Kingshuk Chatterjee; Kumar S. Ray

Watson-Crick pushdown automata

Kybernetika, Vol. 53 (2017), No. 5, 868–876

Persistent URL: http://dml.cz/dmlcz/147098

KYBERNETIKA — VOLUME *53* (2017), NUMBER 5, PAGES 868–876

WATSON–CRICK PUSHDOWN AUTOMATA

Kingshuk Chatterjee and Kumar S. Ray

A multi-head 1-way pushdown automaton with k heads is a pushdown automaton with k 1-way read heads on the input tape and a stack. It was previously shown that the deterministic variant of the model cannot accept all the context free languages. In this paper, we introduce a 2-tape, 2-head model namely Watson-Crick pushdown automata where the content of the second tape is determined using a complementarity relation, similar to Watson–Crick automata. We show computational powers of nondeterministic two-head pushdown automata and nondeterministic Watson–Crick pushdown automata are same. Moreover, deterministic Watson–Crick pushdown automata can accept all the context free languages.

Keywords: deterministic Watson–Crick automata, deterministic Watson–Crick pushdown automata, deterministic multi-head pushdown automata, context free languages

Classification: 68Q45, 68Q10

1. INTRODUCTION

The first significant study on multi-head pushdown automata was done by Harrison et al. [6]. Later Chrobak et al. [2] proved that there are languages accepted by $(k+1)$-head 1-way deterministic pushdown automata but not by k-head 1-way nondeterministic pushdown automata, for every k which was conjectured in [6]. They also gave some significant results regarding multi-head deterministic pushdown automata. One of the important results is that there exists a context free language not accepted by any multi-head deterministic pushdown automaton. A two head pushdown automaton model was also introduced by Samson [12] but in Samson's model the two heads move from the two opposite sides, Nagy did detailed analysis of Samson's model in [8].

A Watson–Crick automaton [5] is a finite automaton 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. Nondeterministic Watson–Crick automata and their properties were discussed in [10]. Deterministic Watson–Crick automata were introduced by Czeizler et al. [4]. Czeizler et al. [3] also carried out a detailed survey on Watson–Crick automata. The State complexity of Watson–Crick automata was reported in [9]

DOI: 10.14736/kyb-2017-5-0868

and [11]. Chatterjee et al. introduced reversible Watson–Crick automata and discussed their properties in [1].

In this paper, we introduce Watson–Crick pushdown automata by equipping Watson–Crick finite automata with stack (See Section 3). We show that the introduction of the DNA properties (two tapes where the content of the second tape is determined by a complementarity relation) in the nondeterministic model does not increase the computational power of the model with respect to traditional nondeterministic two-head pushdown automata, but, deterministic Watson–Crick pushdown automata exploit the complementarity relation property of Watson–Crick automata to accept all the context free languages (See Section 4). The results show that addition of DNA complementarity property to traditional multi-head nondeterministic pushdown automaton has no impact on its computational power but in case of the restricted models (such as deterministic model) the computational power increases significantly.

2. BASIC DEFINITIONS

The symbol V is a finite nonempty set of abstract symbols (alphabet). The set of all finite words over V is denoted by V^*, which includes the empty word λ. The symbol $V^+ = V^* \setminus \{\lambda\}$ denotes the set of all non-empty words over the alphabet V. 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 $u \sim_p v$ if u is a prefix of v or vice versa. Now, let $\rho \subseteq V \times V$ be a symmetric relation, called the Watson–Crick complementarity relation on V. The symbol $\begin{bmatrix} V \\ V \end{bmatrix}_\rho = \{ \begin{bmatrix} a \\ b \end{bmatrix} | a, b \in V, (a, b) \in \rho \}$ and $WK_\rho(V) = \begin{bmatrix} V \\ V \end{bmatrix}_\rho^*$ denotes the Watson–Crick domain associated with V and ρ. The symbol $\begin{pmatrix} w_1 \\ w_2 \end{pmatrix}$ is just a pair of strings written in that form instead of (w_1, w_2) and the symbol $\begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$ denotes that the two strands are of same length i. e. $|w_1| = |w_2|$ and the corresponding symbols in two strands are complementary in the sense given by the relation ρ.

2.1. Nondeterministic Watson–Crick automata

A Watson–Crick automaton is a 6-tuple of the form $M = (V, \rho, Q, q_0, F, \delta)$ where V is an input alphabet, set of states is denoted by Q, $\rho \subseteq V \times V$ is the Watson–Crick complementarity relation, q_0 is the initial state and $F \subseteq Q$ is the set of final states. δ contains a finite number of transition rules of the form $q \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \to q'$, which denotes that the machine in state q parses w_1 in upper strand and w_2 in lower strand and goes to state q' where $w_1, w_2 \in V^*$. A transition in a Watson–Crick finite automaton can be defined as follows: For $\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}, \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \in \begin{pmatrix} V^* \\ V^* \end{pmatrix}$ such that $\begin{bmatrix} x_1 u_1 w_1 \\ x_2 u_2 w_2 \end{bmatrix} \in WK_\rho(V)$ and $q, q' \in Q$, $\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} q \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \Rightarrow \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} q' \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}$ iff there is a transition rule

$q\begin{pmatrix}u_1\\u_2\end{pmatrix} \to q'$ in δ and $\overset{*}{\Rightarrow}$ denotes the transitive and reflexive closure of $\Rightarrow$. The language accepted by a Watson–Crick automaton M is L(M)=$\{w_1 \in V^* | q_0 \begin{bmatrix}w_1\\w_2\end{bmatrix} \overset{*}{\Rightarrow} \begin{bmatrix}\lambda\\\lambda\end{bmatrix} q$, with $q \in F, w_2 \in V^*, \begin{bmatrix}w_1\\w_2\end{bmatrix} \in WK_\rho(V)\}$. A configuration of a Watson–Crick automaton is a pair $(q, \begin{pmatrix}w_1\\w_2\end{pmatrix})$ where q is the current state of the automaton and $\begin{pmatrix}w_1\\w_2\end{pmatrix}$ is the part of the input word which has not been read yet.

2.2. Deterministic Watson–Crick automata

Czeizler et al. [4] introduced the notion of determinism in Watson–Crick automata. Different notions of determinism of Watson–Crick automaton are as follows:

- **weakly deterministic Watson–Crick automaton(WDWK)**: Watson–Crick automaton is weakly deterministic if in every configuration that can occur in some computation of the automaton, there is a unique possibility to continue the computation, i. e. at every step of the automaton there is at most one way to carry on the computation.
- **deterministic Watson–Crick automaton(DWK)**: deterministic Watson–Crick automaton is a Watson–Crick automaton for which if there are two transition rules of the form $q\begin{pmatrix}u\\v\end{pmatrix} \to q'$ and $q\begin{pmatrix}u'\\v'\end{pmatrix} \to q''$ then $u \not\sim_p u'$ or $v \not\sim_p v'$.
- **strongly deterministic Watson–Crick automaton(SDWK)**: strongly deterministic Watson–Crick automaton is a deterministic Watson–Crick automaton where the Watson–Crick complementarity relation is identity.

2.3. Multi-head pushdown automata

Informally, a multi-head 1-way pushdown automaton with k heads is a pushdown automaton with k 1-way read heads on the input tape and a pushdown store (stack for short). The multi-head pushdown automaton accepts by final state. For formal definition of multi-head pushdown automaton see [6].

3. WATSON–CRICK PUSHDOWN AUTOMATA (WKPDA)

In this section, we define the structure of Watson–Crick pushdown automata.

The Watson–Crick pushdown automaton is a 10-tuple system similar to nondeterministic pushdown automaton [7] defined in formal automata theory. A Watson–Crick pushdown automaton $P = (Q, \#, \$, V, \Gamma, \delta, q_0, Z_0, F, \rho)$ where Q is a finite set of states, V is an input alphabet, Γ is a finite stack alphabet that is the set of symbols we are allowed to push into the stack, δ is the set of transition rules which governs the behaviour of the automaton. The rules in δ are of the form $(q_i, \begin{pmatrix}w_1\\w_2\end{pmatrix}, X) \to (q_j, \gamma)$ where q_i, q_j are states in Q, the strings w_1, $w_2 \in V^* \cup V^*\$ \cup \#V^* \cup \#V^*\$$, X is a stack symbol that

is a member of Γ, γ is the string of stack symbols that replaces X at the top of the stack. For instance, if $\gamma = \lambda$ then the stack is popped, if $\gamma = X$, then the stack is unchanged and if $\gamma = YZ$ then X is replaced by Z and Y is pushed onto the stack. q_0 is the start state, the Watson–Crick pushdown automaton is in this state before making any transitions. Z_0 is the start symbol, initially, the Watson–Crick pushdown automaton's stack consists of one instance of this symbol and nothing else. The set of accepting states or final states are denoted by F. ρ is the Watson–Crick complementarity relation similar to Watson–Crick automaton. The symbol $\# \notin V$ is the left end marker in both the tapes and $\$ \notin V$ is the right end marker in both the tapes.

3.1. Instantaneous configuration of Watson–Crick pushdown automaton

Here we state the instantaneous configuration of Watson–Crick pushdown automaton that comprises of the stack contents γ, the remaining input to be read $\begin{pmatrix} x \\ y \end{pmatrix}$ where $x, y \in \#V^*\$ \cup V^*\$ \cup \lambda$ and the current state of the automaton is q. That is, the instantaneous configuration of the automaton is a triple $(q, \begin{pmatrix} x \\ y \end{pmatrix}, \gamma)$. Initially the computation starts with the input of the form $\begin{pmatrix} \#w_1\$ \\ \#w_2\$ \end{pmatrix}$ where $\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \in WK_\rho(V)$, the initial state is q_0 and the stack contains Z_0. The initial configuration of the automaton is the triple $(q_0, \begin{pmatrix} \#w_1\$ \\ \#w_2\$ \end{pmatrix}, Z_0)$. Conventionally the contents of the stack are represented as a string with the top of the stack on the left end and the bottom of the stack on the right. For a Watson–Crick pushdown automaton $P = (Q, \#, \$, V, \Gamma, \delta, q_0, Z_0, F, \rho)$. $\vdash$ is understood as follows: Suppose $(q, \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}, X) \rightarrow (p, \alpha)$, then $(q, \begin{pmatrix} a_1 w_1 \\ a_2 w_2 \end{pmatrix}, X\beta) \vdash (p, \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}, \alpha\beta)$. $\vdash^*$ is used to represent zero or more moves of the Watson–Crick pushdown automaton.

3.2. The acceptance condition of Watson–Crick pushdown automaton

The Watson–Crick pushdown automaton accepts by final state.

Let $P = (Q, \#, \$, V, \Gamma, \delta, q_0, Z_0, F, \rho)$ be a Watson–Crick pushdown automaton. Then the language accepted by P by final state is $L(P) = \{w_1 \in V^* | w_2 \in V^*$,where $\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \in WK_\rho(V) \,|(q_0, \begin{pmatrix} \#w_1\$ \\ \#w_2\$ \end{pmatrix}, Z_0) \vdash^* (q, \begin{pmatrix} \lambda \\ \lambda \end{pmatrix}, \alpha)$, for some state q in F and any stack string $\alpha\}$.

3.3. Deterministic Watson–Crick pushdown automata

We are using the deterministic notions of Watson–Crick automata to define the different notions of determinism in Watson–Crick pushdown automata.

- **deterministic Watson–Crick pushdown automaton(DWKPDA)**: deterministic Watson–Crick pushdown automaton is a Watson–Crick pushdown automaton for which if there are two transition rules of the form $\delta(q, \begin{pmatrix} u \\ v \end{pmatrix}, X) \rightarrow (q', \gamma)$ and $\delta(q, \begin{pmatrix} u' \\ v' \end{pmatrix}, X) \rightarrow (q'', \gamma')$ then $u \not\sim_p u'$ or $v \not\sim_p v'$.

- **strongly deterministic Watson–Crick pushdown automaton(SDWKPDA)**: strongly deterministic Watson–Crick pushdown automaton is a deterministic Watson–Crick pushdown automaton where the Watson–Crick complementarity relation is identity.

4. MAIN RESULT

In this section, we show that the computational power of nondeterministic two-head pushdown automata and nondeterministic Watson–Crick pushdown automata are same. We further show that deterministic Watson–Crick pushdown automata can accept all the context free languages.

Proposition 4.1. For every nondeterministic two-head pushdown automaton that accepts a language L there exists a Watson–Crick pushdown automaton with identity complementarity relation which accepts the same language L.

Proposition 4.1 follows from the fact that when the complementarity relation is identity then the content of the second tape of Watson–Crick pushdown automaton is same as the first head. As a result the head on the first tape of Watson–Crick pushdown automaton can simulate the movement of the first head of the two-head pushdown automaton and the head on the second tape of Watson–Crick pushdown automaton simulates the second head of the two-head pushdown automaton.

Proposition 4.2. For every nondeterministic Watson–Crick pushdown automaton M which accepts a language L, there exists a nondeterministic Watson–Crick pushdown automaton $M^{'}$ which accepts the same language L and all its transitions are of the form $\delta(q, \binom{u}{v}, X) \to q^{'}$ where $|u| \leq 1$ and $|v| \leq 1$.

This can be achieved by breaking the transition $\delta(q, \binom{x}{y}, X) \to q^{'}$ of M into several steps as follows. Let $x = x_1x_2 \dots x_m$ and $y = y_1y_2 \dots y_n$, let $\delta(q, \binom{x}{y}, X) \to q^{'}$ be the i^{th} transition . We introduce the transitions $\delta(q, \binom{x_1}{\lambda}, X) \to i^1$, $\delta(i^1, \binom{x_2}{\lambda}, \lambda) \to i^2$, $\dots$, $\delta(i^{m-1}, \binom{x_m}{\lambda}, \lambda) \to i^m$, $\delta(i^m, \binom{\lambda}{y_1}, \lambda) \to i^{m+1}$, $\delta(i^{m+1}, \binom{\lambda}{y_2}, \lambda) \to i^{m+2}$, $\dots$, $\delta(i^{m+n-1}, \binom{\lambda}{y_n}, \lambda) \to q$, in place of the transition $\delta(q, \binom{x}{y}, X) \to q^{'}$ in $M^{'}$.

Theorem 4.3. For every nondeterministic Watson–Crick pushdown automaton $M^{'}$ which accepts the language L and all its transitions are of the form $\delta^{'}(q, \binom{u}{v}, X) \to (q^{'}, \gamma)$ where $|u| \leq 1$ and $|v| \leq 1$ there exists a nondeterministic two-head pushdown automaton M which accepts the same language L.

P r o o f . Given a nondeterministic Watson–Crick pushdown automaton $M^{'} = (Q, \#, \$, V, \Gamma, \delta^{'}, q_0, Z_0, F, \rho)$, where ρ is the Watson–Crick complementarity relation. We can obtain a nondeterministic two-head pushdown automaton $M = (2, Q, \#, \$, V, \Gamma, \delta, q_0, Z_0, F)$ where δ is formed from $\delta^{'}$ in the following manner for transitions of the form $\delta^{'}(q, \begin{pmatrix} u \\ \lambda \end{pmatrix}, X) \rightarrow (q^{'}, \gamma)$ where $u \in V \cup \{\lambda, \#, \$\}$ we introduce $\delta(q, u, \lambda, X) = (q^{'}, \gamma)$ in $M^{'}$. For transitions of the form $\delta^{'}(q, \begin{pmatrix} u \\ v \end{pmatrix}, X) \rightarrow (q^{'}, \gamma)$ where $u \in V \cup \{\lambda, \#, \$\}$, $v \in V$, we introduce transitions of the form $\delta(q, u, a, X) = (q^{'}, \gamma)$ in δ where $\rho(a) = v$ and a must be present in the upper strand in the corresponding position as contents of the lower strand is obtained by applying the complementarity relation ρ to each position of the upper strand. The nondeterministic multi-head pushdown automaton nondeterministically guesses the complementarity element in the lower head for a and executes that particular transition. For transitions of the form $\delta^{'}(q, \begin{pmatrix} u \\ v \end{pmatrix}, X) \rightarrow (q^{'}, \gamma)$ where $u \in V \cup \{\lambda, \#, \$\}$, $v \in \{\#, \$\}$, we introduce transitions of the form $\delta(q, u, v, X) = (q^{'}, \gamma)$ in M. As the transitions in M mimic the transitions in $M^{'}$ so M behaves in the same manner as $M^{'}$. Moreover, both M and $M^{'}$ have the same set of final states therefore they both accept the same set of strings L. □

The following corollary follows from Proposition 4.1, Proposition 4.2 and Theorem 4.3.

Corollary 4.4. The computational power of nondeterministic two-head pushdown automata and nondeterministic Watson–Crick pushdown automata are same.

A **λ-free nondeterministic pushdown automaton** is a pushdown automaton which does not have any transition defined on λ.

Theorem 4.5. For every λ-free nondeterministic pushdown automaton NP that accepts a language $L \subseteq V^*$ where V is an alphabet there exists an alphabet $V_{NP} \supseteq V$ and a complementarity relation ρ_{NP} such that a deterministic Watson–Crick pushdown automaton accepts L as a language over V_{NP}(Both V_{NP} and ρ_{NP} depends on NP).

P r o o f . To prove the above theorem, we first give a construction to obtain a deterministic Watson–Crick pushdown automaton M from a λ-free nondeterministic pushdown automaton NP which accepts a language L by empty stack and then we show that the deterministic Watson–Crick pushdown automaton M obtained from the λ-free nondeterministic pushdown automaton NP accepts the same language L.

First part: Given a λ-free nondeterministic pushdown automaton $NP = (Q, V, \Gamma, \delta, q_0, Z_0)$ which accepts a language L by empty stack. We construct a Watson–Crick pushdown automaton $M = (Q^{'}, \#, \$, V_{NP}, \Gamma, \delta^{'}, q_0, Z_0, F, \rho_{NP})$ from NP in the following manner:

M has the same initial stack symbol and stack alphabet as NP. We list all the transitions in NP in a particular order. Each transition is assigned a symbol t_i where $t_i \notin V$ and i is the position of the transition in the list. If there are n transitions in NP then

$V_{NP} = V \cup \{t_i | 1 \leq i \leq n\}$ and complementarity relation $\rho_{NP} = \{(x, t_1), (t_1, x), \ldots, (x, t_i), (t_i, x), \ldots, (V\}$. For each transition $\delta(q, x, Y) = (q', Z)$ in δ having symbol t_i assigned to it, we introduce the transition $\delta'(q, \binom{x}{t_i}, Y) = (q', Z)$ in δ'.

The following transitions are also added to δ'.

- $\delta'(q_0', \binom{\#}{\#}, Z_0) = (q_0, Z_0)$ this transition ensures that M enters the start state of NP at the beginning of the input word w.
- $\delta'(q, \binom{\$}{\$}, \lambda) = (q_f, \lambda)$ for all $q \in Q$. These set of transitions ensure that if stack of NP is empty after completely consuming the input word w then M also enters its final state after completely consuming its input $\#w\$$.

The start state of M is q_0', the set of states is $Q' = Q \cup \{q_0', q_f\}$, and the set of final states is $F = \{q_f\}$.

Second Part: In the second part of the proof, we show that both M and NP accepts the same language L. If NP accepts w then there is a sequence of transitions of length $|w|$ that takes NP to an empty stack after consuming w. The complementarity relation ρ of M relates each symbol to all the transitions in δ, therefore the set of possible complementarity strings for w comprises of all sequences of transitions of length $|w|$. As NP 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 NP to empty stack. Now M armed with this correct sequence of transitions can deterministically decide which transitions to take and following this sequence of transitions reaches a position when its stack is empty and both its head are on \$. Then the transition $\delta'(q, \binom{\$}{\$}, \lambda) = (q_f, \lambda)$ for all $q \in Q$ takes M to its final state after completely consuming its input and thus M also accepts w. If NP does not accept w then there is no sequence of transitions that takes NP to an empty stack after consumption of w. Thus, no matter what the complementarity string w' of w is, M simulating NP based on w' will never be in a position where its stack is empty and the two heads are on \$. Thus the transitions of the form $\delta'(q, \binom{\$}{\$}, \lambda) = (q_f, \lambda)$ for all $q \in Q$ cannot be applied to M, so M will never enter its final state q_f as a result M also rejects w.

□

Lemma 4.6. For every context free language L there exists a grammar G in Greibach Normal Form such that $L(G) = L \setminus \{\lambda\}$.

Lemma 4.7. For every grammar G in Greibach Normal Form there exists a λ-free nondeterministic pushdown automaton which accepts the language $L(G)$ by empty stack.

Lemma 4.6 and 4.7 are stated in [7].

Corollary 4.8 follows from Lemma 4.6 and 4.7.

Corollary 4.8. For every context free language L, there exists a λ-free nondeterministic pushdown automaton which accepts the language $L \setminus \{\lambda\}$ by empty stack.

The following theorem follows from Theorem 4.5 and Corollary 4.8.

Theorem 4.9. For every context free language L, there exists a deterministic Watson–Crick pushdown automaton which accepts the language $L \setminus \{\lambda\}$.

Lemma 4.10. The context free language $L = \{x_1\$y_1 * x_2\$y_2 * \dots x_n\$y_n\# \ (x_1^{'}\$y_1^{'})^R * (x_2^{'}\$y_2^{'})^R * \dots (x_n^{'}\$y_n^{'})^R \mid$ there exist i and j such that $x_i = x_j^{'}$ and $y_i \neq y_j^{'}$, where w^R is the reverse of the string w$\}$ is not accepted by any multi-head deterministic pushdown automaton.

The proof of Lemma 4.10 is in [2].

The following corollary follows from Lemma 4.10 and Theorem 4.9.

Corollary 4.11. Deterministic Watson–Crick pushdown automaton accepts a language which is not accepted by any multi-head deterministic pushdown automaton.

5. CONCLUSION

In this paper, we add a stack to Watson–Crick automaton to obtain Watson–Crick pushdown automaton. We show that the nondeterministic variant of the model has the same computational power as 2-head nondeterministic pushdown automata. We also show the deterministic variant of the model accepts all the context free languages by exploiting the complementarity relation property of Watson–Crick automata. Such a deterministic pushdown model is of significant interest because it is the only multi-head deterministic pushdown automata that accept all the context free languages.

(Received December 30, 2016)

REFERENCES

[1] K. Chatterjee and K. S Ray: Reversible Watson–Crick automata. Acta Informatica *54* (2017), 5, 487–499. DOI:10.1007/s00236-016-0267-0

[2] M. Chrobak and M. Li: k+1 heads are better than k for PDA's. In: Proc. 27th Annual Symp. on Foundations of Computer Science 1986, pp. 361–367. DOI:10.1109/sfcs.1986.27

[3] E. Czeizler and E. Czeizler: A short survey on Watson–Crick automata. Bull. EATCS *88* (2006), 104–119.

[4] E. Czeizler, E. Czeizler, L. Kari, and K. Salomaa: On the descriptional complexity of Watson–Crick automata. Theoretical Computer Science *410* (2009), 3250–3260. DOI:10.1016/j.tcs.2009.05.001

[5] R. Freund, G.Paun, G.Rozenberg, and A.Salomaa: Watson–Crick finite automata. In: Proc. 3rd DIMACS Workshop on DNA Based Computers, Philadelphia 1997, pp. 297–328. DOI:10.1090/dimacs/048/22

[6] M. A. Harrison and O. H. Ibarra: Multi-head and multi-tape pushdown automata. Inform. Control *13* (1968), 433–470. DOI:10.1016/s0019-9958(68)90901-7

[7] J. E Hopcroft, R. Motwani, and J. D. Ullman: Introduction to Automata Theory, Languages and Computation. Third edition. Prentice Hall 2007.

[8] B. Nagy: A family of two-head pushdown automata. In: NCMA 2015: 7th Workshop on Non Classical Models of Automata and Applications, Porto 2015, pp. 177–191.

[9] A. Paun and M. Paun: State and transition complexity of Watson–Crick finite automata. Fundamentals of Computation Theory, Lecture Notes in Computer Science *1684* (1999), 409–420. DOI:10.1007/3-540-48321-7_34

[10] G. Paun, G. Rozenberg, and A. Salomaa: DNA Computing: New Computing Paradigms. Springer-Verlag, Berlin, 1998. DOI:10.1007/978-3-662-03563-4

[11] K. S. Ray, K. Chatterjee, and D. Ganguly: State complexity of deterministic Watson–Crick automata and time varying Watson–Crick automata. Nat. Comput. *14* (2015), 691–699. DOI:10.1007/s11047-015-9494-5

[12] A. A. Samson: 2-head pushdown automata. Procedia – Social and Behavioral Sciences *195* (2015), 2037–2046. DOI:10.1007/s11047-015-9494-5

Kingshuk Chatterjee, Electronics and Communication Sciences Unit, Indian Statistical Institute, Kolkata-700108. India.
e-mail: kingshukchaterjee@gmail.com

Kumar S. Ray, Electronics and Communication Sciences Unit, Indian Statistical Institute, Kolkata-700108. India.
e-mail: ksray@isical.ac.in