

Static Watson-Crick Context-Free Grammars

<https://doi.org/10.3991/ijoe.v15i10.10878>

Wan Heng Fong (✉), Aqilahfarhana Abdul Rahman, Nor Haniza Sarmin
Universiti Teknologi Malaysia, Johor, Malaysia.
fwh@utm.my

Sherzod Turaev
International University of Sarajevo, Sarajevo, Bosnia and Herzegovina.

Abstract—Sticker systems and Watson-Crick automata are two modellings of DNA molecules in DNA computing. A sticker system is a computational model which is coded with single and double-stranded DNA molecules; while Watson-Crick automata is the automata counterpart of sticker system which represents the biological properties of DNA. Both of these models use the feature of Watson-Crick complementarity in DNA computing. Previously, the grammar counterpart of the Watson-Crick automata have been introduced, known as Watson-Crick grammars which are classified into three classes: Watson-Crick regular grammars, Watson-Crick linear grammars and Watson-Crick context-free grammars. In this research, a new variant of Watson-Crick grammar called a static Watson-Crick context-free grammar, which is a grammar counterpart of sticker systems that generates the double-stranded strings and uses rule as in context-free grammar, is introduced. The static Watson-Crick context-free grammar differs from a dynamic Watson-Crick context-free grammar in generating double-stranded strings, as well as for regular and linear grammars. The main result of the paper is to determine the generative powers of static Watson-Crick context-free grammars. Besides, the relationship of the families of languages generated by Chomsky grammars, sticker systems and Watson-Crick grammars are presented in terms of their hierarchy.

Keywords—DNA Computing, Watson-Crick Grammar, Context-Free Grammar, Sticker Systems, Generative Power

1 Introduction

DNA (Deoxyribonucleic Acid) molecule plays an important role in DNA computing. DNA is a polymer which is constructed from monomers namely deoxyribonucleotides. Each deoxyribonucleotide consists of three parts of components; a sugar, a phosphate group, and a nitrogenous base. The four nitrogenous bases are adenine (A), thymine (T), guanine (G), and cytosine (C) which are paired as A-T and C-G according to the Watson-Crick (WK) complementarity. DNA computing is a branch of biomolecular computing which concerns with the utilization of DNA as an information carrier. The birth of this field has been marked by Adleman [1] in 1994. By using

DNA strands in his experiment, he was able to solve the Hamiltonian path problem for a simple graph with the sticker operation.

Sticker systems and Watson-Crick automata are DNA computing models which are based on different principles, but the complementarity relation do exist in a computation or derivation step. In 1998, sticker systems were introduced by Kari et al. [2] as language generating devices based on the sticking operation which is a model of techniques used by Adleman [1]. The operation starts from a finite set of axioms and then prolongs to the right of the generated sequences (single or double) symbols by using single-stranded strings, either to the upper or lower strands, therefore matching the sequence based on the complementarity relation. Some variants of sticker systems have been defined such as one-sided, regular, simple, simple one-sided and simple regular sticker systems [3]. In order to increase the generative power of sticker systems, some additional restrictions have been imposed such as by assigning an element of a monoid to the sticker operation [4], by introducing probabilistic sticker systems [5] and by including the presence of weight for the variant of sticker systems [6].

Watson-Crick automata was proposed by Freund et al. in 1997 [7] which is an extension of finite automata with the addition of two reading heads on double-stranded sequences. Some restrictions and extensions have been proposed on the basic model of Watson-Crick automata in order to achieve higher generative powers of Watson-Crick automata. Some variants of Watson-Crick automata have been proposed such as Watson-Crick transducers [8], Watson-Crick omega-automata [9] and weighted Watson-Crick automata [10].

On the other hand, formal language theory is a natural framework in formalizing and investigating DNA computing models. Grammars act as language generator, and besides grammar, automata are devices for defining language which work differently from the grammar. Historically, earlier grammar models introduced in DNA computing did not utilise Watson-Crick complementarity of DNA molecules [11, 12]. Following that, a grammar model that uses this feature has been proposed, known as Watson-Crick grammars which produce each stranded string independently [13]. In this research, a new variant of Watson-Crick grammars, called a static Watson-Crick context-free grammar is introduced as an analytical counterpart of sticker systems.

This paper is organized as follows: Section 1 introduces the background of the research. In Section 2, some necessary definitions and notations used in this research are presented. Next, the definition of Watson-Crick grammars, the concepts of sticker systems and Watson-Crick Chomsky normal form are discussed and shown in Section 3. In Section 4, the results on the static Watson-Crick context-free grammar with some of the generative powers are given.

In the next section, some preliminaries concepts which are used in this paper are stated.

2 Literature Review

In the following section, some information on grammars, static Watson-Crick regular and linear grammars are stated. In this paper, the symbol \subseteq denotes the inclusion

while \subset denotes the strict (proper) inclusion. The membership of an element to a set is denoted by \in . 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 $T^+ = T^* - \{\lambda\}$.

In formal language theory, a grammar acts as a mechanism to describe languages mathematically; in other words, it acts as a language generator. A Chomsky grammar (sometimes simply called a grammar) is a set of rule formation for rewriting strings. Chomsky grammar is classified depending on their respective form of production rules. The definition of Chomsky grammar is stated as follows.

Definition 1 [3]: Chomsky Grammar. A Chomsky grammar is a quadruple $G = (N, T, S, P)$, where N, T are disjoint alphabets, $S \in N$ and P is a finite subset of $(N \cup T)^* N (N \cup T)^* \times (N \cup T)^*$.

The Chomsky grammar is classified depending on their respective form of production rules. A grammar $G = (N, T, S, P)$ is called:

1. context-sensitive, if each rule $u \rightarrow v \in P$ has $u = u_1 A u_2, v = u_1 x u_2$, for $u_1, u_2 \in (N \cup T)^*, A \in N$ and $x \in (N \cup T)^+$.
2. context-free, if each rule $u \rightarrow v \in P$ has $u \in N$.
3. linear, if each rule $u \rightarrow v \in P$ has $u \in N$ and $v \in T^* \cup T^* N T^*$.
4. right-linear, if each rule $u \rightarrow v \in P$ has $u \in N$ and $v \in T^* \cup T^* N$.
5. left-linear, if each rule $u \rightarrow v \in P$ has $u \in N$ and $v \in T^* \cup N T^*$.
6. regular, if each rule $u \rightarrow v \in P$ has $u \in N$ and $v \in T \cup T N \cup \{\lambda\}$.

The family of languages generated by regular grammars is equal to the family of languages generated by the right- or left-linear grammars. The families of context-sensitive, context-free, linear and regular languages are denoted as **CS**, **CF**, **LIN** and **REG** respectively. Other than that, **RE** and **FIN** represent the family of recursive enumerable languages, i.e., arbitrary languages and finite language. Hence, the following strict inclusion holds for Chomsky hierarchy, where **FIN** \subset **REG** \subset **LIN** \subset **CF** \subset **CS** \subset **RE** [3].

Next, we recall the definition of static Watson-Crick regular and linear grammars. In this paper, we state only for static Watson-Crick right-linear grammars for the regular grammars since the definition is almost similar to left-linear grammars.

Definition 2 [14] Static Watson-Crick right-linear grammar. A static Watson-Crick right-linear grammar is a 5-tuple $G = (N, T, \rho, S, P)$ where N, T are disjoint nonterminal and terminal alphabets respectively, $\rho \subseteq T \times T$ is a symmetric relation (Watson-Crick complementarity), $S \in N$ is a start symbol (axiom) and P is a finite set of production rules in the form of

1. $S \rightarrow \begin{bmatrix} u \\ v \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} A$ where $A \in N - \{S\}, \begin{bmatrix} u \\ v \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \in R_\rho(T)$;
7. $A \rightarrow \begin{pmatrix} x \\ y \end{pmatrix} B$ where $A, B \in N - \{S\}, \begin{pmatrix} x \\ y \end{pmatrix} \in LR_\rho^*(T)$; or
8. $A \rightarrow \begin{pmatrix} x \\ y \end{pmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$ where $A \in N - \{S\}, \begin{pmatrix} x \\ y \end{pmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \in L_\rho(T)$.

Definition 3 [15] Static Watson-Crick Linear Grammar. A static Watson-Crick linear grammar is a 5-tuple $G = (N, T, \rho, S, P)$ where N, T are disjoint nonterminal and terminal alphabets respectively, $\rho \subseteq T \times T$ is a symmetric relation (Watson-Crick complementarity), $S \in N$ is a start symbol (axiom) and P is a finite set of production rules in the form of

1. $S \rightarrow \begin{bmatrix} u_1 \\ v_1 \end{bmatrix} \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} A \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} \begin{bmatrix} u_2 \\ v_2 \end{bmatrix}$ where $A \in N - \{S\}$, $\begin{bmatrix} u_1 \\ v_1 \end{bmatrix} \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} \in R_\rho(T)$ and $\begin{pmatrix} x_2 \\ y_2 \end{pmatrix} \begin{bmatrix} u_2 \\ v_2 \end{bmatrix} \in L_\rho(T)$;
2. $A \rightarrow \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} B \begin{pmatrix} x_2 \\ y_2 \end{pmatrix}$ where $A, B \in N - \{S\}$ and $\begin{pmatrix} x_1 \\ y_1 \end{pmatrix}, \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} \in LR_\rho^*(T)$; or
3. $A \rightarrow \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}$ where $A \in N - \{S\}$ and $\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} \in LR_\rho^*(T)$.

In the next section, the concepts of sticker systems, the definition of Watson-Crick grammars and Watson-Crick Chomsky normal form are presented.

3 Methodology

In this research, the static Watson-Crick context-free grammars are introduced by referring from the Watson-Crick grammars with some modifications and by using the concept of sticker systems. Besides, the generative power of these grammars are classified through comparison with the Chomsky hierarchy and Watson-Crick languages. In the following subsections, some information on sticker systems and Watson-Crick grammars are stated.

3.1 Sticker Systems

Let V be an alphabet and let ρ be a symmetric relation for $\rho \subseteq V \times V$ over V (of complementarity). The symbol V^* represents a set of all strings which is composed of elements of V and the empty string denoted as λ , and V^+ is the set $V^* - \{\lambda\}$. The set of all pairs of strings over V is denoted as $\begin{pmatrix} V^* \\ V^* \end{pmatrix}$. To represent DNA molecules as the string, the elements $(x_1, x_2) \in V^* \times V^*$ are written in the form of $\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \in \begin{pmatrix} V^* \\ V^* \end{pmatrix}$. The set $WK_\rho(V) = \begin{bmatrix} V \\ V \end{bmatrix}_\rho^*$ where $\begin{bmatrix} V \\ V \end{bmatrix}_\rho = \left\{ \begin{bmatrix} a \\ b \end{bmatrix} \mid a, b \in V, (a, b) \in \rho \right\}$ denotes the *Watson-Crick domain* which is associated to the alphabet V and the complementarity relation ρ . The elements $\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \in WK_\rho(V)$ are called well-formed double-stranded sequences. The set of incomplete molecules is denoted as $W_\rho(V) = L_\rho(V) \cup R_\rho(V) \cup LR_\rho(V)$, where

$$L_\rho(V) = \left(\begin{pmatrix} \lambda \\ V^* \end{pmatrix} \cup \begin{pmatrix} V^* \\ \lambda \end{pmatrix} \right) \begin{bmatrix} V \\ V \end{bmatrix}_\rho^*$$

$$R_\rho(V) = \left[\begin{matrix} V \\ V \end{matrix} \right]_\rho^* \left(\left(\begin{matrix} \lambda \\ V^* \end{matrix} \right) \cup \left(\begin{matrix} V^* \\ \lambda \end{matrix} \right) \right),$$

$$LR_\rho(V) = \left(\left(\begin{matrix} \lambda \\ V^* \end{matrix} \right) \cup \left(\begin{matrix} V^* \\ \lambda \end{matrix} \right) \right) \left[\begin{matrix} V \\ V \end{matrix} \right]_\rho^+ \left(\left(\begin{matrix} \lambda \\ V^* \end{matrix} \right) \cup \left(\begin{matrix} V^* \\ \lambda \end{matrix} \right) \right)$$

In this research, the definition of $LR_\rho(V)$ is modified according to our grammar, where

$$LR_\rho^*(T) = \left(\left(\begin{matrix} \lambda \\ T^* \end{matrix} \right) \cup \left(\begin{matrix} T^* \\ \lambda \end{matrix} \right) \right) \left[\begin{matrix} T \\ T \end{matrix} \right]_\rho^* \left(\left(\begin{matrix} \lambda \\ T^* \end{matrix} \right) \cup \left(\begin{matrix} T^* \\ \lambda \end{matrix} \right) \right),$$

$$LR_\rho^+(T) = \left(\left(\begin{matrix} \lambda \\ T^* \end{matrix} \right) \cup \left(\begin{matrix} T^* \\ \lambda \end{matrix} \right) \right) \left[\begin{matrix} T \\ T \end{matrix} \right]_\rho^+ \left(\left(\begin{matrix} \lambda \\ T^* \end{matrix} \right) \cup \left(\begin{matrix} T^* \\ \lambda \end{matrix} \right) \right),$$

and the alphabet V which was defined in $W_\rho(V)$ is changed to alphabet T according to the definition in the Chomsky grammar.

Definition 4 [3] A **sticker system** is a construct $\gamma = (V, \rho, A, D)$ where V is an alphabet, $\rho \subseteq V \times V$ is a symmetric relation, A is a finite subset of $LR_\rho(V)$ (called *axioms*) and D is a finite subset of $W_\rho(V) \times W_\rho(V)$ (called *dominoes*).

3.2 Watson-crick grammars

In this subsection, the definitions of Watson-Crick grammars are stated as follows.

Definition 5 [13] A **Watson-Crick (WK) grammar** $G = (N, T, \rho, S, P)$ is called

- Regular if each production has the form $A \rightarrow \langle u/v \rangle$ where $A, B \in N$ and $\langle u/v \rangle \in \langle T^*/T^* \rangle$
- Linear if each production has the form $A \rightarrow \langle u_1/v_1 \rangle B \langle u_2/v_2 \rangle$ or $A \rightarrow \langle u/v \rangle$ where $A, B \in N$ and $\langle u/v \rangle, \langle u_1/v_1 \rangle, \langle u_2/v_2 \rangle \in \langle T^*/T^* \rangle$
- Context-free if each production has the form $A \rightarrow \alpha$ where $A \in N$ and $\alpha \in (N \cup \langle T^*/T^* \rangle)^*$

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 Watson-Crick Chomsky normal form is stated.

Definition 6 [16] A **Watson-Crick (WK) context-free grammar** $G = (N, T, \rho, S, P)$ is said to be in Watson-Crick Chomsky normal form if all productions are of the form

- $A \rightarrow BC$,
- $A \rightarrow \langle u/v \rangle$, or
- $S \rightarrow \langle \lambda/\lambda \rangle$,

Where $A \in N$, $B, C \in N - \{S\}$ and $\langle u/v \rangle \in \langle T/\lambda \rangle \cup \langle \lambda/T \rangle$.

In the next section, the definition and the generative power of static Watson-Crick context-free grammars are presented.

4 Results and Findings

In this research, the definition of static Watson-Crick context-free grammar is introduced whereby it is a grammar counterpart of the sticker system that has rules as in context-free grammars.

4.1 Definition 7. A static Watson-Crick context-free grammar

Is a 5-tuple $G = (N, T, \rho, S, P)$ where N, T are disjoint nonterminal and terminal alphabets, respectively, $\rho \in T \times T$ is a symmetric relation (Watson-Crick complementarity), $S \in N$ is a start symbol (axiom) and P is a finite set of production rules in the form of

1. $S \rightarrow x_1 A_1 x_2 A_2 \cdots x_k A_k x_{k+1}$ where $A_i \in N - \{S\}$ for $1 \leq i \leq k$, $x_1 \in R_\rho(T)$, $x_i \in LR_\rho^+(T)$ for $2 \leq i \leq k$ and $x_{k+1} \in L_\rho(T)$;
2. $A \rightarrow y_1 B_1 y_2 B_2 \cdots y_t B_t y_{t+1}$ where $A, B_i \in N - \{S\}$ for $1 \leq i \leq t$, $y_i \in LR_\rho^+(T)$ for $2 \leq i \leq t$; or
3. $A \rightarrow x$ where $A \in N - \{S\}$ and $x \in LR_\rho^*(T)$.

Remark 1. The elements $\begin{bmatrix} u \\ v \end{bmatrix}$ in the set of all pairs of strings $T \times T$ can be classified into two cases, whether in the form of $\begin{bmatrix} u \\ v \end{bmatrix} \neq \begin{bmatrix} \lambda \\ \lambda \end{bmatrix}$ or $\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \lambda \\ \lambda \end{bmatrix}$.

The derivation step for the static Watson-Crick context-free grammar is shown in the following. Let $G = (N, T, \rho, S, P)$ be a static Watson-Crick context-free grammar. We say that α derives β in G , denoted or written as $\alpha \Rightarrow \beta$ such that

1. $\alpha = S$ and $\beta = x_1 A_1 x_2 A_2 x_3 \cdots x_k A_k x_{k+1}$ where $\alpha \Rightarrow \beta \in P$;
2. $\alpha = xAy$ and $\beta = xx_1 B_1 x_2 B_2 x_3 \cdots x_k B_k x_{k+1} y$ where $A, B \in N - \{S\}$, $x, y \in (LR_\rho(T) \cup N)^*$ and $A \Rightarrow x_1 B_1 x_2 B_2 x_3 \cdots x_k B_k x_{k+1} \in P$; or
3. $\alpha = xAy$ and $\beta = x \begin{bmatrix} u \\ v \end{bmatrix} y$ where $A, B \in N - \{S\}$, $x \in R_\rho(T)$, $y \in L_\rho(T)$ and $\begin{bmatrix} u \\ v \end{bmatrix} \in LR_\rho^*(T)$.

The reflexive and transitive closure of \Rightarrow or (\Rightarrow) is denoted by \Rightarrow_G^* or $(\Rightarrow)^*$. The language generated by a static WK context-free grammar G , denoted by $L(G)$, is defined as $L(G) = \left\{ u: \begin{bmatrix} u \\ v \end{bmatrix} \in WK_\rho(T) \text{ and } S \Rightarrow_G^* \begin{bmatrix} u \\ v \end{bmatrix} \right\}$. The family of languages generated by static WK context-free grammar is denoted by **SCF**.

Next, an example is illustrated to show the family of languages generated by **SCF**.

Example 1. Let $G = (\{S, A, B\}, \{a, b, c, d\}, \{(a, a), (b, b), (c, c), (d, d)\}, S, P)$ be a static WK context-free grammar, where P consists of the following rules:

$$\begin{aligned}
 S &\rightarrow \begin{bmatrix} a \\ a \end{bmatrix} \binom{\lambda}{a} A \binom{d}{\lambda} \begin{bmatrix} d \\ d \end{bmatrix}, \\
 A &\rightarrow \binom{a}{\lambda} \begin{bmatrix} a \\ a \end{bmatrix} \binom{\lambda}{a} A \binom{d}{\lambda} \begin{bmatrix} d \\ d \end{bmatrix} \binom{\lambda}{d}, \\
 A &\rightarrow \binom{a}{\lambda} \begin{bmatrix} b \\ b \end{bmatrix} \binom{\lambda}{b} B \binom{c}{\lambda} \begin{bmatrix} c \\ c \end{bmatrix} \binom{\lambda}{d}, \\
 B &\rightarrow \binom{b}{\lambda} \begin{bmatrix} b \\ b \end{bmatrix} \binom{\lambda}{b} B \binom{c}{\lambda} \begin{bmatrix} c \\ c \end{bmatrix} \binom{\lambda}{c}, \\
 B &\rightarrow \binom{b}{\lambda} \binom{\lambda}{c}.
 \end{aligned}$$

From this, we obtain the derivation:

$$\begin{aligned}
 S &\Rightarrow^* \begin{bmatrix} a^{n-1} \\ a^{n-1} \end{bmatrix} \binom{\lambda}{a} A \binom{d}{\lambda} \begin{bmatrix} d^{n-1} \\ d^{n-1} \end{bmatrix} \\
 &\Rightarrow \begin{bmatrix} a^n b \\ a^n b \end{bmatrix} \binom{\lambda}{b} B \binom{c}{\lambda} \begin{bmatrix} c d^n \\ c d^n \end{bmatrix} \\
 &\Rightarrow^* \begin{bmatrix} a^n b^{m-1} \\ a^n b^{m-1} \end{bmatrix} \binom{\lambda}{b} B \binom{c}{\lambda} \begin{bmatrix} c^{m-1} d^n \\ c^{m-1} d^n \end{bmatrix} \\
 &\Rightarrow \begin{bmatrix} a^n b^m c^m d^n \\ a^n b^m c^m d^n \end{bmatrix}.
 \end{aligned}$$

Hence, G generates the language $L(G) = \{a^n b^m c^m d^n \mid n, m \geq 1\}$.

In the investigation of the generative power of static Watson-Crick context-free grammars, we observe the results on the relationships between the families of languages generated by static Watson-Crick grammars to the families of Chomsky languages and Watson-Crick languages.

The following two lemmas immediately follow from the definition of static Watson-Crick context-free grammars. Lemma 1 shows the inclusion between context-free languages with static WK context-free languages; while Lemma 2 presents the inclusion between static Watson-Crick regular, linear and context-free languages.

4.2 Lemma 1. $CF \subseteq SCF$.

Proof. For a context-free grammar $G = (N, T, S, P)$, its static WK variant $G' = (N, T, \rho, S, P')$ is defined where $\rho = \{(a, a) \mid a \in T\}$ and for each production $A \rightarrow \alpha \in P$, every terminal string x in α is changed to $\begin{bmatrix} x \\ x \end{bmatrix}$. Then, it is easy to see that $L(G') = L(G)$. ■

4.3 Lemma 2. The following inclusion holds:

$$SREG \subseteq SLIN \subseteq SCF.$$

Proof. The inclusion follows from the definition of static Watson-Crick grammars and by referring to the Chomsky hierarchy.

Next, we show that the language generated by static Watson-Crick context-free grammar can generate some non-context-free languages to relate the generative power in Lemma 2 with the result in [14, 15].

4.4 Lemma 3. SCF - CF $\neq \emptyset$

Proof. Consider the static WK context-free grammar $G = (\{S, A, B, C, D, E\}, \{a, b, c\}, \rho, S, P)$ where P contains the following productions:

1. $S \rightarrow SS,$
2. $S \rightarrow \begin{bmatrix} a \\ a \end{bmatrix} \begin{pmatrix} a \\ \lambda \end{pmatrix} A \begin{pmatrix} c \\ \lambda \end{pmatrix} \begin{bmatrix} c \\ c \end{bmatrix},$
3. $A \rightarrow \begin{pmatrix} a \\ \lambda \end{pmatrix} A \begin{pmatrix} c \\ \lambda \end{pmatrix},$
4. $A \rightarrow \begin{pmatrix} b \\ \lambda \end{pmatrix} B,$
5. $B \rightarrow \begin{pmatrix} \lambda \\ a \end{pmatrix} C,$
6. $C \rightarrow \begin{pmatrix} b \\ \lambda \end{pmatrix} B,$
7. $C \rightarrow \begin{pmatrix} \lambda \\ b \end{pmatrix} D \begin{pmatrix} \lambda \\ c \end{pmatrix},$
8. $D \rightarrow \begin{pmatrix} \lambda \\ b \end{pmatrix} D \begin{pmatrix} \lambda \\ c \end{pmatrix},$
9. $D \rightarrow \begin{bmatrix} b \\ b \end{bmatrix}.$

Thus, the derivation for each of the production rules is defined as follows:

Step 1. From rule (i):

$$S \Rightarrow SS. \quad (1)$$

Step 2. Derivation (1) can be continued with rule (ii):

$$S \Rightarrow \begin{bmatrix} a \\ a \end{bmatrix} \begin{pmatrix} a \\ \lambda \end{pmatrix} A \begin{pmatrix} c \\ \lambda \end{pmatrix} \begin{bmatrix} c \\ c \end{bmatrix} S \quad (2)$$

Step 3. Derivation (2) can be continued with rule (iii) or rule (iv). Without the loss of generality, we apply rule (iii) $k \geq 0$ times and apply rule (iv):

$$S \Rightarrow^* \begin{bmatrix} a \\ a \end{bmatrix} \begin{pmatrix} a \\ \lambda \end{pmatrix} \begin{pmatrix} a^k \\ \lambda \end{pmatrix} \begin{pmatrix} b \\ \lambda \end{pmatrix} A \begin{pmatrix} c^k \\ \lambda \end{pmatrix} \begin{pmatrix} c \\ \lambda \end{pmatrix} \begin{bmatrix} c \\ c \end{bmatrix} S = \begin{bmatrix} a \\ a \end{bmatrix} \begin{pmatrix} a^{k+1} \\ \lambda \end{pmatrix} \begin{pmatrix} b \\ \lambda \end{pmatrix} A \begin{pmatrix} c^{k+1} \\ \lambda \end{pmatrix} \begin{bmatrix} c \\ c \end{bmatrix} S. \quad (3)$$

Step 4. Derivation (3) can only be continued with rule (v):

$$S \Rightarrow^* \begin{bmatrix} a \\ a \end{bmatrix} \begin{pmatrix} a^{k+1} \\ \lambda \end{pmatrix} \begin{pmatrix} b \\ \lambda \end{pmatrix} \begin{pmatrix} \lambda \\ a \end{pmatrix} C \begin{pmatrix} c^{k+1} \\ \lambda \end{pmatrix} \begin{bmatrix} c \\ c \end{bmatrix} S = \begin{bmatrix} aa \\ aa \end{bmatrix} \begin{pmatrix} a^k \\ \lambda \end{pmatrix} \begin{pmatrix} b \\ \lambda \end{pmatrix} C \begin{pmatrix} c^{k+1} \\ \lambda \end{pmatrix} \begin{bmatrix} c \\ c \end{bmatrix} S. \quad (4)$$

Step 5. Derivation (4) can be continued with rule (vi) and rule (vii). Rule (vi) must be applied k times to complete the lower strand of $\binom{a^k}{\lambda}$, which results in applying rule (v) k times, and then we apply rule (vii):

$$S \Rightarrow^* \begin{bmatrix} aa \\ aa \end{bmatrix} \binom{a^k}{\lambda} \binom{b}{\lambda} \binom{b^k}{\lambda} c \binom{c^{k+1}}{\lambda} \begin{bmatrix} c \\ c \end{bmatrix} S = \begin{bmatrix} a^{k+2} \\ a^{k+2} \end{bmatrix} \begin{bmatrix} b \\ b \end{bmatrix} \binom{b^k}{\lambda} D \binom{c^k}{\lambda} \begin{bmatrix} cc \\ cc \end{bmatrix} S. \quad (5)$$

Step 6. To complete derivation (5), we apply rule (viii) k times to complete the lower strand of $\binom{b^k}{\lambda}$ and $\binom{c^k}{\lambda}$. The derivation is completed with rule (ix) and we apply again the similar step to complete the whole derivation of $L(G)$:

$$S \Rightarrow^* \begin{bmatrix} a^{k+2} b^{k+2} c^{k+2} \\ a^{k+2} b^{k+2} c^{k+2} \end{bmatrix} S = \begin{bmatrix} a^{k+2} b^{k+2} c^{k+2} & a^{k+2} b^{k+2} c^{k+2} \\ a^{k+2} b^{k+2} c^{k+2} & a^{k+2} b^{k+2} c^{k+2} \end{bmatrix}.$$

Thus, $L(G) = \{a^{n_1} b^{n_1} c^{n_1} a^{n_2} b^{n_2} c^{n_2} \mid n_1, n_2 \geq 2\}$. ■

Next, the result shows that the family of Watson-Crick context-free languages is included in the family of static Watson-Crick context-free languages.

4.5 Lemma 4. The following inclusion holds:

WKCF \subseteq SCF.

Proof. Let $G = (N, T, \rho, S, P)$ be a WK context-free grammar (WKCF) in a modified Chomsky normal form. We construct a static WK context-free grammar (SCF) where $G' = (N', T, \rho, S, P')$ such that $L(G) = L(G')$. Without loss of generality, we assume that $m \geq n$. We set $N' = N \cup \{A' \mid A \in N\}$ where A' 's are new nonterminals and P' contains the productions of P in the form of $A \rightarrow BC, A \rightarrow \binom{\lambda}{\lambda}$ and the set of productions

$$A \rightarrow \binom{u_1 u_2 \cdots u_m}{\lambda} A', \quad A' \rightarrow \binom{\lambda}{v_1 v_2 \cdots v_n},$$

$$A \rightarrow \binom{u_1 u_2 \cdots u_{m-1}}{\lambda} \begin{bmatrix} u_m \\ v_1 \end{bmatrix} \binom{\lambda}{v_2 \cdots v_n},$$

$$A \rightarrow \binom{u_1 u_2 \cdots u_{m-2}}{\lambda} \begin{bmatrix} u_{m-1} u_m \\ v_1 v_2 \end{bmatrix} \binom{\lambda}{v_3 \cdots v_n},$$

$$A \rightarrow \binom{\lambda}{v_1 v_2 \cdots v_n} A', \quad A' \rightarrow \binom{u_1 u_2 \cdots u_m}{\lambda},$$

$$A \rightarrow \binom{\lambda}{v_1 v_2 \cdots v_{n-1}} \begin{bmatrix} u_1 \\ v_n \end{bmatrix} \binom{u_2 \cdots u_m}{\lambda},$$

for each production $A \rightarrow \binom{u_1 u_2 \cdots u_m}{v_1 v_2 \cdots v_n} \in P$.

First, we need to show that $L(G) \subseteq L(G')$. For $w \in L(G)$, we consider the following derivation where $S \Rightarrow w_1 \Rightarrow w_2 \Rightarrow \cdots \Rightarrow w_k \Rightarrow w_{k+1} \Rightarrow \cdots \Rightarrow w$ for $k \geq 1$ and

w_{k+1} is obtained from w_k by applying rule $A \rightarrow \langle u/v \rangle \in P$ where $u = u_1 u_2 \cdots u_m$, $v = v_1 v_2 \cdots v_n$ and $m \geq n$. There are some cases in the “sticking” of the symbols, u and v whereby

1. The symbols u and v are not complementary to each other.
2. The symbols u and v are complement to each other partially or completely.

For example, let $w = x \begin{pmatrix} u_{i_1} u_{i_2} \cdots u_{i_s} \\ v_{j_1} v_{j_2} \cdots v_{j_s} \end{pmatrix} y$. In this case, we replace $A \rightarrow \langle u/v \rangle$ in the corresponding derivation in G' with production

$$\begin{aligned}
 A &\rightarrow \begin{pmatrix} u_{i_1} u_{i_2} \cdots u_{i_s} \\ \lambda \end{pmatrix} A', \quad A' \rightarrow \begin{pmatrix} \lambda \\ v_{j_1} v_{j_2} \cdots v_{j_s} \end{pmatrix}, \\
 A &\rightarrow \begin{pmatrix} u_{i_1} u_{i_2} \cdots u_{i_{1-1}} \\ \lambda \end{pmatrix} \begin{matrix} [u_{i_s}] \\ [v_{j_1}] \end{matrix} \begin{pmatrix} \lambda \\ v_{j_2} \cdots v_{j_s} \end{pmatrix}, \\
 A &\rightarrow \begin{pmatrix} u_{i_1} u_{i_2} \cdots u_{i_{1-2}} \\ \lambda \end{pmatrix} \begin{matrix} [u_{i_{1-1}} u_{i_s}] \\ [v_{j_1} v_{j_2}] \end{matrix} \begin{pmatrix} \lambda \\ v_{j_3} \cdots v_{j_s} \end{pmatrix}, \\
 A &\rightarrow \begin{pmatrix} \lambda \\ v_{j_1} v_{j_2} \cdots v_{j_{1-1}} \end{pmatrix} \begin{matrix} [u_{i_1}] \\ [v_{j_s}] \end{matrix} \begin{pmatrix} u_{i_2} \cdots u_{i_s} \\ \lambda \end{pmatrix}.
 \end{aligned}$$

In a derivation $S \Rightarrow^* w$ in G , every application of production rule $A \rightarrow \langle u/v \rangle$ is replaced with all the possibilities of production rules as stated above which results in the derivation $S' \Rightarrow S \Rightarrow^* w$ in G' . The similar arguments also holds to show for $L(G') \subseteq L(G)$.

Combining the results above with the results from [3, 15], we obtain the following relation in the hierarchy form. The relation in Figure 1 holds where the solid arrows represent the proper inclusions of the lower families into the upper families, while the dotted arrow represents the inclusions.

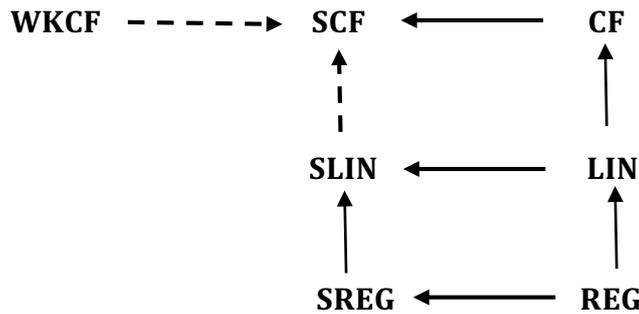


Fig. 1. The hierarchy of static Watson-Crick, Watson-Crick and Chomsky language families.

5 Conclusion

In this paper, we defined a new theoretical model known as the static Watson-Crick context-free grammar and investigated some of their generative powers. Based on the results obtained, we can conclude that:

1. The family of context-free languages is strictly included in the family of static Watson-Crick context-free languages,
2. Static Watson-Crick context-free grammars can generate non context-free languages,
3. The family of Watson-Crick context-free languages is included in the family of static Watson-Crick context-free languages.

The contribution from the results of this research leads to the development of different variants of WK grammars which are useful for DNA based computing devices and algorithmic techniques.

6 Acknowledgements:

This research work is supported by Ministry of Education (MOE) and Research Management Centre (RMC), Universiti Teknologi Malaysia (UTM) through Research University Grant Vote No. 5F022 and UTM under Zamalah Scholarship.

7 References

- [1] Adleman, L.M., Molecular Computation of Solutions to Combinatorial Problems. *Science*, 1994. 266(5187): 1021-1024. <https://doi.org/10.1126/science.7973651>
- [2] Kari, L., Paun, G., Rozenberg, G., Salomaa, A and Yu, S. DNA Computing, Sticker Systems and Universality. *Acta Informatica*. 1998. 35: 401- 420. <https://doi.org/10.1007/s002360050125>
- [3] Paun, G., Rozenberg, G. and Salomaa, A. *DNA Computing: New Computing Paradigms*. New York: Springer-Verlag Berlin Heidelberg. 1998.
- [4] Mohd Sebry, N. A., Hamzah, N. Z. A., Sarmin, N. H., Fong, W. H. and Turaev, S. Sticker System over Monoid. *Malaysian Journal of Fundamental and Applied Sciences*, 2012. 8(3): 127-132. <https://doi.org/10.11113/mjfas.v8n3.136>
- [5] Mathuri, S., Fong, W. H., Sarmin, N.H. and Turaev, S. Probabilistic Sticker Systems. *Malaysian Journal of Fundamental and Applied Sciences*, 2013. 9(3): 150-155.
- [6] Gan, Y.S., Fong, W. H., Sarmin, N.H. and Turaev, S. The Generative Power of Weighted One-Sided and Regular Sticker Systems. *AIP Conference Proceedings*, 2014, 1602:855-862. <https://doi.org/10.1063/1.4882584>
- [7] Freund, R., Paun, G. Rozenberg, and Salomaa, A. Watson-Crick Finite Automata. *Proc. 3rd DIMACS Workshop on DNA based Computers*. 1997. Philadelphia: 297-328. <https://doi.org/10.1090/dimacs/048/22>
- [8] Czeizler, E., Czeizler, E., Kari, L. and Salomaa, K. On the descriptive complexity of Watson-Crick automata. *Theoretical Computer Science*. 2009. 410(35): 3250-3260. <https://doi.org/10.1016/j.tcs.2009.05.001>

- [9] Petre, E. Watson-Crick Omega-automata. *Journal of Automata Languages and Combinatorics*. 2003. 8(1): 59-70.
- [10] Mohd Tamrin, M.I., Turaev, S. and Tengku Sembok, T.M. Weighted Watson-Crick Automata. *AIP Conference Proceedings*. 2014. 1605(302): 302-306.
<https://doi.org/10.1063/1.4887606>
- [11] Sutapa, D. and Mukhopadhyay, S. A Composite Method Based on Formal Grammar and DNA Structural Features in Detecting Human Polymerase II Promoter Region. *PLOS One*. 2013. 8(2): e54843. <https://doi.org/10.1371/journal.pone.0054843>
- [12] Algwaiz, A., Ammar, R., and Rajasekaran, S. Framework for Data Mining of Big Data Using Probabilistic Grammars. In *e-Learning (econf), 2015 Fifth International Conference on*. IEEE, 2015. 241-246. <https://doi.org/10.1109/econf.2015.50>
- [13] Zulkufli, N.L.M., Turaev, S., Tamrin, M.I.M. and Azeddine, M., Closure Properties of Watson-Crick Grammars. In *AIP Conference Proceedings*. 2015 December 11. 2015 Dec 11. AIP Publishing. 040032. <https://doi.org/10.1063/1.4937082>
- [14] Rahman, A. A., Fong, W. H., Sarmin, N. H., Turaev, S. and Zulkufli, N. L. M. Static Watson-Crick regular grammar, *Malaysian Journal of Applied and Fundamental Sciences*. 2018. 14: 457-462. <https://doi.org/10.11113/mjfas.v14n0.1282>
- [15] Rahman, A. A., Fong, W. H., Sarmin, N. H., Turaev, S. and Zulkufli, N. L. M. Static Watson-Crick linear grammars and its computational power. *International Graduate Conference on Engineering, Science and Humanities (IGCESH 2018)* (accepted).
- [16] Zulkufli, N.L.M., Turaev, S., Tamrin, M.I.M. and Azeddine, M., Watson-Crick Context-Free Grammars: Grammar Simplifications and a Parsing Algorithm. *The Computer Journal*. 2018. 61(9):1361-1373. <https://doi.org/10.1093/comjnl/bxx128>

8 Authors

Wan Heng Fong is with the Department of Mathematical Sciences, Faculty of Science, Universiti Teknologi Malaysia, UTM, Johor Bahru, Johor, Malaysia. E mail fwh@utm.my

Aqilahfarhana Abdul Rahman is with the Department of Mathematical Sciences, Faculty of Science, Universiti Teknologi Malaysia, UTM, Johor Bahru, Johor, Malaysia.

Nor Haniza Sarmin is with the Department of Mathematical Sciences, Faculty of Science, Universiti Teknologi Malaysia, UTM, Johor Bahru, Johor, Malaysia.

Sherzod Turaev is with the Faculty of Engineering and Natural Sciences, International University of Sarajevo, Hrasnicka cesta 15, 71210, Ilidza, Sarajevo, Bosnia and Herzegovina.

Article submitted 2019-03-12. Resubmitted 2019-04-24. Final acceptance 2019-05-02. Final version published as submitted by the authors.