IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 7, No. 4, December 2018, pp. 179~184
ISSN: 2252-8938, DOI: 10.11591/ijai.v7.i4.pp179-184

M-ITRS: Mathematical Model for Identification of Tandem Repeats in DNA sequence

Ajay Kumar, Sunita Garhwal
Department of Computer Science and Engineering, Thapar University, Patiala, India

Article Info

Article history:

Received Jul 11, 2018
Revised Sept 15, 2018
Accepted Sept 30, 2018

Keyword:

Deep pushdown automata
DNA
Formal grammar
K-copy language
Tandem repeats

ABSTRACT

In DNA, tandem repeat consists of two or more contiguous copies of a pattern of nucleotides. Tandem repeats of the motif are useful in many applications like molecular biology (related to genetic information of inherited diseases), forensic medicines, DNA fingerprinting and molecular markers for cancer. Various researchers designed formal models and grammars to identify two contiguous copies of the pattern. Tree-adjoining grammar cannot be designed for k-copy language. There is a need to design a formal model which will work for more than two contiguous copies of the pattern. In this paper, we have designed deep pushdown automata for k-continuous copies of the pattern for $k \geq 2$. The proposed formal model will also identify the tandem repeats without specifying the pattern and its size.

Corresponding Author:

Ajay Kumar,
Department of Computer Science and Engineering,
Thapar University,
Patiala, India.
Email: ajayloura@gmail.com

1. INTRODUCTION

Deoxyribonucleic acid (DNA) is a nucleic acid which consists of genetic instructions used in the development and functioning of all living organisms and viruses. Tandem repeats in DNA consist of two or more contiguous copies of a pattern of nucleotides. Repeating patterns are also known as motifs. The motif can occur in different lengths and repetitions can be exact or approximate copies. Repeats of the motif are classified into short tandem repeats or microsatellites (length 10 or shorter) and minisatellites (repeats of 10-60 nucleotides) [1],[2]. Lalioti et al. [3] and Wren et al. [4] observed that the repeats in DNA sequence associated with the neurological disorder. Huang et al. [5], Richard et al. [6] and Mcmurray [7] investigated the repeat of tandem repeats play a significant role in the formation of hairpin structures. Motivated by the applications of tandem repeats in DNA sequence in the area of molecular biology, orensic medicines, DNA fingerprinting and molecular markers for cancer [8]-[11], in this paper we have designed deep pushdown automata for k-copy language.

Related work: The K-copy language can be described by k-copy $= \{x^k \mid x \in \{0,1\}^*\}$. Various researchers carried out work to represent tandem repeats using formal grammar, but the major limitation of their work is that using their formal grammar, we can able to recognize only $L_1 = \{ww \mid w \in \{a,b\}^*\}$ and their grammar cannot generate the languages for www, $wwww$ and so on. $L = \{www \mid w \in \{a,b\}^*\}$ cannot be generated by tree adjoining grammar [12]. Kalra and Kumar [13], [14] introduced the concept of fuzzy deep pushdown automata and deterministic deep pushdown automata. Kalra and Kumar [15] designed the state grammar and deep pushdown automata for Tandem repeats, inverted repeats and interleaved repeats. This proposed approach work for a subset of tandem repeat motif.

Inspired by various applications of k-copy language, a generalized deep pushdown automaton has been designed for tandem repeat motif.

The paper is organized as follows: In Section 2, some preliminaries concept of DNA, tandem repeats and deep pushdown automata are given. Section 3 consists of deep pushdown automata for k-copy languages and tandem repeats followed by conclusions in section 4.

2. PRELIMINARIES

Let $\Sigma = \{a, g, c, t\}$ denotes the DNA alphabet. Purines are classified into guanine (g) and adenine a , whereas pyrimidines are classified into uracil (u), thymine (t) and cytosine (c). Pairing occurs between pyrimidines and purines. The complement of a symbol a is represented by a'. In DNA, $c' = g, g' = c, a' = t,$ and $t' = a$. Deep pushdown automaton is a formal model to represent cross-dependencies in natural and formal languages. It is a counterpart of state grammar.

Def. 1: A deep pushdown automaton [16] is a septuple $(Q, \Sigma, \Gamma, s, S, R, F)$ where Q is a finite set of states, Σ is an alphabet, Γ is a set of stack symbol such that $\Sigma \subseteq \Gamma$, s is a start state, S is a starting pushdown symbol, R is a transition relation defined by $R \subseteq (N \times Q \times (\Gamma - (\Sigma \cup \{\#\})) \times Q \times (\Gamma - \{\#\})^+ \cup (N \times Q \times \{\#\} \times Q \times (\Gamma - \{\#\})^* \{\#\})$ and F is a set of final states such that $F \subseteq Q$. Here $\# \in (\Gamma - \Sigma)$ is a special symbol. The configuration of the deep pushdown automaton is represented by $Q \times \Gamma^* \times (\Gamma - \{\#\})^* \{\#\}$.

In this paper, we have made following changes to the original definition of the deep pushdown automata:

Transition relation R is defined by $R \subseteq (N \times Q \times \Sigma \times (\Gamma - (\Sigma \cup \{\#\})) \times Q \times (\Gamma - \{\#\})^+ \times \{0,1\}) \cup (\{N\} \times Q \times \Sigma \times \{\#\} \times Q \times (\Gamma - \{\#\})^* \{\#\} \times \{0,1\}) \cup (\{0\} \times Q \times \Sigma \times \Sigma \times Q \times \{\lambda\} \times \{0,1\})$.

The sub-relation $(\{0\} \times Q \times \Sigma \times \Sigma \times Q \times \{\lambda\} \times \{0,1\}$ explicitly represents pop of terminal symbol from the top of the stack. Here $\{0,1\}$ represent whether the R/W head remains stationary or point to the symbol on the input tape. Terminal symbol presented on top of the stack are considered as of depth 0, whereas Non-terminals presented on the stack are considered as of depth 1, 2, 3 and so on.

We have explicitly represented the symbol reading from R/W head.

Example 1: Deep pushdown automata for the language $L = \{a^n b^m c^n d^m \mid n, m \geq 0\}$ ${}_2M = (\{q_0, q_1, q_2, q_3, q_4, q_f\}, \{a, b, c, d\}, \{A, S, a, b, c, d, \#\}, q_0, S, \{q_f\}, R)$ and the transition relation R is defined by

$(1, q_0, a, S) \rightarrow (q_1, AA, 0)$
$(1, q_1, a, A) \rightarrow (q_1, Ac, 1)$
$(2, q_1, b, A) \rightarrow (q_2, Ad, 1)$
$(2, q_2, b, A) \rightarrow (q_2, Ad, 1)$
$(1, q_2, c, A) \rightarrow (q_3, \lambda, 0)$
$(0, q_3, c, c) \rightarrow (q_3, \lambda, 1)$
$(1, q_3, d, A) \rightarrow (q_4, \lambda, 0)$
$(0, q_4, d, d) \rightarrow (q_4, \lambda, 1)$
$(0, q_4, \Delta, \#) \rightarrow (q_f, \#, 1)$

For input string $w = aabbbccddd$

$(q_0, aabbbccddd, S\#) \vdash (q_1, aabbbccddd, AA\#) \vdash (q_1, abbbccddd, AcA\#) \vdash (q_1, bbbccddd, AccA\#)$
$\vdash (q_2, bbccddd, AccAd\#) \vdash (q_2, bccddd, AccAdd\#) \vdash (q_2, ccddd, AccAddd\#) \vdash (q_3, ccddd, ccAddd\#)$
$\vdash (q_3, cddd, cAddd\#) \vdash (q_3, ddd, Addd\#) \vdash (q_4, ddd, ddd\#) \vdash (q_4, dd, dd\#) \vdash (q_4, d, d\#) \vdash (q_4, \Delta, \#)$
$\vdash (q_f, \Delta, \#)$

Def. 2: A state grammar is a quintuple (V, Q, Σ, S, P), where V is a total alphabet, Q is a finite set of states, Σ is an alphabet, also known terminal symbol such that $\Sigma \subset V$, $S \in V - T$ is start symbol and P is a finite relation such that $P \subseteq (Q \times (V - \Sigma)) \times (Q \times V^{+})$.

3. DEEP PUSHDOWN AUTOMATA FOR TANDEM REPEATS

In this section, we will design deep pushdown automata for k-copy language and tandem repeats in DNA.

3.1. Deep Pushdown Automata for K-Copy Language

Figure 1 represents the deep pushdown automata for k-copy language. In deep pushdown automata, the element can be pushed on to a deeper part of the stack also. Deep pushdown automata represented in Figure 1 is of depth 2, which means that at a particular point of time, two topmost non-terminals can be expanded. Deep Pushdown automaton for k-copy language is defined by ${}_2M = (\{p_0, p_1, p_2, p_3, p_4, p_f\}, \{a, b\}, \{S, A, B, a, b, \#\}, p_0, S, \{q_f\}, R)$ where the transition relation R is defined by:

$(1, p_0, a, S) \rightarrow (p_0, AB, 0)$
$(1, p_0, b, S) \rightarrow (p_0, AB, 0)$
$(2, p_0, a, B) \rightarrow (p_0\ aB, 1)$
$(2, p_0, b, B) \rightarrow (p_0, bB, 1)$
$(1, p_0, a, A) \rightarrow (p_1, \lambda, 0)$
$(1, p_0, b, A) \rightarrow (p_1, \lambda, 0)$
$(0, p_1, a, a) \rightarrow (p_2, \lambda, 1)$
$(0, p_1, b, b) \rightarrow (p_2, \lambda, 1)$
$(1, p_1, b, B) \rightarrow (p_3, BC, 0)$
$(1, p_1, a, B) \rightarrow (p_3, BC, 0)$
$(0, p_3, a, B) \rightarrow (p_1, \lambda, 0)$
$(0, p_3, b, B) \rightarrow (p_1, \lambda, 0)$
$(0, p_3, a, a) \rightarrow (p_4, \lambda, 0)$
$(0, p_3, b, b) \rightarrow (p_4, \lambda, 0)$
$(2, p_4, a, C) \rightarrow (p_3, aC, 1)$

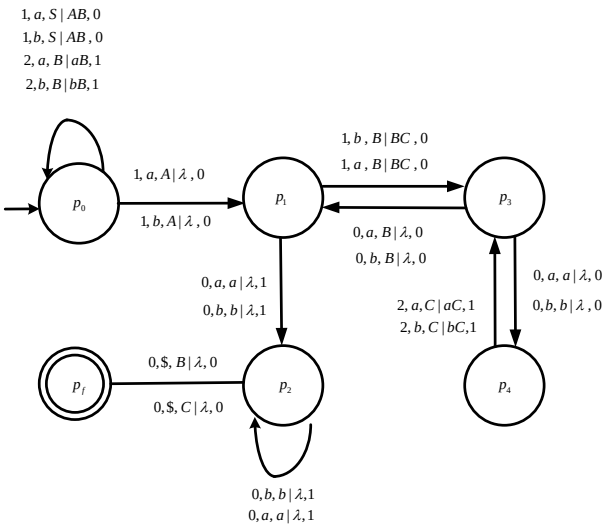

Figure 1. Deep pushdown automata for k-copy language

$(2, p_4, b, C) \rightarrow (p_3, bC, 1)$
$(0, p_2, b, b) \rightarrow (p_2, \lambda, 1)$
$(0, p_2, a, a) \rightarrow (p_2, \lambda, 1)$
$(0, p_2, \$, B) \rightarrow (p_f, \lambda, 0)$
$(0, p_2, \$, C) \rightarrow (p_f, \lambda, 0)$

Derivation of the string, $w = ababab$.

$(p_0, ababab, S) \vdash (p_0, ababab, AB) \vdash (p_0, babab, AaB) \vdash (p_0, abab, AabB) \vdash (p_1, abab, abB)$
$\vdash (p_3, abab, abBC) \vdash (p_4, abab, bBC) \vdash (p_3, bab, bBaC) \vdash (p_4, bab, BaC) \vdash (p_3, ab, BabC)$
$\vdash (p_1, ab, abC) \vdash (p_2, b, bC) \vdash (p_2, \Delta, C) \vdash (p_f, \Delta, C)$

3.2. Deep Pushdown Automata for Tandem Repeats in DNA

The transition diagram of deep pushdown automata for tandem repeats is shown in Figure 2. It is defined by ${}_2M = (\{p_0, p_1, p_2, p_3, p_4, p_f\}, \{a, b\}, \{S, A, B, a, b, \#\}, p_0, S, \{q_f\}, R)$ where the transition relation R is defined by:

Transition from p_0 to p_0
$(1, p_0, g, S) \rightarrow (p_0, AB, 0)$
$(1, p_0, c, S) \rightarrow (p_0, AB, 0)$
$(1, p_0, a, S) \rightarrow (p_0, AB, 0)$
$(1, p_0, t, S) \rightarrow (p_0, AB, 0)$
$(2, p_0, g, B) \rightarrow (p_0, gB, 1)$
$(2, p_0, c, B) \rightarrow (p_0, cB, 1)$
$(2, p_0, a, B) \rightarrow (p_0, aB, 1)$
$(2, p_0, t, B) \rightarrow (p_0, tB, 1)$

Transition from p_0 to p_1
$(1, p_0, g, A) \rightarrow (p_1, \lambda, 0)$
$(1, p_0, c, A) \rightarrow (p_1, \lambda, 0)$
$(1, p_0, a, A) \rightarrow (p_1, \lambda, 0)$
$(1, p_0, t, A) \rightarrow (p_1, \lambda, 0)$

Transition from p_1 to p_3
$(1, p_1, t, B) \rightarrow (p_3, BC, 0)$
$(1, p_1, a, B) \rightarrow (p_3, BC, 0)$
$(1, p_1, c, B) \rightarrow (p_3, BC, 0)$
$(1, p_1, g, B) \rightarrow (p_3, BC, 0)$

Transition from p_3 to p_1
$(0, p_3, g, B) \rightarrow (p_1, \lambda, 0)$
$(0, p_3, c, B) \rightarrow (p_1, \lambda, 0)$
$(0, p_3, a, B) \rightarrow (p_1, \lambda, 0)$
$(0, p_3, t, B) \rightarrow (p_1, \lambda, 0)$

Transition from p_3 to p_4
$(0, p_3, g, g) \rightarrow (p_4, \lambda, 0)$
$(0, p_3, c, c) \rightarrow (p_4, \lambda, 0)$
$(0, p_3, a, a) \rightarrow (p_4, \lambda, 0)$
$(0, p_3, t, t) \rightarrow (p_4, \lambda, 0)$

Transition from p_4 to p_3

$(2, p_4, t, C) \rightarrow (p_3, tC, 1)$
$(2, p_4, a, C) \rightarrow (p_3, aC, 1)$
$(2, p_4, c, C) \rightarrow (p_3, cC, 1)$
$(2, p_4, g, C) \rightarrow (p_3, gC, 1)$

Transition from p_1 to p_2
$(0, p_1, t, t) \rightarrow (p_2, \lambda, 1)$
$(0, p_1, a, a) \rightarrow (p_2, \lambda, 1)$
$(0, p_1, c, c) \rightarrow (p_2, \lambda, 1)$
$(0, p_1, g, g) \rightarrow (p_2, \lambda, 1)$

Transition from p_2 to p_2
$(0, p_2, g, g) \rightarrow (p_2, \lambda, 1)$
$(0, p_2, c, c) \rightarrow (p_2, \lambda, 1)$
$(0, p_2, a, a) \rightarrow (p_2, \lambda, 1)$
$(0, p_2, t, t) \rightarrow (p_2, \lambda, 1)$

Transition from p_2 to p_f
$(0, p_2, \Delta, C) \rightarrow (p_f, \lambda, 0)$
$(0, p_2, \Delta, B) \rightarrow (p_f, \lambda, 0)$

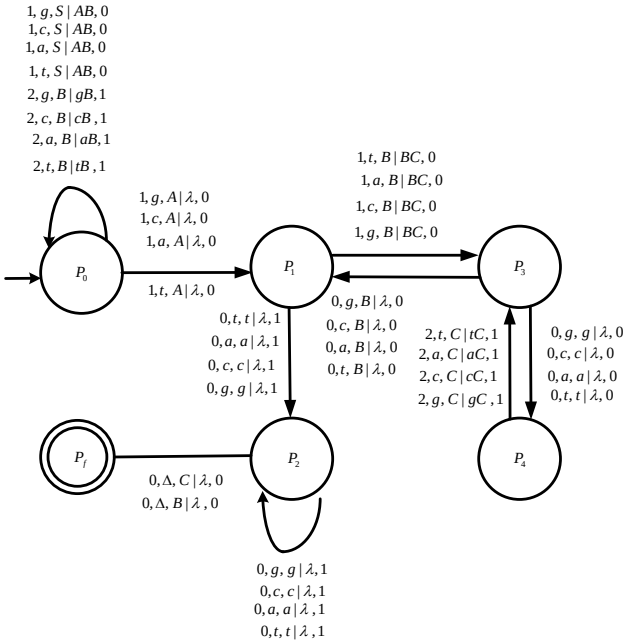

Figure 2. Deep pushdown automata for tandem repeats of DNA

Derivation of the input string $w = gatgatgat$

$(p_0, gatgatgat\Delta, S\#) \square (p_0, gatgatgat\Delta, AB\#) \square (p_0, atgatgat\Delta, AgB\#) \square (p_0, tgatgat\Delta, AgaB\#)$
$(p_0, gatgat\Delta, AgatB\#) \square (p_1, gatgat\Delta, gatB\#) \square (p_3, gatgat\Delta, gatBC\#) \square (p_4, gatgat\Delta, atBC\#)$
$(p_3, atgat\Delta, atBgC\#) \square (p_4, atgat\Delta, tBgC\#) \square (p_3, tgat\Delta, tBgaC\#) \square (p_4, tgat\Delta, BgaC\#)$
$(p_3, gat\Delta, BgatC\#) \square (p_1, gat\Delta, gatC\#) \square (p_2, at\Delta, atC\#) \square (p_2, t\Delta, tC\#) \square (p_2, \Delta, C\#) \square (p_f, \Delta, \#)$

4. CONCLUSION

In this paper, we will design the deep pushdown automata for tandem repeats. The designed deep pushdown automata will work for k-copy language. The major advantage of the proposed approach over the existing approach is that it will work for the languages $ww, www, wwww, \ldots$. Following are the research direction in which work can be carried out in the near future:

1. Parsing of k-copy language and tandem repeats.
2. Design of a tool for identifying tandem repeat pattern in DNA sequence.
3. This model can be extended to k-approximate tandem repeats and multiple length tandem repeats.

REFERENCES

[1] H. Ellegren, Microsatellites: simple sequences with complex evolution, *Nat. Rev. Genet.*, vol. 5, pp. 435-445, 2004.
[2] F. Denoeud, G. Vergnaud and G. Benson, Predicting human minisatellite polymorphism, *Genome Res.*, vol. 13, pp. 856-867, 2003.
[3] M. D. Lalioti, H. S. Scott, C. Buresi, A. Bottani, M. A. Norris, A. Malafosse and S. E. Antonarakis, Dodecamer repeat expansion in cystatin B gene in progressive myoclonus epilepsy, *Nature*, vol. 386, pp. 847-852, 1997.
[4] J. D. Wren, E. Forgacs, J. W. Fondon, A. Pertsemlidis, S. Y. Cheng and T. Gallardo et al., Repeat polymorphisms within gene regions: phenotypic and evolutionary implications, *Am. J. Hum. Genet.* vol. 67, pp. 345-356, 2000.
[5] C. Huang, Y. Lin, Y. Yang, S. Huang and C. Chen, The telomeres of Streptomyces chromosomes contain conserved palindromic sequences with potential to form complex secondary structures, *Mol. Microbiol.*, vol. 28, pp. 905-916, 1998.
[6] G. F. Richard, C. Hennequin, A. Thierry and B. Dujon, Trinucleotide repeats and other microsatellites in yeasts, *Res. Microbiol.*, vol.150, pp. 589-602, 1999.
[7] C. T. McMurray, DNA secondary structure: a common and causative factor for expansion in human disease. *Proc. Natl. Acad. Sci. USA*, vol. 96, pp. 1823-1825, 1999.
[8] P. Gill, C. P. Kimpton, A. Urquhart, N. Oldroyd, E. S. Millican, S. K. Watson and T. J. Downes, Automated short tandem repeat (STR) analysis in forensic casework-a strategy for the future, *Electrophoresis*, vol. 16 (1), pp. 1543-1552.
[9] B. Yuan, D. Vaske, J.L. Weber, J. Beck and V.C. Sheffield, Improved set of short-tandem repeat polymorphisms for screening the human genome, *Am. J. Hum. Genet.* vol. 60 (2), pp. 459-460, 1997.
[10] W. Parson, R. Kirchebner, R. Mühlmann, K. Renner, A. Kofler, S. Schmidt and R. Kofler, Cancer cell line identification by short tandem repeat profiling: power and limitations, *FASEB J.*, vol. 19 (3), pp. 434-436, 2005.
[11] S. Pelotti, S. Ceccardi, M. Alu, F. Lugaresi, R. Trane, M. Falconi, C. Bini and A. Cicognani, Cancerous tissues in forensic genetic analysis, *Genet. Test.*, vol. 11 (4), pp. 397-400, 2008.
[12] A. K. Joshi and L. Levy and M. Takahashi, Tree adjunct grammars, *Journal of Computer and System Sciences*, vol. 10, pp. 136-163, 1985.
[13] N. Kalra and A. Kumar, Deterministic Deep Pushdown Transducer and its Parallel version, *The Computer Journal*, Accepted 2017, doi:10.1093/comjnl/bxx036.
[14] N. Kalra and A. Kumar, Fuzzy state grammar and fuzzy deep pushdown automaton, *Journal of Intelligent & Fuzzy Systems*, vol. 31(1), pp. 249-258, 2016.
[15] N. Kalra and A. Kumar, State Grammar and Deep Pushdown Automata for Biological Sequences of Nucleic Acids, *Current Bioinformatics*, vol. 11(4), pp.470-479, 2016.
[16] A. Meduna and P. Zemek, Regulated Grammar and Automata, 1st edition, *Springer-Verlag*: New York, 2014.

BIOGRAPHIES OF AUTHORS

Dr Ajay Kumar is working as an assistant professor in Computer science and Engineering department, Thapar University, Patiala, india. He is having 13 years of teaching and reserach experience. He has published several reserach paper in SCI journals including The computer journal, Quantum information processing, Expert system, current bioinformatics, Journal of intelligent and fuzzy system, applied mathematics & information Science. He is a active reviewer and program committee members of various international conferences and Journals. He is working in the area of Theoretical Computer science, Quantum Computing, Cognitive Science.

Dr Sunita Garhwal is working as an assistant professor in Computer science and Engineering department, Thapar University, Patiala, india. Shee is having 9 years of teaching and reserach experience. She has published several reserach paper in SCI journals in the area of fuzzy automata and software testing. She is working in the area of Fuzzy Automata and Software testing.