International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8, Issue-1, May 2019

A Comprehensive Study of Insertion-Deletion System in DNA Computing

V. Sudha, K.S. Easwarakumar

Abstract*: Problem-solving is an evergreen area due to the needs of the fast-growing electronics world. Thus, it is important to find an efficient solution for any given problem. The progress does not stop by finding an efficient solution, but it continues by proving the solution correctness. Thus, it is important to prove the correctness of the proposed solution. Either practical or theoretical justification can be used for proving the correctness of a solution. In a molecular computing field like DNA computing, proving the correctness of a solution is a time consuming and expensive task. Hence, in DNA computing, using insertion and deletion operations a number of theoretical models are proposed. These insertion and deletion operations are practically realized using well-known biological operations. In literature, it is also proved that these insertion-deletion systems are Turing equivalent. Also, the closure properties of these systems are studied. In this regard, it is important to study the existing insertion-deletion system in the literature. This study may be useful to define a new system in the future or to solve new problems using the existing insertion-deletion system. In this paper, a survey on the existing insertion-deletion system is done.

***Index Terms*: context-free, context-sensitive, insertion operation, deletion operation.**

I. INTRODUCTION

In later days, computation became fast and efficient with the help of computers. Now the computer has become a part of human life. The processing speed of the computer is improved to meet the growing needs of the users. Due to the inbuilt constraints of the electronic equipment's, these improvements are possible only to a certain extent. To overcome this disadvantage, a number of other types of computing techniques such as molecular computing, membrane computing, etc., are proposed in the literature.

One of such computing technique is DNA computing. DNA computing is molecular computing that makes use of the parallelism and storage capacity of the DNA for its computing. Due to the high cost involved in the experiments involving DNA, most of the problems are theoretically solved. All these theoretical models involve insertion and deletion operations. These insertion-deletion systems as powerful as that of the Turing Machine. Hence, to solve a problem in DNA computing either the operations of the existing system can be used or a new insertion-deletion system can be proposed.

Turing machine is a computational model with the highest computing power among all mathematical models. This model gives either yes or no as the output. A Turing Machine consists of finite control and a tape of infinite size. For a given instance, a Turing machine takes a state, an input symbol and makes a transition to another state by overwriting the input symbol on the tape and moving the tape head either left or right. It has the capability of solving all computational problems. In this regard, if we want to show the computational power of any proposed mathematical model, then it is enough to prove that it is equal to Turing Machine. In this paper, a survey is done on the existing insertion-deletion system along with their properties. The organization of the paper is as follows. Section II contains the discussion about the existing insertion-deletion system in the literature; Section III discusses the points inferred from the previous section; Conclusion and Future scope are given in Section IV.

Revised Manuscript Received on 30 May 2019.
***** Correspondence Author
V. Sudha, Computer Science and Engineering, Kumaraguru College of Technology, Coimbatore, India.
K. S. Easwarakumar, Computer Science and Engineering, Anna University, Chennai, India.

II. PROPOSED METHODOLOGY IN INSERTION-DELETION SYSTEM

This section contains the description of various insertion-deletion systems in DNA computing. All these insertion-deletion systems make use of the biological operations defined in [1].

Block diagram

Given a string, the insertion operation inserts a substring between the two contexts, while a deletion operation deletes the substring that lies between the given two contexts. Here, starting from the axiom, a set of strings (language) is derived by applying one or more insertion and deletion operations. The above process is described by the block diagram in Fig. 2.1.

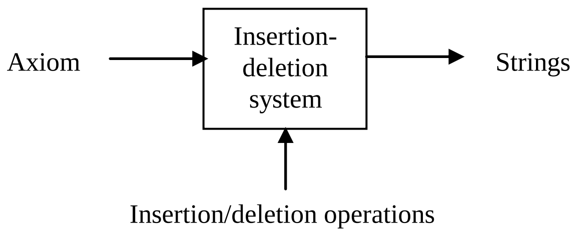

Fig. 2.1. Structure of insertion-deletion system

The context for an insertion operation can consist of left substring only, right substring only or it can contain both left and right substring.

A. Context-sensitive insertion-deletion system INSDEL system

In [2], a context-sensitive insertion-deletion model (INSDEL) is proposed based on the biological operation: PCR-site specific mutagenesis. In the proposed system, the insertion and deletion operations are double contextual. That is, a substring is inserted/deleted into/from a given string only if the given set of contexts exists in the given string. Let u and v be the context and x be the strings to be inserted. Then, applying the insertion operation $(u,x,v)_I$, the string *uv* becomes *uxv* after insertion. Similarly, applying the deletion operation $(u,x,v)_D$, the string *auxvb* becomes *auvb* after deletion. Here, if there is no context, then no insertion or deletion operation takes place. It means there is no change in the input string. Also, it proved that the language accepted by this proposed insertion-deletion system is Turing equivalent. The following algorithm describes the process of this system

```
While(context exists)
 {
   Apply the insertion and deletion operations
 }
```

B. Context-free insertion-deletion system

Margenstern et.al, in [3] proved that context-free insertion and deletion operation generates recursively enumerable language. In [4], the computational power of context-free insertion-deletion system is studied. It is shown that the system is universal computable if the length of the inserted and deleted string is at least 3.

In [5], verlan identified that context-free insertion-deletion system with context length at least three is universal computable. Also, in [6] he has discussed the recent developments in the insertion and deletion systems such as context-free insertion-deletion system, one-sided insertion-deletion systems, etc. The family of languages generated by the context-free insertion-deletion system is indicated by $INS^0_*DEL^0_*$. It is proved that this language is equivalent to the recursively enumerable language. It is also showed that the system with one or two axioms is Turing equivalent.

C. Circular insertion-deletion system

It is known that DNA exists in both linear and circular forms. In [7], a theoretical model for inserting a circular DNA strand into a linear DNA strand is proposed. To insert the circular strand, first, the circular strand is linearized by cutting the circular strand between some specified locations. Next, the linearized circular strand is inserted into the given context. Let c_1 and l_1 be the circular and the linear DNA strand. Let *(x, y)* and *(u, v)* be the site and context in the circular and linear DNA strand respectively. It is also proved that the proposed system is Turing equivalent.

D. *SInsDelP* System

A insertion-deletion system based on replicative transposition and PCR-site specific mutagenesis (*SInsDelP*) is proposed in [8]. Here insertion operation is based on a single context while deletion operation on double context. Given a string *utx*, the content *s* and the context *t*, then insertion operation performed using replicative transposition results in the string *utstx*. Note here that insertion operation results in duplication of contexts at either side of the content to be inserted. Deletion operation is performed using PCR-site specific mutagenesis which is double contextual in nature. It is the same as the deletion operation in the *INSDEL* system. It is proved that the regular, context-free, context-sensitive and recursively enumerable languages are closed under single-contextual insertion operation. Turing equivalence of the proposed system is also proved.

E. Insertion only system

Fujiko in [9] extended the insertion operation of the following form. If *uv* is the context and *x* is the string to be inserted (content) then after insertion, we get the string *uxv* where *x* be the content to be inserted between *u* and *v* and proposed picture-insertion system for generating two dimensional picture languages.

F. Graph controlled insertion-deletion system

A graph-controlled insertion-deletion system is proposed in [10]. In this proposed system, the application of the insertion and deletion operations depends on the current node of the graph. It is proved that the graph with four nodes is computable. In a graph-controlled insertion-deletion system, the instantaneous description is indicated by the pair (t, s) where t is the rule label and s is the current input string. It is also proved that the language generated by this graph-controlled insertion-deletion system and the language accepted by the Turing machine is equivalent.

G. Insertion-deletion system with one-sided context

In [11], Matveevici et.al proposed an insertion-deletion system with one-sided context. This proposed system is more or less similar to that of the context-free insertion and deletion i.e. here also there is no restriction on the number of times an operation is performed. But the context is one-sided (either left or right). This system is more or less same as that of the context-free insertion-deletion system. It is also shown that minimal deletion generates the universal language.

H. Restricted insertion-deletion system

In [12], Katsanyi proposed a restricted insertion-deletion system. The word restricted in the proposed system refers to the minimal use of symbols from the given alphabet. The proposed system generates the language using a homomorphism operation.

In this paper, it is proved that the traditional way of generating the language is the same as the language generated by the insertion-deletion operation over two letter alphabet. Closure properties of controlled insertion-deletion are studied in [13].

I. Tile-assembly model

DNA Self assembly is a model in which molecules self-assemble automatically to do computation. Each tile/DNA strand will have a glue that attaches automatically to generate the needed structure. The self-assembly model proposed by Winfree [14]. The self-assembly model consists of a set of monomers called tiles. All tiles will be of uniform in size which cannot be rotated. Each side of the tile has a unique label. All these tiles will be placed in the test tube and allowed to bond each other locally without external influence. Tiling is the arrangement of basic shapes that fits perfectly in a given infinite plane. The growth of the tiling system starts from the initial tile called seed tile. In the tiling system, tiles may fall off when the bonding is weak. This makes use of hybridization operation. The problem with the self-assembly model is that the algorithms are usually slow due to the low growth rate. When the temperature is raised to accelerate the process, the error rate increases exponentially and goes out of control. In DNA self-assembly, the required target is achieved using a single operation.

III. RESULT ANALYSIS

As a result of this study it is known that using either insertion or deletion or both of these operations, universal languages can be generated. Similarly to apply these operations either context may be needed or may not be needed. When context is applied, then we may provide some restriction in applying insertion and deletion operations.

IV. CONCLUSION AND FUTURE SCOPE

Thus a survey on the existing insertion-deletion system in DNA computing is done in this paper. The following points are inferred from the survey. Insertion and deletion operation in DNA computing is nothing but an insertion and deletion of a string within another string. Both these operations are realized using the well-defined biological operations. It is well understood that using these insertion and deletion operations we can generate language that is equivalent to the language of Turing machine.

REFERENCES

1. L. Kari, "DNA computing: Arrival of Biological Mathematics", The Mathematical Intelligencer, vol. 19, 1997, pp. 9–22.
2. L. Kari and P. Sosik, "On the weight of universal insertion grammars", Theoretical Computer Science, 2008, pp. 264–270.
3. M. Margenstern, G. Paun, Y. Rogozhin and S. Verlan, "Context-free insertion-deletion systems", Theoretical Computer Science, 2005, pp. 339–348.
4. A. Salomaa, Formal Languages, Academic Press, New York, 1973 .
5. S. Verlan, "On minimal context-free insertion-deletion systems", Journal of Automata, Languages and Combinatorics, vol. 12, 2007, pp. 317–328.
6. S. Verlan, "Recent Developments on Insertion-Deletion Systems", Computer Science Journal of Moldova, vol. 18, 2010, pp. 317–328.
7. M. Daley, L. Kari, G. Gloor, and R. Sironmoney, "Circular contextual insertions/deletions with applications to biomolecular computation", SPIRE, 1999, pp. 47–54.
8. A. Murugan, and K. S. Easwarakumar, "Transposition based contextual insertion on linear DNA strands with deletion precedence", International Journal of Computer Mathematics, vol. 81, 2004, pp. 647–660.
9. K. Fujioka, "A two-dimensional extension of insertion systems", Theory and Practice of Natural Computing, Third International Conference,, 2014, pp. 181–192.
10. R. Freud, M. Kogler, Y. Rogozhin and S. Verlan, "Graph-Controlled Insertion-Deletion Systems",12th International Workshop on Descriptional Complexity of Formal Systems(DCFS 2010), 2010, pp. 88–98.
11. A. Matveevici, Y. Rogozhin and S. Verlan, "Insertion-deletion systems with one-sided contexts", Lecture Notes in Computer Science, vol. 4664, 2007, pp. 205–217.
12. I. Katsanyi, "Restricted insertion-deletion systems", Annales Uiv. Sci. Budapest., Sect. Comp., vol. 25, 2005, pp. 67–74.
13. L. Kari, "Power of controlled insertion and deletion", Results and Trends in Theoretical Computer Science, 2005, pp. 197–212.
14. Erik Winfree, "Simulations of Computing by Self-Assembly", 1998.

AUTHORS PROFILE

V. Sudha received her BE degree in Computer Science and Engineering from VMKV Engineering College, Salem and ME degree in Computer Science and Engineering from CEG Campus, Anna University. She is currently pursuing her PhD in Anna University in the area of Theoretical Computer Science.

K. S. Easwarakumar is a Professor at the Department of Computer Science and Engineering at Anna University, Chennai. He received his M. Tech in Computer and Information Sciences from Cochin University of Science and Technology, Cochin and PhD in Computer Science and Engineering from Indian Institute of Technology, Madras.
He was the Head of the Department of Computer Science, in Anna University, Chennai for three years. For the past 25 years, he has dedicated himself to academia in Computer Science. His research interests include parallel and distributed computing, data structures, graph algorithms, parallel algorithms, computational geometry, theoretical computer science, molecular computing, and pattern recognition.