İSTANBUL GELİŞİM ÜNİVERSİTESİ

ICAAMM 2018

Proceedings of the

7th International Conference on Applied Analysis and Mathematical Modeling

Editors
Mustafa Bayram
Aydin Secer

Istanbul Gelisim University, Istanbul, Turkey

PROCEEDINGS OF THE 7th INTERNATIONAL CONFERENCE ON APPLIED ANALYSIS

and

MATHEMATICAL MODELING, ICAAMM2018

June 20-24, 2018, Istanbul-Turkey

Editors

Mustafa Bayram
Istanbul Gelisim University

Aydin Secer
Yildiz Technical University

TABLE OF CONTENTS

Closure properties of static Watson-Crick regular grammars

Aqilahfarhana Abdul Rahman, Wan Heng Fong, Nor Haniza Sarmin, Sherzod Turaev and Nurul Liyana Mohamad Zulkufli

Department of Mathematical Sciences, Faculty of Science, Universiti Teknologi Malaysia, 81310 UTM Skudai, Johor, Malaysia
Department of Computer Science, Faculty of Information and Communication Technology, International Islamic University Malaysia, 53100 Kuala Lumpur, Malaysia
E-mail: aqilahfarhana_13@yahoo.com, fwh@utm.my, nhs@utm.my, sherzod@iium.edu.my and liyanazulkufli@iium.edu.my

Abstract: In DNA computing, formal language theory is a suitable tool to analyze different kinds of biological sequences. Some of the computational models based on different operations of DNA molecules have been developed using formal language theory such as automata and sticker systems. Automata are represented as mathematical models of abstract computing devices to recognize the input words, where Watson-Crick automata is the type of automata that relate to DNA computing devices. Meanwhile, a sticker system is a computational model which uses sticker operation on DNA molecules. A static Watson-Crick regular grammar is a grammar counterpart of sticker system that generates the double stranded strings and has rules as in a regular grammar. In this paper, some closure properties of static Watson-Crick regular grammar are presented. We investigate and show that the static Watson-Crick regular grammars are closed under the different operations which are similar to the ones in Chomsky grammars and Watson-Crick grammars.

Keywords: Static Watson-Crick grammar, regular grammar, closure properties.

1 Introduction

Deoxyribonucleic acid (DNA) is a double stranded molecule which consists of four types of nucleotides as its bases known as *adenine*(A), *thymine*(T), *cytosine*(C) and *guanine*(G). These four bases are pairing according to the Watson-Crick (WK) complementarity, one of the unique features of DNA molecules where the base A is paired with the base T (A-T); and the base C is paired with the base G (C-G). Another feature called massive parallelism allows the construction of many copies of DNA strands and carries out of operations on encoded information simultaneously.

The birth of a new field, called DNA computation has been marked by Adleman [1] in 1994 . By using the DNA strand in his experiment, he was able to solve the Hamiltonian path problem (HPP) for a simple graph with the method of sticker operation. Then, Kari *et al.* [2] proposed a mathematical model known as a sticker system which uses the sticker operation on DNA to form a complete double stranded sequence. In 1997, Freund *et al.* [3] proposed the Watson-Crick automata (WKA) which is one of the mathematical models used in DNA computation. WKA is an extension of finite automata with the addition of two reading heads on double stranded sequences. The difference between the sticker systems and WKA is that the sticker systems generate complete double-stranded molecules by using sticker operation while WKA which is the accepting counterpart, parse a given complete double-stranded molecule and determine whether the input is accepted or not [4].

The first grammar model based on WKA has been introduced in [5], which is a Watson-Crick regular grammar. The research has been extensively studied with Watson-Crick grammars introduced in

[6, 7]. Watson-Crick (WK) grammars which consist of regular grammar, linear grammar and context-free grammar produce each stranded string independently and can only check the WK complementarity of a generated complete double-stranded string at the end. Besides, they do not fully illustrate the synthesis of DNA molecules and it affects the computation efficiency of the models. As improvement of WK grammars, a new variant of WK grammars is proposed, called static Watson-Crick (sWK) grammars [8, 9] as an analytical counterpart of sticker systems. In this research, we mainly focused on static Watson-Crick regular (SREG) grammars.

A language can be shown as a regular language by using regular expression, regular grammar and finite automata. Another way to show whether a given language is regular or not is by studying the closure properties of regular languages [10]. In 1998, Paun *et al.* [11] found the closure properties of the families in the Chomsky hierarchy. It is proven that the regular languages are closed under all operations in formal language theory such as union, intersection, complement, concatenation, Kleene-star closure, mirror image and so on. According to Zulkufli *et al.* [12], the closure properties of WK grammars help to find the correctness of the results when performing the operations on the sets of DNA molecules generated by some WK grammars. They proved that the WK regular (WKREG) grammars preserve almost all of the closure properties of regular grammars.

This paper is organized as follows: Section 1 gives the background and introduction of the paper. Section 2 presents some necessary definitions and notations from the theories of formal languages, WKREG grammars and SREG grammars. The closure properties of SREG grammars are presented in the last section.

2 Preliminaries

In this section, some preliminary concepts which involve the basic notations of formal language theory and definition of WKREG grammars and SREG grammars are included in this paper. For further details on automata and sticker system, the reader is referred to [2, 10, 11, 13]. Throughout the paper, we use the following general notation.

The membership of an element to a set is denoted by $\in$ and the empty set is denoted by $\emptyset$. An alphabet is a finite nonempty set of symbols. Let T be a finite alphabet. Then, T^* is the set of all finite strings (words) over T. A string with no symbols, or we called it as empty string is denoted by λ. The set T^* always contains λ and to exclude the empty string, the symbol T^+ is defined as the set of all nonempty finite strings over T where $T^+ = T^* - \{\lambda\}$. Therefore, a language is defined as a subset of the set of strings where $L \subseteq T^*$.

Recall some basic terms that we used for closure properties of languages. The union of two languages L_1 and L_2, denoted as $L_1 \cup L_2$, is the set of all strings which includes the elements in both L_1 and L_2. Next, the concatenation of two languages L_1 and L_2 is the set of all strings that is obtained by concatenating any element of L_1 with any element L_2 where $L_1L_2 = \{xy : x \in L_1, y \in L_2\}$. The star-closure or Kleene-star closure of a language is defined as $L^* = L^0 \cup L^1 \cup L^2 \cdots$ and the mirror image (reverse) of a language is the set of all string reversals where $L^R = \{w^R : w \in L\}$.

A grammar acts as a mechanism to describe the languages mathematically, known as a language generator. A Chomsky grammar (sometimes simply called a grammar) is a set of rules formation for rewriting strings which is defined by a quadruple $G = (N, T, S, P)$ where the alphabet N is defined as the *nonterminal* alphabet, T is the *terminal* alphabet, $S \in N$ is the axiom or start alphabet, and $P \subseteq (N \cup T)^*N(N \cup T)^* \times (N \cup T)^*$ is the set of production rules of G. The rules of $(x, y) \in P$ are written in the form of $x \to y$ where $x \in (N \cup T)^+$ and $y \in (N \cup T)^*$. Here, u *directly derives* v or v is derived from u with respect to G written as $u \Rightarrow v$ if and only if $u = u_1xu_2, v = u_1yu_2$, for some $u_1u_2 \in (N \cup T)^*$ and $x \to y \in P$. The set of all terminal strings is the language generated by the grammar which defined by $L(G) = \{w \in T^* : S \Rightarrow^* w\}$. A grammar $G = (N, T, S, P)$ is called:

(i) *regular*, if each rule $u \to v \in P$ has $u \in N$ and $v \in T \cup TN \cup \{\lambda\}$.

(ii) *right-linear*, if each rule $u \to v \in P$ has $u \in N$ and $v \in T^* \cup T^*N$.

(iii) *left-linear*, if each rule $u \to v \in P$ has $u \in N$ and $v \in T^* \cup NT^*$.

The families of languages generated by regular grammars are equal to the families of languages generated by *right-* or *left-*linear grammars. The language generated by regular grammar is denoted as **REG**.

Recall the definition of Watson-Crick regular grammars [6, 7]:

Definition 2.1. A WK grammar $G = (N, T, S, P, \rho)$ is called *regular* if each production has the form $A \longrightarrow \langle u/v\rangle B$ or $A \longrightarrow \langle u/v\rangle$ where $A, B \in N$ and $\langle u/v\rangle \in \langle T^*/T^*\rangle$.

The notation $\langle u/v\rangle$ represents the element $(u, v) \subseteq V \times V$ in the set of pairs of strings and $\langle T^*/T^*\rangle$ is written instead of $V^* \times V^*$.

Next, the definition of static Watson-Crick regular grammars which consist of right-linear and left-linear introduced in [8] are given as follows:

Definition 2.2. A static Watson-Crick (sWK) *right-linear* grammar is a 5-tuple $G = (N, T, \rho, S, P)$ where N, T, S are defined as for a regular grammar, $\rho \in T \times T$ is a symmetric relation (WK complementary) and P is a finite set of productions in the form of

(i) $S \to \begin{bmatrix} u \\ v \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} A$ where $A \in N - \{S\}$, $\begin{bmatrix} u \\ v \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \in R_\rho(T)$;

(ii) $A \to \begin{pmatrix} x \\ y \end{pmatrix} B$ where $A, B \in N - \{S\}$ and $\begin{pmatrix} x \\ y \end{pmatrix} \in LR_\rho^*(T)$; or

(iii) $A \to \begin{pmatrix} x \\ y \end{pmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$ where $A \in N - \{S\}$, $\begin{pmatrix} x \\ y \end{pmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \in L_\rho(T)$.

Definition 2.3. A static Watson-Crick (sWK) *left-linear* grammar is a 5 -tuple $G = (N, T, \rho, S, P)$ where N, T, ρ, S are defined as for a sWK right-linear grammar and P is a finite set of productions in the form of

(i) $S \rightarrow A\binom{x}{y}\begin{bmatrix}u\\v\end{bmatrix}$ where $A \in N - \{S\}$, $\binom{x}{y}\begin{bmatrix}u\\v\end{bmatrix} \in L_\rho(T)$;

(ii) $A \rightarrow B\binom{x}{y}$ where $A, B \in N - \{S\}$ and $\binom{x}{y} \in LR^*_\rho(T)$; or

(iii) $A \rightarrow \begin{bmatrix}u\\v\end{bmatrix}\binom{x}{y}$ where $A \in N - \{S\}$, $\begin{bmatrix}u\\v\end{bmatrix}\binom{x}{y} \in R_\rho(T)$.

Definition 2.4. The language generated by a sWK regular grammars G, denoted by $L(G)$, is defined as

$$L(G) = \{u : \begin{bmatrix}u\\v\end{bmatrix} \in WK_\rho(T) \text{ and } S \underset{G}{\Rightarrow}^* \begin{bmatrix}u\\v\end{bmatrix}\}.$$

The families of languages generated by sWK regular grammars are denoted by **SREG**.

In the next section, the closure properties of sWK regular grammars are presented.

3 Closure properties

In this section, the closure properties of SREG grammars are introduced. Let $L_1, L_2 \in$ SREG and $G_1 = (N_1, T, \rho, S_1, P_1), G_2 = (N_2, T, \rho, S_2, P_2)$ be SREG grammars generating languages L_1 and L_2 respectively, such that $L_1 = L(G_1)$ and $L_2 = L(G_2)$. Assume that $N_1 \cap N_2 \neq \emptyset$.

Proposition 3.1. *(Union) If L_1 and L_2 are SREG languages, then $L_1 \cup L_2$ is also a SREG language.*

Proof Let $G = (N, T, \rho, S, P)$ be a SREG grammar where $N = N_1 \cup N_2 \cup \{S\}$, $T = T_1 \cup T_2$, $\rho \in \begin{bmatrix}T\\T\end{bmatrix}^*$ with $S \notin N_1 \cup N_2$, and $P = P_1 \cup P_2 \cup \{S \longrightarrow S_1\} \cup \{S \longrightarrow S_2\}$. Therefore, it is obvious that $L(G) = L_1 \cup L_2$. Thus, SREG language is closed under union. ■

Proposition 3.2. *(Concatenation) If L_1 and L_2 are SREG languages, then $L_1 \cdot L_2$ is also a SREG language.*

Proof Let $G = (N, T, \rho, S, P)$ be a SREG grammar where $N = N_1 \cup N_2$ and $S = S_1$. Define the production rule in the form of $P = (P_1 - \{A \longrightarrow \binom{x}{y}\begin{bmatrix}u\\v\end{bmatrix} \in P_1\}) \cup$ $\{A \longrightarrow \binom{x}{y}\begin{bmatrix}u\\v\end{bmatrix} S_2 : A \longrightarrow \binom{x}{y}\begin{bmatrix}u\\v\end{bmatrix} \in P_1\} \cup P_2$. Hence, the concatenation of the language is $L(G) = L_1 \cdot L_2$. Therefore, SREG language is closed under concatenation. ■

Proposition 3.3. *(Star-Closure) If L_1 is a SREG language, then L_1^* is also a SREG language.*

Proof Let $G = (N_1, T, \rho, S, P_1)$ be a SREG grammar in which $S = S_1$ and P_1 is the production rule in the form of $P_1 = (\{P_R - P_T\}) \cup P_S \cup \{S_1 \longrightarrow \lambda\}$, where

(i) $P_R = \{S \longrightarrow \begin{bmatrix}u\\v\end{bmatrix}\binom{x}{y}A : \begin{bmatrix}u\\v\end{bmatrix}\binom{x}{y} \in R_\rho(T)\}$ and $\{A \longrightarrow \binom{x}{y}B : \binom{x}{y} \in LR^*_\rho(T)\}$,

(ii) $P_T = \{A \longrightarrow \binom{x}{y}\begin{bmatrix}u\\v\end{bmatrix} \in P : \binom{x}{y}\begin{bmatrix}u\\v\end{bmatrix} \in L_\rho(T)\}$,

(iii) $P_S = \{A \longrightarrow \binom{x}{y}\begin{bmatrix}u\\v\end{bmatrix} S_1 : \binom{x}{y}\begin{bmatrix}u\\v\end{bmatrix} \in P_T\}$.

Hence, $L(G) = L_1^*$. Thus, SREG language is closed under star-closure. ∎

Proposition 3.4. *(Mirror Image) If $L_1(G)$ is a SREG language, then $L_1^R(G)$ is also a SREG language.*

Proof Let $L_1(G) \in$ SREG. We need to show that $L_1^R(G) \in$ SREG too. Define $G = (N_1, T, \rho, S_1, P_1)$ which generates the language $L_1^R(G)$ where P_1 consists of production rules in the form of : $S \longrightarrow A\binom{x}{y}\begin{bmatrix}u\\v\end{bmatrix}$, where $A \in N - \{S\}$, $\binom{x}{y}\begin{bmatrix}u\\v\end{bmatrix} \in L_\rho(T)$ and $S \longrightarrow \binom{x}{y}\begin{bmatrix}u\\v\end{bmatrix} A \in P$. $A \longrightarrow B\binom{x}{y}$, where $A, B \in N - \{S\}$, $\binom{x}{y} \in LR_\rho^*(T)$ and $A \longrightarrow \binom{x}{y} B \in P$ or $A \longrightarrow \begin{bmatrix}u\\v\end{bmatrix}\binom{x}{y}$, where $A \in N - \{S\}$, $\begin{bmatrix}u\\v\end{bmatrix}\binom{x}{y} \in R_\rho(T)$ and $A \longrightarrow \begin{bmatrix}u\\v\end{bmatrix}\binom{x}{y} \in P$.

Hence, SREG is closed under mirror image. ∎

Based on the results shown above, we can conclude that the family of static Watson-Crick regular language is closed under union, concatenation, Kleene-star closure and mirror image similar to the result in [12]. Other properties such as intersection, complement, homomorphism and others are still in progress.

4 Conclusion

In this paper, we have shown that the static Watson-Crick regular grammars are closed under union, concatenation, Kleene-star closure and mirror image. We believe that the static Watson-Crick regular grammars have similar closure properties as the ones in Chomsky grammars and Watson-Crick grammars. This research can be further studied by finding the closure properties of static Watson-Crick linear grammars and static Watson-Crick context-free grammars.

5 Acknowledgement

The first author would like to thank UTM Zamalah for funding her studies at Universiti Teknologi Malaysia (UTM) under the Zamalah Scholarship. The second and third authors would also like to thank the Ministry of Higher Education (MOHE) and Research Management Centre (RMC), UTM for the financial funding through Research University Grant Vote No. 13H18. The fourth author is grateful to IIUM for the financial support through Research Initiative Grant Scheme RIGS16-368-0532.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

All authors have contributed to all parts of the article. All authors read and approved the final manuscript.

References

[1] Adleman, L., Molecular computation of solutions to combinatorial problems. Science, 266(5187), pp. 1021-1024. 1994.

[2] Kari, L., Paun, G., Salomaa, A. ansd Yu, S., DNA Computing, sticker systems and universality. Acta Informatica, 35, pp. 401-420. 1998.

[3] Freund, R., Paun, G., Rozenberg,G. and Salomaa, A., Proc. 3rd DIMACS Workshop on DNA based Computers, Philadelphia, 1997, p. 297-328.

[4] Kari, L., Seki, S. and Sosik, P., Handbook of Natural Computing 1073âĂŞ1127 (2012).

[5] Subramanian, K.G., Hemalatha, S. and Venkat, I., On Watson-Crick Automata, In Proc. The Second International Conference on Computational Science, Engineering and Information Technology, pp. 151-156, Coimbatore, India, 2012.

[6] Zulkufli, N.L.M., Turaev, S., Tamsin, M.I.M. and Messikh, A. The Computational Power of Watson-Crick Grammars: Revisited. In Bio-Inspired Computing-Theories and Applications, Springer, Singapore (2016), 215-225.

[7] Zulkufli, N.L.M., Turaev, S., Tamrin, M.I.M. and Azeddine, M., Generative Power and Closure Properties of Watson-Crick Grammars. Applied Computational Intelligence and Soft Computing, 2016.

[8] Abdul Rahman, A.F., Fong, W.H., Sarmin, N. H., Turaev, S. and Zulkufli, N.L.M. Static Watson-Crick Regular Grammars, UTM-Emerging Scientists Conference (UTM-ESCon) 2018. (Submitted).

[9] Abdul Rahman, A.F., Fong, W.H., Sarmin, N. H., Turaev, S. and Zulkufli, N.L.M. Static Watson-Crick Linear Grammars, International Graduate Conference on Engineering, Science and Humanities (IGCESH) 2018. (Submitted).

[10] Linz, P., An Introduction to Formal Languages and Automata, Jones and Barlett Publishers, 2006.

[11] Paun, G., Rozenberg, G. and Salomaa, A., DNA Computing: New Computing Paradigms, New York, Springer-Verlag, 1998.

[12] Zulkufli, N.L.M., Turaev, S., Tamrin, M.I.M. and Azeddine, M., Closure Properties of Watson-Crick Grammars. In AIP Conference Proceedings, 1691, pp. 040032, 2015.

[13] Czeizler, E. and Czeizler, E., A short survey on Watson-Crick Automata. Bulletin of the EATCS. 88(3),pp. 104-119. 2006.