REVIEW OF INTERNATIONAL GEOGRAPHICAL EDUCATION

ISSN: 2146-0353 • © RIGEO • 11(4), WINTER, 2021

www.rigeo.org Research
Article

# Application of Automata Theory On n-th Order Limit Language

**Siti Hajar Mohd Khairuddin[1]***
Centre for Mathematical Sciences, Universiti Malaysia Pahang, 26300 Lebuhraya Tun Razak, UMP Gambang, Pahang, Malaysia

**Muhammad Azrin Ahmad[2]**
Centre for Mathematical Sciences, Universiti Malaysia Pahang, 26300 Lebuhraya Tun Razak, UMP Gambang, Pahang, Malaysia

**Noraziah Adzhar[3]**
Centre for Mathematical Sciences, Universiti Malaysia Pahang, 26300 Lebuhraya Tun Razak, UMP Gambang, Pahang, Malaysia

[1] Corresponding Author: Email: sitihajarmohdkhairuddin96@gmail.com

**Abstract**
The application of automata theory on the DNA splicing system is rapidly growing from time to time. The idea of a splicing system is formalized by Tom Head in 1987. There are three essential parts in the splicing system models, which are the alphabets, initial strings, and the rules. The alphabets represent the nucleotides or the DNA, known as Adenine, Thymine, Guanine, and Cytosine, which are later abbreviated as a, t, g, c following Watson-Cricks complementary. On the other hand, the set of rules represents the restriction enzyme used for the splicing process. In this research, automata theory is used to transform the limit language into a transition graph. The n-th order limit language is then derived from grammar shown as an automated diagram and shown by transition graphs, which represent the language of transitional labels of DNA molecules derived from the respective splicing system.

**Keywords**
Formal Language Theory, DNA Splicing System, DNA Splicing Language, Automata Theory.

**To cite this article:** Khairuddin, S, H, M.; Ahmad, M, A.; and Adzhar, N. (2021) Application of Automata Theory On n-th Order Limit Language. *Review of International Geographical Education (RIGEO), 11(4), 817-824. doi:* 10.48047/rigeo.11.04.75

**Submitted:** 20-03-2021 • **Revised:** 15-04-2021 • **Accepted:** 17-05-2021

# Introduction

DNA is an abbreviation of deoxyribonucleic acid (Alberts et al., 2014). Long-term storage of information is the principal function of DNA in cells. DNA is a long polymer made up of basic nucleotides known by sugar and phosphate groups in the backbone (Picardi, 2015). This backbone is made up of four groups of molecules known as adenine, thymine, guanine and cytosine. An additional DNA line can be built on the complementarity of the nucleobase (Watson & Crick, 1953). Each base pair $A=T$ and $G\equiv C$ occupy almost the same space and thus makes a two-helix twisted DNA formation. The double helix DNA is also reinforced by hydrogen bonding between the nucleobases.

A splicing system is a formal model of a recombinant activity for double-stranded DNA (dsDNA) molecules in combination with a ligase and restriction enzyme (Head, 1987). The introduction of new computational models, inspired by bio-processes which contribute to its renovation, was a positive influence on formal language theory. The mathematical formalism for biological phenomena of recombinant processes proposed by Head in 1987 is a specific example of the evolution of splicing systems (Head, 1987). Head proposed combining the formal language theory and molecular biology. The DNA strand is made from a spinal cord of connected deoxyribose molecules, each of which holds an adenine, guanine, cytosine, and thymine in one of four bases. Thus, it is seen as a four-letter alphabet string (or nucleotides *a, g, c, t*), and hence it is within the scope of Formal Language Theory to shape the natural way for DNA computation (Head, 1998). This idea led to the introduction of a language that defines by splicing system or known as splicing language.

Inert, transient and limit language are the three types of splicing language (Goode 2004). Meanwhile, in this research, a limit language is used. Limit language is the number of terms that will occur if a certain quantity of each original molecule occurs and ample time to achieve equilibrium, independent of the equilibrium of the reactants in a certain experimental period (Goode, 2014). Thus, this language is generated by grammar since grammar is a generator of the splicing language.

Automata theory is a concept of abstract machines and automatons and the questions of programming which can be overcome by them. (Peter, 2012). Fundamentally, it is a philosophy of computer science. Deterministic finite automata (dfa) and non-deterministic finite automata (nfa) are the types of automata. Both of these forms can be depicted in the form of a transition graph. The transition graph can be interpreted as a flowchart for an algorithm recognizing a language (Peter, 2012). Based on the previous study in (Yusof et al., 2011), the transition graph was used by eliminating the vertices to obtain the inert persistence splicing languages which lie in limit languages. Then, in (Ahmad, 2016) the transition graph is used to show the flow of the language. In this paper, transition graphs are given to show the difference between splicing language and the third, fourth and fifth-order limit language. (Fong & Ismail, 2018) described that the concepts in automata theory and grammar are applied in DNA splicing systems with a single cutting site, using finite automated dfa, in each palindrome and non-palindrome regulations for the same and different crossings.

The researchers previously had discussed until second order limit language by using a transition graph. In this research, the higher order limit language other than the second order limit language is discussed and presented using a transition graph. The higher order limit language can be determined using the definition of *n*-th order limit language which have been improvised from the original definition given by (Goode, 2004). In this research, automata are used in presenting the transition graph from the grammar which has been generated by the limit languages.

## Preliminaries

**Definition 1 (Peter, 2012) : Alphabets, A**

An alphabet $A$ is a finite non-empty set of symbols.

**Definition 2 (Peter, 2012) : Strings**

A string is a finite set of alphabet symbols.

**Definition 3 (Peter, 2012): Language, L**

A set of strings that have been selected from $A^*$, which $A$ is a single alphabet is known as a language.

**Definition 4 (Goode 2004): *n*-th Order Limit Language**

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 $L_{n-1}$.

Subsequently, it can be 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 contexts, 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 site, $x$ 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 $L_{n-1}(S)$, $L_{n-2}(S)$, …, such that $L_n(S) \cap L_{n-1}(S) \cap L_{n-2}(S), \ldots = \varnothing$ and $L_n(S) \not\subset L_{n-1}(S) \not\subset L_{n-2}(S), \ldots$. (M. Khairuddin, 2020)

Next, the definition of grammar and dfa is presented as follows. The grammar is used as a generator while the acceptor is dfa.

**Definition 5 (Peter, 2012): Grammar, Language Generated by a Grammar**

Grammar $G$ shall be described as a quadruple $G = (V, T, S, P)$ where
$V$ is a finite set of objects that have been called variables
$T$ is a finite set of objects recognised as terminal symbols
$S \in V$ is a special symbol recognised as the start variables
$P$ is a finite set of productions

**Definition 6 (Peter, 2012): Deterministic Finite Automata**

Formally, a DFA is a 5-tuple $M = (Q, \Sigma, \delta, q_0, F)$ where
$Q$ is a finite set recognized as the states
$\Sigma$ is a finite set recognized as the alphabets
$\delta : Q \times \Sigma \to Q$ is the transition function
$q_0 \in Q$ is the starting states
$F \subseteq Q$ is the set of accepting states
Then, we proceed with the results and discussions.

## Results and Discussion

This section presents several cases that will be discussed to form the theorems. So, the explanation of the cases is first presented, followed by the discussion of the theorem mentioned above. Three cases are presented.

| Case | Order | Limit Languages | The language that generates by grammar |
|---|---|---|---|
| 1 | 3 | $L_3=\begin{Bmatrix}\mu(J\cup H\cup K\cup M)\gamma,\mu(J\cup H\cup K\cup M)\mu',\\ \gamma'(J\cup H\cup K\cup M)\gamma\end{Bmatrix}$ | $G_1=(\{V=S_o,S_1,S_2,S_3,S_4,S_5,S_6,S_7\},\{S=S_0\},\{T=\mu,\mu'\gamma,\gamma',J,H,K,M\},P)$ *P* consisting of the productions $S_0\to\mu S_1\mid\gamma' S_5$, $S_1\to J,H,K,MS_2$, $S_2\to J,H,K,MS_1\mid\gamma S_3\mid\mu' S_4$, $S_5\to J,H,K,MS_6$, $S_6\to J,H,K,MS_1\mid\gamma S_7$, $S_{3,4,7}\to\lambda$ |
| 2 | 4 | $L_4=\begin{Bmatrix}\mu(J\cup H\cup K\cup M\cup L\cup N)\gamma,\\ \mu(J\cup H\cup K\cup M\cup L\cup N)\mu',\\ \gamma'(J\cup H\cup K\cup M\cup L\cup N)\gamma\end{Bmatrix}$ | $G_2=(\{V=S_o,S_1,S_2,S_3,S_4,S_5,S_6,S_7\},\{S=S_0\},\{T=\mu,\mu',\gamma,\gamma',J,H,K,M,L,N\},P)$ *P* consisting of the productions $S_0\to\mu S_1\mid\gamma' S_5$, $S_1\to J,H,K,M,L,NS_2$, $S_2\to J,H,K,M,L,NS_1\mid\gamma S_3\mid\mu' S_4$, $S_5\to J,H,K,M,L,NS_6$, $S_6\to J,H,K,M,L,NS_1\mid\gamma S_7$, $S_{3,4,7}\to\lambda$ |
| 3 | 5 | $L_5=\begin{Bmatrix}\mu(J\cup H\cup K\cup M\cup L\cup N\cup O\cup I)\gamma,\\ \mu(J\cup H\cup K\cup M\cup L\cup N\cup O\cup I)\mu',\\ \gamma'(J\cup H\cup K\cup M\cup L\cup N\cup O\cup I)\gamma\end{Bmatrix}$ | $G_3=(\{V=S_o,S_1,S_2,S_3,S_4,S_5,S_6,S_7\},\{S=S_0\},\{T=\mu,\mu',\gamma,\gamma',J,H,K,M,L,N,O,I\},P)$ *P* consisting of the productions $S_0\to\mu S_1\mid\gamma' S_5$, $S_1\to J,H,K,M,L,N,O,IS_2$, $S_2\to J,H,K,M,L,N,O,IS_1\mid\gamma S_3\mid\mu' S_4$, $S_5\to J,H,K,M,L,N,O,IS_6$, $S_6\to J,H,K,M,L,N,O,IS_1\mid\gamma S_7$, $S_{3,4,7}\to\lambda$ |

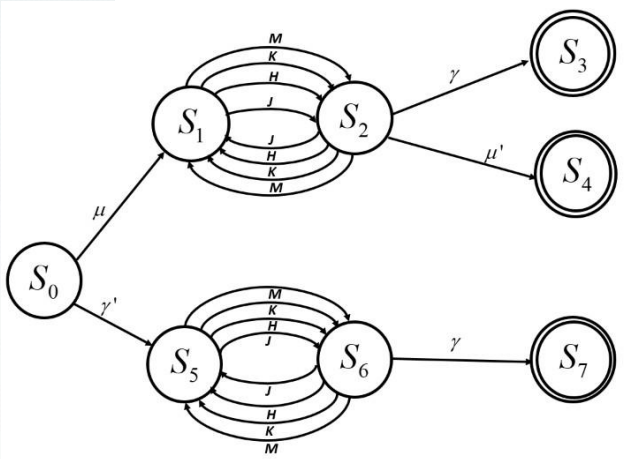
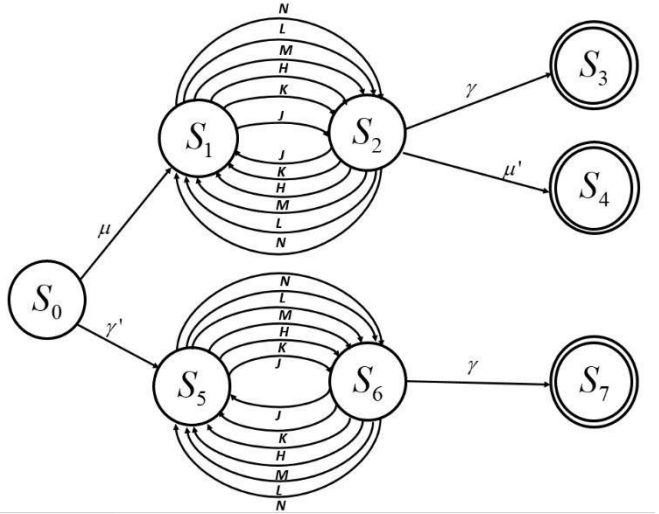
Based on the discussion and explanation above, Theorem 1 and 2 are introduced as follows.

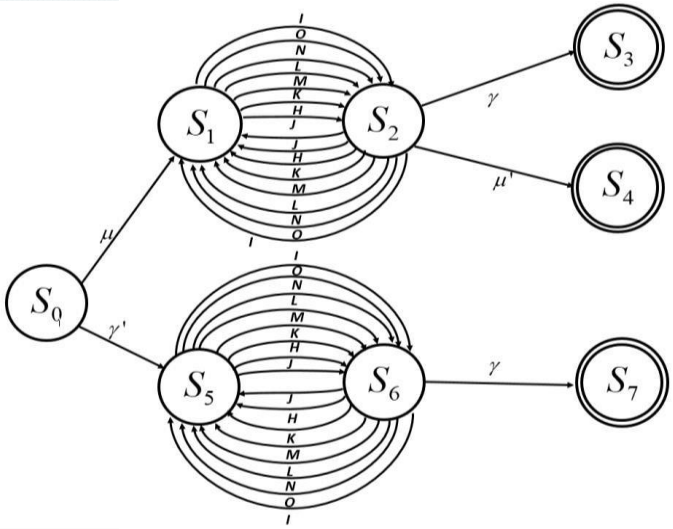
**Theorem 1**

If the combination of strings produced by the limit language is $2n-2$ then the number of lopping will follow the number of the combination of strings.

**Proof:**

The combination of strings of the limit language that can be produced by the splicing system depends on the number of rules in the splicing system. So, in this case, if the number of rules used is $n$, the *n* combination of the string produced is $2n-2$. Based on this theorem, the number of looping will also be $2n-2$.

| Cases | Deterministic Finite Automaton | Transition Graphs |
|---|---|---|
| 1 | $M_1 = Q = \{S_o, S_1, S_2, S_3, S_4, S_5, S_6, S_7\}, \Sigma = \{\mu, \gamma', J, H, K, M, \mu', \gamma\},$ $\delta, S_0, F = \{S_3, S_4, S_7\}$ where $\delta(S_0, \mu) = S_1,\ \delta(S_1, \mathrm{H}) = S_2,\ \delta(S_2, \gamma) = S_3,\ \delta(S_5, H) = S_1,$ $\delta(S_0, \gamma') = S_5,\ \delta(S_1, \mathrm{K}) = S_2,\ \delta(S_2, \mu') = S_4,\ \delta(S_5, \mathrm{K}) = S_1,$ $\delta(S_1, \mathrm{J}) = S_2,\ \delta(S_1, \mathrm{M}) = S_2,\ \delta(S_5, \mathrm{J}) = S_1,\ \delta(S_5, M) = S_1,$ $\delta(S_6, \gamma) = S_7,$ |  |
| 2 | $M_1 = Q = \{S_o, S_1, S_2, S_3, S_4, S_5, S_6, S_7\}, \Sigma = \{\mu, \gamma', J, H, K, M, L, N, \mu', \gamma\},$ $\delta, S_0, F = \{S_3, S_4, S_7\}$ where $\delta(S_0, \mu) = S_1,\ \delta(S_1, \mathrm{K}) = S_2,\ \delta(S_2, \gamma) = S_3,\ \delta(S_5, \mathrm{K}) = S_1,$ $\delta(S_0, \gamma') = S_5,\ \delta(S_1, \mathrm{M}) = S_2,\ \delta(S_2, \mu') = S_4,\ \delta(S_5, M) = S_1,$ $\delta(S_1, \mathrm{J}) = S_2,\ \delta(S_1, \mathrm{L}) = S_2,\ \delta(S_5, \mathrm{J}) = S_1,\ \delta(S_5, L) = S_1,$ $\delta(S_1, \mathrm{H}) = S_2,\ \delta(S_1, \mathrm{N}) = S_2,\ \delta(S_5, H) = S_1,\ \delta(S_5, M) = S_1,$ $\delta(S_6, \gamma) = S_7,$ |  |

| Cases | Deterministic Finite Automaton | Transition Graphs |
|---|---|---|
| 3 | $M_2 = Q = \{S_o, S_1, S_2, S_3, S_4, S_5, S_6, S_7\}, \Sigma = \{\mu, \gamma', J, H, K, M, L, N, O, I, \mu', \gamma, \mu', \gamma\},$ $\delta, S_0, F = \{S_3, S_4, S_7\}$ where $\delta(S_0, \mu) = S_1,\ \delta(S_1, \mathrm{M}) = S_2,\ \delta(S_2, \gamma) = S_3,\ \delta(S_5, M) = S_1,$ $\delta(S_0, \gamma') = S_5,\ \delta(S_1, \mathrm{L}) = S_2,\ \delta(S_2, \mu') = S_4,\ \delta(S_5, L) = S_1,$ $\delta(S_1, \mathrm{J}) = S_2,\ \delta(S_1, \mathrm{N}) = S_2,\ \delta(S_5, \mathrm{J}) = S_1,\ \delta(S_5, N) = S_1,$ $\delta(S_1, \mathrm{H}) = S_2,\ \delta(S_1, \mathrm{O}) = S_2,\ \delta(S_5, H) = S_1,\ \delta(S_5, O) = S_1,$ $\delta(S_1, \mathrm{K}) = S_2,\ \delta(S_1, \mathrm{I}) = S_2,\ \delta(S_5, \mathrm{K}) = S_1,\ \delta(S_5, I) = S_1,\quad \delta(S_6, \gamma) = S_7,$ |  |

For Case 1, the splicing system uses three rules which resulted in the third-order limit language. From this result, we can get $n = 3, 2(3) - 2 = 4.$ So, the splicing language has four combinations of strings of the language. Then, it can be observed that there are existences of four looping parts. Second, in Case 2, the splicing system are using four rules which resulted in the fourth-order limit language. From that, we can get $n = 4, 2(4) - 2 = 6.$ So the splicing language has six combinations of strings of the language. So that, we can see that there are existences of six looping part. Thirdly, in Case 3, the splicing system is using five rules which resulted in the fifth-order limit language. From that, we can get $n = 5, 2(5) - 2 = 8.$ Therefore, the splicing language has eight combinations of strings of the language. Thus, there are existences of eight looping part. Then, the theorem is proven. We then proceed to the second theorem.

**Theorem 2**

The number of end states of the transition graphs depends on the number of the pattern of strings of the limit language produced.

**Proof:**

In this theorem, the focus is on the number of end states in the transition graph. The pattern of the string of limit language is the strings produce after the splicing process. The third, fourth and fifth-order limit language are presented as follows.

$$L_3(S) = \{\mu(H \cup I \cup J \cup K)\gamma, \mu(H \cup I \cup J \cup K)\mu', \gamma'(H \cup I \cup J \cup K)\gamma\}$$

$$L_4(S) = \begin{Bmatrix} \mu(H \cup I \cup J \cup K \cup L \cup M)\gamma, \mu(H \cup I \cup J \cup K \cup L \cup M)\mu', \\ \gamma'(H \cup I \cup J \cup K \cup L \cup M)\gamma \end{Bmatrix}$$

$$L_5(S) = \begin{Bmatrix} \mu(H \cup I \cup K \cup K \cup L \cup M \cup N \cup O)\gamma, \mu(H \cup I \cup K \cup K \cup L \cup M \cup N \cup O)\mu', \\ \gamma'(H \cup I \cup K \cup K \cup L \cup M \cup N \cup O)\gamma \end{Bmatrix}$$

Based on the above limit language, we can see that the pattern of limit language is $\mu(\ldots)\gamma, \mu(\ldots)\mu'$ and $\gamma'(\ldots)\gamma$. All of the above cases used only one initial string in the splicing system. Thus, the end state of all transition graph for all the cases are three which are $S_3, S_4$ and $S_7$. Then the theorem is proven.

## Conclusion

In this research, the concepts in automata theory and grammar are applied in DNA splicing for defining the *n*-th order limit language to show the flow of language by using transition graph. The languages generated by the grammars depict the Generalized splicing languages consisting of the dsDNA strings. The splicing system used is Head splicing system that involves four cases which are an initial string and a rule, an initial sting and two rules, two initial strings and a rule and two initial strings and two rules. Two theorems were presented with the overall stated number of looping in the transition graph which is affected by the number of string combination and the number of end states of the transition graph depending on the number of string patterns of the limit language. Based on Theorem 1, we can see that the number of rules, *n* will affect the looping part in the transition graph. So that, if the number of rules used in the splicing system is *n* the combination of strings of the limit language and the number of lopping in the transition graph are $2n-2$. Secondly, based on Theorem 2, the end state can be determined by the number of patterns of the limit languages produce by the splicing system. From what we can see, the number of end states of each case is the same with the number of patterns of the string of the limit languages.

## Acknowledgement

All authors are indebted to the Ministry of Education (MOE) Malaysia for funding through the Fundamental Research Grant Scheme (FRGS), FGRS/1/2018/STG06/UMP/03/1-RDU190118.

## References

Ahmad, M. A. (2016). *Second Order Limit Language and its Properties in Yusof-Goode Splicing System*. Universiti Teknologi Malaysia.

Alberts B, Johnson A, Lewis J, Raff M, Roberts K, W. P. (2014). *Molecular Biology of the Cell (6th ed.)*. Garland. p.

Fong, W. H., & Ismail, N. I. (2018). *Generalisations of DNA Splicing Systems with One Palindromic Restriction Enzyme*. *34*(1), 59–71.

Goode, E. (2004). *Splicing to the Limit*. 189–201.

Goode, T. E. (2004). Splicing to the Limit. *Lecture Notes in Computing Science*.

Goode, T. E. (2014). *DNA Splicing Systems. DNA Splicing Systems an Ordinary Differential Equations Model and Simulation*. (January 2007).

Head, T. (1987). Formal Language Theory and DNA: An Analysis of the Generative Capacity of Specific Recombinant Behaviors. *Bulletin of Mathematical Biology*, *49*(6), 737–759.

Head, T. (1998). *Splicing representations of strictly locally testable languages*. (98).

Peter, L. (2012). *An introduction to formal languages and automata (5th ed.)*. USA:Jones and Bartlett, LLC.

M. Khairuddin, S.H. (2020). *Effect of m Number of Initial String on the n-th Order Limit Language*. Technology Reports of Kansai University. 62(7), 681-3689. Crossmark.

Picardi, E. (2015). RNA bioinformatics. *RNA Bioinformatics*, (January), 1–415. https://doi.org/10.1007/978-1-4939-2291-8

Watson, & Crick. (1953). *A structure for deoxyribose nucleic acid*. (4356), 737–738.

Yusof, Y., Sarmin, N. H., Goode, T. E., Mahmud, M., & Heng, F. W. (2011). An extension of DNA splicing system. *Proceedings - 2011 6th International Conference on Bio-Inspired Computing: Theories and Applications, BIC-TA 2011*, 246–248.