Modelling Intermolecular Structures and Defining Ambiguity in Gene Sequences using Matrix Insertion-Deletion Systems

Lakshmanan Kuppusamy, Anand Mahendran, Éric Villemonte de La Clergerie

Modelling Intermolecular Structures and Defining Ambiguity in Gene Sequences using Matrix Insertion-Deletion Systems[1]

Lakshmanan Kuppusamy [a,2], Anand Mahendran [a] and Eric Villemonte de la Clergerie [b]

[a] *School of Computing Science and Engineering*
VIT University
Vellore-632 014, India.
Email:{klakshma, manand}@vit.ac.in

[b] *INRIA-Rocquencourt*
B.P. 105, Le Chesnay - 78153, France.
Email: Eric.De_La_Clergerie@inria.fr

Abstract. Gene insertion and deletion are considered as the basic operations in DNA processing and RNA editing. Based on these evolutionary transformations, a computing model has been formulated in formal language theory known as *insertion-deletion* systems. Recently, in [6], a new computing model named *Matrix insertion-deletion system* has been introduced to model various bio-molecular structures such as *hairpin*, *stem and loop*, *pseudoknot*, *attenuator*, *cloverleaf*, *dumbbell* that occur at intramolecular level. In this paper, we model some of the intermolecular structures such as *double strand languages*, *nick languages*, *hybrid molecules* (*with R-loops*), *holliday structure*, *replication fork* and *linear hybridization* (*ligated*) *languages* using Matrix insertion-deletion system. In [2], the ambiguity in gene sequence was defined as deriving more than one structure for a single gene sequence. Here, we propose a different view of understanding the ambiguity in gene sequences: A gene sequence is obtained by more than one way such that their intermediate sequences are different. We further classify the ambiguity into many levels based on the components *axiom*, *string* (order of deletion/insertion) and *contexts* (order of the used contexts). We notice that some of the inter and intramolecular structures obey the newly defined ambiguity levels.

Keywords. insertion-deletion systems, intermolecular structures, ambiguity, matrix grammars, gene sequences

Introduction

Insertion-deletion systems are introduced to theoretically analyze the insertion and deletions operations that take place in gene sequences. Informally, the insertion and deletion

[1]This work was partially supported by the project SR/S3/EECE/054/2010, Department of Science and Technology (DST), New Delhi, India.

[2]Corresponding Author: School of Computing Science and Engineering, VIT University, Vellore 632 014, India.E-mail: klakshma@vit.ac.in

operations of an insertion-deletion system is defined as follows: If a string β is inserted between two parts w_1 and w_2 of a string w_1w_2 to get $w_1\beta w_2$, we call the operation as insertion, whereas if a substring α is deleted from a string $w_1\alpha w_2$ to get w_1w_2, we call the operation as deletion. These systems opened a particular attention in the field of formal languages as the system is not exactly based on rewriting systems.

As DNA and RNA molecules can be considered as a sequence of symbols (i.e., strings) over $\{a, t, g, c\}$ and $\{a, u, g, c\}$ respectively, problems that exist in such molecules can be studied using formal grammars. The following example witnesses the correlation between formal grammars and bio-molecular structures. Consider a context-free language $\{ww^R \mid w \in \{a, b\}^*\}$ where w^R is the reverse of w. Consider the following gene sequence $gcatgcgcatgc$. As the complementary pairs $\bar{a} = t$, $\bar{t} = a$, $\bar{g} = c$ and $\bar{c} = g$, the above gene sequence resembles a word in the palindrome language $\{w\bar{w}^R \mid w \in \{a, t, g, c\}^*\}$.

There exist some relations between intramolecular gene sequences and natural language constructions such as *triple agreements :* $\{a^nb^nc^n \mid n \geq 1\}$, *crossed dependencies:* $\{a^nb^mc^nd^m \mid n, m \geq 1\}$ and *copy language:* $\{ww \mid w \in \{a, b\}^*\}$ [1], [2]. In below, we discuss briefly some of the important intra and intermolecular structures that are predominantly available in bio-molecules such as protein, DNA and RNA. Fig.1 shows the intramolecular structures (a) *stem and loop*, (b) *cloverleaf* and (c) *dumbbell*. Note that these structures can be represented by context-free grammars. However, there are some more structures that occur frequently in bio-molecules which cannot be modelled by context-free grammars. Fig.2 represents such intramolecular structures (a) *pseudoknot* and (b) *attenuator*. In Fig. 1(a) and Fig. 2, the strings are obtained by reading the symbols as per directed dotted lines. The string $ugcucaag$ (refer Fig. 2a) represents the pseudoknot structure, the string $agaucuaga$ (refer Fig. 2b) represents the attenuator structure and they resemble the crossed dependency language and the copy language respectively.

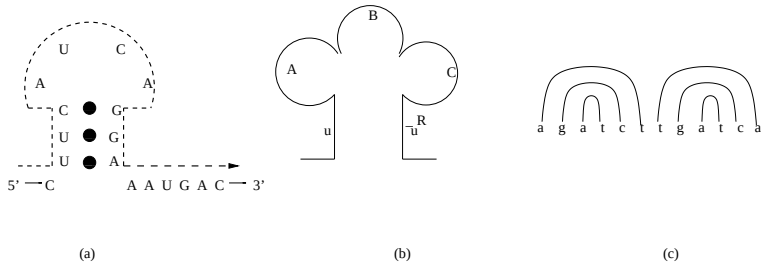

Figure 1. Intramolecular structures: (a) stem and loop (b) cloverleaf (c) dumbbell

In RNA, the gene sequences are mostly intermolecular rather than intramolecular. Such intermolecular structures are modelled by cutting the intramolecular secondary structures randomly by leaving some open points such that base pairing can be done at these open points. Fig.3 represents some of the intermolecular structures (a) *double strand languages* and (b) *nick languages* where the cut takes place at arbitrary position is represented by a $\bullet$. In Fig.3 S, stands for the non-terminal in their corresponding context-free grammar.

In the last two decades, attempts have been used to model these intra and inter molecular structures by defining new grammar formalisms like *crossed-interaction grammar*

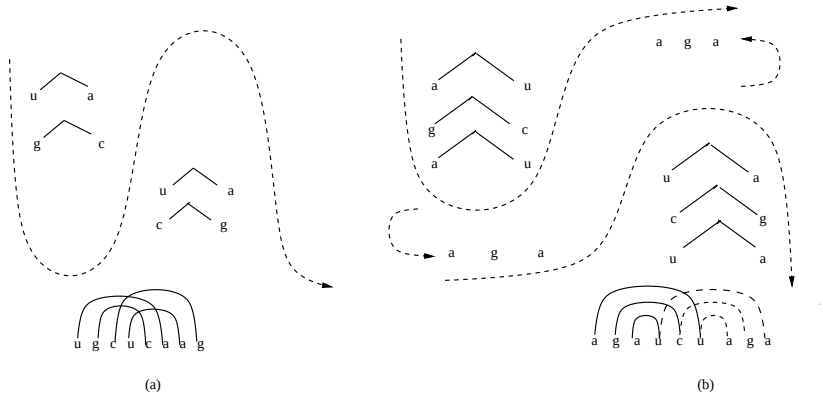

Figure 2. Intramolecular structures: (a) pseudoknot structure (b) attenuator

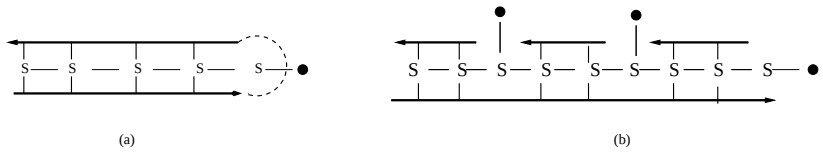

Figure 3. Intermolecular structures: (a) double stranded molecule (b) nick language

[4], *simple linear tree adjoining grammars* and *extended simple linear tree adjoining grammars* [11]. In [3], *cut grammars, ligation grammars* were defined specifically to model intermolecular structures. However, there is no unique grammar system that encapsulates all essential and important (intra/inter) bio-molecular structures. For example double copy language cannot be modelled by a simple linear tree adjoining grammar [11].

Very recently, in [6] a new simple and powerful biologically inspired computing model namely *Matrix insertion-deletion* system has been introduced by combining insertion-deletion system and matrix grammar with a strong motivation to model the above mentioned intramolecular structures in terms of languages. Though, insertion-deletion system itself is computationally complete, and therefore possible to generate any recursive enumerable language, constructing insertion-deletion grammars for some languages (that form bio-molecular structures) is not easy and the procedure to simulate the grammar is very tedious. For example, consider the triple agreement language $\{a^n b^n c^n \mid n \geq 1\}$ that forms the triple stranded DNA. Defining the insertion-deletion grammar is very difficult since we need to make sure that whenever one a is introduced, simultaneously, one b and c must be introduced at the appropriate places, thus the synchronization between a, b and c is taken care. To do this in insertion-deletion system, we need to introduce a new marker near the inserted a and move the marker towards right by passing all a until a b is encountered (for inserting b) and then move further to encounter a c (to introduce a c). Besides, the marker need to be deleted on crossing each symbol. If not constructing this way, we need to identify the Chomsky grammar which generate this language, and convert it to a Kuroda (or suitable) normal form and from that nor-

mal form, we can obtain the insertion-deletion grammar using the computational completeness result. This is also not an easy way when compared to the grammar of Matrix insertion-deletion system as we are going to see how simply we can construct a grammar in Matrix insertion-deletion system for such languages and how freely the synchronization is achieved. In [7], the computational completeness of Matrix insertion-deletion system has been analyzed. Interestingly, in [8], the same Matrix insertion-deletion system has been introduced independently but without such motivations and its computational completeness has been analyzed. In this paper, we model some of the intermolecular structures that are predominantly available in gene sequences (more precisely in RNA and proteins).

Ambiguity is considered to be one of the important issues not only in natural and programming languages, but also in gene sequences. In [6], the ambiguity in gene sequences are dealt informally by showing that there exists some bio-molecular structures which has the notion of ambiguity. Since the insertion-deletion system can be applied theoretically in DNA processing [5], the ambiguity in DNA processing (which uses the insertion-deletion system) may happen in the following manner. Let W_1W_2 be a DNA strand and suppose we want to insert $W_3W_4W_5$ between W_1 and W_2 to obtain another DNA strand $W_1W_3W_4W_5W_2$. This can be done first by inserting W_3 between W_1 and W_2, followed by inserting W_4 between W_3 and W_2, followed by inserting W_5 between W_4 and W_2. The other sequence would be first by inserting W_5 between W_1 and W_2, followed by inserting W_4 between W_1 and W_5, followed by inserting W_3 between W_1 and W_4. More precisely the derivations are given below: (the underlined string denotes the inserted string) (1) $W_1W_2 \Longrightarrow W_1\underline{W_3}W_2 \Longrightarrow W_1W_3\underline{W_4}W_2 \Longrightarrow W_1W_3W_4\underline{W_5}W_2$ (2) $W_1W_2 \Longrightarrow W_1\underline{W_5}W_2 \Longrightarrow W_1\underline{W_4}W_5W_2 \Longrightarrow W_1\underline{W_3}W_4W_5W_2$. This shows that ambiguity in gene sequences is also possible (i.e., starting from one sequence we are able to get another sequence in more than one way such that the intermediate sequences are different). With this view of ambiguity, in this paper, we formally define the various levels of ambiguity for Matrix insertion-deletion systems. Study of this concept of ambiguity may be useful while considering inheritance properties and phylogenetic trees [10].

1. Preliminaries

Here, we recall the basic notions which are used in the paper. A finite non-empty set V or Σ is called an alphabet. We denote by V^* or Σ^* , the free monoid generated by V or Σ, by λ its identity or the empty string, and by V^+ or Σ^+ the set $V^* - \{\lambda\}$ or $\Sigma^* - \{\lambda\}$. The elements of V^* or Σ^* are called *words* or *strings*. For a string w, $|w|_b$ denotes number of b in w. The language of DNA (RNA) can be considered over $\Sigma^*_{DNA(RNA)} = \{a, t(u), g, c\}$. For more details on formal language theory, we refer to [9].

Next, we look into the basic definitions of insertion-deletion systems. Given an insertion-deletion system $\gamma = (V, T, A, R)$, where V is an alphabet, $T \subseteq V$, A is a finite language over V, R is a finite triples of the form $(u, \alpha/\beta, v)$, where $(u, v) \in V^* \times V^*$, $(\beta, \alpha) \in (V^+ \times \{\lambda\}) \cup (\{\lambda\} \times V^+)$. The pair (u, v) is called contexts. Insertion rule will be of the form $(u, \lambda/\beta, v)$ which means that β is inserted between u and v. Deletion rule will be of the form $(u, \alpha/\lambda, v)$, which means that α is deleted between u and v. In

other words, $(u, \lambda/\beta, v)$ corresponds to the rewriting rule $uv \rightarrow u\beta v$, and $(u, \alpha/\lambda, v)$ corresponds to the rewriting rule $u\alpha v \rightarrow uv$.

Consequently, for $x, y \in V^*$ we can write $x \Longrightarrow^* y$, if y can be obtained from x by using either an insertion rule or a deletion rule which is given as follows: (the down arrow $\downarrow$ indicates the position where the string is inserted, the down arrow $\Downarrow$ indicates the position where the string is deleted and the underlined string indicates the string deleted/inserted)

1. $x = x_1 u^{\downarrow} v x_2,\ y = x_1 u \underline{\beta} v x_2$, for some $x_1, x_2 \in V^*$ and $(u, \lambda/\beta, v) \in R$.
2. $x = x_1 u \underline{\alpha} v x_2,\ y = x_1 u^{\Downarrow} v x_2$, for some $x_1, x_2 \in V^*$ and $(u, \alpha/\lambda, v) \in R$.

The language generated by γ is defined by

$$L(\gamma) = \{w \in T^* \mid x \Longrightarrow^* w,\ for\ some\ x \in A\}$$

where $\Longrightarrow^*$ is the reflexive and transitive closure of the relation $\Longrightarrow$.

Next, we will look into the definition of cut grammars. A cut grammar $G = (N, \Sigma, S, P)$ where N is a finite set of non-terminals, Σ is a finite set of terminals, S is a start symbol and P is a finite set of productions in $(N \cup \Sigma)^* N (N \cup \Sigma)^* \times (N \cup \Sigma \cup \{\diamondsuit\})^*$. where $\diamondsuit$ is a new symbol called cut symbol not in N or Σ. The language generated by cut grammar is defined as $L(G) = \{w \in (\Sigma \cup \diamondsuit)^* \mid S \Longrightarrow^* w\}$. Given any string $w = w_1 \diamondsuit w_2 \diamondsuit \ldots \diamondsuit w_n$ where $w_i \in \Sigma^*$ for $1 \leq i \leq n$, the cut function $\widehat{w} = \{w_1, w_2, \ldots, w_n\}$ and the uncut function $\widetilde{w} = w_1 w_2 \ldots w_n$. For a given cut grammar G and start symbol S, the cut language is defined as $\widehat{L}(G) = \{\widehat{w} \in 2^{\Sigma^*} \mid S \Longrightarrow^* w\}$ and the uncut language $\widetilde{L}(G) = \{\widetilde{w} \in \Sigma^* \mid S \Longrightarrow^* w\}$. A ligation grammar is a cut grammar with an additional new symbol $\sharp$. For any $w = w_1 \sharp w_2 \sharp \ldots \sharp w_n$, where $w_i \in (\Sigma \cup \{\diamondsuit\})^*$ for each $1 \leq i \leq n$, the ligate function is defined as $\breve{w} = \{\widetilde{w}_1, \widetilde{w}_2, \ldots, \widetilde{w}_n\}$. The ligated language $\breve{L}(G) = \{\breve{w} \in 2^{\Sigma^*} \mid S \Longrightarrow^* w\}$.

2. Matrix Insertion-Deletion Systems

In this section, we describe *Matrix insertion-deletion (in short Matrix ins-del) systems*. A Matrix ins-del system is a construct $\Upsilon = (V, T, A, R)$ where V is an alphabet, $T \subseteq V$, A is a finite language over V, R is a finite triples of the form in matrix format $[(u_1, \alpha_1/\beta_1, v_1), \ldots, (u_n, \alpha_n/\beta_n, v_n)]$, where $(u_k, v_k) \in V^* \times V^*$, and $(\beta_k, \alpha_k) \in (V^+ \times \{\lambda\}) \cup (\{\lambda\} \times V^+)$, with $(u_k, \alpha_k/\beta_k, v_k) \in R_{I_i} \cup R_{D_j} \cup R_{I_i/D_j}$, for $1 \leq k \leq n$. Here R_{I_i} denotes the matrix which consists of only insertion rules, R_{D_j} denotes the matrix which consists of only deletion rules and R_{I_i/D_j} denotes the matrix which consists of both insertion and deletion rules.

Consequently, for $x, y \in V^*$ we can write $x \Longrightarrow y$, if y can be obtained from x by using a matrix consisting of insertion or deletion or insertion and deletion rules as follows: In a derivation step the rules in a matrix are applied sequentially one after other in order and no rule is in appearance checking (note that the rules in a matrix are not applied in parallel). The language generated by Υ is defined by $L(\Upsilon) = \{w \in T^* \mid x \Longrightarrow^*_{R_\chi} w,\ \text{for some } x \in A\},\ \text{where } \chi \in \{I_i, D_j, I_i/D_j\}$ where $\Longrightarrow^*$ is the reflexive and transitive closure of the relation $\Longrightarrow$. Note that the string w is collected after applying all the rules in a matrix and also $w \in T^*$ only. Note that the above mentioned cut, uncut

and ligate function are also applicable to Matrix ins-del systems. $\diamondsuit$, # are included in V in Matrix ins-del system.

3. Modelling Intermolecular structures

In this section, we model some of the intermolecular structures using Matrix insertion-deletion systems.

Lemma 1 *The double strand language $L_{ds} = \{u\diamondsuit\bar{u}^R \mid u \in \Sigma^*_{DNA}\}$ can be modelled by Matrix ins-del system.*

Proof. The language L_{ds} can be be modelled by Matrix ins-del system $\Upsilon_{ds} = (\{b, \bar{b}, \diamondsuit\}, \{b, \bar{b}, \diamondsuit\}, \{\diamondsuit\}, R)$ where $b \in \{a, t, g, c\}$, $\bar{b}$ is complement of b and R is given as: $R_{I_1} = [(\lambda, \lambda/b, \diamondsuit), (\diamondsuit, \lambda/\bar{b}, \lambda)]$ A sample derivation is given as follows:

$$^{\downarrow}\diamondsuit^{\downarrow} \Longrightarrow_{R_{I_1}} \underline{a}^{\downarrow}\diamondsuit^{\downarrow}\underline{t} \Longrightarrow_{R_{I_1}} ag\underline{}^{\downarrow}\diamondsuit^{\downarrow}\underline{c}t \Longrightarrow_{R_{I_1}} aga\underline{}^{\downarrow}\diamondsuit^{\downarrow}\underline{t}ct \Longrightarrow_{R_{I_1}} agac\underline{}^{\downarrow}\diamondsuit^{\downarrow}\underline{g}tct$$

□

In the next lemma, we give a cut grammar which generates nick language and we model such cut grammar using Matrix ins-del system. From Fig.3(b) the nick language can be informally described as $\{w_1\diamondsuit w_2 \mid \widetilde{w}_2 = \bar{w_1}^R\}$, where $w_1 \in \Sigma^*$ and $w_2 \in (\Sigma \cup \{\diamondsuit\})^*$ (i.e., w_2 is a string which contains many number of $\diamondsuit$).

Lemma 2 *The nick language L_{nl} can be generated by Matrix ins-del system.*

Proof. The language L_{nl} can be generated by the cut grammar $G_{nl} = S \rightarrow bS\bar{b} \mid S\diamondsuit \mid \diamondsuit$ for each $b \in \Sigma_{DNA}$. The grammar G_{nl} can be modelled by Matrix ins-del system $\Upsilon_{nl} = (\{b, \bar{b}, \dagger, \diamondsuit\}, \{b, \bar{b}, \diamondsuit\}, \{b\dagger\bar{b}, \dagger\diamondsuit, \diamondsuit\}, R)$ where $b \in \{a, t, g, c\}$, $\bar{b}$ is complement of b and R is given as: $R_{I_1} = [(\lambda, \lambda/b, \dagger), (\dagger, \lambda/\bar{b}, \lambda)], R_{I_2} = [(\dagger, \lambda/\diamondsuit, \lambda)], R_{D_1} = [(\diamondsuit, \dagger/\lambda, \lambda)]$. A sample derivation is given as follows:

$$a^{\downarrow}\dagger^{\downarrow}t \Longrightarrow_{R_{I_1}} a\underline{t}^{\downarrow}\dagger^{\downarrow}\underline{a}t \Longrightarrow_{R_{I_1}} at\underline{g}\dagger^{\downarrow}\underline{c}at \Longrightarrow_{R_{I_2}} atg^{\downarrow}\dagger^{\downarrow}\underline{\diamondsuit}cat \Longrightarrow_{R_{I_1}} atg\underline{a}$$
$$\dagger^{\downarrow}\underline{t}\diamondsuit cat \Longrightarrow_{R_{I_2}} atga\dagger\underline{\diamondsuit}t\diamondsuit cat \Longrightarrow_{R_{D_1}} atga^{\Downarrow}\diamondsuit t\diamondsuit cat$$ □

In the next two lemmas we model R-loops and holliday structure as mentioned in Fig. 4 (in Fig 4. • denotes $\diamondsuit$, * denotes λ and S, A, R represents the non-terminals in their corresponding context-free grammars).

Lemma 3 *The hybrid molecule language (with any number of R-loops) $L_{rl} = \{u\diamondsuit v\bar{u}^R \mid u, v \in \Sigma^*_{DNA}\}$ can be generated by Matrix ins-del system.*

Proof. The language L_{rl} can be generated by Matrix ins-del system $\Upsilon_{rl} = (\{b, \bar{b}, \dagger_1, \dagger_2, \diamondsuit\}, \{b, \bar{b}, \diamondsuit\}, \{\dagger_1\diamondsuit\dagger_2\}, R)$ where $b \in \{a, t, g, c\}$, $\bar{b}$ is complement of b and R is given as: $R_{I_1} = [(\lambda, \lambda/b, \dagger_1), (\dagger_2, \lambda/\bar{b}, \lambda)],\ R_{I_2} = [(\lambda, \lambda/b, \dagger_2)],\ R_{D_1} = [(\lambda, \dagger_1/\lambda, \lambda)],\ R_{D_2} = [(\lambda, \dagger_2/\lambda, \lambda)]$. A sample derivation is given as follows:

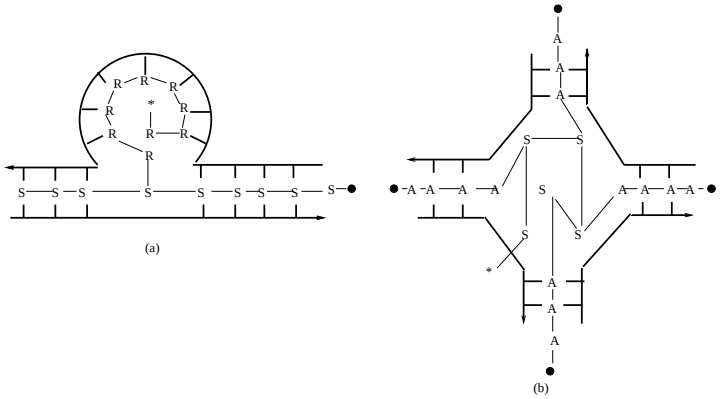

Figure 4. Intermolecular structures: (a) R-loops (b) holliday structure

$$^{\downarrow}\dagger_1 \diamondsuit \dagger_2^{\downarrow} \Longrightarrow_{R_{I_1}} \underline{a}^{\downarrow} \dagger_1 \diamondsuit \dagger_2^{\downarrow} \underline{t} \Longrightarrow_{R_{I_1}} a\underline{t}^{\downarrow} \dagger_1 \diamondsuit \dagger_2^{\downarrow} \underline{a}t \Longrightarrow_{R_{I_1}} atg \dagger_1 \diamondsuit^{\downarrow} \dagger_2 \underline{c}at$$
$$\Longrightarrow_{R_{I_2}} atg \dagger_1 \diamondsuit \underline{a}^{\downarrow} \dagger_2 cat \Longrightarrow_{R_{I_2}} atg \dagger_1 \diamondsuit a\underline{c} \dagger_2 cat \Longrightarrow_{R_{D_1}} atg^{\Downarrow} \diamondsuit ac \dagger_2 cat$$
$$\Longrightarrow_{R_{D_2}} atg \diamondsuit ac^{\Downarrow} cat \qquad \square$$

Lemma 4 *The holliday structure* $L_{hs} = \{u_1 \diamondsuit \bar{u_1}^R u_2 \diamondsuit \bar{u_2}^R \dots u_n \diamondsuit \bar{u_n}^R \mid n \geq 0, u_1, \dots, u_n \in \Sigma_{DNA}^*\}$ *can be generated by Matrix ins-del system.*

Proof. The language L_{hs} (for n=3) can be generated by Matrix ins-del system $\Upsilon_{hs} = (\{b, \bar{b}, \dagger_1, \dagger_2, \dagger_3, \diamondsuit\}, \{b, \bar{b}, \diamondsuit\}, \{\dagger_1 \diamondsuit \dagger_2 \diamondsuit \dagger_3 \diamondsuit\}, R)$ where $b \in \{a, t, g, c\}$, $\bar{b}$ is complement of b and R is given as: $R_{I_1} = [(\lambda, \lambda/b, \dagger_1), (\dagger_1 \diamondsuit, \lambda/\bar{b}, \lambda)],\ R_{I_2} = [(\lambda, \lambda/b, \dagger_2), (\dagger_2 \diamondsuit, \lambda/\bar{b}, \lambda)],\ R_{I_3} = [(\lambda, \lambda/b, \dagger_3), (\dagger_3 \diamondsuit, \lambda/\bar{b}, \lambda)],\ R_{D_1} = [(\lambda, \dagger_1/\lambda, \lambda)]$, $R_{D_2} = [(\lambda, \dagger_2/\lambda, \lambda)],\ R_{D_3} = [(\lambda, \dagger_3/\lambda, \lambda)]$

$$^{\downarrow}\dagger_1 \diamondsuit^{\downarrow} \dagger_2 \diamondsuit \dagger_3 \diamondsuit \Longrightarrow_{R_{I_1}} \underline{a} \dagger_1 \diamondsuit \underline{t}^{\downarrow} \dagger_2 \diamondsuit^{\downarrow} \dagger_3 \diamondsuit \Longrightarrow_{R_{I_2}} a \dagger_1 \diamondsuit t\underline{g} \dagger_2 \diamondsuit \underline{c}^{\downarrow} \dagger_3 \diamondsuit^{\downarrow}$$
$$\Longrightarrow_{R_{I_3}} a \dagger_1 \diamondsuit tg \dagger_2 \diamondsuit c\underline{t} \dagger_3 \diamondsuit \underline{a} \Longrightarrow_{R_{D_1}} a^{\Downarrow} \diamondsuit tg \dagger_2 \diamondsuit ct \dagger_3 \diamondsuit a \Longrightarrow_{R_{D_2}} a \diamondsuit tg^{\Downarrow} \diamondsuit$$
$$ct \dagger_3 \diamondsuit a \Longrightarrow_{R_{D_3}} a \diamondsuit tg \diamondsuit ct^{\Downarrow} \diamondsuit a \qquad \square$$

Lemma 5 *The replication fork language* $\widetilde{L}_{rf} = \{u\bar{u}^R uv\bar{v}^R\bar{u}^R \mid u, v \in \Sigma_{DNA}^*\}$ *can be generated by Matrix ins-del system.*

Proof. The language $\widetilde{L}_{rf}$ can be generated by $\Upsilon_{rf} = (\{\dagger_1, \dagger_2, \dagger_3, \dagger_4, a, t, g, c\}, \{a, t, g, c\}, \{\dagger_1\ \dagger_2\ \dagger_3 \dagger_4\}, R)$, where R is given as follows:
$R_{I_1} = [(\lambda, \lambda/a, \dagger_1), (\dagger_1, \lambda/t, \lambda), (\lambda, \lambda/a, \dagger_2), (\dagger_4, \lambda/t, \lambda)]$,
$R_{I_2} = [(\lambda, \lambda/t, \dagger_1), (\dagger_1, \lambda/a, \lambda), (\lambda, \lambda/t, \dagger_2), (\dagger_4, \lambda/a, \lambda)]$,
$R_{I_3} = [(\lambda, \lambda/g, \dagger_1), (\dagger_1, \lambda/c, \lambda), (\lambda, \lambda/g, \dagger_2), (\dagger_4, \lambda/c, \lambda)]$,
$R_{I_4} = [(\lambda, \lambda/c, \dagger_1), (\dagger_1, \lambda/g, \lambda), (\lambda, \lambda/c, \dagger_2), (\dagger_4, \lambda/g, \lambda)]$,
$R_{I_5} = [(\lambda, \lambda/b, \dagger_3), (\dagger_3, \lambda/\bar{b}, \lambda)]$, $R_{D_1} = [(\lambda, \dagger_1/\lambda, \lambda), (\lambda, \dagger_2/\lambda, \lambda), (\lambda, \dagger_4/\lambda, \lambda)]$,
$R_{D_2} = [(\lambda, \dagger_3/\lambda, \lambda)]$. A sample derivation is given as follows:

$$^{\downarrow}\dagger_1^{\downarrow}\ ^{\downarrow}\dagger_2\ \dagger_3\dagger_4^{\downarrow} \Longrightarrow_{R_{I_1}} \underline{a}^{\downarrow} \dagger_1^{\downarrow} \underline{t}\ \underline{a}^{\downarrow} \dagger_2\ \dagger_3\ \dagger_4^{\downarrow} \underline{t} \Longrightarrow_{R_{I_3}} a\underline{g}^{\downarrow} \dagger_1^{\downarrow} \underline{c}ta\underline{t}^{\downarrow} \dagger_2\ \dagger_3\ \dagger_4^{\downarrow} \underline{c}t$$

$\Longrightarrow_{R_{I_4}} agc\underline{c}\dagger_1 \underline{g}ctat\underline{c}\dagger_2^{\downarrow}\dagger_3^{\downarrow}\dagger_4 \underline{g}ct \Longrightarrow_{R_{I_5}} agc\dagger_1 gctatc\dagger_2 \underline{a}\dagger_3 \underline{t}\dagger_4 gct$

$\Longrightarrow_{R_{D_1}} agc^{\Downarrow}gctatc^{\Downarrow}a\dagger_3 t^{\Downarrow}gct \Longrightarrow_{R_{D_2}} agcgctatca^{\Downarrow}tgct$ □

Lemma 6 *The generalized linear hybridization L_{glh} can be modelled by Matrix ins-del system.*

Proof. The generalized linear hybridization language L_{glh} can be generated by the grammar $G_{glh} = S \to bS\bar{b} \mid \diamondsuit A \mid B\diamondsuit \mid \sharp,\ A \to \sharp Ab \mid bS\bar{b} \mid \sharp,\ \ B \to bB\sharp \mid bS\bar{b} \mid \sharp$ for each $b \in \Sigma_{DNA}$. The grammar G_{glh} can be modelled by Matrix ins-del system $\Upsilon_{glh} = (\{b, \bar{b}, \dagger_1, \dagger_2, \dagger_3, \diamondsuit, \sharp\}, \{b, \bar{b}, \sharp, \diamondsuit\}, \{b\dagger_1\bar{b}, \diamondsuit\dagger_2, \dagger_3\diamondsuit, \sharp\}, R)$, where $b \in \{a, t, g, c\}$, $\bar{b}$ is complement of b and R is given as:
$R_{I_1/D_1} = [(\lambda, \lambda/b\dagger_1\bar{b}, \dagger_1), (\dagger_1\bar{b}, \dagger_1/\lambda, \lambda)], R_{I_2/D_2} = [(\lambda, \lambda/\diamondsuit\dagger_2, \dagger_1), (\diamondsuit\dagger_2, \dagger_1/\lambda, \lambda)]$,
$R_{I_3/D_3} = [(\lambda, \lambda/\dagger_3\diamondsuit, \dagger_1), (\dagger_3\diamondsuit, \dagger_1/\lambda, \lambda)], R_{I_4/D_4} = [(\lambda, \lambda/\sharp, \dagger_1), (\sharp, \dagger_1/\lambda, \lambda)]$,
$R_{I_5} = [(\lambda, \lambda/\sharp, \dagger_2), (\dagger_2, \lambda/b, \lambda)], R_{I_6/D_5} = [(\lambda, \lambda/b\dagger_1\bar{b}, \dagger_2), (\dagger_1\bar{b}, \dagger_2/\lambda, \lambda)]$,
$R_{I_7/D_6} = [(\lambda, \lambda/\sharp, \dagger_2), (\sharp, \dagger_2/\lambda, \lambda)], R_{I_8} = [(\lambda, \lambda/b, \dagger_3), (\dagger_3, \lambda/\sharp, \lambda)]$,
$R_{I_9/D_7} = [(\lambda, \lambda/b\dagger_1\bar{b}, \dagger_3), (\dagger_1\bar{b}, \dagger_3/\lambda, \lambda)], R_{I_{10}/D_8} = [(\lambda, \lambda/\sharp, \dagger_3), (\sharp, \dagger_3/\lambda, \lambda)]$
A sample derivation is given as follows:

$\diamondsuit^{\downarrow}\dagger_1 \Longrightarrow_{R_{I_6/D_5}} \diamondsuit\underline{a^{\downarrow}\dagger_1 t^{\Downarrow}} \Longrightarrow_{R_{I_1/D_1}} \diamondsuit a\underline{g^{\downarrow}\dagger_1 c^{\Downarrow}}t \Longrightarrow_{R_{I_2/D_2}} \diamondsuit ag\underline{\diamondsuit^{\downarrow}\dagger_2}^{\Downarrow}ct$

$\Longrightarrow_{R_{I_5}} \diamondsuit ag\diamondsuit\underline{\sharp}\dagger_2 \underline{c}ct \Longrightarrow_{R_{I_7/D_6}} \diamondsuit ag\diamondsuit\sharp\underline{\sharp}^{\Downarrow}\underline{c}ct$ □

4. New Levels of Ambiguity

In this section, we formally define various ambiguity levels for Matrix ins-del system based on the components used in the derivation such as *axiom*, *contexts*, *strings* (for deletion/insertion).

Consider the following derivation step in a Matrix ins-del system Υ, $\delta : w_1 \Longrightarrow w_2 \Longrightarrow \ldots... \Longrightarrow w_m, m \geq 1$, such that $w_1 \in A$ and the following scenarios can happen (1): $w_k \Longrightarrow w_{k+1}$ can be obtained by using a matrix which consists of only insertion rules (R_{I_i}) (2): $w_k \Longrightarrow w_{k+1}$ can be obtained by using a matrix which consists of only deletion rules (R_{D_j}) (3): $w_k \Longrightarrow w_{k+1}$, such that $1 \leq k \leq m$ can be obtained by using a matrix which consists of both insertion and deletion rules (R_{I_i/D_j}). The sequence which consists of used axiom, strings α_j/β_j to be deleted/inserted is called as *Control Sequence* which is given as follows: $w_1, [(\alpha_1/\beta_1), \ldots, (\alpha_n/\beta_n)], [(\alpha_1/\beta_1), \ldots, (\alpha_n/\beta_n)], [(\alpha_1/\beta_1), \ldots, (\alpha_n/\beta_n)], \ldots,$ $[(\alpha_{m-1}/\beta_{m-1}), \ldots, (\alpha_n/\beta_n)]$. More precisely the control sequence means the order in which the strings are deleted/inserted. The sequence which consists of used axiom, the strings β_j/α_j to be deleted/inserted and the used contexts (u_j, v_j) is called *Complete Control Sequence* which is given as follows: $w_1, [(u_1, \alpha_1/\beta_1, v_1), \ldots, (u_n, \alpha_n/\beta_n, v_n)],$ $[(u_1, \alpha_1/\beta_1, v_1), \ldots, (u_n, \alpha_n/\beta_n, v_n)], \ldots, [(u_{m-1}, \alpha_{m-1}/\beta_{m-1}, v_{m-1}), \ldots, (u_n, \alpha_n/\beta_n, v_n)]$. More precisely the complete control sequence means the order of the contexts used in deletion/insertion rules. Note that one of β_j or α_j is empty for all j in the derivation. The position where insertion β /deletion α takes place can be given by the *description* of δ.

Definition 1:

1. A Matrix ins-del system Υ, is said to be *0-ambiguous*, if there exist at least two different axioms, $w_1, w_2 \in A,\ w_1 \neq w_2$, such that they both derive the same word z, i.e., $w_1 \Longrightarrow^+ z,\ w_2 \Longrightarrow^+ z$.
2. A Matrix ins-del system Υ, is said to be *1-ambiguous*, if there are two different ordered control sequences which derive the same word.
3. A Matrix ins-del system Υ, is said to be *2-ambiguous*, if there are two different ordered complete control sequences which derive the same word.
4. A Matrix ins-del system Υ, is said to be *3-ambiguous*, if there are two different descriptions which derive the same word.

5. Interpretation of Ambiguity in Gene Sequences

In this section, our aim is not to show that there exists inherently ambiguous languages, rather our objective is to present a new interpretation of ambiguity in gene sequences. As an application to the new introduced ambiguity levels, we present some intra and intermolecular structures in gene sequences which exhibits the above introduced new levels of ambiguity for Matrix ins-del systems.

Level 0: The Matrix ins-del system is said to be 0-ambiguous if the same string can be derived from two different axioms.

Definition 2: A string w over a complementary alphabet Σ is called ideal iff $|w|_b = |w|_{\bar{b}}$ for all $b \in \Sigma$. A language is ideal iff it contains only ideal strings.

Lemma 7 *The ideal language L_{id} generated by Matrix ins-del system exhibits level 0 ambiguity.*

Proof. The ideal language L_{id} can be generated by the Matrix ins-del system $\Upsilon_{id} = (\{b, \bar{b}\}, \{b, \bar{b}\}, \{\lambda\}, R)$, where $b \in \{a, t, g, c\}$, $\bar{b}$ is complement of b and R is given as $R_{I_1} = [(\lambda, \lambda/b, \lambda), (\lambda, \lambda/\bar{b}, \lambda)]$. Consider the gene sequence $tactgagcta$ in ideal language. This sequence can be generated by the Matrix ins-del system Υ_{id} from two different axioms at and gc such that the same string is obtained at the end of the derivation. The two different derivations which differ by axioms are given as follows:

$$Derivation\ 1 : at^{\downarrow} \Longrightarrow a^{\downarrow}t^{\downarrow}\underline{gc} \Longrightarrow a\underline{c}\underline{t}\underline{g}^{\downarrow}gc^{\downarrow} \Longrightarrow^{\downarrow} actg\underline{a}g\underline{c}\underline{t}^{\downarrow} \Longrightarrow \underline{t}actgagct\underline{a}$$

$$Derivation\ 2 : gc^{\downarrow} \Longrightarrow^{\downarrow} gct\underline{a} \Longrightarrow \underline{ta}^{\downarrow}g^{\downarrow}cta \Longrightarrow ta^{\downarrow}\underline{tg}\underline{a}^{\downarrow}cta \Longrightarrow ta\underline{c}tg\underline{a}\underline{g}cta \square$$

Level 1: The Matrix ins-del system is said to be 1-ambiguous if there are two different derivations for the same string which differs by the order of string deleted/inserted.

In the next lemma, we show that level 1 ambiguity exists in both intramolecular structure (stem and loop structure) and intermolecular structure (hybrid molecule structure).

Lemma 8 *The stem and loop structure $L_{sl} = \{uv\bar{u}^R \mid u, v \in \Sigma^*_{DNA}\}$ represented by Matrix ins-del system obeys level 1 ambiguity.*

Proof. The stem and loop structure L_{sl} can be generated by the Matrix ins-del system $\Upsilon_{sl} = (\{b, \bar{b}, \dagger_1, \dagger_2, \dagger_3\}, \{b, \bar{b}\}, \{\lambda, b\ \dagger_1\ \dagger_3\ \dagger_2\ \bar{b}\}, R)$, where $b \in \{a, t, g, c\}$, $\bar{b}$ is complement of b and R is given as: $R_{I_1} = [(\lambda, \lambda/b, \dagger_1), (\dagger_2, \lambda/\bar{b}, \lambda)]$, $R_{I_2} = [(\lambda, \lambda/b, \dagger_3)], R_{D_1} = [(\lambda, \dagger_1/\lambda, \lambda), (\lambda, \dagger_2/\lambda, \lambda)]$, $R_{D_2} = [(\lambda, \dagger_3/\lambda, \lambda)]$.
Consider the gene sequence $aagtt$ in stem and loop structure. This sequence can be generated in two different ordered control sequences by the Matrix ins-del system Υ_{sl}. Note that the axiom for both sequence is the same. The two sequences are given as follows:

$$Sequence\ 1 : a^{\downarrow}\ \dagger_1\ \dagger_3\ \dagger_2^{\downarrow}\ t \Longrightarrow_{R_{I_1}} a\underline{a}\ \dagger_1^{\downarrow}\ \dagger_3\ \dagger_2\ \underline{t}t \Longrightarrow_{R_{I_2}} aa\ \dagger_1\ \underline{g}\ \dagger_3\ \dagger_2 tt \Longrightarrow_{R_{D_1}}$$

$$aa^{\Downarrow} g\ \dagger_3^{\Downarrow}\ tt \Longrightarrow_{R_{D_2}} aag^{\Downarrow} tt$$

$$Sequence\ 2 : a\ \dagger_1^{\downarrow}\ \dagger_3\ \dagger_2\ t \Longrightarrow_{R_{I_2}} a^{\downarrow}\ \dagger_1\ \underline{g}\ \dagger_3\ \dagger_2^{\downarrow} t \Longrightarrow_{R_{I_1}} a\underline{a}\ \dagger_1\ g\ \dagger_3\ \dagger_2\underline{t}t \Longrightarrow_{R_{D_2}}$$

$$aa\ \dagger_1\ g^{\Downarrow}\ \dagger_2\ tt \Longrightarrow_{R_{D_1}} aa^{\Downarrow} g^{\Downarrow} tt \qquad \square$$

In sequence 1, the order of strings used for deletion/insertion is $(a, t), g, (\dagger_1, \dagger_2), \dagger_3$. In sequence 2, the order of strings used for deletion/insertion is $g, (a, t), \dagger_3, (\dagger_1, \dagger_2)$. Thus, we obtain two different ordered control sequences which derive the same gene sequence. Therefore, the Matrix ins-del system Υ_{sl} obeys level 1 ambiguity. The Level 1 ambiguity can be pictorially represented as shown in Fig.5. Fig. 5(a) corresponds to sequence 1 and Fig. 5(b) corresponds to sequence 2.

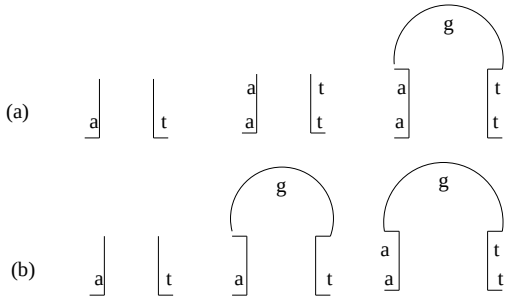

Figure 5. Ambiguity in stem and loop structure

Level 1 ambiguity in Intermolecular structure: Consider the following string $gt\diamondsuit gcac$ in the hybrid molecule structure. This string can be generated in two different ordered control sequences by the Matrix ins-del system Υ_{rl}. The two sequences are given below:

$$Sequence\ 1 :^{\downarrow}\ \dagger_1\diamondsuit\dagger_2^{\downarrow} \Longrightarrow_{R_{I_1}} \underline{g}^{\downarrow}\ \dagger_1\ \diamondsuit\ \dagger_2^{\downarrow}\ \underline{c} \Longrightarrow_{R_{I_1}} g\underline{t}\ \dagger_1\ \diamondsuit^{\downarrow}\ \dagger_2\ \underline{a}c \Longrightarrow_{R_{I_2}} gt\ \dagger_1$$

$$\diamondsuit\underline{g}^{\downarrow}\ \dagger_2\ ac \Longrightarrow_{R_{I_2}} gt\ \dagger_1\ \diamondsuit g\underline{c}\ \dagger_2\ ac \Longrightarrow_{R_{D_1}} gt^{\Downarrow}\diamondsuit gc\ \dagger_2\ ac \Longrightarrow_{R_{D_2}} gt\diamondsuit gc^{\Downarrow} ac$$

$$Sequence\ 2 : \dagger_1\diamondsuit^{\downarrow}\dagger_2 \Longrightarrow_{R_{I_2}} \dagger_1\diamondsuit\underline{g}^{\downarrow}\dagger_2 \Longrightarrow_{R_{I_2}}\ ^{\downarrow}\dagger_1\ \diamondsuit g\underline{c}\dagger_2^{\downarrow} \Longrightarrow_{R_{I_1}} \underline{g}^{\downarrow}\ \dagger_1\ \diamondsuit gc$$

$$\dagger_2^{\downarrow}\underline{c} \Longrightarrow_{R_{I_1}} g\underline{t}\ \dagger_1\ \diamondsuit gc\ \dagger_2\ \underline{a}c \Longrightarrow_{R_{D_2}} gt\ \dagger_1\ \diamondsuit gc^{\Downarrow} ac \Longrightarrow_{R_{D_1}} gt^{\Downarrow}\diamondsuit gcac$$

In sequence 1, the order of strings used for deletion/insertion is $(g, c), (t, a), g, c, \dagger_1, \dagger_2$. In sequence 2, the order of strings used for deletion/insertion is $g, c, (g, c), (t, a), \dagger_2, \dagger_1$. Thus, we obtain two different ordered control sequences which derive the same gene

sequence $gt\diamondsuit gcac$. Therefore, the Matrix ins-del system Υ_{rl} exhibits level 1 ambiguity.

Level 2: The Matrix ins-del system is said to be 2-ambiguous if there are two different derivations for the same string which differs by the order of contexts used for deletion/insertion.

Lemma 9 *The holliday structure* $L_{hs} = \{u_1\diamondsuit\bar{u_1}^R u_2\diamondsuit\bar{u_2}^R \dots u_n\diamondsuit\bar{u_n}^R \mid n \geq 0, u_1, \dots, u_n \in \Sigma^*_{DNA}\}$ *represented by Matrix ins-del system obeys level 2 ambiguity.*

Proof. Consider the following string $t\diamondsuit ac\diamondsuit gg\diamondsuit c$ in holliday structure. This string can be generated in two different ordered complete control sequences by the Matrix ins-del system Υ_{hs}. The two sequences are given below:

$$Sequence\ 1 :^{\downarrow} \dagger_1\diamondsuit^{\downarrow}\dagger_2\diamondsuit\dagger_3\diamondsuit \Longrightarrow_{R_{I_1}} \underline{t}\dagger_1\diamondsuit\underline{a}^{\downarrow}\dagger_2\diamondsuit^{\downarrow}\dagger_3\diamondsuit \Longrightarrow_{R_{I_2}} t\dagger_1\diamondsuit a\underline{c}\dagger_2$$
$$\diamondsuit\underline{g}^{\downarrow}\dagger_3\diamondsuit^{\downarrow} \Longrightarrow_{R_{I_3}} t\dagger_1\diamondsuit ac\dagger_2\diamondsuit g\underline{g}\dagger_3\diamondsuit\underline{c} \Longrightarrow_{R_{D_1}} t^{\Downarrow}\diamondsuit ac\dagger_2\diamondsuit gg\dagger_3\diamondsuit c \Longrightarrow_{R_{D_2}}$$
$$t\diamondsuit ac^{\Downarrow}\diamondsuit gg\dagger_3\diamondsuit c \Longrightarrow_{R_{D_3}} t\diamondsuit ac\diamondsuit gg^{\Downarrow}\diamondsuit c$$
$$Sequence\ 2 : \dagger_1\diamondsuit^{\downarrow}\dagger_2\diamondsuit^{\downarrow}\dagger_3\diamondsuit \Longrightarrow_{R_{I_2}} \dagger_1\diamondsuit\underline{c}\dagger_2\diamondsuit\underline{g}\dagger_3\diamondsuit \Longrightarrow_{R_{D_2}} {}^{\downarrow}\dagger_1\diamondsuit^{\downarrow}c^{\Downarrow}\diamondsuit g$$
$$\dagger_3\diamondsuit \Longrightarrow_{R_{I_1}} \underline{t}\dagger_1\diamondsuit\underline{a}c\diamondsuit g\dagger_3\diamondsuit \Longrightarrow_{R_{D_1}} t^{\Downarrow}\diamondsuit ac\diamondsuit g^{\downarrow}\dagger_3\diamondsuit^{\downarrow} \Longrightarrow_{R_{I_3}} t\diamondsuit ac\diamondsuit g\underline{g}\dagger_3\diamondsuit\underline{c}$$
$$\Longrightarrow_{R_{D_3}} t\diamondsuit ac\diamondsuit gg^{\Downarrow}\diamondsuit c \qquad \square$$

In sequence 1, the order of contexts used is $[(\lambda, \dagger_1), (\dagger_1\diamondsuit, \lambda)], [(\lambda, \dagger_2), (\dagger_2\diamondsuit, \lambda)], [(\lambda, \dagger_3), (\dagger_3\diamondsuit, \lambda)], [(\lambda, \lambda)], [(\lambda, \lambda)], [(\lambda, \lambda)]$. In sequence 2, the order of contexts used is $[(\lambda, \dagger_2), (\dagger_2\diamondsuit, \lambda)], [(\lambda, \lambda)], [(\lambda, \dagger_1), (\dagger_1\diamondsuit, \lambda)], [(\lambda, \lambda)], [(\lambda, \dagger_3), (\dagger_3\diamondsuit, \lambda)], [(\lambda, \lambda)]$. Thus, we obtain two different ordered complete control sequences which derive the same gene sequence $t\diamondsuit ac\diamondsuit gg\diamondsuit c$. Therefore, the Matrix ins-del system Υ_{hs} obeys level 2 ambiguity.

Level 3: The Matrix ins-del systems is said to be 3-ambiguous if there are two different descriptions for the same string which differs by the position where the string is deleted/inserted.

Definition 3: A string w over a complementary alphabet Σ is called orthodox iff it is (i) the empty string λ, or (2) the result of inserting two adjacent complementary element $b\bar{b}$, for some $b \in \Sigma$, anywhere in an orthodox string. A language is orthodox iff it contains only orthodox strings.

Lemma 10 *The orthodox language* L_{od} *generated by Matrix ins-del system obeys level 3 ambiguity.*

Proof. The orthodox language L_{od} can be generated by the Matrix ins-del system $\Upsilon_{od} = (\{b, \bar{b}\}, \{b, \bar{b}\}, \{\lambda\}, R)$, where $b \in \{a, t, g, c\}$, $\bar{b}$ is complement of b and R is given as $R_{I_1} = [(\lambda, \lambda/b\bar{b}, \lambda)]$. Consider the string $tagcgcat$ in orthodox language. This string can be derived in two different descriptions by Υ_{od} which are given as follows:

$$Description\ 1 :^{\downarrow} gc \Longrightarrow_{R_{I_1}} \underline{ta}gc^{\downarrow} \Longrightarrow_{R_{I_1}} tagc\underline{gc}^{\downarrow} \Longrightarrow_{R_{I_1}} tagcgc\underline{at}$$
$$Description\ 2 : gc^{\downarrow} \Longrightarrow_{R_{I_1}} gc\underline{gc}^{\downarrow} \Longrightarrow_{R_{I_1}} {}^{\downarrow}gcgc\underline{at} \Longrightarrow_{R_{I_1}} \underline{ta}gcgcat \qquad \square$$

Note that the axiom, order of insertion of strings, order of contexts (here (λ, λ)) all are same in both derivations, but the position of insertion is different in each derivation. Therefore, the system Υ_{od} is 3-ambiguous.
Note: Consider the orthodox string $gcgcgcgc$. This string can be derived from axiom λ by two different descriptions which are given as follows:

$$Description\ 1 : \lambda \Longrightarrow gc \Longrightarrow \underline{gc}gc \Longrightarrow gc\underline{gc}gc \Longrightarrow gcgc\underline{gc}gc$$

$$Description\ 2 : \lambda \Longrightarrow gc \Longrightarrow gc\underline{gc} \Longrightarrow gcgc\underline{gc} \Longrightarrow gcgcgcgc$$

In one derivation the string gc is used one time and cg is used three times. In another derivation the string gc is used four times. The above string obeys level 1 ambiguity but the intermediate gene sequences are same.

6. Conclusion

In this paper, we discussed the Matrix insertion-deletion grammar systems and using the system we have modelled some intermolecular structures like double strand languages, nick languages, hybrid molecules (with R-loops), holliday structure, replication fork, linear hybridization (ligated) languages. We have formally defined various levels of i $(i = 0, 1, 2, 3)$-ambiguity for Matrix insertion-deletion systems. We have given the application for the introduced levels of ambiguity with an interpretation in gene sequences. We witnessed that the many gene sequences in intra and intermolecular structures like ideal, stem and loop, hybrid molecule, holliday structure, orthodox has a relevance with the introduced levels of ambiguity. Since, in this paper we have shown that ambiguity is possible in gene sequences this research will throw some ideas on how gene sequences in DNA, RNA, protein molecules can be predicted and processed.

References

[1] David B. Searls, *The linguistics of DNA*. American Scientist, (1992) 579–591.
[2] David B. Searls, *The comp. linguistics of biological sequences*. In: Hunter, L.(ed.) Artificial Intelligence and Molecular Biology, AAAI Press, (1993), 47–120.
[3] David B. Searls, *Formal grammars for intermolecular structures*, First Intl. IEEE Symp. on Intelligence and Bio. Systems, (1995), 30–37.
[4] Elena Rivas and Sean R. Reddy, *The language of RNA: A formal grammar that includes pseudoknots*, Bioinformatics, **16** (2000) 334–340.
[5] Gheorghe Păun, Grzegorz Rozenberg and Arto Salomaa, *DNA Computing, New Computing Paradigms*. Springer, 1998.
[6] Lakshmanan Kuppusamy, Anand Mahendran and Shankara Narayanan Krishna, *Matrix insertion-deletion systems for bio-molecular structures*, In: Raja Natarajan and Ojo. (eds.), LNCS proc. of ICDCIT-2011, 301–311.
[7] Lakshmanan Kuppusamy, Anand Mahendran and Shankara Narayanan Krishna, *On representing natural languages and bio-molecular structures using Matrix insertion-deletion systems and its computational completeness*, in Proceedings of BILC-2011, 47–56.
[8] Ion Petre and Sergey Verlan, *Matrix insertion-deletion systems*, arXiv:1012.5248v1[cs.fl], Submitted on 23 Dece, 2010.
[9] Rozenberg and Arto Salomaa, *Handbook of formal languages*, Vol 1, Vol 2, Vol 3, Springer, 1997.
[10] Setubal and Meidanis.: Introduction to computational molecular biology. PWS Pub. Co. 1997.
[11] Yasuo Uemura, Aki Hasegawa, Satoshi Kobayashi and Takashi Yokomori, *TAG. for RNA structure prediction*. Theoretical Computer Science, **210** (1999) 277–303.