IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 29 August 2022, accepted 4 September 2022, date of publication 15 September 2022, date of current version 27 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3206772

RESEARCH ARTICLE

Analyzing the Trade-Off Between Complexity Measures, Ambiguity in Insertion System and Its Applications

ANAND MAHENDRAN[1], KUMAR KANNAN[1], MOHAMED HAMADA[2], (Senior Member, IEEE), AND MANUEL MAZZARA[3]

[1]School of Computer Science and Engineering, Vellore Institute of Technology, Vellore 632014, India
[2]Software Engineering Laboratory, The University of Aizu, Aizuwakamatsu 965-8580, Japan
[3]Institute of Software Development and Engineering, Innopolis University, 420500 Innopolis, Russia

Corresponding author: Anand Mahendran (manand@vit.ac.in)

The work of Anand Mahendran was supported by the Council of Scientific and Industrial Research (CSIR) under Project 25(0291)/18/EMR-II.

ABSTRACT Insertion is one of the basic operations in DNA computing. Based on this operation, an evolutionary computation model, the insertion system, was defined. For the above defined evolutionary computation model, varying levels of ambiguity and basic descriptional complexity measures were defined. In this paper, we define twelve new (descriptional) complexity measures based on the integral parts of the derivation, such as axioms, strings to be inserted, and contexts used in the insertion rules. Later, we analyze the trade-off among the (newly defined) complexity measures and the existing ambiguity levels. Finally, we examine the application of the analyzed trade-off in natural languages and modelling of bio-molecular structures.

INDEX TERMS Insertion systems, complexity measures, ambiguity levels, trade-off, natural languages.

I. INTRODUCTION

In the recent decades, the usage of computer has been increased enormously starting from storing and retrieving of data, manipulating scientific computations and performing other complex operations. To capture the needs of the fast growing world, there is a constant research happening in the domain of computer science. Due to the need of increase in computation speed and storage of data, the computing models used for computation and the technologies used for storage medium needs to be changed rapidly. As nature is always more faster than human brains and the computing devices, researchers felt that the nature would play a critical role, in specific, if biology is introduced in the domain of computer science. This initiated the notion of *natural computing or bio-inspired computing models* which bridged the gap between nature and computer science. As a result, lot of bio-inspired computing models have been defined namely *membrane computing, sticker systems, splicing systems, Watson-Crick automata, insertion-deletion systems, DNA Computing, H-systems* [6], [35], [36]. In formal language theory, the language generation depends on the rewriting operations, which paved a new dimension for insertion systems. If a string β is lodged between two substrings α_1, α_2 of a string $\alpha_1\alpha_2$ to get a new string $\alpha_1\beta\alpha_2$, then the performed operation on the strings is called insertion. Insertion operation was first theoretically studied in [16]. In DNA computing, the insertion operations have (some) biological relevance, which in turn has (some) biological relevant properties in human genetics. In [34], the application of the insertion operation in the domain of genetics has been investigated.

In 1969, Solomon Marcus introduced *Contextual grammars* which are mainly based on the descriptive linguistics [30]. In contextual grammars based on the selector, the context is inserted to the left and right of selectors. Using the adjoining operation, iteratively, the strings are generated in the language, where as in insertion system based on the left and right context, the string is inserted. In [33], different ambiguity levels were defined and studied for external, internal contextual grammars depending on the parts that are used in the derivation. For more details, on the ambiguity issues related to contextual grammars, we cite [18], [21], [22], [32], [34]. As insertion system can be viewed as the counterpart of contextual grammars, in the similar line of

The associate editor coordinating the review of this manuscript and approving it for publication was Yang Li.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

direction, in [23], various ambiguity levels were interpreted for insertion systems. As there will be more than one grammars (systems) $G_1, G_2, \ldots, G_n$ which generates a language L, a situation arises to choose an economical grammar (system) G which generates L. This idea of economical grammar (system) leads to the introduction of the notation called descriptional complexity measures. In [15], the complexity measures were investigated for context-free grammars. Several complexity measures were defined for contextual grammars such as *Ax,MAx,TAx,Con,MCon,TCon,Phi,MSel, TSel* [20], [37], [38]. In [24], depending on the insertion and deletion rules, different complexity measures were introduced and analyzed for insertion-deletion systems.

In programming and natural languages ambiguity is one of the interesting problems that needs to be investigated. First, we will discuss about the ambiguity issues in programming languages. Given a grammar G and an input string $w \in L(G)$, if it has more than one derivation or derivation trees (for the same string w), then the grammar G is said to be ambiguous. On the other hand the grammar G is unambiguous, if there exists only one derivation (derivation tree) for all the words in $L(G)$. The following example shows the importance and necessity of studying about the trade-off in programming languages. Consider the following Context Free Grammar (CFG) G which generates L (the set of arithmetic expressions): Let $G = (\{X, Y\}, \{a, +, *, (,)\}, X, \{X \to Y, X \to X + X, X \to X * X, X \to (X), Y \to a\})$. The grammar G is ambiguous. The string $a + a * a \in L(G)$ can be derived by two (distinct) left most derivations (LMD): LMD 1: $X \Longrightarrow X{*}X \Longrightarrow X{+}X{*}X \Longrightarrow a{+}X{*}X \Longrightarrow a{+}a{*}X \Longrightarrow a{+}a{*}a$. LMD 2: $X \Longrightarrow X + X \Longrightarrow a + X \Longrightarrow a + X * X \Longrightarrow a + a * X \Longrightarrow a + a * a$. The grammar G is ambiguous in addition to that the grammar G is minimal in terms of the measures *non-terminals* and *productions*. The minimal values of the measures based on the grammar G is 2 and 5 respectively. For the same language L, an unambiguous grammar G' can be derived. $G' = (\{X, Y, X', Y'\}, \{a, +, *, (,)\}, X, \{X \to X', X' \to Y', Y' \to Y, X \to X + X', X' \to X' * Y', Y' \to (X), Y \to a\}$. The interesting fact about the above grammar G' is unambiguous but is not minimal with respect to the measures non-terminals and number of productions. The Table.1 shows the comparison between the measures of the grammars G and G' respectively.

Based on the (minimal) measures number of non-terminals and productions a minimal CFG can be given for the expression language, but the given CFG is ambiguous. Whereas if the grammar G is unambiguous, it is not minimal in the measures non-terminals and productions. As insertion system is mainly based on the insertion operation, it has a potential application in generating natural languages and modelling of bio-molecular structures [30], [35]. In general, if we want to store natural languages, we will prefer economical and an unambiguous system. Under these circumstances, a trade-off needs to be performed based on the descriptional complexity measures and the ambiguity levels. As far as considering the research work on insertion systems it is mainly focused on the introduction of variants, reducing the weights towards the

TABLE 1. Comparison of complexity measures.

Measures	Ambiguous Grammar (G)	Unambiguous Grammar(G')
Number of Non-terminals	2	4
Number of Terminals	5	5
Number of productions	5	7

computational completeness, analyzing the relationship with Chomskian hierarchy of grammars, closure properties, ambiguity issues and decidability issues. For more details, we refer to [9], [10], [11], [12], [13], [23], [24], [25], [35]. This motivated to define new decsriptional complexity measures for insertion systems, perform the trade-off and to investigate the application of the analyzed trade-off.

The organization of the paper is given as, the preliminaries are dealt in Section II. The newly introduced descriptional complexity measures of insertion systems were discussed in Section III. The trade-off results between the newly defined complexity measures and various ambiguity levels of insertion systems were investigated in Section IV. The application of the trade-off between ambiguity and measures in natural languages and modelling of bio-molecular structures has been probed in Section V. The comparative study is dealt in Section VI. The conclusion and the future work is dealt in Section VII.

II. PRELIMINARIES

We start with discussing about the fundamental notations used in formal language theory. $V(\Sigma)$ is called an alphabet set. T is called a terminal set. The free monoid generated by $V(\Sigma)$ is represented as $V^*(\Sigma^*)$. The null string is denoted by Λ. *Strings* or *words* are the elements from $V^*(\Sigma^*)$. By eliminating Λ from $V^*(\Sigma^*)$, we can obtain $V^+(\Sigma^*)$. A language L is given as $L \subseteq V^*(\Sigma^*)$. For more details, we refer to [40].

An insertion system is defined as: $\gamma = (V, A, R)$, where V represents an alphabet, A is a finite language over the alphabet (axiom), R is defined as a set of finite insertion rules in the given format (u, β, v). In the above insertion rule (IR) the pair (u, v) is called context and $(u, v) \in V^* \times V^*$. The β represents the string to be inserted (IS) and $\beta \in V^+$. Given an insertion rule, depending on the left context (LC) and right context (RC), (u, v), the string β is inserted. If $(u, v) \in \lambda$, then the insertion of β can be done anywhere in the word.

Given an insertion rule (u, β, v), the y can be derived from x as follows $((x, y) \in V^*$ and $x \Longrightarrow y)$. Consider the following derivation step: $x = x_1u^{\downarrow}vx_2, y = x_1u\underline{\beta}vx_2$, *for some* $x_1, x_2 \in V^*$ *and* $(u, \beta, v) \in R$, $\downarrow$ marks the location of the string to be inserted, the inserted string is represented by a underline. The language generated by γ is defined as: $L(\gamma) = \{w \in V^* \mid x \Longrightarrow^* w,$ *for some* $x \in A\}$, where $\Longrightarrow^*$ is the reflexive and transitive closure of the defined relation $\Longrightarrow$.

In [23], six new levels of ambiguity were defined for insertion systems by considering the parts that are used in the derivation. Given a derivation step in an insertion system (γ) $\delta : w_1 \Longrightarrow w_2 \Longrightarrow \ldots \Longrightarrow w_m, m \geq 1$ where $w_1 \in A$ and the scenario can be

$w_j = x_{1,j}u_jv_jx_{2,j}$, $w_{j+1} = x_{1,j}u_j\beta_jv_jx_{2,j}$, when an insertion rule (u_j, β_j, v_j) is used, where $x_{1,j}, u_j, v_j, x_{2,j} \in V^*$. If the sequence has $w_1, \beta_1, \beta_2, \beta_3, \ldots, \beta_{m-1}$ (axioms and inserted strings) then it is *control sequence (CS)*. If the sequence has $w_1, (u_1, \beta_1, v_1), (u_2, \beta_2, v_2), (u_3, \beta_3, v_3), \ldots, (u_{m-1}, \beta_{m-1}, v_{m-1})$ (axioms, inserted strings and the contexts) then it is *complete control sequence (CCS)*. If the sequence is defined as $w_1, x_{1,1}u_1\beta_1v_1x_{2,1}, x_{1,2}u_2\beta_2v_2x_{2,2}, x_{1,3}u_3\beta_3v_3x_{2,3}, \ldots, x_{1,m-1}u_{m-1}\beta_{m-1}v_{m-1}x_{2,m-1}$, which is mainly based on the position where the string β is inserted, then it is a *description*. Given an insertion system γ, the Table. 2 depicts the various ambiguity levels.

TABLE 2. Different ambiguity levels of insertion system.

Ambiguity Level	Description
$0-ambiguous$	From two different axioms ($A_1, A_2 \in A,\ A_1 \neq A_2$,), a same word w can be derived
$1-ambiguous$	If the same word w can be obtained by two distinct unordered CS
$2-ambiguous$	If the same word w can be obtained by two distinct unordered CCS
$3-ambiguous$	If the same word w can be obtained by two distinct ordered CS
$4-ambiguous$	If the same word w can be obtained by two distinct ordered CCS
$5-ambiguous$	If the derivations are different based on the descriptions

Now, we recall the various complexity measures introduced for insertion-deletion (ins-del) systems. Given an ins-del system $\gamma = (V, T, A, R)$, the existing descriptional complexity measures of ins-del systems are defined as follows in Table. 3. For more details, we refer [20], [24], [37], [38]. Given the measure M and language L, the minimal system γ for the language L is defined as: $M(L) = min\{M(\gamma) \mid L = L(\gamma)\}$. For a given measure M and a language L, we define $M^{-1}(L) = \{\gamma \mid L(\gamma) = L \text{ and } M(\gamma) = M(L)\}$. In the above definition, $M^{-1}(L)$ denotes the set of all minimal systems that generates L which are minimal in the measures M. For a language L, two measures M_1, M_2 are said to be *incompatible* if the following relation $M_1^{-1}(L) \cap M_2^{-1}(L) = \emptyset$ holds true. If $M_1^{-1}(L) \cap M_2^{-1}(L) \neq \emptyset$, then the measures ($M_1$ and M_2) are called *compatible*. Based on the above definition, in [34], any two of the measures Ax, MAx, TAx are proved to be compatible. From the above measures (Table. 3), if the deletion rules were not used by the system γ, the measures $\{TDEL - StrCon, TDEL - Str\}$ are not applicable to insertion systems.

III. NEW DESCRIPTIONAL COMPLEXITY MEASURES

In this section, we introduce twelve new descriptional complexity measures depending on the integrants used in the derivation of the language. The Table. 4, shows the newly introduced measures.

IV. TRADE-OFF RESULTS BETWEEN (DESCRIPTIONAL) COMPLEXITY MEASURES AND AMBIGUITY LEVELS

In this section, we investigate the trade-off for insertion languages by considering the complexity measures and ambiguity levels.

Theorem 1: There are pseudo inherently 5-ambiguous insertion languages which are minimal in $M_1 \in \{TLen - LCon, MLen - LCon, TLen - LCon + InsStr, MLen - LCon + InsStr\}$ and $M_2 \in \{Ax\}$.

TABLE 3. Existing descriptional complexity measures of ins-del system.

S.No	Measure	Notation
1	Ax	$card(A)$
2	MAx	$\max_{w \in A} \lvert w \rvert$
3	TAx	$\sum_{w \in A} \lvert w \rvert$
4	$Prod$	$card(R)$
5	$TLength - Con$	$\sum_{(u,v) \in R} \lvert uv \rvert$
6	$TLength - Str$	$\sum_{(u,\alpha/\beta,v) \in R} \lvert \beta \rvert + \lvert \alpha \rvert$
7	$TINS - StrCon$	$\sum_{(u,\lambda/\beta,v) \in R} \lvert uv \rvert + \lvert \beta \rvert$
8	$TDEL - StrCon$	$\sum_{(u,\alpha/\lambda,v) \in R} \lvert uv \rvert + \lvert \alpha \rvert$
9	$TINS - Str$	$\sum_{(u,\lambda/\beta,v) \in R} \lvert \beta \rvert$
10	$TDEL - Str$	$\sum_{(u,\alpha/\lambda,v) \in R} \lvert \alpha \rvert$

Proof: Let the language $L_1 = \{b^3a^m \mid m \geq 0\}$. The following 5-ambiguous insertion system γ_1 can be used to generate L_1. $\gamma_1 = (\{a, b\}, \{b^3, b^3a\}, \{(a, a, \lambda)\})$. The system γ_1 is minimal in $TLen-LCon$, $MLen-LCon$, $TLen-LCon+InsStr$, $MLen - LCon + InsStr$. Now, we will prove on the minimal measures of the system γ_1. In this regard, first we will prove for the measures $TLen - LCon$, $MLen - LCon$. From the system γ_1, we can see that $TLen - LCon(\gamma_1) = 1$. As the system γ_1, uses only one insertion rule and since $TLen-LCon(\gamma_1) = 1$, which implies $MLen-LCon(\gamma_1) = 1$. Now, we will discuss on the other measures $TLen - LCon + InsStr$, $MLen - LCon + InsStr$. Any system which generates L_1, should have a string to be inserted of minimum length one. Therefore, $TINS - Str(\gamma_1) = 1$. Earlier, we have proved that the $TLen - LCon(\gamma_1) = 1 = MLen - LCon(\gamma_1)$. Therefore, $TLen-LCon+InsStr(\gamma_1) = 2 = MLen-LCon+InsStr(\gamma_1)$. From the above arguments, we can conclude that the system γ_1 is minimal in $TLen-LCon$, $MLen-LCon$, $TLen-LCon+InsStr$, $MLen - LCon + InsStr$.

Consider any γ_1' which generates L_1 which has $TLen - LCon(\gamma_1) = 1 = MLen - LCon(\gamma_1)$ and $TLen - LCon + InsStr(\gamma_1) = 2 = MLen - LCon + InsStr(\gamma_1)$. Consider a word $b^3a^k \in L_1$, for a large value of k. In the derivation of the above words, different a can be chosen, thus producing two different descriptions. Therefore, the system γ_1' is 5-ambiguous.

However, the language L_1 is unambiguous as there exists an unambiguous insertion system γ_1'' which generates L_1. Consider the system $\gamma_1'' = (\{a, b\}, \{b^3\}, \{(b^3, a, \lambda)\})$. The system γ_1'' is unambiguous. From the system γ_1'', it is clear that γ_1'' generates L_1. While deriving $b^3a^r, r \geq 1 \in L_1$, the position of the string a to be inserted is unique in the derivation. As the system γ_1'' has only one axiom b^3, it is minimal with respect to Ax. Note that the system γ_1'' is not minimal in $\{TLen - LCon, MLen - LCon, TLen - LCon + InsStr, MLen - LCon + InsStr\}$. $\square$

TABLE 4. Newly introduced descriptional complexity measures of insertion system.

S.No	Measure	Notation	Description
1	$MLen - InsStr$	$\max_{(u,\beta,v)\in R} \vert\beta\vert$	Maximum length of the IS
2	$mLen - InsStr$	$\min_{(u,\beta,v)\in R} \vert\beta\vert$	Minimum length of the IS
3	$MLen - LCon$	$\max_{(u,\beta,v)\in R} \vert u\vert$	Maximum length of the LC used in the IR
4	$MLen - RCon$	$\max_{(u,\beta,v)\in R} \vert v\vert$	Maximum length of the RC used in the IR
5	$mLen - LCon$	$\min_{(u,\beta,v)\in R} \vert u\vert$	Minimum length of the LC used in the IR
6	$mLen - RCon$	$\min_{(u,\beta,v)\in R} \vert v\vert$	Minimum length of the RC used in the IR
7	$TLen - LCon$	$\sum_{(u,\beta,v)\in R} \vert u\vert$	Total length of all LC used in the IR
8	$TLen - RCon$	$\sum_{(u,\beta,v)\in R} \vert v\vert$	Total length of all RC used in the IR
9	$TLen - LCon + InsStr$	$\sum_{(u,\beta,v)\in R} \vert u\vert + \vert\beta\vert$	Total length of LC + the length of the IS
10	$TLen - RCon + InsStr$	$\sum_{(u,\beta,v)\in R} \vert v\vert + \vert\beta\vert$	Total length of LC + the length of the IS
11	$MLen - LCon + InsStr$	$\max_{(u,\beta,v)\in R} \vert u\vert + \vert\beta\vert$	Maximum length of LC + the length of the IS
12	$MLen - RCon + InsStr$	$\max_{(u,\beta,v)\in R} \vert v\vert + \vert\beta\vert$	Maximum length of RC + the length of the IS

Corollary 1: There are pseudo inherently 5-ambiguous insertion languages with $M_1 \in \{TLen - LCon, MLen - LCon, TLen - LCon + InsStr, MLen - LCon + InsStr\}$ and $M_2 \in \{MAx, TAx\}$.

Theorem 2: There are pseudo inherently 5-ambiguous insertion languages which are minimal in $M_1 \in \{TINS - Str\}$ and $M_2 \in \{TLen - RCon, MLen - RCon, mLen - RCon\}$.

Proof: Let the language $L_2 = \{d(a^3b)^k \mid k \geq 0\} \cup \{(a^3b)^k c \mid k \geq 0\}$. The following 5-ambiguous insertion system γ_2 can be used to generate L_2. $\gamma_2 = (\{a, b, c, d\}, \{d, da^3b, c, a^3bc\}, \{(\lambda, a^3b, a^3b)\})$. The system γ_2 is minimal in $TINS - Str$. Any insertion system γ_2' which generates L_2 should have an insertion string of length ≤ 4. Therefore, γ_2 is minimal in $TINS - Str$ and $TINS - Str(L_2) = 4$.

Consider any γ_2' which generates L_2 which has $TINS - Str = 4$. Consider the words $d(a^3b)^i$ or $(a^3b)^j c \in L_2$, for a large values of i, j. In the derivation of the above word, different a^3b can be chosen, thus producing two different descriptions. Therefore, the system γ_2' is 5-ambiguous.

However, the language L_2 is unambiguous as there exists an unambiguous insertion system γ_2'' which generates L_2. Consider the system $\gamma_2'' = (\{a, b, c, d\}, \{c, d\}, \{(d, a^3b, \lambda), (\lambda, a^3b, c)\})$. The system γ_2'' is unambiguous. With the help of the insertion rule (d, a^3b, λ), $d(a^3b)^k, k \geq 0$ can be generated. Likewise, by using the insertion rule (λ, a^3b, c), $(a^3b)^k c, k \geq 0$ can be generated. While deriving $d(a^3b)^r$ or $(a^3b)^s c, r, s \geq 1 \in L_2$, the position of the string to be inserted a^3b is unique in the derivation. From the system γ_2'', it is easy to see that the γ_2'' is minimal with respect to $\{TLen - RCon, MLen - RCon, mLen - RCon\}$. □

Corollary 2: In the above result, if the insertion rule is changed as (a^3b, a^3b, λ), the language L_2 is represented as L_2'. For the language L_2', there exists a result for the following measures. There are pseudo inherently 5-ambiguous insertion languages with respect to $M_1 \in \{TINS - Str\}$ and $M_2 \in \{TLen - LCon, MLen - LCon, mLen - LCon\}$.

Theorem 3: There are pseudo inherently 5-ambiguous insertion languages which are minimal in $M_1 \in \{TINS - Str, TLen - RCon, MLen - RCon, mLen - RCon, TLen - RCon+InsStr, MLen-RCon+InsStr\}$ and $M_2 \in \{Ax, TLen-LCon, MLen - LCon, TLen - LCon + InsStr\}$.

Proof: Let the language $L_3 = \{cba^2ca^n \mid n \geq 1\} \cup \{dba^2da^n \mid n \geq 1\}$. The following 5-ambiguous insertion system γ_3 can be used to generate L_3. $\gamma_3 = (\{a, b, c, d\}, \{cba^2ca, cba^2ca^2, cba^2ca^3, dba^2da, dba^2da^2, dba^2da^3\}, \{(a^3, a, \lambda)\})$. The system γ_3 is minimal in $TINS - Str, TLen - RCon, MLen - RCon, mLen - RCon, TLen - RCon + InsStr, MLen - RCon + InsStr$. Any insertion system γ_3' which generates L_3 should have an insertion string of length ≤ 1. Therefore, γ_3 is minimal in $TINS - Str$ and $TINS - Str(L_3) = 1$. Now, we will prove for other minimal measures. As the insertion system has only one insertion rule and it uses λ as the right context, the system γ_3 is minimal in the measures $TLen - RCon, MLen - RCon, mLen - RCon$. As the $TINS - Str(L_3) = 1$ and $TLen - RCon(L_3) = \lambda = MLen-RCon(L_3)$, we can conclude the system γ_3 is minimal in the measures $TLen-RCon+InsStr, MLen-RCon+InsStr$.

Consider any γ_3' which generates L_3 which has $TINS-Str = 4$, $\{TLen-RCon, MLen-RCon, mLen-RCon\} = \lambda$, $\{TLen-RCon+InsStr, MLen-RCon+InsStr\} = 1$. Consider the words cba^2ca^s or $dba^2da^t \in L_3$, for a large values of s, t. In the derivation of the above words, different a can be chosen, thus producing two different descriptions. Therefore, the system γ_3' is 5-ambiguous.

However, the language L_3 is unambiguous as there exists an unambiguous insertion system γ_3'' which generates L_3. Consider the system $\gamma_3'' = (\{a, b, c, d\}, \{cb^2ca, db^2da\}, \{(c, a, a), (d, a, a)\})$. The system γ_3'' is unambiguous and minimal in the measures $Ax, TLen-LCon, MLen-LCon, TLen-LCon+InsStr$. With the help of the insertion rule (c, a, a), $cba^2ca^k, k \geq 1$ can be generated. Likewise, by the using the insertion rule (d, a, a), $dba^2da^k, k \geq 1$ can be generated. While deriving cba^2ca^r or $dba^2da^s, r, s \geq 1 \in L_3$, the position of the string to be inserted a is unique in the derivation. Any system which generates L_3 should have minimum two axioms. As the system uses the insertion rules of the form $\{(c, a, a), (d, a, a)\}$, it is easy to see that the γ_3'' is minimal with respect to $\{Ax, TLen-LCon, MLen-LCon, TLen-LCon+InsStr\}$. □

Corollary 3: There are pseudo inherently 5-ambiguous insertion languages with respect to $M_1 \in \{TINS-Str, TLen-RCon, MLen-RCon, mLen-RCon, TLen-RCon+InsStr, MLen-RCon+InsStr\}$ and $M_2 \in \{Ax, MAx, TAx, TLen-LCon, MLen-LCon, TLen-LCon+InsStr\}$.

Theorem 4: There are pseudo inherently 4-ambiguous insertion languages with $M_1 \in \{Ax\}$ and $M_2 \in \{TLen-LCon, MLen-LCon, TINS-Str\}$.

Proof: Let the language $L_4 = \{c^2a^n \mid n \geq 0\} \cup \{d^2a^n \mid n \geq 0\} \cup \{c^2a^nd^2a^m \mid n, m \geq 0\}$. The following 4-ambiguous insertion system γ_4 can be used to generate L_4. $\gamma_4 = (\{a, b, c, d\}, \{c^2, d^2, c^2d^2\}, \{(c^2, a, \lambda), (d^2, a, \lambda)\})$. To generate L_4, any insertion system γ_4' should have the axioms of the form (which should be minimum three) c^2, d^2, c^2d^2. Therefore, $Ax(L_4) = 3$.

Consider any γ_4' which generates L_4, where $Ax(L_4) = 3$. To generate $c^2a^l, l \geq 0$ of L_4, definitely, the insertion system must have an insertion rule of the following form (c^2, a^i, λ), $i \geq 1$. To generate $d^2a^k, k \geq 0$ of L_4, definitely, the insertion system should have a rule of the form (d^2, a^j, λ), $j \geq 1$. In order to prove γ_4' is 4-ambiguous, let us examine a string $c^2a^{l+i}d^2a^{k+j} \in L_4$. The above string can be acquired by two different ordered CCS. In one CCS, first the following insertion rule (c^2, a, λ) can be used, followed by the another insertion rule (d^2, a, λ). In another CCS, first the following insertion rule (d^2, a, λ) can be used, followed by the other insertion rule (c^2, a, λ). As the string to be inserted a^i is same for any arbitrary system, the insertion system γ_4' is 1 and 3-unambiguous.

However, L_4 is unambiguous as there exists an unambiguous system γ_4'' which generates L_4. Consider the system $\gamma_4'' = (\{a, b, c, d\}, \{c^2, d^2, c^2d^2, c^2a, d^2a, c^2ad^2, c^2d^2a, c^2a d^2a\}, \{(a, a, \lambda)\})$. The system γ_4'' is minimal while considering the following measures: $\{TLen-LCon, MLen-LCon, TINS-Str\}$. As the system γ_4'' uses only one insertion rule (a, a, λ), it is clear that the system is 4-unambiguous. □

Corollary 4: There are pseudo inherently 4-ambiguous insertion languages with $M_1 \in \{MAx, TAx\}$ and $M_2 \in \{TLen-LCon, MLen-LCon, TINS-Str\}$.

Theorem 5: There are pseudo inherently 4-ambiguous insertion languages with respect to $M_1 \in \{Ax, mLen-LCon, MLen-LCon+InsStr\}$ and $M_2 \in \{TLen-LCon\}$.

Proof: Let the language $L_5 = \{c^2(ab^3)^n \mid n \geq 1\} \cup \{d^2(ab^3)^n \mid n \geq 1\} \cup \{c^2(ab^3)^nd^2(ab^3)^m \mid n, m \geq 0\}$. The following 4-ambiguous insertion system γ_5 can be used to generate L_5. $\gamma_5 = (\{a, b, c, d\}, \{c^2ab^3, d^2ab^3, c^2d^2\}, \{(c^2, ab^3, \lambda), (d^2, ab^3, \lambda)\})$. First, we will discuss on the measure Ax. The axioms c^2ab^3, d^2ab^3, c^2d^2 can be used to derive the first, second and third part of L_5. It is easy to see that minimum three axioms should be there to generate L_5. Therefore, $Ax(L_5) \leq 3$, which implies $Ax(L_5) = 3$. Next, we will discuss on the measure $mLen-LCon$. The insertion rule (c^2, ab^3, λ) is used to generate the first part of the language L_5. The $d^2(ab^3)^n, n \geq 1$ part of the language L_5 can be derived using the following insertion rule (d^2, ab^3, λ). By using the insertion rules alternatively,the third part of the language L_5 can be generated. Any system which generates L_5, should have insertion rules whose $mLen-LCon(L_5) \leq 2$, which implies $mLen-LCon(L_5) = 2$. Next, we will discuss on the measure $MLen-LCon+InsStr$. Any system which generates L_5 should have the left contexts c^2, d^2 in the insertion rules, which implies $MLen-LCon(L_5) = 2$. Likewise, the inserted string should be ab^3, which implies $MLen-InsStr(L_5) = 4$. As the measure $MLen-LCon+InsStr$ is the combination of the above two measures, we can conclude $MLen-LCon+InsStr(L_5) = 6$.

Consider any γ_5' which generates L_5, where $Ax(L_5) = 3$, $mLen-LCon(L_5) = 2$ and $MLen-LCon+InsStr(L_5) = 6$. Since $c^2(ab^3)^n \in L_5$, the insertion system γ_5' should have an insertion rule which should be of the following form $(c^2, (ab^3)^r, \lambda), r \geq 1$. Likewise, $d^2(ab^3)^n \in L_5$, the system should have an insertion rule of the form $(d^2, (ab^3)^s, \lambda)$, $s \geq 1$. Consider the word $c^2(ab^3)^td^2(ab^3)^u \in L_5$. This word can be generated from the axiom c^2d^2 by means of two different ordered CCS. In one CCS, the insertion rule (c^2, ab^3, λ) is used followed by (d^2, ab^3, λ). In another CCS, first the following insertion rule (d^2, ab^3, λ) is used in the derivation followed by the another insertion rule (c^2, ab^3, λ). Therefore, it implies the insertion system γ_5' is 4-ambiguous. The system γ_5' is 1 and 3-unambiguous, because the same string ab^3 is inserted in both the derivations.

However, the language L_5 is unambiguous since there exists an 4-unambiguous system $\gamma_5'' = (\{a, b, c, d\}, \{c^2ab^3, d^2 ab^3, c^2d^2, c^2ab^3d^2, c^2d^2ab^3, c^2ab^3d^2ab^3\}, \{(b^3, ab^3, \lambda)\})$. The system γ_5'' is 4-unambiguous as it uses only one insertion rule. The system γ_5'' is minimal with respect to $TLen-LCon$. Note that, the insertion system γ_5'' is not minimal in the measures $\{Ax, mLen-LCon, MLen-InsStr\}$. □

Corollary 5: There exists pseudo inherently 4-ambiguous insertion languages with $M_1 \in \{MAx, TAx, mLen-LCon, MLen-InsStr\}$ and $M_2 \in \{TLen-LCon\}$.

Theorem 6: There are pseudo inherently 2-ambiguous insertion languages with $M_1 \in \{Ax\}$ and $M_2 \in \{TLen-LCon, TLen-RCon, MLen-LCon, MLen-RCon, TLen-LCon+InsStr, TLen-RCon+InsStr\}$.

Proof: Let the language $L_6 = \{a^2b^m \mid m \geq 0\} \cup \{b^mc^2 \mid m \geq 0\} \cup \{a^2b^mc^2 \mid m \geq 0\}$. The following 2-ambiguous insertion system γ_6 can be used to generate L_6. $\gamma_6 = (\{a, b, c\}, \{a^2, c^2, a^2c^2\}, \{(a^2, b, \lambda), (\lambda, b, c^2)\})$. The insertion system γ_6 is minimal in Ax. Any system which generates L_6 should have minimum three axioms $\{a^2, c^2, a^2c^2\}$. Therefore, $Ax(L_6) \leq 3$, in turn it infers $Ax(L_6) = 3$.

Consider any insertion system γ_6' which is used to generate L_6, where $Ax(L_6) = 3$. Since the strings of the form $a^2b^i, i \geq 0 \in L_6$, definitely in the insertion rule there should be a context of the form $(a^2, b^u), u \geq 0$. Likewise, since the string of the form $b^jc^2, j \geq 0 \in L_6$, definitely in the insertion rule there should be a context of the form $(b^t, c^2), t \geq 0$. In both the cases, the inserted string will be $b^v, v \geq 1$. To prove the insertion system is γ_6' is 2-ambiguous, lets us take a string $a^2b^{e+f}c^2 \in L_6$. From two (different) unordered CCS, the above string can be obtained from the (same) axiom. In one sequence using the context (a^2, b^u) completely, the string $a^2b^{e+f}c^2$ can be obtained. In another sequence, using the context (b^t, c^2) completely, the string $a^2b^{e+f}c^2$ can be derived. Thus, the same word $a^2b^{e+f}c^2$ is derived from two distinct unordered CCS. Therefore, the language L_6 is 2-ambiguous.

The language L_6 is unambiguous as $L(\gamma_6'') = L_6$. The 2-unambiguous insertion system is given as: $\gamma_6'' = (\{a, b, c\}, \{a^2, c^2, a^2b, bc^2, a^2c^2, a^2bc^2\}, \{(b, b, \lambda)\})$. Since, the system γ_6'' has only one context in the insertion rule (b, λ), it implies γ_6'' is 2-unambiguous. The insertion system γ_6'' is minimal in the measures $\{TLen-LCon, TLen-RCon, MLen-LCon, MLen-RCon, TLen-LCon+InsStr, TLen-RCon+InsStr\}$. From the system γ_6'', it is not minimal in Ax. □

Corollary 6: There are pseudo inherently 2-ambiguous insertion languages in the measures $M_1 \in \{MAx, TAx\}$ and $M_2 \in \{TLen-LCon, TLen-RCon, MLen-LCon, MLen-RCon, TLen-LCon+InsStr, TLen-RCon+InsStr\}$.

Theorem 7: There are pseudo inherently 2-ambiguous insertion languages which are minimal in $M_1 \in \{TLen-LCon, TLen-LCon+InsStr\}$ and $M_2 \in \{TLen-RCon, TLen-RCon+InsStr\}$.

Proof: Let the language $L_7 = \{a(bc^3)^n \mid n \geq 1\} \cup \{(bc^3)^nd \mid n \geq 1\} \cup \{a(bc^3)^nd \mid n \geq 0\}$. The following 2-ambiguous insertion system γ_7 can be used to generate L_7. $\gamma_7 = (\{a, b, c, d\}, \{abc^3, bc^3d, ad\}, \{(a, bc^3, \lambda), (\lambda, bc^3, d)\})$. The system γ_7 is minimal with respect to $\{TLen-LCon, TLen-LCon+InsStr\}$. Any system which generates L_7 should have the following contexts $\{(a, \lambda), (\lambda, d)\}$ in the insertion rules and the inserted string of the form bc^3. From the system γ_7, it is clear that the system γ_7 is minimal in the measures $\{TLen-LCon, TLen-LCon+InsStr\}$. It is easy to see that $TLen-LCon \leq 1$ and $TLen-LCon+InsStr \leq 5$, which implies that $TLen-LCon = 1$ and $TLen-LCon+InsStr = 5$.

Consider any system γ_7' which generates L_7, where $TLen-LCon(L_7) = 1$ and $TLen-LCon+InsStr(L_7) = 5$. Since the strings of the form $a(bc^3)^i, i \geq 1 \in L_7$, definitely in the insertion rule there should be a context of the form $(a, (bc^3)^t), t \geq 0$. Likewise, since the strings of the form $(bc^3)^jd, j \geq 1 \in L_7$, definitely in the insertion rule there should be a context of the form $((bc^3)^s, d), s \geq 0$. In both the cases, the inserted string will be $(bc^3)^k, k \geq 1$. In order to prove the insertion system is γ_7' is 2-ambiguous, lets us take a string $a(bc^3)^{e+f}d \in L_7$. From two (different) unordered CCS, the above word can be obtained from the (same) axiom. In one sequence using the context $(a, (bc^3)^t)$ completely, the string $a(bc^3)^{e+f}d$ can be obtained. In another sequence, using the context $((bc^3)^s, d)$ completely, the string $a(bc^3)^{e+f}d$ can be derived. Thus, the same word $a(bc^3)^{e+f}d \in L_7$, is derived from two different unordered CCS. Therefore, the language L_7 is 2-ambiguous.

However, the language L_7 is 2-unambiguous since there exists an 2-unambiguous insertion system which is minimal in $\{TLen-RCon, TLen-RCon+InsStr\}$. $\gamma_7'' = (\{a, b, c, d\}, \{abc^3, bc^3d, ad, abc^3d\}, \{(c^3, bc^3, \lambda)\})$. As the system γ_7'' uses only one insertion rule, obviously, there will be only one context in the insertion rule (c^3, λ). Therefore,the system γ_7'' is 2-unambiguous. The system γ_7'' is minimal in the measures $\{TLen-RCon, TLen-RCon+InsStr\}$. Note that, the insertion system γ_7'' is not minimal in the measure $\{TLen-LCon, TLen-LCon+InsStr\}$. □

Theorem 8: There are pseudo inherently 0-ambiguous insertion languages with respect to $M_1 \in \{Ax, MLen-InsStr, TINS-Str, mLen-InsStr\}$ and $M_2 \in \{MLen-LCon, MLen-RCon, TLen-LCon, TLen-RCon\}$.

Proof: Let the language $L_8 = \{a^2b^{3n} \mid n \geq 1\} \cup \{b^{2n}c^2 \mid n \geq 1\} \cup \{a^2b^nc^2 \mid n \geq 0\}$. The following 0-ambiguous insertion system γ_8 can be used to generate L_8. $\gamma_8 = (\{a, b, c\}, \{a^2b^3, b^2c^2, a^2c^2, a^2bc^2\}, \{(a^2, b^3, \lambda), (\lambda, b^2, c^2)\})$. The system γ_8 is minimal with respect to $\{Ax, MLen-InsStr, TINS-Str\}$. First, we will prove for the measures $\{MLen-InsStr, TINS-Str\}$. From the system γ_8, it is easy to see that $MLen-InsStr(\gamma_8) \leq 3$ and $TINS-Str(\gamma_8) \leq 5$. To generate the strings of the form $a^2b^kc^2, k \geq 0 \in L_8$, the insertion rule should have the string b. However, if such an insertion string is present in any of the insertion rules, then the system γ_8 may generate some strings $a^2b^{3p}, p \geq 1$ and $b^{2q}c^2, q \geq 1$ which doesn't $\notin L_8$. From the above claim, all the parts of L_8 cannot be produced by the insertion string b, which implies $mLen-InsStr(\gamma_8) = 2$. Next, we will discuss on the following measure $TINS-Str$. Since the strings of the structure $a^2b^{3p}, p \geq 1 \in L_8$, insertion rule will certainly have the string b^3. Likewise, since the strings of the structure $b^{2q}c^2, q \geq 1 \in L_8$, insertion rule will certainly have the string b^2. Therefore, we conclude $MLen-InsStr(\gamma_8) \geq 3$ and $TINS-Str(\gamma_8) \geq 5$.

Next, we will discuss on the measure Axiom. Any system which generates L_8 will have three axioms a^2b^3, b^2c^2, a^2c^2. Next, we will discuss why the system should have an axiom a^2bc^2. If the system is not having the axiom a^2bc^2, probably it can be generated by using the axiom a^2c^2 by inserting the string b. But previously, we have proved that the system cannot have b as the string to be inserted. Therefore, it implies a^2bc^2 should be present in the axiom. Therefore, the system γ_8 is minimal in the measure Ax.

Consider any system γ_8' which generates L_8. The system γ_8' is minimal in the measure $Ax, MLen - InsStr, TINS - Str, mLen - InsStr$. In order to claim γ_8' is 0-ambiguous, let us take the strings a^2b^{3r} and $b^{2s}c^2 \in L_8$, for a large values of r and s. To produce the words of the form $a^2b^{3p}, p \geq 1$ and $b^{2q}c^2, q \geq 1$, the insertion system γ_8' should have strings of the form $b^{3t}, t \geq 1$ and $b^{2u}, u \geq 1$ respectively. Consider a word $a^2b^{3tm+2un}c^2 \in L_8$, for $m \geq 1, n \geq 0$. The above word can be achieved from two unique axioms a^2c^2 and a^2bc^2. Starting from the axiom a^2c^2, the word $a^2b^{3tm+2un}c^2$ can be obtained by inserting the strings b^{3t}, m-times and b^{2u}, n-times. On the other hand, the word $a^2b^{3tm+2un}c^2$ can be derived from the axiom a^2bc^2. In one derivation, the string b^{3t} can be inserted for $m-i_1$-times, $i_1 \geq 1$. In another derivation, the string b^{2u} can be inserted for $n+i_2$-times, $i_2 \geq 1$. Thus, the word $a^2b^{3tm+2un}c^2$ is obtained from two different axioms a^2c^2, a^2bc^2. Therefore, the system γ_8' is 0-ambiguous. For the measures $MLen - InsStr, TINS - Str, mLen - InsStr$, an akin reasoning can be given.

Next, we have to prove the L_8 is 0-unambiguous, by showing there exists an 0-unambiguous system $\gamma_8'' = (\{a, b, c\}, \{a^2b^3, a^2b^6, b^2c^2, b^4c^2, b^6c^2, a^2c^2, a^2bc^2, a^2b^2c^2, a^2b^3c^2, a^2b^4c^2, a^2b^5c^2\}, \{(b, b^6, \lambda)\})$ which generates L_8. The system will produce a unique derivation step for any word $\in L_8$, starting from an axiom by inserting the string b^6, which shows γ_8'' is 0-unambiguous. As the system uses the following insertion rule (b, b^6, λ), the system γ_8'' is minimal in the measures $MLen - LCon, MLen - RCon, TLen - LCon, MLen - RCon$. □

Corollary 7: There are pseudo inherently 0-ambiguous insertion languages in the measures $M_1 \in \{MAx, TAx, MLen-InsStr, TINS - Str, mLen - InsStr\}$ and $M_2 \in \{MLen - LCon, MLen - RCon, TLen - LCon, TLen - RCon\}$.

Theorem 9: There are pseudo inherently 0-ambiguous insertion languages with respect to $M_1 \in \{Ax, mLen-InsStr\}$ and $M_2 \in \{MLen - RCon, mLen - RCon, TLen - RCon + InsStr, MLen - RCon + InsStr\}$.

Proof: Let the language $L_9 = \{ba^{2n} \mid n \geq 1\} \cup \{a^{4n}c \mid n \geq 1\} \cup \{ba^nc \mid n \geq 0\}$. The following 0-ambiguous insertion system γ_9 can be used to generate L_9. $\gamma_9 = (\{a, b, c\}, \{ba^2, a^4c, bc, bac\}, \{(b, a^2, \lambda), (\lambda, a^4, c)\})$. The system γ_9 is minimal with respect to $\{Ax, mLen-InsStr\}$. First, we will prove for the measure $mLen - InsStr$. From the system γ_9, it is easy to see that $mLen - InsStr(\gamma_9) \leq 2$. To generate the strings of the form $ab^k, k \geq 0 \in L_9$, the insertion rule should have the string a. However, if such an insertion string is present in any of the insertion rules, then the system γ_9 may generate some strings $ba^{2p}, p \geq 1$ and $a^{4q}c, q \geq 1$ which doesn't $\notin L_9$. From the above claim, all the parts of L_9 cannot be produced by the insertion string a. So, obvisouly the minimum length of the insertion string should be a^2. Therefore, we conclude $mLen - InsStr(\gamma_9) \geq 2$.

Next, we will discuss on the measure Axiom. Any system which generates L_9 will have three axioms ba^2, a^4c, bc. Next, we will discuss why the system γ_9 should have an axiom bac. If the system is not having the axiom bac, probably it can be generated by using the axiom bc by inserting the string a. But previously we have proved that the system cannot have a as the string to be inserted. Therefore, it implies bac should be present in the axiom. Therefore, the system γ_9 is minimal in the measure Ax.

Consider any system γ_9' which generates L_9. The system γ_9' is minimal in the measure $Ax, mLen - InsStr$. In order to claim γ_9' is 0-ambiguous, let us take the strings ba^{2r} and $a^{4s}c \in L_9$, for a large values of r and s. To produce the words of the form $ba^{2p}, p \geq 1$ and $a^{4q}c, q \geq 1$, the insertion system γ_9' should have strings of the form $a^{2t}, t \geq 1$ and $a^{4u}, u \geq 1$ respectively. Consider a word $ba^{4tm+2un}c \in L_9$, for $m \geq 1, n \geq 0$. The above word can be achieved from two unique axioms bc and bac. Starting from the axiom bc, the word $ba^{4tm+2un}c$ can be obtained by inserting the strings a^{2t}, m-times and a^{4u}, n-times. On the other hand, the word $ba^{4tm+2un}c$ can be derived from the axiom bac. In one derivation, the string a^{2t} can be inserted for $m - i_1$-times, $i_1 \geq 1$. In another derivation, the string a^{4u} can be inserted for $n+i_2$-times, $i_2 \geq 1$. Thus, the word $ba^{4tm+2un}c$ is obtained from two different axioms bc, bac. Therefore, the system γ_9' is 0-ambiguous.

Next, we have to prove the L_9 is 0-unambiguous, by showing there exists an 0-unambiguous system $\gamma_9'' = (\{a, b, c\}, \{ba^2, ba^4, a^4c, bc, bac, ba^2c, ba^3c, ba^4c\}, \{(a, a^4, \lambda)\})$ which generates L_9. The system will produce a unique derivation step for any word $\in L_9$, starting from an axiom by inserting the string a^4, which shows γ_9'' is 0-unambiguous. As the system uses the following insertion rule (a, a^4, λ), the system γ_9'' is minimal in the measures $\{MLen-RCon, mLen-RCon, TLen - RCon + InsStr, MLen - RCon + InsStr\}$. □

Corollary 8: There are pseudo inherently 0-ambiguous insertion languages in the measures $M_1 \in \{MAx, TAx, mLen-InsStr\}$ and $M_2 \in \{MLen - RCon, mLen - RCon, TLen - RCon + InsStr, MLen - RCon + InsStr\}$.

Table.5 shows the different trade-off results acquired for various blends of the ambiguity levels and descriptional complexity measures. The intersecting entry at M_1 and M_2 shows there exists a pseudo inherently ambiguous insertion languages with respect to the measure M_1 and M_2. For example, the intersection of row 7 ($mLen - LCon$) and column 9 ($TLen - LCon$) indicates the language L_5 which is pseudo inherently 4-ambiguous with respect to the measures $M_1 \in mLen - LCon$ and $M_2 \in TLen - LCon$. In Table. 4, the intersection of (M_1, M_2) entries that are empty are left as open problems. The entries which are in diagonal and those marked by $\star$ are not suitable for the trade-off study.

V. APPLICATION OF THE TRADE-OFF RESULTS

In this section, we analyze the application and significance of the trade-off in natural languages, modelling of bio-molecular structures. Before moving on to the application, first, we will discuss about the controlling parameters and limitations of the proposed trade-off study. Given an insertion system, the weight of the system is the sum of n, i, j. The n denotes the maximal length of insertion string and i, j denotes the maximal length of the left and right context used in insertion rules. The weight of a insertion system is given as (n, i, j). Several attempts were made on the insertion systems to characterize recursively enumerable languages with a lesser weights. In addition to that many variants has been introduced such as universal matrix insertion grammars, graph controlled insertion-deletion systems, path controlled insertion systems, insertion-deletion P Systems, Context-free insertion-deletion systems [9], [10], [11], [19], [31]. Natural languages such as English, Dutch has some grammatical structures that are beyond the power of context-free languages. As insertion system can characterize recursively enumerable languages the system can be considered as one of the prominent grammar models in generating natural language constructs and modelling of the bio-molecular structures. The computational completeness of the insertion systems mainly depends on the weights used in the insertion rules. In practical, such weights will play a limiting factor while generating the natural language constructs and modelling the bio-molecular structures. Despite of such practical difficulty the application what we had investigated in this paper will throw a new light on theoretical study on the trade-off.

A. APPLICATION OF THE ANALYZED TRADE-OFF IN NATURAL LANGUAGES

Syntactic and semantic ambiguity deserves a special attention in natural, programming and artificial languages. As the programming language constructs are mainly based on syntax and semantic rules handling these ambiguities is not a great deal of interest, whereas in natural languages handling syntactic ambiguity is easier when compared to semantic ambiguity. The main reason is, while dealing with the natural languages, one sentence (or a word) can convey different meaning. Even in Google translator, if the translation is carried out word by word the meaning may be different from the source to the target language. Under these circumstances, natural languages should be translated (stored) in an unambiguous manner. As we know, for every (natural/programming/artificial) language, there is a grammar G, such that $L(G) = L$. To generate the natural languages such as English, Dutch, we need grammars that are beyond the (generative) capability of Context-free grammar [5], [39]. In addition to that, many natural languages has the existence of sentences beyond context free [7], [28]. In this regard, to generate (store) such natural languages the grammar G which generates L should be unambiguous and at the same time it should be minimal in terms of measures. In practical, such a minimal unambiguous system will not be there for all languages. Under these, circumstances a necessary trade-off needs to be studied between the (descriptional) complexity measures and ambiguity.

To prove why such a trade-off is very important in natural languages, lets consider the following sentence, *They are hunting dogs*. The sentence is syntactically correct, where as the sentence is having semantic ambiguity, as it can be elucidated in a different manner. The different interpretation of the above mentioned sentence can be: *Whether any group is hunting for dogs?* or *Whether the category of dogs belongs to the hunting type* or *Whether the phrase hunting dogs* refers to a *music band* or a *basket ball team* or a *secret code*. In fact, the right phrases of the sentence are *They are*, *They are hunting*, *They are dogs*, *They are hunting dogs*. Assume that, we want to construct an insertion system which generates the above sentence. As there is no concept of non-terminals(variables) in insertion system, it can be called as pure grammars. Since the insertion system is a pure grammar, every derivation step should represent a correct phrase, the correct phrases are *They are*, *They are hunting*, *They are dogs*, *They are hunting dogs*. Consider, '*They are*' is an axiom and the insertion rules are of the form: (*They are*, *dogs*, λ) and (*They are*, *hunting*, λ).

By using the above axiom and the insertion rules, the derivations can be of the forms (the underlined words indicates the inserted string): (1) *They are* $\Longrightarrow$ *They are* $\underline{\textit{dogs}}$ $\Longrightarrow$ *They are* $\underline{\textit{hunting}}$ *dogs*, which gives all the three correct phrases. (2) *They are* $\Longrightarrow$ *They are* $\underline{\textit{hunting}}$ $\Longrightarrow$ *They are* $\underline{\textit{dogs}}$ *hunting*, which is not a correct phrase. So, with the above insertion rules all the correct phrases cannot be generated. However, if we consider three insertion rules (*They are*, *dogs*, λ), (*They are*, *hunting*, λ), (*They are*, *hunting*, *dogs*) all the three correct phrases can be derived from the axiom or else using different insertion rules we may get all correct phrases of the sentence, but the number of insertion rules will be more. So, to derive the above sentence, we need three insertion rules.

Such sentences can be stored compactly if there exists an unambiguous system which generates it, but may happen to be not minimal with respect to measure(s). As insertion systems is found to be one of the prominent (rewriting) grammar mechanisms, the system can be recognized to be one of the fit (rewriting) mechanisms to generate natural languages [30]. The above example clearly shows that the sentence can be generated by an unambiguous system but not minimal in terms of components used to iterate the sentence. The above case study explicit the importance of studying the trade-off in natural languages.

B. APPLICATION OF THE ANALYZED TRADE-OFF IN MODELLING OF BIO-MOLECULAR STRUCTURES

In computational biology there are lot of research problems needs to be addressed based on the gene sequence such as gene structure prediction, gene sequence alignment, bio-molecular modelling, construction of phylogenetic trees. Such gene structure prediction, bio-molecular modelling problems are effectuated by progressing with relevant pattern matching algorithms. The above discussed computational biological problems are somewhat akin to investigating the structural frameworks in computational linguistics. The gene structure prediction, bio-molecular modelling problems can be handled in an effective and succinct manner, if there exists a unique grammar model/system which generates/models it.

To model or predict the structures, first, it should be expressed as a gene sequence. Such sequences can be visualized as strings formed over the four basic chemical symbols a, t, g and c (Σ_{DNA}). The complementary of the above four chemical symbols is given as $\bar{a} = t$, $\bar{g} = c$, $\bar{t} = a$, $\bar{c} = g$. As the bio-molecular structures can be expressed in terms of (gene) sequences it has kindled the researchers to study the connection and application of formal language theory and computational biology [42]. In addition to that, the genetic structural descriptions that are found in the bio-molecular structures has some coherence in natural language constructs such as triple agreements: $L_{ta} = \{a^n b^n c^n \mid n \geq 1\}$, quadruple agreements: $L_{qa} = \{a^n b^n c^n d^n \mid n \geq 1\}$, crossed dependencies: $L_{cd} = \{a^n b^m c^n d^m \mid n, m \geq 1\}$, copy language: $L_{cp} = \{ww \mid w \in \{a, b\}^*\}$. More precisely, L_{ta} and L_{qa} resembles triple and quadruple helix structure. Likewise, L_{cd} and L_{cp} has some pertinence with pseudoknot and attenuator structures respectively [41], [43]. For modelling of the bio-molecular structures that occurs at intramolecular, intermolecular and RNA secondary structures, we refer to [17], [25], [26], [27], [29], [44].

Before we discuss about the application of the trade-off in modelling of the bio-molecular structures, first, we will show that insertion system is capable of modelling some of the bio-molecular structures. Consider the following bio-molecular structures like *hairpin, stem and loop, orthodox*. The language description and modelling of the above structures by using insertion system are given in the following lemmas. In the forthcoming lemmas $y \in \Sigma_{DNA}$, the counterpart of y is y'. $\bar{w}^R$, $\bar{u}^R$ is the complementary reversal of the string w,u respectively.

Lemma 1: The hairpin language $L_{hp} = \{w = \bar{w}^R \mid w \in \Sigma^*_{DNA}\}$ can be spawned by the insertion system $\gamma_{hp} = (\{y, y'\}, \{\lambda, y'\}, \{(y, y', y')\})$.

Lemma 2: The stem and loop language $L_{sl} = \{uv\bar{u}^R \mid u, v \in \Sigma^*_{DNA}\}$ can be achieved by insertion system $\gamma_{sl} = (\{y, y'\}, \{\lambda, yy'\}, \{(y, yy', y'), (y, y, y')\})$.

A string w is said to be orthodox over Σ'_{DNA}(*complementary alphabet*) iff it fulfills the following conditions (i) it should be an empty string λ, or (2) the string obtained by the insertion of yy' anywhere in an orthodox string. A language which contains only orthodox strings is called orthodox language L_{od}.

Lemma 3: The orthodox language L_{od} can be spawned by the insertion system $\gamma_{od} = (\{y, b'\}, \{\lambda\}, \{(\lambda, yy', \lambda)\})$.

1) AMBIGUITY ISSUES IN ORTHODOX LANGUAGE L_{od}

In this subsection, we will discuss about the ambiguity issues in the orthodox language L_{od}. In the derivations/descriptions/sequence the underlined string denotes the inserted gene sequence and $^{\downarrow}$ denotes the position where the gene sequence is to be inserted.

Case 1: Consider the string $cgatatgccg \in L_{od}$,which can be obtained from two different axioms.

$$\textit{Derivation } 1 : cg^{\downarrow} \Longrightarrow cg\underline{at}^{\downarrow} \Longrightarrow cgat\underline{at}^{\downarrow} \Longrightarrow cgatat\underline{gc}^{\downarrow} \Longrightarrow cgatatgc\underline{cg}.$$

$$\textit{Derivation } 2 :^{\downarrow} at \Longrightarrow \underline{cg}at^{\downarrow} \Longrightarrow cgat\underline{gc}^{\downarrow} \Longrightarrow cgat^{\downarrow}gc\underline{cg} \Longrightarrow cgat\underline{at}gccg.$$

In both the derivations, the same sequence $cgatatgccg$ is derived from two different axioms cg and at. Therefore, the system γ_{od} evinces 0-ambiguous.

Case 2: Consider an orthodox string $cgtagccgat \in L_{od}$, which can be obtained by two different ordered CS:

$$\textit{Ordered CS1} :^{\downarrow} \lambda \Longrightarrow \underline{ta}^{\downarrow} \Longrightarrow ta\underline{gc}^{\downarrow} \Longrightarrow tagc\underline{cg}^{\downarrow} \Longrightarrow^{\downarrow} tagccg\underline{at} \Longrightarrow \underline{cg}tagccgat.$$

$$\textit{Ordered CS2} :^{\downarrow} \lambda \Longrightarrow^{\downarrow} \underline{ta} \Longrightarrow \underline{cg}ta^{\downarrow} \Longrightarrow cgta^{\downarrow}\underline{at} \Longrightarrow cgta\underline{gc}^{\downarrow}at \Longrightarrow cgtagc\underline{cg}at.$$

In CS1, the order of gene sequence used by the insertion rules are ta, gc, cg, at, cg, whereas in CS2, the order of gene sequence used by the insertion rule are ta, cg, at, gc, cg. Thus, the gene sequence $cgtagccgat$ can be derived by two different ordered CS. Therefore, the system γ_{od} evinces 1-ambiguous also.

Case 3: Consider the string $atcgcgta \in L_{od}$, which can be derived in two different descriptions by γ_{od} which are given below:

$$\textit{Description } 1 : cg^{\downarrow} \Longrightarrow cg\underline{cg}^{\downarrow} \Longrightarrow^{\downarrow} cgcg\underline{ta} \Longrightarrow \underline{at}cgcgta.$$

$$\textit{Description } 2 :^{\downarrow} cg \Longrightarrow \underline{at}cg^{\downarrow} \Longrightarrow atcg^{\downarrow}\underline{ta} \Longrightarrow atcg\underline{cg}ta.$$

In both the descriptions the axioms are same cg and the contexts used in the insertion rules (λ, λ) are also same, but the position where the inserted gene sequence yy' are different. Therefore, the system γ_{od} is 5-ambiguous also.

The above example shows a clear evidence on the existence of different levels of ambiguity for the same language L_{od} on different gene sequences. In addition to that, the above (ambiguity) example reveals that analysis of the ambiguity issues in gene sequences has to be carried out with utmost care because ambiguity issues plays a pivot role in some of the computational biology problems such as protein sequence analysis, parallel gene recognition, prediction of gene locations. For more practical applications on the importance of ambiguity in gene sequences, we refer to [1], [2], [3], [4]. The evolutionary relationship among the various biological species can be depicted in terms of trees. The trees can be derived based on the differences and similarities among the species. Such trees are called as phylogenetic trees [45]. The phylogenetic trees plays an important role in DNA/Protein sequence divergence problem [8].

The axiom, intermediates sequence and the final sequence to be generated can be represented as a tree. If the intermediate gene sequences are different then we will have more than one phylogenetic trees for the same gene sequence. Such a study on the different intermediate sequences will help us to study more on the inheritance properties. The following example of a phylogenetic tree will give a better understanding on the ambiguity. One such phylogenetic tree is shown

TABLE 5. Obtained trade-off results.

$Unambiguous$ & $M2$ $\rightarrow$ ------- $Ambiguous$ & $M1$ $\downarrow$	Ax MAx TAx	$MLen-$ $InsStr$	$mLen-$ $InsStr$	$MLen-$ $LCon$	$MLen-$ $RCon$	$mLen-$ $LCon$	$mLen-$ $RCon$	$TLen-$ $LCon$	$TLen-$ $RCon$	$TLen-$ $LCon+$ $InsStr$	$TLen-$ $RCon+$ $InsStr$	$MLen-$ $LCon+$ $InsStr$	$MLen-$ $RCon+$ $InsStr$	$TINS$ $-Str$
Ax MAx TAx	$\star$			L_4-4A L_6-2A L_8-0A	L_6-2A L_8-0A L_9-0A		L_9-0A	L_4-4A L_5-4A L_6-2A L_8-0A	L_6-2A L_8-0A	L_6-2A	L_6-2A L_9-0A		L_9-0A	L_4-4A
$MLen-InsStr$		$\star$		L_8-0A	L_8-0A			L_8-0A	L_8-0A					
$mLen-InsStr$			$\star$	L_8-0A	L_8-0A L_9-0A		L_9-0A	L_8-0A	L_8-0A		L_9-0A		L_9-0A	
$MLen-LCon$	L_1-5A			$\star$										
$MLen-RCon$	L_3-5A			L_3-5A	$\star$			L_3-5A		L_3-5A				
$mLen-LCon$						$\star$		L_5-4A						
$mLen-RCon$	L_3-5A			L_3-5A			$\star$	L_3-5A		L_3-5A				
$TLen-LCon$	L_1-5A							$\star$	L_7-2A		L_7-2A			
$TLen-RCon$	L_3-5A			L_3-5A				L_3-5A	$\star$	L_3-5A				
$TLen-LCon+$ $InsStr$	L_1-5A								L_7-2A	$\star$	L_7-2A			
$TLen-RCon+$ $InsStr$	L_3-5A			L_3-5A				L_3-5A		L_3-5A	$\star$			
$MLen-LCon+$ $InsStr$	L_1-5A							L_5-4A				$\star$		
$MLen-RCon+$ $InsStr$	L_3-5A			L_3-5A				L_3-5A		L_3-5A			$\star$	
$TINS-Str$	L_3-5A			$L_2^{'}-5A$ L_3-5A L_8-0A	L_2-5A L_8-0A	$L_2^{'}-5A$	L_2-5A	$L_2^{'}-5A$ L_3-5A L_8-0A	L_2-5A L_8-0A	L_3-5A				$\star$

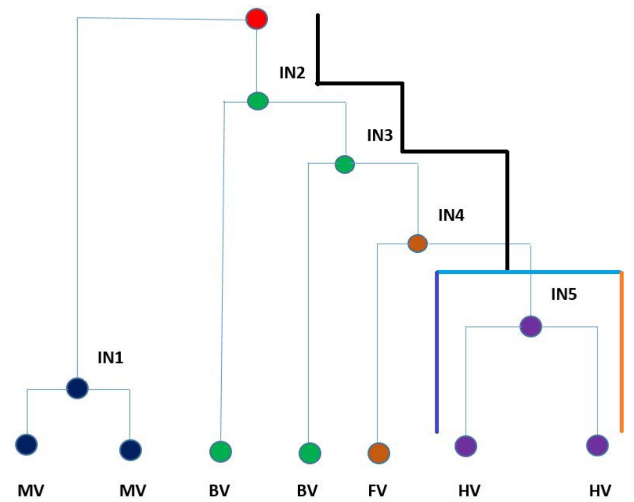

FIGURE 1. Interpretation of ambiguity in Phylogenetic tree.

TABLE 6. Comparative Study.

Paper	Grammar Formalism	Computational Completeness	Trade-off	Applications
Paun, Gh (1997)	Insertion Systems	Y	N	N
Paun, Gh., Rozernberg, G., Salomaa, A (1998)	Insertion-Deletion Systems	Y	N	N
Krishna, S.N. et.al (2001)	Insertion-Deletion P Systems	Y	N	N
Lakshmanan, K et.al (2011)	Matrix Insertion-Deletion Systems	N	N	N
Lakshmanan, K et.al (2011)	Insertion-Deletion Systems	N	Y	N
Fernau, K et.al (2017)	Universal Matrix Insertion Systems	Y	N	N
Fernau, K et.al (2017)	Graph Controlled Insertion-Deletion Systems	Y	N	N
Fernau, K et.al (2019)	Path-structured graph-controlled insertion-deletion systems	Y	N	N
Fernau, K et.al (2019)	Semi-conditional Insertion-Deletion Systems	Y	N	N
Fernau, K et.al (2019)	Path Controlled Insertion-Deletion Systems	Y	N	N
This paper	Insertion System	N	Y	Y

in Figure.1. In Figure.1, IN represents intermediate node, MV represents Monkey Virus, BV represents Bird Virus, FV represents Fish Virus, HV represents Human Virus. To reach the Human Virus leaf node, the path can be explored as: $RootNode \rightarrow IN2 \rightarrow IN3 \rightarrow IN4 \rightarrow IN5$. Starting from the root node there is a unique path up to the intermediate node IN5, whereas, after reaching the purple color intermediate node IN5, there exists two paths. One path will be Voilet line from $IN5 \rightarrow HV$ and another path will be Yellow line from $IN5 \rightarrow HV$. So, the Human Virus node can be reached by two different paths from the root node. The above scenario, clearly shows that a different perspective can be given in the visualization of ambiguity in phylogenetic trees. On the other hand, while predicting the gene structure, we need an optimal system and at the same time the system which generates/models the bio-molecular structure should be unambiguous. Consider the system $\gamma_{od} = (\{y, b'\}, \{\lambda\}, \{(\lambda, yy', \lambda)\})$ which generates the L_{od}. One insertion rule is enough to generate all the strings in L_{od}. The system γ_{od} is minimal $\{Ax, MLen - LCon, MLen - RCon, mLen - LCon, mLen - RCon, TLen-LCon, TLen-TCon\}$. The language L_{od} can be generated by an unambiguous system but definitely the unambiguous system will not be minimal in the above mentioned measures. This example shows the importance and application of the trade-off study between complexity measures and ambiguity levels in modelling of the bio-molecular structures.

VI. COMPARATIVE STUDY

In this section, we discuss about the comparative study of trade-off results obtained for the insertion systems and its applications in natural languages, modelling of bio-molecular structures. Table.6 shows the comparative study of the proposed results and applications with other relevant grammar models. From the comparative study, it has a clear evidence, that the insertion systems, insertion-deletion systems, variants of insertion deletion systems are mainly motivated towards reducing the weights in simulating the recursively enumerable languages by means of suitable normal forms where as, in this paper, we have defined some new descriptional complexity measures, analyzed the trade-off between ambiguity levels and descriptional complexity measures. In addition to that, we have discussed about the application of the analyzed trade-off which was missing in the various research work carried out on insertion systems.

VII. CONCLUSION

In this paper, we defined twelve new descriptional complexity measures namely $MLen - InsStr$, $mLen - InsStr$, $MLen - LCon$, $MLen - RCon$, $mLen - LCon$, $mLen - RCon$, $TLen - LCon$, $TLen - RCon$, $TLen - LCon + InsStr$, $TLen - RCon + InsStr$, $MLen - LCon + InsStr$, $MLen - RCon + InsStr$ based on the components used in the derivations. Later, we discussed the trade-off between the newly defined ambiguity levels and measures in insertion systems by showing that there exists pseudo inherently ambiguous insertion languages which can be generated by an ambiguous system that are minimal in M_1 and unambiguous if they are minimal in M_2. Finally, we have studied the application of the investigated trade-off in natural languages and modelling of bio-molecular structures. Analyzing the trade-off between measures and ambiguity levels which are not considered in this paper is left as a future work. More, precisely it would be interesting such a trade-off results can be obtained for the ambiguity levels 1 and 3. As insertion systems can be recognized as a good model to generate some of the programming language constructs, analyzing the trade-off in programming languages would be an another line of future work.

ACKNOWLEDGMENT

An earlier version of this paper was presented at the International Conference (ICRTC-2022) held at SRMIST, New Delhi.

REFERENCES

[1] M. J. Bishop and C. J. Rawlings, *Nucleic Acid and Protein Sequence Analysis*. London, U.K.: Oxford Univ. Press, 1987.
[2] M. Borodovsky and J. McIninch, "Recognition of genes in DNA sequence with ambiguities," *Biosystems*, vol. 30, nos. 1–3, pp. 161–171, 1993.
[3] M. Borodovsky and J. McIninch, "Prediction of gene locations using DNA Markov chain models," in *Bioinformatics, Supercomputing and Complex Genome Analysis*, 1993, pp. 231–248.
[4] M. Borodovsky and J. McIninch, "GENMARK: Parallel gene recognition for both DNA strands," *Comput. Chem.*, vol. 17, no. 2, pp. 123–133, Jun. 1993.
[5] J. Bresnan, R. M. Kaplan, S. Peters, and S. Zaenen, "Cross-serial dependencies in Dutch," *Linguistic Inquiry*, vol. 13, no. 4, pp. 613–663, 1982.
[6] C. S. Calude and G. Paun, *Computing With Cells and Atoms an Introduction to Quantum, DNA and Membrane Computing*. London, U.K.: Taylor and Francis, 2001.
[7] N. Chomsky, "Formal properties of grammars," in *Handbook of Mathematical Psychology*, R. D. Luce, Ed. New York, NY, USA: Wiley, 1963, pp. 323–418.
[8] S. Choudhuri, *Bioinformatics for Beginners: Genes, Genomes, Molecular Evolution, Databases and Analytical Tools*. Amsterdam, The Netherlands: Elsevier, 2014.
[9] H. Fernau, L. Kuppusamy, and S. Verlan, "Universal matrix insertion grammars with small size," in *Proc. Int. Conf. Unconventional Comput. Natural Comput.* Springer, Cham, Jun. 2017, pp. 182–193.
[10] H. Fernau, L. Kuppusamy, and I. Raman, "Graph-controlled insertion-deletion systems generating language classes beyond linearity," in *Proc. Int. Conf. Descriptional Complex. Formal Syst.* Cham, Switzerland: Springer, Jul. 2017, pp. 128–139.
[11] H. Fernau, L. Kuppusamy, and I. Raman, "On path-controlled insertion–deletion systems," *Acta Inf.*, vol. 56, no. 1, pp. 35–59, Feb. 2019.
[12] H. Fernau, L. Kuppusamy, and I. Raman, "Computational completeness of simple semi-conditional insertion-deletion systems," in *Proc. Int. Conf. Unconventional Comput. Natural Comput.* Cham, Switzerland: Springer, Jun. 2018, pp. 86–100.
[13] H. Fernau, L. Kuppusamy, and I. Raman, "Computational completeness of path-structured graph-controlled insertion-deletion systems," in *Proc. Int. Conf. Implement. Appl. Automata*. Cham, Switzerland: Springer, Jun. 2017, pp. 89–100.
[14] D. Haussler, "Insertion Languages," *Inf. Sci.*, vol. 31, no. 1, pp. 77–89, 1983.
[15] J. Gruska, "Descriptional complexity of context-free languages," in *Mathematical Foundations of Computer Science*. Bratislava, Slovakia: High Tatras, 1973, pp. 71–84.
[16] D. Haussler, "Insertion languages," *Inf. Sci.*, vol. 31, no. 1, pp. 77–89, Oct. 1983.
[17] T. Head, "Formal language theory and DNA: An analysis of the generative capacity of specific recombinant behaviors," *Bull. Math. Biol.*, vol. 49, pp. 737–750, Nov. 1987.
[18] L. Ilie, "On ambiguity in internal contextual languages," in *Proc. 2nd Int. Conf. Math. Ling.*, C. Martin-Vide, Eds. Amsterdam, The Netherlands: John Benjamins, 1997, pp. 29–45.
[19] S. N. Krishna and R. Rama, "Insertion-deletion P systems," in *Proc. Int. Workshop DNA-Based Comput.* Berlin, Germany: Springer, 2001, pp. 362–370.
[20] K. Lakshmanan, "Incompatible measures of internal contextual grammars," in *Proc. DCFS*, C. Mereghetti, Eds. Como, Italy, 2005, pp. 253–260.
[21] K. Lakshmanan, "A note on ambiguity of internal contextual grammars," *Theor. Comput. Sci.*, vol. 369, pp. 436–441, Dec. 2006.
[22] K. Lakshmanan, M. Anand, and K. Krithivasan, "On the trade-off between ambiguity and measures in internal contextual grammars," in *Proc. DCFS*, C. Câmpeanu and G. Pighizinni, Eds., 2008, pp. 216–223.
[23] L. Kuppusamy, A. Mahendran, and K. Krithivasan, "On the ambiguity of insertion systems," *Int. J. Found. Comput. Sci.*, vol. 22, no. 7, pp. 1747–1758, Nov. 2011.
[24] K. Lakshmanan, M. Anand, K. Krithivasan, and K. Mohammed, "On the study of ambiguity and the trade-off between measures and ambiguity in insertion–deletion languages," *Nano Commun. Netw.*, vol. 2, nos. 2–3, pp. 106–118, 2011.
[25] K. Lakshmanan, M. Anand, and S. N. Krishna, "Matrix insertion-deletion systems for bio-molecular structures," in *Proc. ICDCIT* in Lecture Notes in Computer Science, vol. 6536, R. Natarajan and A. Ojo. Eds., 2011, pp. 301–311.
[26] K. Lakshmanan, M. Anand, and E. V. Clergerie, "Modelling intermolecular structures and defining ambiguity in gene sequences using matrix insertion-deletion systems," in *Biology, Computation and Linguistics, New Interdisciplinary Paradigms*, vol. 228. Amsterdam, The Netherlands IOS Press, 2011, pp. 71–85.
[27] L. Kuppusamy and A. Mahendran, "Modelling DNA and RNA secondary structures using matrix insertion–deletion systems," *Int. J. Appl. Math. Comput. Sci.*, vol. 26, no. 1, pp. 245–258, Mar. 2016.
[28] D. T. Langendoen and P. M. Postal, *The Vastness of Natural Language*. Oxford, U.K.: Blackwell, 1984.
[29] A. Mahendran and L. Kuppusamy, "Formal language representation and modelling structures underlying RNA folding process," in *Theoretical Computer Science and Discrete Mathematics* (Lecture Notes in Computer Science), vol. 10398, S. Arumugam, J. Bagga, L. Beineke, and B. Panda, Eds. Cham, Switzerland: Springer, 2017, pp. 20–29.
[30] S. Marcus, "Contextual grammars," *Rev. Roum. Pures. Appl.*, vol. 14, pp. 1525–1534, 1969.
[31] M. Margenstern, G. Paun, Y. Rogozhin, and S. Verlan, "Context-free insertion–deletion systems," *Theor. Comput. Sci.*, vol. 330, no. 2, pp. 339–348, (2005).
[32] C. Martin-Vide, J. Miguel-Verges, A. G. Paun, and A. Salomaa, "Attempting to define the ambiguity in internal contextual languages," in *Proc. 2nd Int. Conf. Math. Ling.*, C. Martin-Vide, Ed. Amsterdam, The Netherlands: John Benjamins, 1997, pp. 59–81.
[33] G. Paun, *Contextual Grammars*. Bucuresti, Romania: Publishing House of the Romanian Academy of Sciences, 1982.
[34] G. Paun, *Marcus Contextual Grammars*. Norwell, MA, USA: Kluwer, 1997.
[35] G. Paun, G. Rozenberg, and A. Salomaa, *DNA Computing, New Computing Paradigms*. Cham, Switzerland: Springer, 1998.
[36] G. Paun, *Membrane Computing An Introduction*. Cham, Switzerland: Springer, 2002.
[37] G. Paun, "On the compl of contextual grammars with choice," *Stud. Cerc. Matem.*, vol. 27, pp. 559–569, 1975.
[38] G. Paun, "Further remarks on the syntactical complexity of Marcus contextual languages," *Ann. Univ. Buc., Ser. Matem.-Inform.*, pp. 72–82, 1991.
[39] G. H. Pullum, "On two recent attempts to show that english is not a context-free language," *Comput. Linguistics*, vol. 10, nos. 3–4, pp. 182–186, 1980.
[40] G. Rozenberg and A. Salomaa, *Handbook of Formal Languages, Word, Language, Grammar*, vol. 1. Cham, Switzerland: Springer, (1997).
[41] E. Rivas and S. R. Eddy, "The language of RNA: A formal grammar that includes pseudoknots," *Bioinformatics*, vol. 16, no. 4, pp. 334–340, Apr. 2000.
[42] D. B. Searls, "The linguistics of DNA," *Amer. Scientist*, vol. 80, no. 6, pp. 579–591, 1992.
[43] D. B. Searls, "The computational linguistics of biological sequences," in *Artificial Intelligence and Molecular Biology*, L. Hunter. Menlo Park, CA, USA: AAAI Press, 1993, pp. 47–120.
[44] D. B. Searls, "Formal grammars for intermolecular structure," in *Proc. 1st Int. Symp. Intell. Neural Biol. Syst. (INBS)*, May 1995, pp. 30–37.
[45] J. C. Setubal and J. Meidanis, *Introduction to Computational Molecular Biology. Brooks*, vol. 5, no. 9. Pacific Grove, CA, USA: Cole Publishing Company, 1997, p. 18.

ANAND MAHENDRAN received the B.E. degree in computer science and engineering from Madras University, India, in 2003, the M.E. degree in computer science and engineering from Anna University, India, in 2005, and the Ph.D. degree in computer science and engineering from the Vellore Institute of Technology (VIT), Vellore, India, in 2012. He worked as a Postdoctoral Research Fellow with the Laboratory of Theoretical Computer Science, National Research University, Higher School of Economics (HSE), Moscow, Russia. He is currently an Associate Professor (Senior) with the School of Computer Science and Engineering, VIT. He has published more than 50 papers in international journals and refereed international conferences. His research interests include formal language and automata theory, and bio-inspired computing models.

KUMAR KANNAN received the bachelor's degree in computer science and engineering (CSE) from Madras University, in 1998, the master's degree in CSE from Pondicherry University, in 2004, and the Ph.D. degree in CSE from the Vellore Institute of Technology, Vellore, in 2016. From 1998 to 2001, he worked as a Programmer/Software Engineer at Computer Access Ltd., and Bhari Information Technology, and later he worked as a Lecturer by passion. Since 2005, he has been an Associate Professor with the VIT. He is currently a Researcher and an Engineer who has been working in software engineering, recommender patterns, context-aware patterns, and machine learning applications. He has presented several works in the area of software patterns and machine learning, and published various articles, and book chapters in international and national level.

MOHAMED HAMADA (Senior Member, IEEE) received the Ph.D. degree from the University of Tsukuba, Japan, under the scholarship from the Japanese Government (MEXT). He got a Japan International Cooperation Agency (JICA) Fellowship for six months. He is currently a Senior Associate Professor at The University of Aizu, Aizuwakamatsu, Fukushima, Japan. He is a Regular Visiting Professor at Fatih University, Istanbul, Turkey, and the African University of Science and Technology, Abuja, Nigeria. He leaded several funded research projects and supervised several graduate (M.Sc. and Ph.D.) and undergraduate students. He edited three books and has more than 100 papers in major international journals and conferences published by major publishers, such as IEEE, ACM, Elsevier, and Springer. His research interests include artificial intelligence and learning technologies. He is also interested in smart devices (such as smartphones and tablets) applications development and innovation. He is a member of the editorial board of several international journals and a program committee member of several international conferences. He is a member of the IEEE Technical Committee on Multimedia, and the IEEE Technical Committee on Learning Technologies. He is a Senior Member of IEEE and ACM Computer Societies.

MANUEL MAZZARA received the Ph.D. degree in computing science from the University of Bologna, Italy. He is currently a Professor of computer science at Innopolis University, Russia, with a research background in software engineering, service-oriented architecture, concurrency theory, formal methods, and software verification. He is also the Director of the Institute of Software Development and Engineering and the Head of the International Cooperation Office at Innopolis University. He published many relevant and highly-cited papers, in particular in the field of service engineering and software architectures. He has collaborated with European and U.S. industries, plus governmental and inter-governmental organizations, such as the United Nations, always at the edge between science and software production. The work conducted by Dr. Manuel Mazzara and his team in recent years focuses on the development of theories, methods, tools, and programs covering the two major aspects of software engineering: the process side, related to how we develop software, and the product side, concerning the results of this process.