ASM Sc. J., 13, 2020
https://doi.org/10.32802/asmscj.2020.sm26(5.1)

1-Normal Form for Static Watson-Crick Regular and Linear Grammars

Aqilahfarhana Abdul Rahman[1], Wan Heng Fong[2*], Nor Haniza Sarmin[3] and Sherzod Turaev[4]

[1,2,3]*Department of Mathematical Sciences, Faculty of Science, Universiti Teknologi Malaysia, 81310 UTM Skudai, Johor, Malaysia.*

[4]*Sherzod Turaev Department of Computer Science & Software Engineering, College of Information Technology, United Arab Emirates University, Al Ain 15551, United Arab Emirates*

In DNA computing, there are various formal language theoretical approaches that involves the recombinant behavior of DNA sequences. A Watson-Crick automaton is a mathematical model that represents the biological properties of DNA based on the Watson-Crick complementarity of DNA molecules. Meanwhile, a sticker system is another DNA computing models which uses the sticker operation to form complete double-stranded sequences. Previously, Watson-Crick grammars have been introduced as the grammar counterparts of Watson-Crick automata which generate double-stranded strings using the Watson-Crick complementarity rules. Following that, static Watson-Crick grammars are introduced, where both stranded strings are generated dependently by checking for the Watson-Crick complementarity of each complete substring. In formal language theory, normal forms, such as Chomsky Normal Form (CNF) are defined by imposing the restrictions on the rules contained in context-free grammars. However, in previous research, 1-normal form for a Watson-Crick linear grammar was defined. In this research, 1-normal forms are introduced for both static Watson-Crick regular and linear grammars. Moreover, the implementation of 1-normal form is also presented by investigating the computational properties between the static Watson-Crick regular and linear grammars. The results from this research, hence, simplify the length of the rules in the grammars, which are useful for studying computational properties of Watson-Crick grammars.

Keywords: DNA computing, formal languages, context-free grammars, static Watson-Crick grammars, normal form

I. INTRODUCTION

DNA (Deoxyribonucleic acid) computing contains various formal language theoretical approaches that broadly use the recombinant behavior of DNA sequences under the effect of enzymatic activities. DNA is a polymer which is constructed from monomers namely deoxyribonucleotides.

Each deoxyribonucleotides consists of three components; a sugar, a phosphate group, and a nitrogenous base. The four nitrogenous bases are adenine (A), thymine (T), guanine (G), and cytosine (C). The adenine (A) and guanine (G) bases are double-ring molecules called purines; whereas the cytosine (C) and thymine (T) bases are single-ring molecules called pyrimidine (P ãun et al., 1998).

A DNA molecule is composed of two DNA strands which are held together by the hydrogen bonds between the paired bases. There are two fundamental features of DNA molecules known as Watson-Crick (WK)

*Corresponding author's e-mail: fwh@utm.my

complementarity and massive parallelism of DNA strands. WK complementarity is a base pairing where a purine always binds with a pyrimidine, but each purine binds to one particular type of pyrimidine only; meanwhile, massive parallelism of DNA strands allows construction of many copies of DNA strands where numerous operations are carried out on the encoded information simultaneously (Pãun et al., 1998; Ng et al., 2007).

DNA computation has been marked by Adleman (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. (1998) proposed a mathematical model known as a sticker system which uses the sticker operation on DNA to form complete double-stranded sequences. Following that, Freund et al. (1997) 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 grammatical studies of DNA strands started in 2012 when Subramanian et al. (2012) introduced the WK regular grammar, and later, modified variants for different types of grammars were defined in (Zulkufli et al., 2016). The research is motivated by the synthesis processes in DNA replication which can be simulated by derivations in the WK grammars. Although these WK grammars use different restriction of production rules, all of them generate double-stranded strings dynamically: the WK complementarity can only be checked once generating both strands of a complete double-stranded string. Motivated by the WK grammars, the static WK grammars are proposed as a grammar counterpart of the sticker systems (Abdul Rahman et al., 2018a; 2018b). This new theoretical model generates both stranded strings dependently by checking for WK complementarity of each complete substring and also illustrate the replication of DNA in DNA molecules.

Next, a normal form is defined by imposing the restrictions of the rules contained in the grammar. In Chomsky grammars, the normal forms are implemented for context-free grammars and context-sensitive grammars (Ito et al., 2010). The most important normal forms of context-free grammars are the Chomsky Normal Form and Greibach Normal Form (Levelt, 1974). For WK grammars, Zulkufli et al. (2016) have introduced 1-normal form for WK linear grammars and showed that for every WK linear grammar, there exist an equivalent WK linear grammar in 1-normal form. In this research, we investigate the 1-normal form for each static WK regular and linear grammars.

This paper is organized as follows: Section 1 introduces the background of the research. In Section 2, some preliminary concepts involve the basic terms of strings, languages and grammars, sticker systems as well as dynamic and static WK grammars are presented. In Section 3, the normal forms for static WK regular and linear grammars are introduced.

II. PRELIMINARIES

This section includes some preliminary concepts which involves the basic terms and definitions that are used in this paper. The reader may refer to (Pãun et al., 1998; Linz, 2006) for detailed information regarding on the basic concepts of strings, languages, grammars and sticker systems.

In this paper, the symbol $\in$ denotes the membership of an element to a set. Let T be an alphabet which is a nonempty finite set of abstract symbols, then T^* is a set of all strings, a finite sequence of symbols (words) over T. A string with no symbol is called the empty string and denoted by λ. The set T^+ is defined as the set of all nonempty finite strings over T, i.e., $T^+ = T^* - \lambda$.

A *Chomsky grammar* is defined as a quadruple $G = (N, T, \rho, S, P)$ where the alphabet N is defined as the *nonterminal* alphabet, T is the *terminal* alphabet, $S \in N$ is the axiom or the start symbol, and $P \subseteq (N \cup T)^* N (N \cup T)^*$ is a set of production rules of G. The rules $(x, y) \in P$ are written in the form of $x \to y$. We say that u *directly derives* v or v is derived from u with respect to G, which is written as $u \Rightarrow v$, if and only if $u = u_1 x u_2$, $v = u_1 y u_2$, for some $u_1, u_2 \in (N \cup T)^*$ and $x \to y \in P$. The reflexive and transitive closure of $\Rightarrow$ is denoted by $\Rightarrow^*$.

A grammar can normally generate many strings by applying the rules in arbitrary order. The set of all terminal strings is the language generated by the grammar which is defined by $L(G) = \{w \in T^*: S \Rightarrow^* w\}$.

The *Chomsky grammars* are classified depending on their respective form of production rules. A grammar $G = (N, T, \rho, S, P)$ is called *context-sensitive*, if each rule $u \to v \in P$ 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 $x \in (N \cup T)^+$; *context-free*, if each rule $u \to v \in P$ has $u \in N$; *linear*, if each rule $u \to v \in P$ has $u \in N$ and $v \in T^* \cup T^*NT^*$; *regular*, if each rule $u \to v \in P$ has $u \in N$ and $v \in T \cup TN \cup \{\lambda\}$ (P ãun et al., 1998).

All those families of languages generated by *context-sensitive*, *context-free*, *linear* and *regular* grammars 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 languages.

Further, we recall the definitions of Watson-Crick grammars:

Definition 1. (Zulkufli et al., 2016)
A Watson-Crick (WK) grammar $G = (N, T, \rho, S, P)$ is called *regular,* if each production has the form $A \to \langle u/v \rangle$ where $A, B \in N$ and $\langle u/v \rangle \in \langle T^*/T^* \rangle$; *linear,* if each production has the form $A \to \langle u_1/v_1 \rangle B \langle u_2/v_2 \rangle$ or $A \to \langle u/v \rangle$ 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 $A \to \alpha$ where $A \in N$ and $\alpha \in (N \cup \langle T^*/T^* \rangle)^*$.

In order to generate or form a complete double-stranded sequence of DNA, the sticker system uses a sticker operation on DNA molecules (P ãun and Rozenberg, 1998). Let V be an alphabet for a symmetric relation $\rho \in V \times V$ over V. The set $WK_\rho(V) = \begin{bmatrix} V \\ V \end{bmatrix}_\rho^*$ where $\begin{bmatrix} V \\ V \end{bmatrix}_\rho = \left\{ \begin{bmatrix} a \\ b \end{bmatrix} \mid a, b \in V, \begin{pmatrix} a \\ b \end{pmatrix} \in \rho \right\}$ denotes the Watson-Crick domain associated to 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 strings w_1 is the upper strand and w_2 is the lower strand of the molecule.

Apart from that, the set of incomplete molecules are denoted as $W_\rho(V) = L_\rho(V) \cup R(V) \cup LR_\rho(V)$, where

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

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

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

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

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

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

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

Next, we recall the definition of static WK regular grammar and static WK linear grammar. Since static WK regular grammar consists of right-linear and left-linear grammar, then we state only for right-linear grammar (Rahman et al., 2018a) in this paper.

Definition 2. (Rahman et al., 2018a)
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, $\rho \in T \times T$ 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

(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 $A, B \in N - \{S\}$, $\begin{pmatrix} x \\ y \end{pmatrix} \in LR_\rho{}^*(T)$; 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 3. (Rahman et al., 2018b)
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, $\rho \in T \times T$ 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

(i) $S \to \begin{bmatrix} u_1 \\ v_1 \end{bmatrix} \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} A \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} \begin{bmatrix} u_2 \\ v_2 \end{bmatrix}$ where $A \in N - \{S\}$, $\begin{bmatrix} u_1 \\ v_1 \end{bmatrix} \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} \in R_\rho(T)$, $\begin{pmatrix} x_2 \\ y_2 \end{pmatrix} \begin{bmatrix} u_2 \\ v_2 \end{bmatrix} \in L_\rho(T)$;

(ii) $A \to \binom{x_1}{y_1} B \binom{x_2}{y_2}$ where $A, B \in N - \{S\}$, $\binom{x_1}{y_1}, \binom{x_2}{y_2} \in LR_\rho{}^*(T)$; or

(iii) $A \to \binom{x_1}{y_1}$ where $A \in N - \{S\}$, $\binom{x_1}{y_1} \in LR_\rho{}^*(T)$.

Next, the 1-normal form for static WK regular and linear grammars are discussed in the following section.

III. RESULTS AND DISCUSSIONS

The normal form represents the standardized form for the production rules in the grammars. Therefore in this section, we define a 1-normal form for static WK regular and linear grammars where the order of each upper and lower strand is less than or equal to one, such that $|u_i| \leq 1, |v_i| \leq 1$. The following shows a 1-normal form for static WK regular grammar as stated in Definition 4 and Lemma 1.

Definition 4. A static WK regular grammar $G = (N, T, \rho, S, P)$ is said to be in 1-normal form if each production in P has one of the following forms:

(i) $A \to \binom{a}{\lambda} B \mid \binom{a}{\lambda}$,

(ii) $A \to \binom{\lambda}{a} B \mid \binom{\lambda}{a}$,

(iii) $A \to \binom{\lambda}{\lambda} B \mid \binom{\lambda}{\lambda}$,

(iv) $A \to \begin{bmatrix} a \\ b \end{bmatrix} B \mid \begin{bmatrix} a \\ b \end{bmatrix}$,

where $A, B \in N$ and $(a, b) \in \rho$.

Lemma 1. For every static WK regular grammar, there exists a static WK regular grammar in 1-normal form.

Proof. Let $G = (N, T, \rho, S, P)$ be a static WK regular grammar. By definition, P can be divided into three subsets:

(i) $P_1 = \left\{ S \to \begin{bmatrix} u \\ v \end{bmatrix} \binom{x}{y} A \in P \mid \begin{bmatrix} u \\ v \end{bmatrix} \binom{x}{y} \in R_\rho(T), A \in N - \{S\} \right\}$;

(ii) $P_2 = \left\{ A \to \binom{x}{y} B \in P \mid \binom{x}{y} \in LR_\rho{}^*(T), A, B \in N - \{S\} \right\}$; or

(iii) $P_3 = \left\{ A \to \binom{x}{y} \begin{bmatrix} u \\ v \end{bmatrix} \in P \mid \binom{x}{y} \begin{bmatrix} u \\ v \end{bmatrix} \in L_\rho(T), A \in N - \{S\} \right\}$;

i.e., $P = P_1 \cup P_2 \cup P_3$. Without loss of generality, consider $r: A \to \binom{x}{y} B$ where $\binom{x}{y} = \binom{a_1 a_2 \cdots a_k}{\lambda} \begin{bmatrix} b_1 b_2 \cdots b_n \\ c_1 c_2 \cdots c_n \end{bmatrix} \binom{\lambda}{d_1 d_2 \cdots d_m}$. We construct the sequence of new productions $\{r\}: A \to \binom{a_1}{\lambda} B_r{}^1, \cdots, B_r{}^{k-1} \to \binom{a_k}{\lambda} B_r{}^k, B_r{}^k \to \begin{bmatrix} b_1 \\ c_1 \end{bmatrix} C_r{}^1, \cdots, C_r{}^{n-1} \to \begin{bmatrix} b_n \\ c_n \end{bmatrix} C_r{}^n, C_r{}^n \to \binom{d_1}{\lambda} D_r{}^1, \cdots, D_r{}^{m-1} \to \binom{d_m}{\lambda} B$ where $B_r{}^i, C_r{}^j$ and $D_r{}^k$ are new nonterminals that are only used in the rule r.

Next, we define a static WK regular grammar $G' = (N', T, \rho, S, P')$ where N' contains the nonterminals of N and all new nonterminals introduced above and P' contains the productions constructed above. Hence, every production r in P can be replaced with the corresponding sequence $\{r\}$ of productions in P' and vice versa. Therefore, $L(G) = L(G')$. ∎

Next, the following definition and lemma of 1-normal form for static WK linear grammars are given.

Definition 5. A static WK linear grammar $G = (N, T, \rho, S, P)$ is said to be in 1-normal form if each production in P has one of the following forms:

(i) $A \to \binom{a}{\lambda} B \mid B \binom{a}{\lambda} \mid \binom{a}{\lambda}$,

(ii) $A \to \binom{\lambda}{a} B \mid B \binom{\lambda}{a} \mid \binom{\lambda}{a}$,

(iii) $A \to \binom{\lambda}{\lambda} B \mid B \binom{\lambda}{\lambda} \mid \binom{\lambda}{\lambda}$,

(iv) $A \to \begin{bmatrix} a \\ b \end{bmatrix} B \mid B \begin{bmatrix} a \\ b \end{bmatrix} \mid \begin{bmatrix} a \\ b \end{bmatrix}$,

where $A, B \in N$ and $(a, b) \in \rho$.

Lemma 2. For every static WK linear grammar, there exists a static WK linear grammar in 1-normal form.

Proof. Let $G = (N, T, \rho, S, P)$ be a static WK linear grammar. By definition, P can be divided into three subsets:

(i) $P_1 = \left\{ S \to \begin{bmatrix} u_1 \\ v_1 \end{bmatrix} \binom{x_1}{y_1} A \binom{x_2}{y_2} \begin{bmatrix} u_2 \\ v_2 \end{bmatrix} \in P \mid \begin{bmatrix} u_1 \\ v_1 \end{bmatrix} \binom{x_1}{y_1} \in R_\rho(T), \binom{x_2}{y_2} \begin{bmatrix} u_2 \\ v_2 \end{bmatrix} \in L_\rho(T) \text{ and } A \in N - \{S\} \right\}$;

(ii) $P_2 = \left\{A \to \binom{x_1}{y_1} B \binom{x_2}{y_2} \in P \mid \binom{x_1}{y_1}, \binom{x_2}{y_2} \in LR_\rho{}^*(T), A, B \in N - \{S\}\right\}$; or

(iii) $P_3 = \left\{A \to \binom{x_1}{y_1} \in P \mid \binom{x_1}{y_1} \in LR_\rho{}^*(T), A \in N - \{S\}\right\}$;

i.e., $P = P_1 \cup P_2 \cup P_3$. Without loss of generality, consider $r: S \to \begin{bmatrix}u_1\\v_1\end{bmatrix} \binom{x_1}{y_1} A \binom{x_2}{y_2} \begin{bmatrix}u_2\\v_2\end{bmatrix} = \begin{bmatrix}a_1a_2\cdots a_n\\a_1a_2\cdots a_n\end{bmatrix} \binom{b_1b_2\cdots b_k}{\lambda} A \binom{c_1c_2\cdots c_l}{\lambda} \begin{bmatrix}d_1d_2\cdots d_m\\d_1d_2\cdots d_m\end{bmatrix}$.

We construct the sequence of new productions $\{r\}: S \to \begin{bmatrix}a_1\\a_1\end{bmatrix} B_r{}^1, B_r{}^1 \to \begin{bmatrix}a_2\\a_2\end{bmatrix} B_r{}^2, \cdots,$

$B_r{}^{n-1} \to \begin{bmatrix}a_n\\a_n\end{bmatrix} B_r{}^n, B_r{}^n \to \binom{b_1}{\lambda} C_r{}^1, C_r{}^1 \to \binom{b_2}{\lambda} C_r{}^2, \cdots, C_r{}^{k-1} \to \binom{b_k}{\lambda} C_r{}^k, C_r{}^k \to D_r{}^1 \begin{bmatrix}d_m\\d_m\end{bmatrix}, \cdots,$

$D_r{}^{m-1} \to D_r{}^m \begin{bmatrix}d_1\\d_1\end{bmatrix}, D_r{}^m \to A_r{}^1 \binom{c_l}{\lambda}, \cdots, A_r{}^{l-1} \to A \binom{c_1}{\lambda}$

where $A_r{}^i, B_r{}^j, C_r{}^p$ and $D_r{}^q$ are new nonterminals that are used in this production with $1 \le i \le l-1, 1 \le j \le k, 1 \le p \le k$ and $1 \le q \le m$.

Next, we define a static WK linear grammar $G' = (N', T, \rho, S, P')$ where N' contains the nonterminals of N and all new nonterminals introduced above and P' contains the productions constituted above. Hence, every production r in P can be replaced with the corresponding sequence $\{r\}$ of productions in P' and vice versa. Therefore, $L(G) = L(G')$. ■

To further investigate on the computational properties between the static WK regular and linear grammars, the next lemma shows that there exist a static WK linear language which cannot be generated by static WK regular grammar. We consider a language $L(G) = \{a^n b^m c^m d^n \mid n, m \ge 1\}$ where the idea of 1-normal form is used to simplify the length of the rules in the grammars as shown in Lemma 3.

Lemma 3. $L(G) = \{a^n b^m c^m d^n \mid n, m \ge 1\} \in$ **SLIN**−**SREG**.

Proof. Let $G = (\{S, A, B, C, D, E\}, \{a, b, c, d\}, \{(a, a), (b, b), (c, c), (d, d)\}, S, P)$ be a static WK linear grammar, where P consists of the following rules:

$S \to \begin{bmatrix}a\\a\end{bmatrix} A \begin{bmatrix}d\\d\end{bmatrix}$, $A \to \binom{a}{\lambda} A \binom{d}{\lambda} \mid \binom{a}{\lambda} B \binom{d}{\lambda}$, $B \to \binom{\lambda}{a} B \binom{\lambda}{d} \mid \binom{\lambda}{a} C \binom{\lambda}{d}$, $C \to \binom{b}{\lambda} C \binom{c}{\lambda} \mid \binom{b}{\lambda} D \binom{c}{\lambda}$, $D \to \binom{\lambda}{b} D \binom{\lambda}{c} \mid \binom{\lambda}{b} E \binom{\lambda}{c}$, $E \to \binom{\lambda}{\lambda}$.

From this, we obtain the derivation:

$S \Rightarrow \begin{bmatrix}a\\a\end{bmatrix} A \begin{bmatrix}d\\d\end{bmatrix} \Rightarrow^* \begin{bmatrix}a\\a\end{bmatrix} \binom{a^n}{\lambda} B \binom{d^n}{\lambda} \begin{bmatrix}d\\d\end{bmatrix} \Rightarrow^* \begin{bmatrix}a^{n+1}\\a^{n+1}\end{bmatrix} C \begin{bmatrix}d^{n+1}\\d^{n+1}\end{bmatrix} \Rightarrow^* \begin{bmatrix}a^{n+1}\\a^{n+1}\end{bmatrix} \binom{b^m}{\lambda} C \binom{c^m}{\lambda} \begin{bmatrix}d^{n+1}\\d^{n+1}\end{bmatrix} \Rightarrow^* \begin{bmatrix}a^{n+1}b^m\\a^{n+1}b^m\end{bmatrix} E \begin{bmatrix}c^m d^{n+1}\\c^m d^{n+1}\end{bmatrix} \Rightarrow \begin{bmatrix}a^{n+1}b^m c^m d^{n+1}\\a^{n+1}b^m c^m d^{n+1}\end{bmatrix}$.

Therefore, $L(G) = \{a^n b^m c^m d^n \mid n, m \ge 1\}$.

Next, we need to show that $L(G) = \{a^n b^m c^m d^n \mid n, m \ge 1\} \notin$ **SREG**. By contradiction, suppose that $L(G)$ can be generated by a static WK regular grammar $G' = (N', \{a, b, c, d\}, \rho, S, P')$. Without loss of generality, assume that G' is in 1-normal form. Let $w = a^j b^k c^k d^j$ be a string in $L(G)$. Then, the grammar G' can generate the complete sequences $\begin{bmatrix}a^j b^k c^k d^j\\a^j b^k c^k d^j\end{bmatrix}$ as follows:

Case 1: In the derivation of the string, the first b can occur in the upper (or lower) strand if a^j has already been generated in the upper (or lower) strand. Then, some of the possible derivations are:

$$S \Rightarrow^* \begin{bmatrix}a^i\\a^i\end{bmatrix} \binom{a^p}{\lambda} \binom{b}{\lambda}; \quad \text{or} \tag{5}$$

$$S \Rightarrow^* \begin{bmatrix}a^j\\a^j\end{bmatrix} \binom{b}{\lambda} \quad \text{or} \quad S \Rightarrow^* \begin{bmatrix}a^j b\\a^j b\end{bmatrix}, \tag{6}$$

where $i + p = j$. For derivation (5), we can continue the derivation by using the lower strand of a and upper strand of b to generate the upper strand of b^k:

$$S \Rightarrow^* \begin{bmatrix}a^i\\a^i\end{bmatrix} \binom{a^p}{\lambda} \binom{b}{\lambda} \Rightarrow^* \begin{bmatrix}a^j\\a^j\end{bmatrix} \binom{b^k}{\lambda}. \tag{7}$$

Following that, derivation (7) is continued by generating the first c in the upper strand and use the lower strand of b to control their number of occurrences:

$$S \Rightarrow^* \begin{bmatrix}a^j\\a^j\end{bmatrix} \binom{b^k}{\lambda} \Rightarrow^* \begin{bmatrix}a^j b^k\\a^j b^k\end{bmatrix} \binom{c^k}{\lambda}. \tag{8}$$

Next, derivation (8) is continued by generating the first d in the upper strand and use the strand of c to control their number of occurrences:

$$S \Rightarrow^* \begin{bmatrix}a^j b^k\\a^j b^k\end{bmatrix} \binom{c^k}{\lambda} \Rightarrow^* \begin{bmatrix}a^j b^k c^k\\a^j b^k c^k\end{bmatrix} \binom{d^l}{\lambda}. \tag{9}$$

The derivation can be completed by generating the lower strand of d. Thus, $\begin{bmatrix} a^j b^k c^k d^j \\ a^j b^k c^k d^j \end{bmatrix}$ where $l \neq j$. For derivation (6), we can continue the derivation by using the same idea as in (5) and get the possible derivation as in (7) until (9).

Case 2: The number of a in the lower strand is controlled by the upper strand of c. In this case, the number of c's cannot be related to the number of b such that $S \Rightarrow^* \begin{bmatrix} a^j \\ a^j \end{bmatrix} \begin{pmatrix} b^k c^j \\ \lambda \end{pmatrix}$. Thus, we can conclude that $L(G) \notin$ **SREG** since the number of b, c and d are difficult to control at the same time using **SREG** rules.

IV. CONCLUSION

In this paper, the 1-normal form for static WK regular and linear grammars are defined. We show that for each grammar, there exists the equivalent grammars by using the concept of 1-normal form. In addition, the implementation of 1-normal form has been shown in Lemma 3 to investigate the computational properties between the static WK regular and linear grammars. It has been found that there exist a static WK linear language which cannot be generated by static WK regular grammar. There are some interesting topic that can be explored in the future research such as to study the computational properties for the static WK grammars, define static WK context-free grammar and introduce the normal forms for static WK context-free grammar.

V. 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 Education (MOE) and Research Management Centre (RMC), UTM for the financial funding through Fundamental Research Grant Scheme (FRGS) Vote No. 5F022.

VI. REFERENCES

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

Freund, R., Pãun, G., Rozenberg, G. & Salomaa, A. (1997). Watson-Crick finite automata. Proceeding 3rd DIMACS Workshop on DNA based Computers, Philadelphia, 297-328.

Hg, W. D. & Wong, C. K. B. (2007). Self-Recognition of DNA from life processes to DNA computation. Biophysical reviews and letters, 2(02), 123-137.

Ito, M., Kobayashi, Y. & Shoji, K. (2010). Automata, Formal languages and algebraic systems: Proceedings of AFLAS 2008, Kyoto, Japan, World Scientific.

Kari, L., Pãun, G., Rozenberg, G., Salomaa, A. & Yu, S. (1998). DNA computing, sticker systems, and universality. Acta Informatica, 35(5), 401-420.

Levelt, W. J. (1974). Formal grammars in linguistics and psycholinguistics. Vol. III, Psycholinguistic Applications. Mouton.

Linz, P. (2006). An introduction to formal languages and automata, Jones and Bartlett Publishers, USA.

Pãun, G. & Rozenberg, G. (1998). Sticker systems. Theoretical Computer Science, 204, 183-203.

Pãun, G., Rozenberg, G. & Salomaa, A. (1998). DNA computing: new computing paradigms. Springer-Verlag & Business Media.

Rahman, A. A., Fong, W. H., Sarmin, N. H., Turaev, S. & Zulkufli, N. L. M. (2018a). Static Watson-Crick regular grammar, Malaysian Journal of Applied and Fundamental Sciences, 14, 457-462.

Rahman, A. A., Fong, W. H., Sarmin, N. H., Turaev, S. & Zulkufli, N. L. M. (2018b). Static Watson-Crick linear grammars and its computational power, International Graduate Conference on Engineering, Science and Humanities (IGCESH) (accepted).

Subramanian, K. G., Hemalatha, S. & Venkat, I. (2012). On Watson-Crick automata. In Proceedings of the Second International Conference on Computational Science, Engineering and Information Technology, 151-156.

Zulkufli, N. L. M., Turaev, S., Mohd Tamrin, M. & Messikh, A. (2016). Generative power and closure properties of Watson-Crick grammars. Applied Computational Intelligence and Soft Computing, 1-12