Nano Communication Networks 2 (2011) 106–118

Contents lists available at SciVerse ScienceDirect

Nano Communication Networks

journal homepage: www.elsevier.com/locate/nanocomnet

On the study of ambiguity and the trade-off between measures and ambiguity in insertion–deletion languages☆

Lakshmanan Kuppusamy [a,*], Anand Mahendran [a], Kamala Krithivasan [b], Khalid Mohammed [a]

[a] School of Computing Science and Engineering, VIT University, Vellore - 632 014, India

[b] Department of Computer Science and Engineering, IIT Madras, Chennai - 600 036, India

ARTICLE INFO

Article history:
Received 18 April 2011
Received in revised form 28 May 2011
Accepted 9 June 2011
Available online 2 July 2011

Keywords:
Insertion–deletion systems
Inherently ambiguous languages
Unambiguous system
Undecidability
Bio-molecular structures
Descriptional complexity measures

ABSTRACT

Gene insertion and deletion are the operations that occur commonly in DNA processing and RNA editing. Based on these operations, a computing model has been formulated in formal language theory known as *insertion–deletion* systems. In this paper we study about ambiguity issues of these systems. First, we define six levels of ambiguity for insertion–deletion systems that are based on the components used in the derivation such as *axiom*, *contexts* and *strings*. Next, we show that there are inherently i-ambiguous insertion–deletion languages which are j-unambiguous for the combinations $(i, j) \in \{(5, 4), (4, 3), (4, 2), (3, 1), (2, 1), (1, 0), (0, 1)\}$. As an application, we discuss with an example that how some of these ambiguity levels can be interpreted in gene sequences. Further, we prove an important result that the ambiguity problem of insertion–deletion systems is undecidable. Then, we define six new measures for insertion–deletion systems based on used contexts and strings. Finally, we analyze the trade-off between ambiguity levels and measures. We note that there are languages which are inherently i-ambiguous (for $i = 5, 4, 2, 0$) when a measure M is minimal for the languages but they are i-unambiguous otherwise.

1. Introduction

In the past few decades, natural computing is an area being pursued with great interest. It includes evolutionary computing [2] and biologically inspired computing such as DNA computing [11] and membrane computing [9]. The developments in DNA computing inspired the study of new theoretical models in formal language theory known as *sticker systems, splicing systems, Watson–Crick automata* and *insertion–deletion systems* [1,11]. Insertion and deletion operations were studied first in [4,5] and based on these operations, insertion–deletion systems were introduced in [6]. Informally, the insertion and deletion operations in an insertion–deletion system is defined as follows: if a string β is inserted between two parts w_1 and w_2 of a string w_1w_2 to get $w_1\beta w_2$, we call the operation as insertion, whereas if a substring α is deleted from a string $w_1\alpha w_2$ to get w_1w_2, we call the operation as deletion.

Insertion–deletion operations have relevances to some phenomena in human genetics. In Fig. 1 we show how the insertion–deletion systems are applied in the field of genetics. Consider a single strand DNA sequence $S_1 = xuvyz$, where x, u, v, y, z are all strings. Add a single stranded DNA sequence $u'w'v'$ to the sequence $xuvyz$, where u', v' are the Watson–Crick complements of the strings u, v and w' is the complement of some string w (see Fig. 1(a)). First, annealing will take place such that u' will stick to u and v' to v, thus we obtain the scenario as in Fig. 1(b). Next a cut by a restriction enzyme to the

☆ This paper is the revised and extended version of the paper that appeared in the proceedings of Bionetics-2010 held in Boston, USA, December 2010.

* Corresponding author. Tel.: +91 8903988685.
E-mail addresses: klakshma@vit.ac.in, klachu@gmail.com (L. Kuppusamy), manand@vit.ac.in (A. Mahendran), kamala@iitm.ac.in (K. Krithivasan), mkhalid@vit.ac.in (K. Mohammed).

1878-7789/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.nancom.2011.06.001

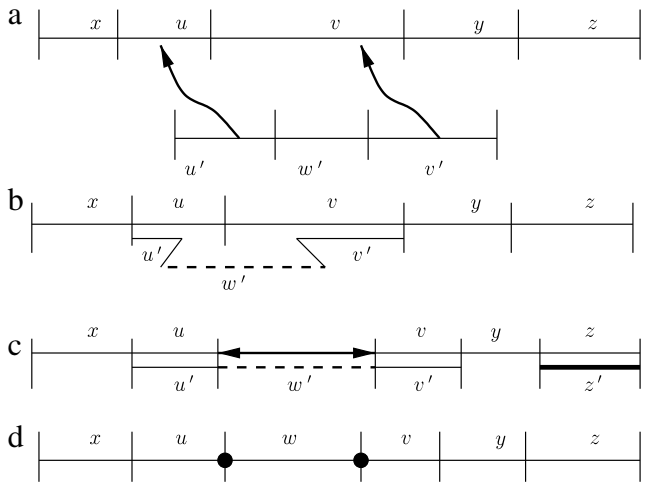

Fig. 1. Insertion by annealing.

double stranded DNA sequence uv will be done (shown in Fig. 1(c) where the cut is denoted by a thick double directed arrow $\longleftrightarrow$). By adding a primer z', we obtain another double stranded sequence. Finally, by melting the double stranded sequence, the two strands will be separated, thus we obtain two strings in which one is of the form $S_2 = xuwvyz$ (refer Fig. 1(d)). Thus, the string S_2 obtained from S_1 shows the use of insertion operation in DNA sequences.

Ambiguity is considered as one of the fundamental problems in linguistics. A grammar is said to be ambiguous, if there exists more than one distinct derivation of words in the generated language. As we have seen above that the insertion–deletion system can be applied theoretically in DNA processing, the ambiguity in DNA processing (which uses the insertion–deletion system) may happen in the following manner. Let W_1W_2 be a DNA strand and suppose we want to insert $W_3W_4W_5$ between W_1 and W_2 to obtain another DNA strand $W_1W_3W_4W_5W_2$. This can be done first by inserting W_3 between W_1 and W_2, followed by inserting W_4 between W_3 and W_2, followed by inserting W_5 between W_4 and W_2. The other sequence would be first by inserting W_5 between W_1 and W_2, followed by inserting W_4 between W_1 and W_5, followed by inserting W_3 between W_1 and W_4. The two distinct derivations of above are given as: (1) $W_1W_2 \Longrightarrow W_1\underline{W_3}W_2 \Longrightarrow W_1W_3\underline{W_4}W_2 \Longrightarrow W_1W_3W_4\underline{W_5}W_2$ (2) $W_1\underline{W_2} \Longrightarrow W_1\underline{W_5}\underline{W_2} \Longrightarrow W_1\underline{W_4}W_5W_2 \Longrightarrow W_1\underline{W_3}W_4W_5W_2$ (the underlined string denotes the inserted string). This shows that ambiguity in gene sequences is also possible in the sense that starting from one sequence we are able to get another sequence in more than one way such that the intermediate sequences are different. This motivates us to define formally the ambiguity for insertion–deletion systems.

Fig. 1 shows the applicability of insertion–deletion systems in gene sequences, storing of such sequences of large data can be minimized if the corresponding insertion–deletion system can be identified by means of minimal measures. Such systems are called minimal systems (with respect to the measure). In [10], the following measures are defined for insertion systems: *Ax*, *MAx*, *TAx*, *Prod*, *Symbol*. *Ax* denotes the number of axioms, *MAx* denotes the maximum length of an axiom, *TAx* denotes total length of all axioms, *Prod* denotes the number of insertion rules and *Symbol* denotes the number of symbols in the insertion rules. As the insertion–deletion systems are extended models of insertion systems, the measures *Ax*, *MAx*, *TAx*, *Prod* are even applicable to insertion–deletion systems. In this paper, we introduce a few more descriptional complexity measures for insertion–deletion systems: *TLength-Con* (total length of contexts used in insertion and deletion rules), *TLength-Str* (total length of the strings to be inserted plus the total length of the strings to be deleted), *TINS-StrCon* (total length of the contexts used in insertion rules plus the total length of the strings to be inserted), *TDEL-StrCon* (total length of the contexts used in deletion rules plus the total length of the strings to be deleted), *TINS-Str* (total length of the strings to be inserted), *TDEL-Str* (total length of the strings to be deleted). It is preferable that a minimal system is also unambiguous as it will help to predict the gene structure without any ambiguity. Unfortunately, such a system may not exist for all languages and in such cases, a trade-off between ambiguity and measures need to be analyzed. In this paper, we identify some languages for which the trade-off has to be made. We notice that all the minimal systems for some languages turn to be ambiguous and when the minimal condition is relaxed, there exist unambiguous systems for such languages. We call these languages as pseudo inherently ambiguous languages.

In this paper, we extend the work carried out in [7]. In [7], six levels of ambiguity of insertion–deletion systems have been defined and it is shown that there are inherently i-ambiguous insertion–deletion languages which are j-unambiguous for the combinations $(i, j) \in \{(5, 4), (4, 2), (3, 1), (3, 2), (2, 1), (0, 1)\}$. Also, three new measures *TLength-Con*, *TLength-Ins*, *TLength-Del* are introduced and we discussed the trade-off between ambiguity and measures of insertion–deletion languages. More specifically, it is shown that if there are languages for which a minimal measure M is chosen, then all the corresponding minimal grammar systems are ambiguous and