International Journal of Engineering, Mathematical and Physical Sciences
ISSN: 2517-9934
Vol:4, No:4, 2010

Some Relationships between Classes of Reverse Watson-Crick Finite Automata

Kazuki Murakami, Takashige Nakamura, Noriko Sakamoto, and Kunio Aizawa

***Abstract*—**A Watson-Crick automaton is recently introduced as a computational model of DNA computing framework. It works on tapes consisting of double stranded sequences of symbols. Symbols placed on the corresponding cells of the double-stranded sequences are related by a complimentary relation. In this paper, we investigate a variation of Watson-Crick automata in which both heads read the tape in reverse directions. They are called *reverse Watson-Crick finite automata (RWKFA).* We show that all of following four classes, i.e., *simple, 1-limited, all-final, all-final and simple*, are equal to non-restricted version of RWKFA.

***Keywords*—**automaton, DNA computing, formal languages, Watson-Crick automaton

I. Introduction

DNA computing provides new paradigms of computation [1]. The *Watson-Crick automata*, introduced in [2], provide computational models of the DNA computing framework. They use a double-stranded tape called Watson-Crick tape, whose strands are separately scanned by read-only heads. The symbols placed on the corresponding cells of the double-stranded sequences are related by a complimentary relation similar to a DNA molecule. The relationships between classes of the automata are investigated in [2 - 5].

The two strands of a DNA molecule have opposite $5' \rightarrow 3'$ orientation. This suggests considering a variant of Watson-Crick finite automata that read the two strands of its input tape in opposite direction. Such automata are called *reverse Watson-Crick automata* and introduced in [2]. Some variations of the reverse Watson-Crick automata which have sensing power that tells the upper and the lower heads are within a fixed distance (or meet at the same position) are discussed in [6, 7].

In this paper, we investigate the relationship between classes of the reverse Watson-Crick automata. We show that $RWK = SRWK = 1RWK = FRWK = FSRWK$ as in the case of the Watson-Crick automata.

The rest of the paper is organized as follows. In Section II, basic definitions of the Watson-Crick finite automata are reviewed. Definitions of the reverse version are given in Section III. Some relationships between classes of reverse version are investigated in Section IV. Finally a brief conclusion is given in Section V.

Manuscript received February 20, 2010.

K. Murakami is with Department of Mathematics and Computer Science, Interdisciplinary Graduate School of Science and Engineering, Shimane University, Matsue, Shimane 690-8504 Japan.

T. Nakamura is with Department of Mathematics and Computer Science, Interdisciplinary Faculty of Science and Engineering, Shimane University, Matsue, Shimane 690-8504 Japan.

N. Sakamoto is with Department of Mathematics and Computer Science, Interdisciplinary Graduate School of Science and Engineering, Shimane University, Matsue, Shimane 690-8504 Japan.

K. Aizawa is with Department of Mathematics and Computer Science, Interdisciplinary Faculty of Science and Engineering, Shimane University, Matsue, Shimane 690-8504 Japan (phone 0852 32 6475; e-mail: aizawa@cis.shimane-u.ac.jp).

II. Watson-Crick Finite Automaton

A. Complementarity Relation

DNA strands consist of polymer chains which are composed of nucleotides. There are four types of nucleotides: A (Adenine), G (Guanine), C (Cytosine), and T (Thymine). The double helix of DNA arises by the bonding of two separate strands. Bonding occurs by the pairwise attraction of types: A always bonds with T, and G with C. This phenomenon is known as Watson-Crick complementarity.

Consider an alphabet V and a symmetric relation $\rho \subseteq V \times V$ over V. We consider the complementarity relation as a generalization of Watson-Crick complementarity.

B. Watson-Crick Domain

We associate with V the monoid $V^* \times V^*$, of pair of strings. In accordance with the way of representing DNA molecules, where one considers the two strands placed one over the other, we write the elements $(x_1, x_2) \in V^* \times V^*$ in the form $\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$. The concatenation of two pairs $\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ and $\begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$ is $\begin{pmatrix} x_1 y_1 \\ x_2 y_2 \end{pmatrix}$. We also write $\begin{pmatrix} V^* \\ V^* \end{pmatrix}$ instead of $V^* \times V^*$.

We denote $\begin{bmatrix} V \\ V \end{bmatrix}_\rho = \{ \begin{bmatrix} a \\ b \end{bmatrix} \mid a, b \in V, (a, b) \in \rho \}$ and $WK_\rho(V) = \begin{bmatrix} V \\ V \end{bmatrix}_\rho^*$. The set $WK_\rho(V)$ is called the Watson-Crick domain associated to the alphabet V and the complementarity relation ρ.

For $w_1 = a_1 a_2 \L\ a_n$, $w_2 = b_1 b_2 \L\ b_n$, the elements $\begin{bmatrix} a_1 \\ b_1 \end{bmatrix}\begin{bmatrix} a_2 \\ b_2 \end{bmatrix}\L\begin{bmatrix} a_n \\ b_n \end{bmatrix} \in WK_\rho(V)$ are written in the form $\begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$. We call the elements *double stranded sequences*. The string w_1 is called the *upper strand* and w_2 is called the *lower strand*.

C. Definition of Watson-Crick finite Automaton

A Watson-Crick automaton is a construct

$$M = (V, \rho, Q, s_0, F, \delta)$$

where V is an alphabet, $\rho \subseteq V \times V$ is a complementarity relation, Q is a set of states, $s_0 \in Q$ is the initial state, $F \subseteq Q$ is the set of final states, and $\delta : Q \times \begin{pmatrix} V^* \\ V^* \end{pmatrix} \to P\ (Q)$ is a transition function. The interpretation of $s' \in \delta(s, \begin{pmatrix} x_1 \\ x_2 \end{pmatrix})$ is that the automaton M, in the state s, passes over x_1 in the upper strand and x_2 in the lower strand of a double stranded sequence, and enters the state s'.

A transition in a Watson-Crick finite automaton can be defined as follows: For $\begin{pmatrix} u_1 \\ u_2 \end{pmatrix}, \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \in \begin{pmatrix} V^* \\ V^* \end{pmatrix}$ such that $\begin{bmatrix} u_1 x_1 v_1 \\ u_2 x_2 v_2 \end{bmatrix} \in WK_\rho(V)$ and $s, s' \in Q$, we write

$$\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} s \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \Rightarrow \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} s' \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$$

iff $s' \in \delta(s, \begin{pmatrix} x_1 \\ x_2 \end{pmatrix})$.

We denote by $\Rightarrow^*$ the reflexive and transitive closure of the relation $\Rightarrow$.

The language $L(M)$ of M is defined as follows:

$$L(M) = \{w_1 \mid s_0 \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \Rightarrow^* \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} s_f,$$

$$\text{for some } s_f \in F \text{ and } \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \in WK_\rho(V)\}.$$

There are some restricted variations of the Watson-Crick finite automata.

- N: *stateless*, i.e., with only one state: if $Q = F = \{q_0\}$;
- F: *all-final*, i.e., with only final states: if $Q = F$;
- S: *simple*, if $\delta : (Q \times \left(\begin{pmatrix} V^* \\ \{\lambda\} \end{pmatrix} \cup \begin{pmatrix} \{\lambda\} \\ V^* \end{pmatrix} \right) \to P\ (Q)$;
- 1: *1-limited*, if $\delta : (Q \times \left(\begin{pmatrix} V \\ \{\lambda\} \end{pmatrix} \cup \begin{pmatrix} \{\lambda\} \\ V \end{pmatrix} \right) \to P\ (Q)$.

Further variations, such as NS, FS, N1 and F1 Watson-Crick automata can be defined in a straightforward way by using multiple constrains. We denote by *WK, NWK, FWK, SWK, 1WK, NSWK, N1WK, FSWK, F1WK*, the families of languages of Watson-Crick automata which are arbitrary, stateless, all-final, simple, 1-limited, stateless and simple, stateless and 1-limited, all-final and simple, and all-final and 1-limited, respectively.

III. Reverse Watson-Crick Automaton

The two strands of a DNA molecule have opposite 5' → 3' orientation. This suggests considering a variant of Watson-Crick finite automata that read the two strands of its input tape in opposite direction. Such automata are called *reverse Watson-Crick finite automata*. Fig. 1 illustrates the initial configuration of such an automaton.

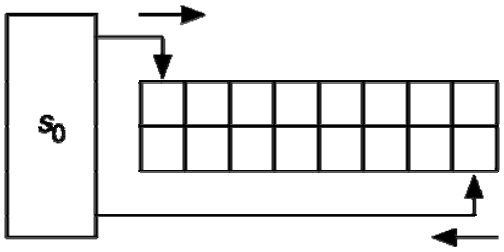

Fig. 1. A reverse Watson-Crick finite automaton

Formally, a reverse Watson-Crick finite automaton is a construct

$$M = (V, \rho, Q, s_0, F, \delta)$$

with the components defined exactly as for Watson-Crick finite automata, but the relation $\Rightarrow$ defined as follows:

For $\begin{pmatrix} u_1 \\ u_2 \end{pmatrix}, \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \in \begin{pmatrix} V^* \\ V^* \end{pmatrix}$ such that $\begin{bmatrix} u_1 x_1 v_1 \\ u_2 x_2 v_2 \end{bmatrix} \in WK_\rho(V)$ and $s, s' \in Q$, we write

$$\begin{pmatrix} u_1 \\ u_2 x_2 \end{pmatrix} s \begin{pmatrix} x_1 v_1 \\ v_2 \end{pmatrix} \Rightarrow \begin{pmatrix} u_1 x_1 \\ u_2 \end{pmatrix} s' \begin{pmatrix} v_1 \\ x_2 v_2 \end{pmatrix}$$

iff $s' \in \delta(s, \begin{pmatrix} x_1 \\ x_2 \end{pmatrix})$.

The language $L(M)$ of M is defined as follows:

$$L(M) = \{w_1 \mid \begin{pmatrix} \lambda \\ w_2 \end{pmatrix} s_0 \begin{pmatrix} w_1 \\ \lambda \end{pmatrix} \Rightarrow^* \begin{pmatrix} w_1 \\ \lambda \end{pmatrix} s_f \begin{pmatrix} \lambda \\ w_2 \end{pmatrix},$$

$$\text{for some } s_f \in F \text{ and } \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \in WK_\rho(V)\}.$$

All restriction of the Watson-Crick finite automata can be defined for reverse Watson-Crick automata. We denote by *RWK, NRWK, FRWK, SRWK, 1RWK, NSRWK, N1RWK, FSRWK, F1RWK*, the families of languages of reverse Watson-Crick automata which are arbitrary, stateless, all-final, simple, 1-limited, stateless and simple, stateless and 1-limited, all-final and simple, and all-final and 1-limited, respectively.

IV. Relations Between Classes of Reverse Watson-Crick Automata

Directly from the definitions we obtain:

- $XRWK \subseteq RWK$, for $X \in \{N, F, S, 1, NS, N1, FS, F1\}$,
- $NRWK \subseteq FRWK$, $NSRWK \subseteq FSRWK$, and $N1RWK \subseteq F1RWK$,
- $XSRWK \subseteq SRWK$, $X1RWK \subseteq 1RWK$, $1RWK \subseteq SRWK$, and $X1RWK \subseteq XSRWK \subseteq XRWK$ for $X \in \{N, F\}$.

The following relationships are proved in [1]:

- $NSRWK \subseteq REG$,
- $NRWK - CF \neq \emptyset$ and $F1RWK - CF \neq \emptyset$,
- $NSRWK \subset FSRWK$ and $NSRWK \subset NRWK$.

These relations are summarized in Fig. 2.

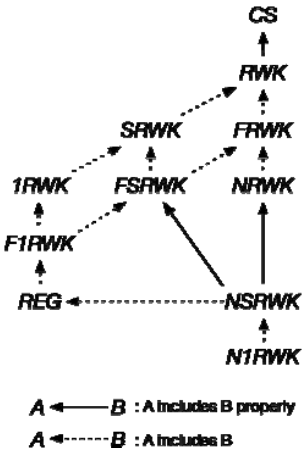

Fig. 2. Known relations on classes of reverse Watson-Crick finite automaton

Now, we show some relationships between families of RWK languages.

Theorem 1: $1RWK = SRWK = RWK$.

Proof. For any unrestricted reverse Watson-Crick automaton $M = (V, \rho, Q, s_0, F, \delta)$, there exists an 1-limited reverse Watson-Crick automaton $M' = (V, \rho, Q', s_0, F', \delta')$ such that $L(M) = L(M')$.

For each transition rule

$$t : s' \in \delta(s, \begin{pmatrix} a_1 a_2 \dots a_n \\ b_1 b_2 \dots b_m \end{pmatrix}),\ n, m \geq 0,\ n + m \geq 2,$$

we introduce the following rules of δ' :

$$s_{t,1} \in \delta(s, \begin{pmatrix} a_1 \\ \lambda \end{pmatrix}),$$

$$s_{t,i+1} \in \delta(s_{t,i}, \begin{pmatrix} a_{i+1} \\ \lambda \end{pmatrix}),\ 1 \leq i \leq n-1,$$

$$s'_{t,1} \in \delta(s_{t,n}, \begin{pmatrix} \lambda \\ b_m \end{pmatrix}),$$

$$s'_{t,i+1} \in \delta(s'_{t,i}, \begin{pmatrix} \lambda \\ b_{m-i} \end{pmatrix}),\ 1 \leq i \leq m-2,$$

$$s' \in \delta(s'_{t,m-1}, \begin{pmatrix} \lambda \\ b_1 \end{pmatrix}),$$

where states $s_{t,i}$, $s'_{t,i}$ are newly introduced in *Q'*.

From this construction, it is easy to see that $RWK \subseteq 1RWK$. So we conclude that $1RWK = SRWK = RWK$. □

Theorem 2: $FSRWK = RWK$.

Proof. For any unrestricted reverse Watson-Crick automaton $M = (V, \rho, Q, s_0, F, \delta)$, there exists an all-final and simple reverse Watson-Crick automaton $M' = (V, \rho, Q', i, Q', \delta')$ such that $L(M) = L(M')$.

To handle "all-final" situation, the upper head of *M'* always rest on odd positions while the lower head always rest on even position.

Let *m* be larger than the maximum length of strands which appear in δ . So for each $\delta(s, \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}) \neq \emptyset$, $|x_1| < m$ and $|x_2| < m$ hold. Then we define the set of states *Q'* as follows: $Q' = \{(s, u, v) \mid s \in Q, u, v \in V^*, |u|, |v| \leq m\} \cup \{i, f\}$, where $i, f \notin Q$. The transition function δ' is constructed as follows:

- For any $a \in V$, we introduce a transition rule $\delta'(i, \begin{pmatrix} a \\ \lambda \end{pmatrix}) = \{(s_0, a, \lambda)\}$. At this point, the upper head moves one step and the state of *M'* memorizes the letter *a* scanned by the head.
- For any $(s, ux, v) \in Q'$, $|x| = 2$, we introduce a transition $(s, ux, v) \in \delta'((s, u, v), \begin{pmatrix} x \\ \lambda \end{pmatrix})$. Symmetrically, for any $(s, u, yv) \in Q'$, $|y| = 2$, we introduce a transition $(s, u, yv) \in \delta'((s, u, v), \begin{pmatrix} \lambda \\ y \end{pmatrix})$. These transitions keep track

the scanned symbols which lengths are at most m. Since $|x|, |y| = 2$, the upper head always stays on odd positions and the lower head always on even positions.

- For any $(s, xu, vy) \in Q'$ and $s' \in \delta(s, \begin{pmatrix} x \\ y \end{pmatrix})$, we introduce a transition $(s', u, v) \in \delta'(s, \begin{pmatrix} xu \\ vy \end{pmatrix})$. These transitions simulate the corresponding transitions in M without any moves of heads of M'.
- For any $(s, u, v) \in Q'$ and $s' \in \delta(s, \begin{pmatrix} ua \\ v \end{pmatrix})$, $a \in V$, $s' \in F$, then we introduce a transition $f \in \delta'((s, u, v), \begin{pmatrix} a \\ \lambda \end{pmatrix})$. Symmetrically, For any $(s, u, v) \in Q'$ and $s' \in \delta(s, \begin{pmatrix} u \\ bv \end{pmatrix})$, $b \in V$, $s' \in F$, then we introduce a transition $f \in \delta'((s, u, v), \begin{pmatrix} \lambda \\ b \end{pmatrix})$. These transitions simulate the acceptance process of M. At this stage, the upper (lower) head moves one step if the length of input tape is even (odd), respectively.

At the final stage, M' read its input tape completely, if and only if M accepts the input tape. Since M' is all-final, it accepts the tape, if and only if M accepts the input tape.

From this construction, it is easy to see that $RWK \subseteq FSRWK$. So we conclude that $FSRWK = RWK$. □

Corollary 1: $RWK = SRWK = 1RWK = FRWK = FSRWK$.

Our results together with the results in Fig. 2 are summarized in Fig. 3.

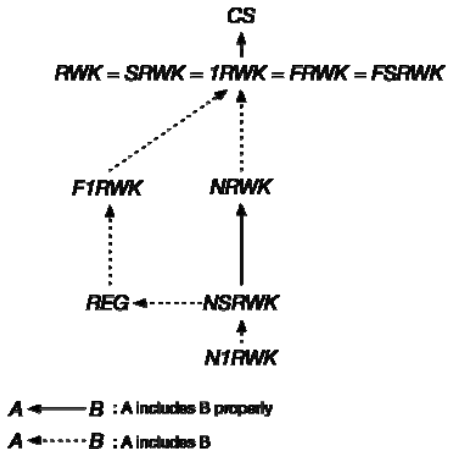

Fig. 3. Refined relations on classes of reverse Watson-Crick finite automaton

V. Conclusion

We have investigated some relationships between classes of reverse Watson-Crick automata. The relationships between $F1$ and FS in both normal and reverse Watson-Crick automata will be very interesting at next phase.

Acknowledgment

Kunio Aizawa thanks Prof. Akira Nakamura for his valuable suggestions.

References

[1] G. Păun, G. Rozenberg, and A. Salomaa, *DNA Computing, New Computing Paradigms*. Springer-Verlag, 1998, ch. 5.

[2] R. Freund, G. Păun, G. Rozenberg, and A. Salomaa, "Watson-Crick finite automata," in *Proc. of the Third Annual DIMACS Symp on DNA Based Computers*, Philadelphia, 1997, pp. 305-317.

[3] S. Hirose, K. Tsuda, Y. Ogoshi, and H. Kimura, "Some relations between Watson-Crick finite automata and Chomsky hierarchy," IEICE Trans. *on Information and Systems*, E89-D (10), 2006, pp. 2591-2599.

[4] S. Okawa and S. Hirose, "The relations among Watson-Crick automata and their relations with context-free languages," *IEICE Trans. on Information and Systems*, E87-D (5), 2004, pp. 1261-1264.

[5] D. Kuske and P. Weigel, "The role of the complementarity relations in Watson-Crick automata and sticker systems," in *Proc. of DLT 2004, Lecture Notes in Computer Science*, 3340, Springer-Verlag, 2004, pp. 272-283.

[6] B. Nagy, "On 5'→3' sensing Watson-Crick finite automata," in *Proc. of DNA 13, Lecture Notes in Computer Science*, 4848, Springer-Verlag, 2007, pp. 256-262.

[7] B. Nagy, "On a hierarchy of 5'→3' Sensing WK finite automata langugaes," in *Proc. of 5th Conference on Computability in Europe, CiE 2009*, Hidelberg, Germany, 2009, 266-275.