A, R)$, comprising V as the set of alphabet, T denotes a terminal symbol, A represents the set of axiom, while R refers to the set of rules.

The language production by extended-H splicing systems employed in this research are then shown below.

Example 3 $\gamma = (\{a, b, c, d, e\}$
$\{a, b, c, d\}, \{cabd, caebd\}, \{c\#ca\#ebd, ce\#\$b\#d\})$

Provided that the first rule is applied to the axioms, the resulting language is illustrated as follows.

$$(c|a^n b^n d, ca|ebd) \longrightarrow (cebd, ca^{n+1} b^n d)$$

Therefore, the axiom generated from the 1st splicing will be spliced by employing the 2nd rule, as shown below.

$$(ce|bd, ca^{n+1} b^n|d) \longrightarrow (ced, ca^{n+1} b^{n+1} d)$$

The language generated by the extended H splicing system is expressed by $L(\gamma) = \{ca^n b^n d | n \geq 1\}$.

Then, two lemmas recalled from n-th order limit language generalization, involving the combination of the string as well as the pattern of the string, are presented below.

Lemma 1, [7]: If $b \geq 1$ rule is employed in the splicing system, then $2(b - 1)$ different combination of the string in the language is generated.

Lemma 2, [7]: If c number of initial strings are employed in the splicing system, then $c(2c + 1)$ pattern of strings pertaining to the language is achieved.

The next section introduces the *n*-th order limit language based on the perspective of the extended-H (EH) splicing system.

3 *n*-th order Limit Language in Extended-H (EH) Splicing System Perspectives

In this section, we revisit and refine the concept of the *n*-th order limit language, reinterpreting it in the framework of the EH splicing system. The revised definition not only clarifies the original concept but also lays the groundwork for introducing a new understanding of the *n*-th order limit language generated by the EH splicing system. By framing the definition within this context, we aim to provide a deeper insight into how these languages evolve under different splicing system in term of language generation. Additionally, several theorems and lemmas are presented to explore and solidify this new definition, offering a thorough examination from a rule-based perspective in EH splicing system. These theoretical tools help to formalize the behaviour of the *n*-th order limit language, ensuring a comprehensive understanding of its structure and the mechanisms that govern its generation of language in the EH splicing system.

The *n*-th order limit language definition from the generation of language perspectives built upon the EH splicing system is described as follows.

Definition: Suppose $\gamma = (V, T, A, R)$ is an EH splicing system in which V is an alphabet of the system, $T \subseteq V$ denotes a terminal alphabet, $A \subseteq V^*$ represents a set of axioms, while R refers to a set of rules over A, where $R_n, 1 \leq n \leq k$.
We define
$L(\gamma) = \sigma^*(A) \cap T^* \neq L_n(\gamma) = \sigma^*(A) \cap T^*$ and L_n can be obtained in $L(\gamma)$ by removing string that can be spliced using rules.

Suppose $L(\gamma)$ represents the splicing language from the splicing system. Then, we express $L_n(\gamma)$ given that n represents the limit language order that is defined by the number of rules utilised in the splicing system. Thus, the rule must be different from each other, and the length of the rule must be the same. The language formed by splicing system is $L(\gamma) = \sigma^*(A) \cap T^*$. Therefore, $L_n(\gamma)$ is different from the set of string $L_{(n+1)}(\gamma), L_{(n+2)}(\gamma), \ldots$, provided that $\bigcap_{i=1}^{n} L_i = \emptyset$ with $L_1(\gamma) \not\subset L_2(\gamma) \not\subset \ldots \not\subset L_n(\gamma)$.

This definition illustrates the language formed by the EH splicing system, a language-based splicing system, which bears similarities to the language established by the Head splicing system. These similarities will be proven in the theorem below.

Theorem 1. For every language produced by the Head splicing system, there is an equivalent language in the EH splicing system.

Proof. Suppose $S = (A, I, B, C)$ represents a Head splicing system while $y = (V, T, A, R)$ is the EH splicing system. By contrasting the tuples that exist in the splicing system, A (in the Head splicing system) is a set of alphabets similar to the tuple V (in the EH splicing system), which is the variables. For instance, I (in the Head splicing system) is the initial string, which is also homogeneous with the tuple A (in the extended H splicing system). Since the tuple B and C are equivalent to the rule used in the Head splicing system, they are also similar to the R tuple in the extended H splicing system. The claim is that both splicing system generates the same splicing language, and this claim will be proved by direct proof for the initial string and an axiom in the Head and EH splicing system, respectively.

Let S be a Head splicing system having an initial string and a rule, $S = (w, x, y, z, \mu wxzz\gamma, w, xz, z, \emptyset)$, in which w and z as well as x and y, represent the complements of each other with $\mu, \gamma, w, x, y, z \in A^*$. The initial string is described below.

$$5' - \mu wxzz\gamma - 3'$$
$$3' - \mu' zyww\gamma' - 5'$$

The language resulting from the splicing process appears as given below.

$$L(S) = \{\mu wxzz\gamma, \mu wwyz\mu', \gamma' wxzz\gamma\},$$

where $\mu', \gamma' \in A^*$. It can be generalized as $L(S) = \{\mu \ldots \gamma, \mu \ldots \mu', \gamma' \ldots \gamma\}$, where a part of the string, $wxzz, wwyz$ can be any alphabets.

The order is determined by the number of rules. Thus, the given expression for the first-order limit language or the limit language of the splicing system is given below.

$$L_1(S) = \{\mu \ldots \mu', \gamma' \ldots \gamma\},$$

This EH splicing system uses a rule and an axiom. The following can be obtained such as follows.

Let $\gamma = (a, b, a, b, ab, ab)$ represent an extended H splicing system having one axiom as well as a rule. Following the implementation of the rule, the axiom will be divided into 2 parts.

$$a|b$$

In that case, the splicing language formed by the aforementioned splicing system can be restated as follows.

$$L(\gamma) = \sigma^*(A) \cap T = ab, aa, bb$$

Following the utilization of a rule within the splicing system, the 1st-order limit language is delineated below.

$$L_1(\gamma) = \{aa, bb\}$$

For comparing both splicing systems in Case 1, an alternative writing approach is suggested for the splicing language formed from the Head splicing system (1) and the EH splicing system (2), including:

$$L(S) = \{\mu \ldots \gamma, \mu \ldots \mu', \gamma' \ldots \gamma\} \quad (1)$$

$$L(\gamma) = \{ab, aa, bb\} \quad (2)$$

The first order limit language generated from the Head splicing system (3) as well as the EH splicing system (4) can be presented as follows.

$$L_1(S) = \{\mu \ldots \mu', \gamma' \ldots \gamma\} \quad (3)$$

$$L_1(\gamma) = \{aa, bb\} \quad (4)$$

Based on the explanations above, the language pattern formed by the Head splicing system is similar to the language produced by the EH splicing system. The same pattern can be expected for the different number of axiom/initial strings as well as rules in the Head splicing system for the respective language in EH SS.

Theorem 2. For every *n*-th order limit language generated by the Head splicing system, there exists an equivalent *n*-th order limit language in the EH splicing system.

Proof. By referring to Equation 1 and 2 for the splicing language and Equation 3 and 4 for limit language obtained by the splicing systems, the splicing language as well as the order of the limit language is equivalent. It utilises the same number of rules in both spicing systems. The claim is that the spicing system $\gamma = (V, T, A, R)$ forms the same splicing language and limit language as splicing system $\gamma = (V, T, A, R)$, and this claim will be proven by induction:

Let $S = (\{w, x, y, z\}, \{\mu wxzz\gamma\}, \{w, xz, z\}, \emptyset)$ be Head splicing system as well as $\gamma = (a, b, a, b, ab, ab)$ be EH splicing system which operates by a rules, w, xz, z and ab respectively. We can observe that using the same rules results in the splicing language as well as the limit language being identical. Furthermore, the order of the limit language remains consistent.

Assume that the *k*-th iteration splicing in $S = (A, I, B, C)$, every $L(S)$ that generates by splicing in $S = (A, I, B, C)$ also exist by splicing in $\gamma = (V, T, A, R)$ in *k* steps and will introduce as $L(\gamma)$. Suppose $L(S)$ be a splicing language obtained by splicing in *k* steps in $S = (A, I, B, C)$ and assume that $L(S)$ may be spliced using rules which can be B or C, generating a splicing language $L(S)$ in $(k+1)^{th}$ iteration of splicing.

By hypothesis, it is known that $L(\gamma)$ exists in language obtained by k iteration of splicing in

$\gamma = (V, T, A, R)$. The arguments show that initial I can be spliced in $S = (A, I, B, C)$ to obtain the same product $L(\gamma)$ in $(k+1)^{th}$ iteration in $\gamma = (V, T, A, R)$ in exactly the same argument in basic steps of the proof.

Recalling Lemma 1 and Lemma 2, which discuss the number of combinations and patterns of strings in the limit language generated by the Head splicing system as described in the preliminaries section, the following lemmas demonstrate that using the EH splicing system results in the same number of combinations and patterns of strings in the limit language. This conclusion is drawn because the same string patterns are observed in Theorem 1 and Theorem 2.

Lemma 3. Every combination of the language produced by the EH splicing system and Head splicing system is $2(b-1)$, where b denotes a set of axioms.

Proof. b number of rules in the EH splicing system and Head splicing system will produce the same $2(b-1)$ combination of the strings since $B \cup C \in R$. Hence, the theorem is proved.

Lemma 4. Every pattern of the language produced by the Head splicing system and EH splicing system $\gamma = (V, T, A, R)$ is $c(2c+1)$ in which c represents a set of rules.

Proof. c number of axiom in the EH splicing system and c number of initial string in the Head splicing system will produce the same $c(2c+1)$ pattern of the string since $I \in A$. Hence, the theorem is proved.

According to the lemma provided, the rules employed in the splicing system will influence the synthesis pertaining to the splicing language, while the initial strings or axiom utilized in the splicing system will determine the language pattern generated. Then, the theorem below is presented.

Theorem 3. The EH splicing system producing n-th order limit language ,$\gamma = (V, T, A_c, R_b)$ contain of c number of axiom and b number of rules based on the enhanced n-th order limit language definition relying on the rules perspectives.

Proof. The n-th order limit language yields $2(b-1)$ combinations of strings and $c(2c+1)$ patterns of strings relying on the Lemma 3 as well as Lemma 4. The theorem has been demonstrated for both EH, and it has been shown that the Head splicing system will generate the identical splicing language. Thus, the following generalization is obtained: the language as follows is produced by combining the lemmas for b rules and c axioms in order to generalize the n-order of the limit languages.

4 Conclusion

The transition of this study from a purely biological perspective to a language-based splicing system is clearly articulated through the various definitions, theorems, and lemmas discussed earlier. This shift represents a significant change in focus, moving away from models that are solely grounded in biological contexts and instead embracing the principles of language generation.

In this new framework, the study explores how splicing systems can be applied to understand language generation. The theorem presented effectively shows that the results obtained from the Head splicing system are remarkably similar to those derived from the Extended-H splicing system. Specifically, it demonstrates that both systems produce comparable outcomes when it comes to splicing language, limit language, and the order of the limit language.

By highlighting these similarities, the study underscores the potential for applying splicing systems to a broader range of contexts beyond biological applications. This approach not only validates the use of language-based splicing systems but also illustrates their versatility and relevance in various fields of study. The comparison of the Head and Extended-H splicing systems provides valuable insights into how different splicing systems can yield similar results, thereby enhancing our understanding of language generation processes and their underlying structures.

Acknowledgment:
The authors express their gratitude to the Universiti Malaysia Pahang Al-Sultan Abdullah for its financial support through RDU223229 and Tabung Persidangan Luar Negara (TPLN) for this research.

Declaration of Generative AI and AI-assisted Technologies in the Writing Process
During the preparation of this work, the authors used gemini.google.com and ChatGPT.com in order to improve and rephrase the sentence of the article. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

References:

[1] J. Dassow and G. Vaszil, Multiset splicing systems. *Biosystems*, Vol.74, No.1-3, 2003, pp. 1-7.

[2] A. Condon and G. Rozenberg, DNA Splicing Systems, *DNA Computing*, Springer Berlin, Heidelberg, 2008.

[3] K. Ruohonen, Formal languages, *Advanced Textbooks in Control and Signal Processing*, Springer London, United Kingdom, 2007.

[4] T. Head, Formal language theory and DNA: An analysis of the generative capacity of specific recombinant behaviour, *Bulletin of Mathematical Biology*, Vol.49, No.6, 1987, pp. 737-759.

[5] J. Berstel, L. Boasson, and I. Fagnot, Splicing Systems and the Chomsky Hierarchy. *Theoretical Computer Science*, Vol. 436, pp. 2-22.

[6] E. Goode and D. Pixton, Splicing to the Limit, *Aspects of Molecular Computing*, Springer, Berlin, Heidelberg, 2003.

[7] S.H. Mohd Khairuddin, Modelling of *n*-th Order Limit Language using Automata in DNA Splicing System, Master's thesis, Universiti Malaysia Pahang Al-Sultan Abdullah, Pahang, Malaysia, 2022.

[8] E. Csuhaj Varjú, L. Kari and G. Paun, Test Tube Distributed Systems based on Splicing, *Computing and Informatics*, Vol. 28, No. 1, 1998, pp. 1-21.

[9] L. Peter, *An Introduction to Formal Languages and Automata*, 6th ed. Jones and Bartlett, LLC, Massachusetts, United States, 2017.

[10] A. Pettorossi, *Automata Theory and Formal Languages*, Springer Natural Switzerland, Switzerland, 2022.

[11] G. Rozenberg and A. Salomaa, *Handbook of Formal Languages*.Springer-Verlag Berlin Heidelberg, 1997.

[12] P. V. Chavan and A. Jadhav, *Automata Theory and Formal Languages*, Elsevier Science, United Kingdom, 2023.

[13] G. Păun, Regular Extended H Systems are Computationally Universal, *Journal of Automata, Languages and Combinatorics*, Vol. 1, No. 1, 1996, pp. 27-36.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors equally contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

This research is fully funded by grant RDU223229 and Tabung Persidangan Luar Negara (TPLN) from Universiti Malaysia Pahang Al-Sultan Abdullah.

Conflicts of Interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.