Identifying Ambiguity Levels in Gene Sequences using Matrix Ins-Del Systems

Lakshmanan K, Anand Mahendran

Abstract*—Ambiguity is one of the important issues not only in natural and programming languages, but also in gene sequences. In programming languages, the ambiguity is defined as existence of (at least) two distinct derivations that yield a same word. Considering in that line, ambiguity in gene sequences may be interpreted as a gene sequence can be obtained by more than one way such that its intermediate gene sequences are different. Analyzing the ambiguity issues in gene sequences will help us to know the evolution of gene sequences. Recently, in [9] a new variant called Matrix insertion-deletion systems has been introduced as a biologically inspired computing model to represent various bio-molecular structures such as pseudoknot, hairpin, stem and loop, attenuator, dumbbell and cloverleaf. But the ambiguity issues of Matrix insertion-deletion systems has not been analyzed in detail yet. In this paper, we formally define various levels (0,1,2,3) of ambiguity for Matrix insertion-deletion systems based on the components used in the derivation such as axiom, context, string (used for insertion/deletion). Next, we relate the newly defined ambiguity levels of Matrix insertion-deletion systems with bio-molecular structures and analyze their ambiguity issues. We notice that ideal language obeys the level 0-ambiguity, stem and loop structure obeys level 1-ambiguity, cloverleaf structure obeys level 2-ambiguity and orthodox language obeys level 3-ambiguity.

***Index Terms* — bio-molecular structures, pseudoknot, stem and loop, Matrix insertion-deletion systems, ambiguity, gene sequence**

I. INTRODUCTION

In the last three decades, biology played a great role in the field of formal languages by being the root for the development of various biologically inspired computing models such as *sticker systems, splicing systems, Watson-Crick automata, insertion-deletion systems and p systems* [1], [6], [7]. Since, insertion-deletion systems are not exactly based on rewriting systems, the insertion-deletion systems opened a particular attention in the field of formal languages. Informally, the insertion and deletion operations of an insertion-deletion system is defined as follows: If a string α is inserted between two parts w_1 and w_2 of a string $w_1 w_2$ to get $w_1 \alpha w_2$, we call the operation as insertion, whereas if a substring $\beta$is deleted from a string $w_1 \beta w_2$ to get $w_1 w_2$, we call the operation as deletion. DNA molecules may be considered as strings over the alphabet consisting of four symbols namely *a*, *t* , *g* and *c*. Similarly, RNA molecules may be considered as strings over alphabet consisting of four symbols namely *a*, *u*, *g* and *c*. Since the bio-molecular structures can be defined in terms of sequence of symbols (i.e., strings) there exists a correlation between formal grammars and bio-molecular structures. The following example witnesses this correlation.

Manuscript Received on September 2014.

Dr. Lakshmanan K, School of Computing Science and Engineering, VIT University, Vellore, India.

Dr. Anand Mahendran, School of Computing Science and Engineering, VIT University, Vellore, India.

Consider a context-free language $\{ww^R | w \in \{a, b\}^*\}$ where w^R is the reversal of *w*. Consider the following gene sequence *ctatcgcgatag*. As $\bar{a} = t$, $\bar{t} = a$,$\bar{g} = c$ and $\bar{c} = g$, the above gene sequence resembles a word in the palindrome language $\{w\bar{w}^R \mid w \in \{a, t, g, c\}^*\}$. Researchers shown that there exists a relevance between the gene sequences and natural language constructs such as *triple agreements* : $\{a^n b^n c^n \mid n \geq 1\}$, *crossed dependencies:* $\{a^n b^m c^n d^m \mid n, m \geq 1\}$ and *copy language:* $\{ww \mid w \in \{a,b\}^*\}$ [3], [4]. We discuss below in brief some of the important structures seen in bio-molecules such as protein, DNA and RNA. Fig.1 shows the 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 are predominantly available in bio-molecules which cannot be modelled by context-free grammars. Fig.2 represents such structures (a) *pseudoknot* and (b) *attenuator*.

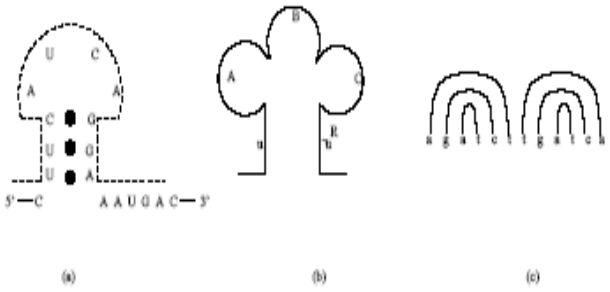

Fig. 1 Bio-Molecular Structures: (a) Stem and Loop (b) Cloverleaf (c) Dumbbell

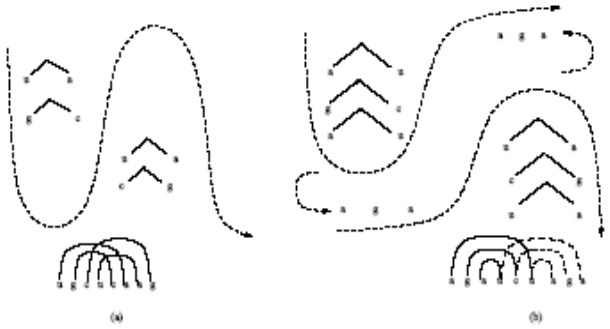

Fig. 2 Bio-Molecular Structures: (a) Pseudoknot Structure (b) Attenuator Structure

In the last two decades or so, many attempts have been made to establish the linguistic behaviour of biological sequences by defining new grammar formalisms like *cut grammars* [2], *crossed-interaction grammar* [5],

simple linear tree adjoining grammars and *extended simple linear tree adjoining grammars* [12] which are capable of generating some of the biological structures mentioned above. However, there is no unique grammar system that encapsulate all essential and important bio-molecular structures. For example double copy language cannot be modelled by a simple linear tree adjoining grammar [12]. In order to find a unique system that encapsulates the above mentioned bio-molecular structures very recently, a new biologically inspired computing model namely *Matrix insertion-deletion* system has been introduced in [9] by combining insertion-deletion system and matrix grammars. This system represents all the above discussed bio- molecular structures and also the other structures such as *non-ideal attenuator, ideal strings, orthodox strings*. In this paper, we briefly recall this system and model a few structures using the system. Ambiguity is considered as one of the fundamental problems in formal language theory. A grammar is said to be ambiguous, if there exists more than one distinct derivation of the words in the generated language. For example, consider the following ambiguous context free grammar $G = (\{E\}, \{id, +, *, (,)\}, E, \{E \to E + E, E \to E * E, E \to (E), E \to id\})$ which generates the set of simple arithmetic expressions over + and *. Consider a word $w = id + id * id$ in the arithmetic expression. The word w can be derived in two distinct (leftmost) derivations (1): $E \Rightarrow E * E \Rightarrow E + E * E \Rightarrow id + E * E \Rightarrow id + id * E \Rightarrow id + id * id$ (2): $E \Rightarrow E + E \Rightarrow id + E \Rightarrow id + E * E \Rightarrow id + id * E \Rightarrow id + id * id$. Since the insertion-deletion system can be applied theoretically to DNA processing [6], the ambiguity in DNA processing (which uses the insertion-deletion system) can 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 bold string denotes the inserted string in the derivation steps).

(1):$W_1W_2 \Rightarrow W_1\mathbf{W_3}W_2 \Rightarrow W_1W_3\mathbf{W_4}W_2 \Rightarrow W_1W_3W_4\mathbf{W_5}W_2$

(2):$W_1W_2 \Rightarrow W_1\mathbf{W_5}W_2 \Rightarrow W_1\mathbf{W_4}W_5W_2 \Rightarrow W_1\mathbf{W_3}W_4W_5W_2$

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). Study of this concept of ambiguity may be useful in considering inheritance properties and phylogenetic trees [11]. More specifically, when these intermediate sequences are represented as phylogenetic trees, we can see that the trees are different and thus it might help us to identify the inheritance properties. Since Matrix insertion-deletion systems is an extension of insertion-deletion systems this motivates us to define formally the various levels of ambiguity for Matrix insertion- deletion systems and how they can be interpreted in gene sequences. This paper is organized as follows. In Section 2, we give the preliminaries. In Section 3, we discuss the Matrix insertion-deletion systems and model some bio-molecular structures. In Section 4, we introduce the various levels of ambiguity for Matrix insertion-deletion systems and finally in Section 5, as an application we show that how the introduced levels of ambiguity can be interpreted in gene sequences.

II. PRELIMINARIES

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 more details on formal language theory, we refer to [8], [10]. 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 $(u, \beta/\alpha, v)$, where $(u, v) \in V^* \times V^*$, $(\alpha, \beta) \in (V^+ \times \{\lambda\}) \cup (\{\lambda\} \times V^+)$. The pair (u, v) is called contexts. Insertion rule will be of the form $(u, \lambda/\alpha, v)$ which means that α is inserted between u and v. Deletion rule will be of the form $(u, \beta/\lambda, v)$, which means that β is deleted between u and v. In other words, $(u, \lambda/\alpha, v)$ corresponds to the rewriting rule $uv \to u\alpha v$, and $(u, \beta/\lambda, v)$ corresponds to the rewriting rule $u\beta v \to uv$. Consequently, for $x, y \in V^*$ we can write $x \Rightarrow^* y$, 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).

1. $x = x_1u\downarrow vx_2$, $y = x_1u\alpha vx_2$, for some $x_1, x_2 \in V^*$ and $(u, \lambda/\alpha, v) \in R$.

2. $x = x_1u\beta vx_2$, $y = x_1u\Downarrow vx_2$, for some $x_1, x_2 \in V^*$ and $(u, \beta/\lambda, v) \in R$.

The language generated by γ is defined by

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

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

The language of DNA can be considered over $\Sigma_{DNA} = \{a, t, g, c\}$, where the complementary can be given as: $\bar{a} = t$, $\bar{t} = a$, $\bar{g} = c$ and $\bar{c} = g$. Similarly, the language of RNA can be considered over $\Sigma_{RNA} = \{a, u, g, c\}$, where the complementary can be given as: $\bar{a} = u$, $\bar{u} = a$, $\bar{g} = c$ and $\bar{c} = g$.

III. 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 $[(u_1, \beta_1/\alpha_1, v_1), \ldots, (u_n, \beta_n/\alpha_n, v_n)]$, where $(u_k, v_k) \in V^* \times V^*$, and $(\alpha_k, \beta_k) \in (V^+ \times \{\lambda\}) \cup (\{\lambda\} \times V^+)$, with $(u_k, \beta_k/\alpha_k, v_k) \in R_{Ii} \cup R_{Dj} \cup R_{Ii/Dj}$, for $1 \le k \le n$.

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 $x \Rightarrow y$, 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 $L(\Upsilon)=\{w \in T^* | x \Rightarrow_{\prod}^* w$, for some $x \in A\}$, where $\prod \in \{I_i, D_j, I_i/D_j\}$ and $\Rightarrow^*$ is the reflexive and transitive closure of the relation $\Rightarrow$. Note that the string w is collected after applying all the rules in a matrix and also w is a terminal string (*i.e.*, $w \in T^*$ only).

3.1 Modelling Bio-Molecular Structures

In this subsection, we give a few examples to represent how some bio-molecular structures can be modelled using this Matrix ins-del systems. These examples also provide us a better understanding of how the system works. We refer to [9] for modeling several other bio-molecular structures.

Example 1:

The pseudoknot structure language $L_{ps} = \{uvu^{\bar{R}}v^{\bar{R}} \mid u, v \in \Sigma^*_{DNA}\}$ can be generated by Matrix ins-del system.

Proof. The language L_{ps} can be generated by the Matrix ins-del system $\Upsilon_{ps} = (\{b, \bar{b}, \dagger_1, \dagger_2, \dagger_3, \dagger_4\}, \{b, \bar{b}\}, \{\lambda, \dagger_1\dagger_2\dagger_3\dagger_4\}, R)$, where $b \in \{a, t, g, c\}$, $\bar{b}$ is complement of b and R is given as follows:

$R_{I_1} = [(\lambda, \lambda/b, \dagger_1), (\lambda, \lambda/\bar{b}, \dagger_3)]$,
$R_{I_2} = [(\lambda, \lambda/b, \dagger_2), (\lambda, \lambda/b, \dagger_4)]$,
$R_{D_1} = [(\lambda, \dagger_1/\lambda, \lambda), (\lambda, \dagger_3/\lambda, \lambda)]$,
$R_{D_2} = [(\lambda, \dagger_2/\lambda, \lambda), (\lambda, \dagger_4/\lambda, \lambda)]$.

A sample derivation is given as follows:

$\dagger_1\dagger_2\dagger_3\dagger_4 \Rightarrow_{R_{I_1}} \mathbf{a}\dagger_1\dagger_2\mathbf{t}\dagger_3\dagger_4 \Rightarrow_{R_{I2}} a\dagger_1\mathbf{g}\dagger_2t\dagger_3\mathbf{c}\dagger_4$
$\Rightarrow_{R_I2} a\dagger_1g\mathbf{a}\dagger_2t\dagger_3c\mathbf{t}\dagger_4 \Rightarrow_{RD_1} aga\dagger_2tct\dagger_4 \Rightarrow_{RD_2} agatct$

Example 2:

The attenuator language $L_{an} = \{uu^{\bar{R}}uu^{\bar{R}} \mid u \in \Sigma_{DNA}^*\}$ can be generated by Matrix ins-del system.

Proof. The language L_{an} can be generated by the Matrix ins-del system $\Upsilon_{an} = (\{a, t, g, c, \dagger_1, \dagger_2\}, \{a, t, g, c\}, \{\lambda, \dagger_1\dagger_2\}, R)$, where R is given as follows:

$R_{I_1}=[(\lambda,\lambda/a,\dagger_1),(\dagger_1,\lambda/t,\lambda),(\lambda,\lambda/a,\dagger_2),(\dagger_2,\lambda/t,\lambda)]$,
$R_{I_2}=[(\lambda,\lambda/t,\dagger_1),(\dagger_1,\lambda/a,\lambda),(\lambda,\lambda/t,\dagger_2),(\dagger_2,\lambda/a,\lambda)]$,
$R_{I_3}=[(\lambda,\lambda/c,\dagger_1),(\dagger_1,\lambda/g,\lambda),(\lambda,\lambda/c,\dagger_2),(\dagger_2,\lambda/g,\lambda)]$
$R_{I_4}=[(\lambda,\lambda/g,\dagger_1),(\dagger_1,\lambda/c,\lambda),(\lambda,\lambda/g,\dagger_2),(\dagger_2,\lambda/c,\lambda)]$,
$R_{D_1}=[(\lambda,\dagger_1/\lambda,\lambda),(\lambda,\dagger_2/\lambda,\lambda)]$.

A sample derivation is given as follows:

$\dagger_1\dagger_2 \Rightarrow_{R_{I_1}} \mathbf{a}\dagger_1\mathbf{ta}\dagger_2\mathbf{t} \Rightarrow_{R_{I_2}} a\mathbf{t}\dagger_1atat\dagger_2at \Rightarrow_{R_{I_3}} atc\dagger_1gatatc\dagger_2gat \Rightarrow_{R_{I_4}} atcg\dagger_1cgatatcg\dagger_2cgat \Rightarrow_{RD_1} atcgcgatatcgcgat$

3.2 New Levels of Ambiguity

Now, we define various ambiguity levels for Matrix ins-del system based on the components used in the derivation. Consider the following derivation step in a Matrix ins-del system $\Upsilon, \delta: w_1 \Rightarrow w_2 \Rightarrow \ldots\ldots \Rightarrow w_m, m \geq 1$, such that $w_1 \in A$ and the following scenarios can happen (1): $w_k \Rightarrow w_{k+1}$ can be obtained by using a matrix which consists of only insertion rules R_{I_i} (2): $w_k \Rightarrow w_{k+1}$ can be obtained by using a matrix which consists of only deletion rules R_{D_j} (3): $w_k \Rightarrow w_{k+1}$, 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 inserted/deleted is called *Control Sequence* which is given as follows: $w_1, [(\alpha_1/\beta_1), \ldots, (\alpha_n/\beta_n)], \ldots, [(\alpha_{m-1}/\beta_{m-1}), \ldots, (\alpha_n/\beta_n)]$. The sequence which consists of used axiom, the strings α_j/β_j to be inserted/deleted and the used contexts (u_j, v_j) is called *Complete Control Sequence* which is given as follows: $w_1, [(u_1, \alpha_1/\beta_1, v_1), \ldots, (u_n, \alpha_n/\beta_n, v_n)], \ldots, [(u_{m-1}, \alpha_{m-1}/\beta_{m-1}, v_{m-1}), \ldots, (u_n, \alpha_n/\beta_n, v_n)]$. Informally, the control sequence means the order in which the strings are inserted/deleted and complete control sequence means the order of the contexts used in insertion/deletion rules. Note that one of α_j/β_j is empty for all j in the derivation. The position where insertion (α) /deletion (β) 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, $w1, w2 \in A$, $w1 \neq w2$, such that they both derive the same word z, i.e., $w_1 \Rightarrow^+ z$, $w_2 \Rightarrow^+ z$.
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.

IV. INTERPRETATION OF AMBIGUITY IN GENE SEQUENCES

In this section we show the application of the introduced new levels of ambiguity for Matrix ins-del systems with gene sequences.

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 1: *The ideal language L_{id} generated by Matrix ins-del system obeys 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 the ideal language. This sequence can be generated by the Matrix ins-del system Υ_{id} from two different axioms *at* and *ta* 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:

Derivation1: $\mathbf{at} \Rightarrow at\mathbf{gc} \Rightarrow a\mathbf{ct}g\mathbf{gc} \Rightarrow actga\mathbf{gc}\mathbf{t} \Rightarrow \mathbf{t}actgagc\mathbf{t}a$

Derivation2: $\mathbf{ta} \Rightarrow \mathbf{t}a\mathbf{gc} \Rightarrow \mathbf{ct}g\mathbf{gc} \Rightarrow \mathbf{a}ctgagc\mathbf{t} \Rightarrow \mathbf{t}actgagc\mathbf{ta}$

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 inserted/deleted.

Lemma 2 *The stem and loop language* $L_{sl} = \{uv\bar{u}^R \mid u,v \in \Sigma_{DNA}{}^*\}$ *represented by Matrix ins-del system obeys level 1 ambiguity.*

Proof. The stem and loop language 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 follows:

$R_{I_1} = [(\lambda, \lambda/b, \dagger 1), (\dagger 2, \lambda/\bar{b}, \lambda)]$,
$R_{I_2} = [(\lambda, \lambda/b, \dagger 3)]$,
$R_{D_1} = [(\lambda, \dagger 1/\lambda, \lambda), (\lambda, \dagger 2/\lambda, \lambda)]$,
$R_{D_2} = [(\lambda, \dagger 3/\lambda, \lambda)]$.

Consider the gene sequence *aagtt* in stem and loop language. 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 same. The two sequences are given as follows:

Sequence 1: a†1†3†2t $\Rightarrow_{R_{I_1}}$ a**a**†1†3†2**t**t $\Rightarrow_{R_{I_2}}$ aa†1**g**†3†2tt $\Rightarrow_{R_{D_1}}$ aag†3tt $\Rightarrow_{R_{D2}}$ aagtt

Sequence 2: a†1†3†2t$\Rightarrow_{R_{I_2}}$ a†1**g**†3†2t $\Rightarrow_{R_{I_1}}$ a**a**†1g†3†2**t**t $\Rightarrow_{R_{D_2}}$ aa†1g†2tt $\Rightarrow_{R_{D1}}$ aagtt

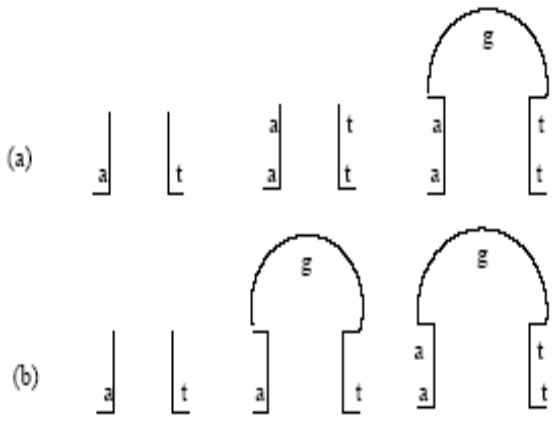

Fig. 3 Ambiguity in Stem and Loop Structure

In sequence 1, the order of strings used for insertion/deletion is $[(a,t)], [g], [(\dagger 1 \dagger 2)], [\dagger 3]$. In sequence2, the order of strings used for insertion/deletion 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} is 1- ambiguous. The Level 1 ambiguity can be pictorially represented as shown in Fig.3. Fig. 3(a) corresponds to sequence 1 and Fig.3(b) corresponds to sequence 2.

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 insertion/deletion.

Lemma 3: *The cloverleaf language* $Lcl = \{ uv1\bar{v}1^R v2\bar{v}2^R, \ldots, vn\bar{v}n^R u\bar{u}^R \mid u, v1, v2, \ldots, vn \in \Sigma_{DNA}{}^*, n \geq 0 \}$ *modelled by Matrix ins-del system obey level 2 ambiguity.*

Proof. The cloverleaf language L_{Cl} (for $n = 3$) can be generated by the Matrix ins-del system $\Upsilon_{Cl} = (\{b, \bar{b}, \dagger 1, \dagger 2, \dagger 3, \dagger 4, \dagger 4, \dagger 5\}, \{b, \bar{b}\}, \{\lambda, b\dagger 1 \dagger 2 \bar{b}, \dagger 3 \dagger 4 \dagger 5, b \dagger 1 \dagger 3 \dagger 4 \dagger 5 \dagger 2 \bar{b}\}, R)$, where $b \in \{a, t, g, c\}$, $\bar{b}$ is complement of b and R is

$R_{I_1} = [(\lambda, \lambda/b, \dagger 1), (\dagger 2, \lambda/\bar{b}, \lambda)]$,
$R_{I_2} = [(\lambda, \lambda/b, \dagger 3), (\dagger 3, \lambda/\bar{b}, \lambda)]$,
$R_{I_3} = [(\lambda, \lambda/b, \dagger 4), (\dagger 4, \lambda/\bar{b}, \lambda)]$,
$R_{I_4} = [(\lambda, \lambda/b, \dagger 5), (\dagger 5, \lambda/b, \lambda)]$,
$R_{D_1} = [(\lambda, \dagger 1/\lambda, \lambda), (\lambda, \dagger 2/\lambda, \lambda)]$,
$R_{D_2} = [(\lambda, \dagger 3/\lambda, \lambda)]$,
$R_{D_3} = [(\lambda, \dagger 4/\lambda, \lambda)]$,
$R_{D_4} = [(\lambda, \dagger 5/\lambda, \lambda)]$.

Consider the gene sequence *ctaatcgg* in cloverleaf language. This sequence can be generated in two different ordered complete control sequences by the Matrix ins-del system Υ_{Cl}. The two sequences are given as follows:

Sequence 1: c†1 †3 †4 †5 †2g $\Rightarrow_{R_{I_2}}$ c†1 **t**†3 **a**†4 †5 †2g$\Rightarrow$ $_{R_{I_3}}$ c†1 t†3 a**a**†4**t** †5 †2g $\Rightarrow_{R_{I_4}}$ c†1 t†3 aa†4t**c** †5**g** †2g$\Rightarrow$ $_{RD_1}$c t†3 aa†4tc †5g g$\Rightarrow_{R_{D_2}}$ c taa†4tc †5g g$\Rightarrow_{R_{D_3}}$ c taatc †5g g$\Rightarrow_{R_{D_4}}$c taatcgg

Sequence 2: c†1 †3 †4 †5 †2g $\Rightarrow_{R_{I_4}}$ c†1†3 †4 **c**†5**g** †2g$\Rightarrow$ $_{R_{I_3}}$ c†1†3 **a**†4**t**c †5g†2g $\Rightarrow_{R_{I_2}}$ c†1 **t**†3 **a**a†4tc†5g †2g$\Rightarrow_{RD_1}$c t†3 aa†4tc †5g g$\Rightarrow_{R_{D_2}}$ c taa†4tc †5g g$\Rightarrow_{R_{D_3}}$ c taatc †5g g$\Rightarrow_{R_{D_4}}$ctaatcgg

Note that in sequence 1, the order of contexts chosen is $[(\lambda, \dagger 3), (\dagger 3, \lambda)], [(\lambda, \dagger 4), (\dagger 4, \lambda)], [(\lambda, \dagger 5), (\dagger 5, \lambda)], [(\lambda, \lambda), (\lambda, \lambda)], [(\lambda, \lambda)], [(\lambda, \lambda)], [(\lambda, \lambda)]$. In another sequence the order of contexts chosen is $[(\lambda, \dagger 5), (\dagger 5, \lambda)], [(\lambda, \dagger 4), (\dagger 4, \lambda)], [(\lambda, \dagger 3), (\dagger 3, \lambda)], [(\lambda, \lambda), (\lambda, \lambda)], [(\lambda, \lambda)], [(\lambda, \lambda)], [(\lambda, \lambda)]$. Thus we are able to give two different ordered complete control sequences which derive the same gene sequence *ctaatcgg*. Therefore, the system Υ_{Cl} is 2-ambiguous. The Level 2 ambiguity can be pictorially represented as shown in Fig.4. Fig. 4(a) corresponds to sequence 1 and Fig.4(b) corresponds to sequence 2. This picture suggests a way of handling ambiguity issues in gene sequences and how they can be interpreted and what could be the intermediate sequences of genes in its sequence process.

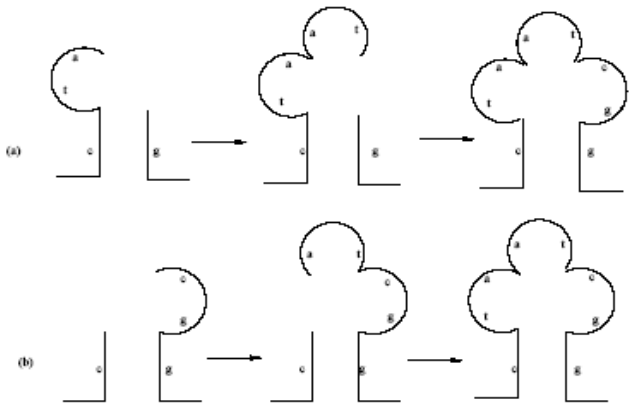

Fig. 4 Ambiguity in Cloverleaf Language

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 inserted/deleted.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

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

Lemma 4: The orthodox string Lod generated by Matrix ins-del systems obeys level 3-ambiguity.

Proof. The orthodox language Lod 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 $R_{I_1} = [(\lambda, \lambda / bb, \lambda)]$. Consider the string *gctagcat* in orthodox language. This string can be derived in two different descriptions by Υ_{od}. The two different descriptions are given as follows:

Description 1: ta⇒ **gc**ta ⇒gcta**gc** ⇒gctagc**at**

Description 2: ta ⇒ta**gc** ⇒**gc**tagc ⇒gctagc**at**

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.

V. CONCLUSION

In this paper, we discussed the Matrix insertion-deletion systems and using the system we have modelled some bio-molecular structures like pseudoknot and attenuator. We have introduced 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 bio-molecular structures like ideal, stem and loop, clover-leaf, orthodox has a relevance with the introduced levels of ambiguity. As we have now shown that ambiguity is possible in gene sequences, the research will throw some ideas on how gene sequences in DNA, RNA, protein molecules can be processed and synthesized.

ACKNOWLEDGMENT

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

REFERENCES

1. Cristian S. Calude and Gheorghe Paun, Computing with cells and atoms, An introduction to Quantum, DNA and Membrane Computing, London: Taylor and Francis, 2001.
2. David B. Searls, "Representing genetic information with formal grammars", in Proceedings of the National Conference on Artificial Intelligence, 1988, pp. 386–391.
3. David B. Searls, "The linguistics of DNA", in American Scientist, 1992, pp. 579–591.
4. David B. Searls, "The computational linguistics of biological sequences (Hunter, L.ed.)", in Artificial Intelligence and Molecular Biology, AAAI Press, 1993, pp.47–120.
5. Elena Rivas and Sean R. Reddy, "The language of RNA: A formal grammar that includes pseudoknots", in Bioinformatics, vol. 16., 2000, pp. 334–340.
6. Gheorghe Paun, Grzegorz Rozenberg and Arto Salomaa, DNA Computing, New Computing Paradigms. Springer, 1998.
7. Gheorghe Paun, Membrane Computing-An introduc tion. Springer, 2002.
8. John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman, Introduction to Automata Theory, Languages and Computation. Addison-Wesley, 2006.
9. Lakshmanan Kuppusamy, Anand Mahendran and Krishna S,`` Matrix Insertion-Deletion Systems for Bio- molecular Structures", in Proceedings of ICDCIT-2011, LNCS proceedings #6536, 2011, pp. 301-312.
10. Rozenberg and Arto Salomaa, Handbook of formal languages, Vol 1, Vol 2, Vol 3, Springer, 1997.
11. Setubal., Meidanis.: Introduction to Computational Molecular Biology. PWS Publishing Company, 1997.
12. Yasuo Uemura, Aki Hasegawa, Satoshi Kobayashi and Takashi Yokomori," Tree adjoining Grammars for RNA structure prediction", in Theoretical Computer Science, vol. 210, 1999, pp. 277–303.

AUTHORS PROFILE

Dr. Lakshmanan K, completed his bachelors from Bharathiar University, Coimbatore and masters from Bharathidasan University, Tirchirappalli, and obtained his Ph.D. in formal languages and automata theory. He works in bio-computing models, formal languages and automata theory. He has published more than 30 papers in peer reviewed international journals and in internal conferences. He is currently working as a professor in VIT University, Vellore, India.

Dr. Anand Mahendran, completed his Bachelors from VIT University, Vellore and Masters from Manonmainam Sundaranar University, Tirunelvelli. He obtained his Ph.D. degree from VIT University, Vellore under the guidance of Dr. Lakshmanan K. He has published more than 10 papers in reviewed international Journals and international conferences. He works in compilers, formal languages and automata theory. He currently works as an Associate Professor in VIT University, Vellore.