Theoretical Computer Science ••• (••••) •••–•••

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Note

Non-regular unary language and parallel communicating Watson–Crick automata systems

Kingshuk Chatterjee, Kumar Sankar Ray *

Electronics and Communication Sciences Unit, Indian Statistical Institute, Kolkata, India

A R T I C L E I N F O

Article history:
Received 4 December 2015
Received in revised form 10 March 2016
Accepted 14 September 2017
Available online xxxx
Communicated by N. Ollinger

Keywords:
Non-deterministic Watson–Crick automata
Parallel communicating Watson–Crick automata systems
Multi-head finite automata
Non-regular unary languages

A B S T R A C T

In 2006, Czeizler et al. introduced parallel communicating Watson–Crick automata system. They showed that parallel communicating Watson–Crick automata system can accept the non-regular unary language $L = \{a^{n^2}, \text{ where } n > 1\}$ using non-injective complementarity relation and three components. In this paper, we improve on Czeizler et al.'s work by showing that parallel communicating Watson–Crick automata system can accept the same language L using just two components.

1. Introduction

Martin-Vide et al. [1] introduced parallel communicating automata systems. The system consists of many finite automata communicating with states. They also established that the computational power of such systems is equivalent to non-deterministic finite automata with multiple heads.

Watson–Crick automata 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. Freund et al. [2] introduced Watson–Crick automata. Its deterministic variants were introduced by Czeizler et al. [3]. Work on state complexity of Watson–Crick automata is discussed in [4] and [5].

Czeizler et al. introduced parallel Communicating Watson–Crick automata systems (PCWKS) [6] and further showed that with non-injective complementarity relation parallel communicating Watson–Crick automata system can accept the non-regular unary language $L = \{a^{n^2}, \text{ where } n > 1\}$ with three components [7].

A parallel communicating Watson–Crick automata system [6] consists of several Watson–Crick automata each on its own input tape, and communicating by states. Every component of parallel communicating Watson–Crick automata system has its own double-stranded tape; the input is the same on all of them. An input is accepted by the system if all components are in final state and they completely parse the tape. Moreover, if one of the components stops before the others, the system halts and rejects the input.

* Corresponding author.
E-mail addresses: kingshukchaterjee@gmail.com (K. Chatterjee), ksray@isical.ac.in (K.S. Ray).

https://doi.org/10.1016/j.tcs.2017.09.008

In this paper we show that parallel communicating Watson–Crick automata systems can accept the language $L = \{a^{n^2},$ where $n > 1\}$ using just two components which is an improvement on the work by Czeizler et al. (see Section 4).

2. 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 λ. The symbol $V^+ = V^* - \{\lambda\}$ 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 $u \le v$. Two words, u and v are prefix comparable denoted by $u \sim_p v$ if u is a prefix of v or vice versa.

A Watson–Crick automaton is a 6-tuple of the form $M = (V, \rho, Q, q_0, F, \delta)$ where V is an alphabet set, set of states is denoted by Q, $\rho \subseteq V \times V$ is the symmetric complementarity relation similar to Watson–Crick complementarity relation, q_0 is the initial state and $F \subseteq Q$ is the set of final states. The function δ contains a finite number of transition rules of the form $q\binom{w_1}{w_2} \to q'$, 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 $w_1, w_2 \in V^*$. The symbol $\left[\begin{smallmatrix} w_1 \\ w_2 \end{smallmatrix}\right]$ is different from $\binom{w_1}{w_2}$. While $\binom{w_1}{w_2}$ is just a pair of strings written in that form instead of (w_1, w_2), the symbol $\left[\begin{smallmatrix} w_1 \\ w_2 \end{smallmatrix}\right]$ denotes that the two strands are of same length i.e. $|w_1| = |w_2|$ and the corresponding symbols in two strands are complementarity in the sense given by the relation ρ. The symbol $\left[\begin{smallmatrix} V \\ V \end{smallmatrix}\right]_\rho = \{\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right] \mid a, b \in V, (a,b) \in \rho\}$ and $WK_\rho(V) = \left[\begin{smallmatrix} V \\ V \end{smallmatrix}\right]_\rho^*$ denotes the Watson–Crick domain associated with V and ρ.

A transition in a Watson–Crick finite automaton can be defined as follows:

For $\binom{x_1}{x_2}, \binom{u_1}{u_2}, \binom{w_1}{w_2} \in \binom{V^*}{V^*}$ such that $\left[\begin{smallmatrix} x_1u_1w_1 \\ x_2u_2w_2 \end{smallmatrix}\right] \in WK_\rho(V)$ and $q, q' \in Q$, $\binom{x_1}{x_2}q\binom{u_1}{u_2}\binom{w_1}{w_2} \Rightarrow \binom{x_1}{x_2}\binom{u_1}{u_2}q'\binom{w_1}{w_2}$ iff there is transition rule $q\binom{u_1}{u_2} \to q'$ in δ. We denote by $\stackrel{*}{\Rightarrow}$ the reflexive and transitive closure of $\Rightarrow$. The language accepted by a Watson–Crick automaton M is $L(M) = \{w_1 \in V^* \mid q_0\left[\begin{smallmatrix} w_1 \\ w_2 \end{smallmatrix}\right] \stackrel{*}{\Rightarrow} q\left[\begin{smallmatrix} \lambda \\ \lambda \end{smallmatrix}\right]$, with $q \in F$, $w_2 \in V^*$, $\left[\begin{smallmatrix} w_1 \\ w_2 \end{smallmatrix}\right] \in WK_\rho(V)\}$.

3. Parallel communicating Watson–Crick automata system

A parallel communicating Watson–Crick automata system of degree n, denoted by PCWK(n), is a $(n+3)$-tuple

$$A = (V, \rho, A_1, A_2, \ldots, A_n, K),$$

where

- V is the input alphabet;
- ρ is the complementarity relation;
- $A_i = (V, \rho, Q_i, q_i, F_i, \delta_i)$, $1 \le i \le n$, are Watson–Crick finite automata, where the sets Q_i are not necessarily disjoint;
- $K = \{K_1, K_2, \ldots, K_n\} \subseteq \bigcup_{i=1}^n Q_i$ is the set of query states.

The automata $A_1, A_2, \ldots, A_n$ are called the *components* of the system A. Note that any Watson–Crick finite automaton is a parallel communicating Watson–Crick automata system of degree 1.

A configuration of a parallel communicating Watson–Crick automata system is a $2n$-tuple $(s_1, \binom{u_1}{v_1}, s_2, \binom{u_2}{v_2}, \ldots, s_n, \binom{u_n}{v_n})$ where s_i is the current state of the component i and $\binom{u_i}{v_i}$ is the part of the input word which has not been read yet by the component i, for all $1 \le i \le n$. We define a binary relation $\vdash$ on the set of all configurations by setting

$$\left(s_1, \binom{u_1}{v_1}, s_2, \binom{u_2}{v_2}, \ldots, s_n, \binom{u_n}{v_n}\right) \vdash \left(r_1, \binom{u_1'}{v_1'}, r_2, \binom{u_2'}{v_2'}, \ldots, r_n, \binom{u_n'}{v_n'}\right)$$

if and only if one of the following two conditions holds:

- $K \cap \{s_1, s_2, \ldots, s_n\} = \emptyset$, $\binom{u_i}{v_i} = \binom{x_i}{y_i}\binom{u_i'}{v_i'}$, and $r_i \in \delta_i(s_i, \binom{x_i}{y_i})$, $1 \le i \le n$;
- for all $1 \le i \le n$ such that $s_i = K_{j_i}$ and $s_{j_i} \notin K$ we have $r_i = s_{j_i}$, whereas for all the other $1 \le \ell \le n$ we have $r_\ell = s_\ell$. In this case $\binom{u_i'}{v_i'} = \binom{u_i}{v_i}$, for all $1 \le i \le n$.

If we denote by $\vdash^*$ the reflexive and transitive closure of $\vdash$, then the language recognized by a PCWKS is defined as:

$$L(A) = \left\{ w_1 \in V^* \,\middle|\, \left(q_1, \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}, q_2, \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}, \ldots, q_n, \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}\right) \vdash^* \left(s_1, \begin{bmatrix} \lambda \\ \lambda \end{bmatrix}, s_2, \begin{bmatrix} \lambda \\ \lambda \end{bmatrix}, \ldots, s_n, \begin{bmatrix} \lambda \\ \lambda \end{bmatrix}\right), \right.$$
$$\left. s_i \in F_i, 1 \le i \le n \right\}.$$

Please cite this article in press as: K. Chatterjee, K.S. Ray, Non-regular unary language and parallel communicating Watson–Crick automata systems, Theoret. Comput. Sci. (2017), https://doi.org/10.1016/j.tcs.2017.09.008