Title: State Complexity of Reversible Watson-Crick Automata

Authors: Kingshuk Chatterjee¹, Debayan Ganguly (corresponding author)², Kumar Sankar Ray³

Affiliations: ¹Government College of Engineering and Ceramic technology, Kolkata-700010
²Government College of Engineering and Leather technology, Kolkata-700106
³Electronics and Communication Sciences Unit, Indian Statistical Institute, Kolkata-700108.
State complexity of Reversible Watson-Crick automata

Kingshuk Chatterjee¹, Debayan Ganguly², Kumar Sankar Ray³

¹Government College of Engineering and Ceramic technology, Kolkata-700010
²Government College of Engineering and Leather technology, Kolkata-700106
³Electronics and Communication Sciences Unit, Indian Statistical Institute, Kolkata-700108.

¹ kingshukchatterjee@gmail.com, ² debayan3737@gmail.com, ³ ksray@isical.ac.in

Abstract: Reversible Watson-Crick automata introduced by Chatterjee et.al. is a reversible variant of an Watson-Crick automata. It has already been shown that the addition of DNA properties to reversible automata significantly increases the computational power of the model. In this paper, we analyze the state complexity of Reversible Watson-Crick automata with respect to non-deterministic finite automata. We show that Reversible Watson-Crick automata in spite of being reversible in nature enjoy state complexity advantage over non deterministic finite automata. The result is interesting because conversion from non deterministic to deterministic automata results in exponential blow up of the number of states and classically increase in number of heads of the automata cannot compensate for non-determinism in deterministic and reversible models.

Keywords: Reversible Watson-Crick automata, non-deterministic finite automata, deterministic finite automata, reversible automata, state complexity.

I. INTRODUCTION

Interest in Reversible automata started when Bennet[1] showed that Reversible Turing machine has the same computational power as a non-deterministic Turing machine. Researchers started exploring restricted versions of reversible automata Pin[2] showed that languages accepted by reversible automata is strictly a proper subset of regular languages. Morita[3] explored the two way variant of multi-head reversible automata and showed that it has the same computational power as two way multi-head deterministic finite automata. Kutrib et.al.[4] showed that one way multi-head reversible finite automata is strictly weaker than its deterministic variant. Watson-Crick automata[5] are finite automata having two independent heads working on double strands where the characters on the corresponding positions of the two strands are connected by a complementarity relation similar to the Watson-Crick complementarity relation. The movement of the heads although independent of each other is controlled by a single state. Chatterjee et.al.[6] introduced a reversible variant of Watson-Crick automata and showed that it can accept all regular languages. Moreover, reversible Watson-Crick automata were able to accept languages which are not accepted by multi-head deterministic finite automata. State and transition complexity of Watson-Crick automata are discussed in detail in[4]. In this paper, we compare the state complexity of Reversible Watson-Crick automata with respect to non-deterministic finite automata.

The main claims of this paper are as follows:

1. For every non deterministic finite automaton accepting a language L using n states there exist a reversible Watson-Crick automata which accepts the same language L using n+2 states.
2. Non-deterministic finite automaton requires at least $2^k+1$ states to accept the language $L_k=\{x^n\mid n\geq 1, x\in \{a,b\}^k\}$ whereas reversible Watson-Crick automata requires $2k+2$ states to accept $L_k$.

II. BASIC TERMINOLOGY

The symbol V denotes a finite alphabet. The set of all finite words over V is denoted by $V^*$, which includes the empty word $\lambda$. The set of all non-empty words over the alphabet V is denoted by $V^*=V^*-\{\lambda\}$. For $w\in V^*$, the length of w is denoted by $|w|$. Let $u\in V^*$ and $v\in V^*$ be two words and if there is some word $x\in V^*$, such that $v=ux$, then u is a prefix of v, denoted by $u\leq v$. Two words, u and v are prefix comparable denoted by $u\sim v$ if u is a prefix of v or vice versa.

Reversible Watson-Crick automata

A one-way reversible Watson-Crick automaton (Rev-WKA) is a system $M= (Q, V, \# , S, q_0, F, \rho, \delta)$ where

1) $Q$ is the finite set of states.
2) $V$ is the finite set of input symbols.
3) $\#\in V$ is the left end marker for both upper and lower strand.
4) $S\in V$ is the right end marker for both upper and lower strand.
5) $q_0\in Q$ is the initial state.
6) $F\subseteq Q$ is the set of accepting states.
Given a non-deterministic finite automaton $\mathcal{M} = (Q, \Sigma, \delta, q_0, F)$, we construct a reversible automaton $\mathcal{M}'$ from $\mathcal{M}$ in the following manner:

1. Convert the automaton $\mathcal{M}$ to its deterministic version $\mathcal{M}^d = (Q^d, \Sigma^d, \delta^d, q_0^d, F^d)$.
2. For each state $q \in Q$, construct a new state $q'$ such that $\delta^d(q, a, b) = (q', c_0, c_1)$.
3. Define the partial transition function $\delta'$ as follows:
   - For each transition $(q, a, b) \rightarrow (q', c_0, c_1)$ in $\delta^d$, add the reverse transition $(q', c_0, c_1) \rightarrow (q, a, b)$ to $\delta'$.
4. Define the final states as $F' = \{ q' \mid q \in F \}$.
5. Define the initial state as $q_0' = q_0$.

The constructed automaton $\mathcal{M}' = (Q', \Sigma', \delta', q_0', F')$ is reversible if and only if $\mathcal{M}$ is strongly reversible. The proof of the above Theorem is similar to the proof in [6] for deterministic finite automata and has been slightly modified to consider state complexity of the constructed reversible Watson-Crick automata and is in two parts. In the first part given a non-deterministic finite automaton $\mathcal{M}$ which accepts a language $L$, we give a construction to obtain a reversible Watson-Crick automaton $\mathcal{M}'$ from $\mathcal{M}$ and in the second part we show that $\mathcal{M}'$ accepts the same language as $\mathcal{M}$.

First Part: Given a non-deterministic finite automaton $\mathcal{M} = (Q, \Sigma, \delta, q_0, F)$. We construct a reversible Watson-Crick automaton $\mathcal{M}' = (Q', \Sigma', \delta', q_0', F')$ from $\mathcal{M}$ in the following manner:

We list all the transitions in $\mathcal{M}$ in a particular order. Each transition is assigned a symbol 't' where 't' $\in \Sigma$ and 't' is the
position of the transition in the list. If there are m transitions in M then $V = V \cup \{1 \leq i \leq m\}$ and complementarity relation $\rho = \{(x,t_3),(t_3,x),...,(x,t_4),(t_4,x),...,(x,t_0),(t_0,x) \} \forall x \in V$. For each transition $q' \in \delta(q,x)$ in $\delta$ having symbol 'i' assigned to it, we introduce the transition $\delta'(q,(x,t_0)) = (q',(1,1))$ in $\delta'$.

Moreover the following transitions are also added to $\delta'$.

1. $\delta'(q_0,(#, #)) = (q_0(1,1))$
2. $\delta'(q,S, S) = (q_0(0,0))$ for all $q \in F$

The set of states of $M'$ i.e. $Q' = Q \cup \{q_0, q_1\}$. If $M$ has n states, $M'$ has n+2 states.

The set of final states of $M'$ i.e. $F' = \{q_1\}$.

The start state of $M'$ is $q_0$.

**Second Part:** In this part we show that $M'$ constructed from $M$ accepts the same language $L$.

If $M$ accepts $w$ then there is a sequence of transitions of length $|w|$ that takes $M$ to a final state after consuming $w$. The complementarity relation $\rho$ of $M'$ relates each symbol to all the transitions in $\delta$, therefore the set of possible complementarity strings for $w$ comprises of all sequences of transitions of length $|w|$. As $M$ accepts $w$ among the set of possible complementarity strings for $w$, we will find a complementarity string $w'$ which resemble the sequence of transitions that takes $M$ to a final state after consuming $w$. Now $M'$ armed with this correct sequence of transitions can deterministically and reversibly (as each transition $q' \in \delta(q,x)$ of $M$ is assigned a unique number $\delta(q,(x,t_0)) = (q',(1,1))$ in $M'$) decide which transitions to take and following this sequence of transitions $M'$ reaches the final state of $M$, and both its heads are on $$. Then the transition $\delta(q,(S, S)) = (q_0(0,0))$ for all $q \in F$ takes $M'$ to its final state after completely consuming its input and thus $M'$ also accepts $w$.

If $M$ does not accept $w$ then there is no sequence of transitions that takes $M$ to a final state after consumption of $w$. Thus, no matter what the complementarity string $w'$ of $w$ is, $M'$ simulating $M$ based on $w'$ will never be in a position where it is in a final state of $M$ and the two heads of $M'$ are on $$. Thus the transitions of the form $\delta(q,(S, S)) = (q_0(0,0))$ for all $q \in F$ cannot be applied to $M'$, so $M'$ will never enter its final state $q_1$ as a result $M'$ also rejects $w$.

**Example 1:** Consider the language $L_1 = \{x^n \mid n \geq 1, x \in \{a,b\}^+\}$, reversible Watson-Crick automaton $M= (Q, V, \#, S, q_0, F, \delta, \rho)$ can accept the language $L_2$ using $2(2+1)=6$ states.

$Q = \{q_0, q_1, q_2, q_3, q_4\}$, $V = \{a, b, a_1, b_1\}$, $\rho = \{(a,a),(a,a_1),(a_1,a),(b,b_1),(b_1,b),(b,b_1)\}$, $F = \{q_1\}$. The transitions $\delta$ of $M$ are as follows:

$\delta(q_0(#,#)) = (q_0(1,1))$, $\delta(q_0(x,x)) = (q_1(0,1)) \ x \in \{a,b\}$, $\delta(q_1(x,x)) = (q_2(0,1)) \ x \in \{a,b\}$, $\delta(q_2(x,x)) = (q_3(1,1))$, $\delta(q_3(x,x)) = (q_4(1,1)) \ x \in \{a,b\}$, $\delta(q_4(x,x)) = (q_4(1,1)) \ x \in \{a,b\}$, $\delta(q_4(x,x)) = (q_4(1,1)) \ x \in \{a,b\}$

The above automaton works in the following manner:

After reading the beginning marker in both the tapes, the lower head of $M$, first shifts 2 places, to the right whereas the upper head remains in its position. Now for every position of the upper head we match with the lower head to see whether they are reading the same values or not, we repeat this process for 2 positions as length of $x$ is 2, if there is a mismatch we reject the input string as there is no transition defined when there is a mismatch and the automaton halts in a non-final state. If there is no mismatch then we repeat this procedure of checking lower head and upper head values repeatedly for 2 times by entering state $q_3$. If the lower head reaches $\$ then we accept the string by going to state $q_1$ and halting. To ensure reversibility of our model when transitioning from state $q_1$ to $q_3$ we use the transitions $\delta(q_3(x,a)) = (q_3(1,1))$, $\delta(q_3(x,b)) = (q_3(1,1))$ instead of $\delta(q_2(x,a)) = (q_3(1,1))$, $\delta(q_2(x,b)) = (q_3(1,1))$ so that when the loop again comes back to state $q_3$ using the transition $\delta(q_3(x,x)) = (q_4(1,1)) \ x \in \{a,b\}$ from $q_1$ there is no conflict in reversibility condition.

**Theorem 2:** Consider the language $L_3 = \{x^n \mid n \geq 1, x \in \{a,b\}^+\}$, reversible Watson-Crick automaton $M= (Q, V, \#, S, q_0, F, \delta, \rho)$ can accept the language $L_4$ using $2(2+1) = 2k+2$ states.

**Proof:** This proof is a generalisation of the construction used in Example 1, in Example 1, $k$ was equal to 2. The reversible Watson-Crick automaton $M= (Q, V, \#, S, q_0, F, \delta, \rho)$ is $Q = \{q_0, q_1, ..., q_{2(k+1)}\}$, $V = \{a, b, a_1, b_1\}$, $\rho = \{(a,a),(a,a_1),(a_1,a),(b,b_1),(b_1,b),(b,b_1)\}$, $F = \{q_{2k+2}\}$.

The transitions $\delta$ of $M$ are as follows:

$\delta(q_0(#,#)) = (q_0(1,1))$, $\delta(q_m(x,x)) = (q_{m+1}(0,1)) \ x \in \{a,b\}$ where $m = 0, 1, 2, ..., (k-1)$, $\delta(q_m(x,a)) = (q_{k+1}(1,1))$, $\delta(q_m(x,b)) = (q_{k+1}(1,1))$, $\delta(q_m(x,x)) = (q_{k+1}(1,1)) \ x \in \{a,b\}$ where $m = (k+1), (k+2), 2k$, $\delta(q_{2k+1}(x,x)) = (q_{k+1}(1,1)) \ x \in \{a,b\}$, $\delta(q_{2k+1}(x,x)) = (q_{k+1}(1,1)) \ x \in \{a,b\}$

The above automaton works in the following manner:
After reading the beginning marker in both the tapes using state \(q_0\) and transition \(\delta(q_0(#,#)) = (q_0(1,1))\), the lower head of \(M\), first shifts \(k\) places to the right whereas the upper head remains in its position using the states \(q_0, q_1, \ldots, q_k\) and the transitions \\
\(\delta(q_0(x,x)) = (q_{m+1}, (0,1)) \forall x \in \{a, b\}\) where \(m = 0,1,2,\ldots,(k-1)\). Now for every position of the upper head we match with the lower head to see whether they are reading the same values or not, we repeat this process for \(k\) positions as length of \(x\) is \(l\) using states \(q_{k+1}, q_{k+2}, \ldots, q_{2k+1}\). If there is a mismatch we reject the input string as there is no transition defined when there is a mismatch and the automaton halts in a non-final state. If there is no mismatch then we repeat this procedure of checking lower head and upper head values repeatedly for \(k\) times by entering state \(q_{2k+1}\) again and again and repeating the above procedure. If the lower head reaches \$ then we accept the string by going to state \(q_{2k+2}\) and halting. To ensure reversibility of our model when transitioning from state \(q_{k}\) to \(q_{k+1}\) we use the transitions \(\delta(q_k(a,a)) = (q_{k+1}, (1,1)), \delta(q_k(b,b)) = (q_{k+1}, (1,1))\) instead of \(\delta(q_k(a,a)) = (q_{k+1}, (1,1)), \delta(q_k(b,b)) = (q_{k+1}, (1,1))\) so that when the loop again comes back to state \(q_{k}\) using the transition \(\delta(q_{k+1}, (x,x)) = (q_{k+1}, (1,1)) \forall x \in \{a, b\}\) from \(q_{2k+1}\) there is no conflict in reversibility condition. Thus the above automaton accepts \(L_4\) using just \(2k+2\) states.

**Lemma 2:** Number states required for non-deterministic finite automata to accept \(L_4 = \{x^n \mid n \geq 1, x \in \{a, b\}^k\}\) is greater than \(2^k+1\).

**Proof:** The proof of this lemma is in [7].

**Theorem 3:** Reversible Watson-Crick automata has state complexity advantage over non-deterministic finite automata

**Proof:** From, Theorem 2 and Lemma 2 we see that to accept \(L_4 = \{x^n \mid n \geq 1, x \in \{a, b\}^k\}\) a non-deterministic finite automaton requires at least \(2^k+1\) states whereas Reversible Watson-Crick automaton accepts it using \(2k+2\) states. Moreover, from Theorem 1 we see no matter the language accepted by non-deterministic finite automaton, Reversible Watson-Crick automaton always accepts it using just two more states than the non-deterministic version, so there is just a difference of a constant number of states, whereas for some languages such as \(L_4\) the difference in number of states is exponential to linear. Thus, we can say that Reversible Watson-Crick automata have state complexity advantage over non-deterministic finite automata.

IV. CONCLUSIONS

In this paper we have compared the state complexity of Reversible Watson-Crick automata with non-deterministic finite automata. We have shown for every non-deterministic finite automaton accepting a language \(L\) using \(n\) states there exist a reversible Watson-Crick automaton which accepts the same language \(L\) using \(n+2\) states. Moreover we also showed Reversible Watson-Crick automaton requires \(2k+2\) states to accept \(L_4 = \{x^n \mid n \geq 1, x \in \{a, b\}^k\}\). We already know that a non-deterministic finite automaton requires at least \(2^k+1\) states to accept \(L_4\). Thus we, concluded that Reversible Watson-Crick automata has state complexity advantage over non-deterministic finite automata.

REFERENCES

[1] C. H. Bennett, “Logical reversibility of computation,” *IBM Journal of Research and Development*, vol. 17, pp. 525–532, nov 1973.

[2] J. E. Pin, “On the languages accepted by finite reversible automata,” in Automata, Languages and Programming, pp. 237–249, Springer Berlin Heidelberg, 1987.

[3] K. Morita, “Two-way reversible multi-head finite automata,” *Fundam. Inf.*, vol. 110, no. 1-4, pp. 241-254, 2011.

[4] M. Kutrib and A. Malcher, “One-way reversible multi-head finite automata,” *Theoretical Computer Science*, vol. 682, pp. 149–164, jun 2017.

[5] R. Freund, G. Păun, G. Rozenberg, and A. Salomaa, “Watson-crick finite automata,” in *DNA Based Computers III*, pp. 297–327, American Mathematical Society, apr 1999.

[6] K. Chatterjee and K. S. Ray, “Reversible watson-crick automata,” *Acta Inf.*, vol. 54, no. 5, pp. 487–499, 2017.

[7] A. Păun and M. Păun, “State and transition complexity of watson-crick finite automata,” in *Fundamentals of Computation Theory*, pp. 409–420, Springer Berlin Heidelberg, 1999.