Annals of Pure and Applied Mathematics
Vol. 8, No. 2, 2014, 75-82
ISSN: 2279-087X (P), 2279-0888(online)
Published on 17 December 2014
www.researchmathsci.org

Timed Watson-Crick Online Tessellation Automaton

Mary Jemima Samuel[1], V.R. Dare[2]* and *D.G. Thomas[3]

[1]Department of Mathematics, Anna Adarsh College for Women, Anna Nagar
Chennai – 600040, India. Email:dr_pjaya@yahoo.com
[2,3]Department of Mathematics, Madras Christian College, Tambaram East
Chennai – 600059, India
Email: [2]rajkumardare@yahoo.com, [3]dgthomasmcc@yahoo.com

Received 13 September 2014; accepted 21 November 2014

Abstract. Watson-Crick finite automata are language recognizing devices similar to finite automata introduced in DNA computing area. We define Watson-Crick online tessellation automata which work on double-stranded arrays where the two strands relate to each other through a complementary relation inspired by the DNA complementarity. Also we define timed Watson-Crick online tessellation automaton and some equivalence results have been established.

Keywords: Watson-Crick online tessellation automaton, equivalence

AMS Mathematics Subject Classification (2010): 03D05

1. Introduction

The remarkable progress made by molecular biology and biotechnology in the last couple of decades, particularly in sequencing, synthesizing and manipulating DNA molecules, gave rise to the possibility of using DNA as a support for computation. The computer science community quickly reacted to this challenge and many computational models were built to exploit the advantages of nano-level biomolecular computing. One of them is Watson-Crick automata.

Watson-Crick automata, introduced in [3] represent one instance of mathematical model abstracting biological properties for computational purposes. They are finite automata with two reading heads, working on double stranded sequences. One of the main features of these automata is that characters on corresponding positions from the two strands of the input are related by a complementarity relation similar with the Watson-Crick complementarity of DNA nucleotides. The two strands of the input are separately scanned from left to right by read only heads controlled by a common state.

Online tessellation automaton is a recognizing device for accepting finite arrays [4]. Online tessellation automaton reads elements diagonalwise. In this paper we define Watson-Crick online tessellation automaton and the language accepted by it.

Timed (finite) automaton is defined to accept timed words [1]. It is a finite automaton with a finite set of real-valued clocks. It accepts timed words which are infinite sequences in which a real valued time of occurrences is associated with each symbol. The infinite word α over an alphabet Σ is represented as a function from the set

N of all positive integers to Σ. We represent the infinite word α as $\alpha = a_1 a_2 \ldots$ where $\alpha(i) = a_i \in \Sigma$. The set of all infinite words over Σ is denoted as Σ^ω. If $L \subset \Sigma^\omega$ then L is called an ω-language. Timed ω-double-stranded sequence and timed Watson-Crick ω-automaton and the language accepted by it are defined. In this paper we extend the concept of timed online tessellation automaton [5] to Watson-Crick online tessellation automaton.

2. Preliminaries

Let Σ be a finite alphabet. Σ^* denotes the set of all finite words over Σ, λ is the empty word and Σ^+ is the set of all non empty finite words over Σ. i.e., $\Sigma^+ = \Sigma^* \setminus \{\lambda\}$.

Definition 1. [7] We now define a "complementarity" relation on the alphabet Σ (like the Watson-Crick complementarity relation among the four DNA nucleotides), $\rho \subseteq \Sigma \times \Sigma$ which is symmetric. Denote $WK_\rho(\Sigma) = \left\{ \begin{pmatrix} a \\ b \end{pmatrix} / \; a, b \in \Sigma, (a,b) \in \rho \right\}^*$.

The set $WK\rho(\Sigma)$ is called the Watson-Crick domain associated to Σ and ρ. The elements $\begin{pmatrix} a_1 \\ b_1 \end{pmatrix}\begin{pmatrix} a_2 \\ b_2 \end{pmatrix}\cdots\begin{pmatrix} a_n \\ b_n \end{pmatrix} \in WK_\rho(\Sigma)$ are also written in the form $\begin{pmatrix} w_1 \\ w_2 \end{pmatrix}$ for $w_1 = a_1 a_2 \ldots a_n$ and $w_2 = b_1 b_2 \ldots b_n$. According to the usual way of representing DNA molecules as double-stranded sequence, we also write the product monoid $\Sigma^* \times \Sigma^*$ in the form $\begin{pmatrix} \Sigma^* \\ \Sigma^* \end{pmatrix}$ and its elements in the form $\begin{pmatrix} x \\ y \end{pmatrix}$.

Definition 2. [6] A Watson-Crick finite automaton is a 6-tuple $M = (\Sigma, \rho, Q, s_0, F, \delta)$ where Σ - a finite alphabet, Q - a finite set of states, $\rho \subseteq \Sigma \times \Sigma$ - is a symmetric relation (the complementarity relation), $s_0 \in Q$ is the initial state, $F \subseteq Q$ is the set of final states and $\delta : Q \times \begin{pmatrix} \Sigma^* \\ \Sigma^* \end{pmatrix} \to 2^Q$ is a mapping such that $\delta(s, \begin{pmatrix} x \\ y \end{pmatrix}) \neq \phi$ only for finitely many triples $(s, x, y) \in Q \times \Sigma^* \times \Sigma^*$.

We can also write the transition of M as rewriting rules of the form $s\begin{pmatrix} x \\ y \end{pmatrix} \to s'$ such a rule having the same meaning as $s' \in \delta(s, \begin{pmatrix} x \\ y \end{pmatrix})$.

For $\begin{pmatrix} u_1 \\ u_2 \end{pmatrix}, \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \in \begin{pmatrix} \Sigma^* \\ \Sigma^* \end{pmatrix}$ and $s, s' \in Q$ we write $s\begin{pmatrix} u_1 \\ u_2 \end{pmatrix}\begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \Rightarrow s'\begin{pmatrix} w_1 \\ w_2 \end{pmatrix}$ iff $s\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \to s' \in \delta$.

If $\Rightarrow^*$ is the reflexive and transitive closure of the relation $\Rightarrow$, then the language accepted by a Watson-Crick automaton is

$$L(M) = \{w_1 \in \Sigma^*/s_0 \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \Rightarrow^* s_f \text{ for some } w_2 \in \Sigma^* \text{ and } s_f \in F\}$$

We see that a WK automaton is a finite automaton with a double-stranded tape (and two read heads, one for each strand of the tape).

Definition 3. [4] A two dimensional online tessellation automaton A = (Q, Σ, Q_0, δ, F) where Q - is a finite set of states, Σ - is a input alphabet, $Q_0 \subseteq Q$ - is a set of initial states, $F \subseteq Q$ - is a set of final states and $\delta : Q \times Q \times \Sigma \to 2^Q$ is a transition function. Computation by a two dimensional online tessellation automaton on an infinite array is as follows.

We introduce a special boundary symbol $\# \notin \Sigma$ below the first row and to the left of the first column of p and denote the resulting array by b(p). At time t = 0, an initial state $q_0 \in Q_0$ is associated with all positions of #. At time t = 1, a state from $\delta(q_0, q_0, a_{11})$ is associated with the position (1, 1) holding a_{11}. At time t = 2, states are associated simultaneously with positions (1, 2) and (2, 1) respectively holding a_{12} and a_{21}. If s_{11} is the state associated with the position (1, 1) then the states associated with the position (2, 1) is an element of $\delta(q_0, s_{11}, a_{21})$ and to the position (1, 2) is an element of $\delta(s_{11}, q_0, a_{12})$. We then proceed to the next diagonal and so on.

3. Watson-Crick online tessellation automaton

Definition 4. Watson-Crick online tessellation automaton is a 6-tuple M = (Σ, ρ, Q, q_0, F, δ), where Σ - is the input alphabet, $\rho \subseteq \Sigma \times \Sigma$ is a symmetric relation called the complementary relation on Σ, Q - is a set of states, $q_0 \in Q$ - inital state, $F \subseteq Q$ - is a set of final states

$\delta : Q \times Q \times \begin{pmatrix} \Sigma' \\ \Sigma' \end{pmatrix} \to 2^Q$ is a transition function where $\Sigma' = \Sigma \cup \{\lambda\}$.

i.e., $\delta(q_0, q_0, \begin{pmatrix} x \\ y \end{pmatrix}) = q_1$, such that $\delta(q, q', \begin{pmatrix} x \\ y \end{pmatrix}) \neq \phi$ only for finitely many $(q, q'\ x, y) \in Q \times Q \times \Sigma' \times \Sigma'$.

There are two planes. The automaton reads x from the 1st plane and y from the 2nd plane and enters state q_1. The 1st plane and 2nd plane corresponds to the upper level strand and lower level strand of a double stranded sequence.

Computation on the two planes follow the pattern of online tessellation automaton. A special boundary symbol $\# \notin \Sigma$ is below the first row and to the left of the first column.

At time t = 0, an initial state q_0 is associated with all positions of #. At time t = 1, a state from $\delta(q_0, q_0, \begin{pmatrix} x_{11} \\ y_{11} \end{pmatrix})$ is associated with the position (1, 1) holding x_{11} in 1st plane and y_{11} in 2nd plane. At time t = 2, states are associated simultaneously with positions (1, 2) and (2, 1) respectively both in the 1st plane and 2nd plane. If q_1 is the state associated with the position (1, 1) in both the planes then the state associated with the position (2, 1) is an element of $\delta(q_1, q_0, \begin{pmatrix} x_{21} \\ y_{21} \end{pmatrix})$ and to the position (1, 2) is an element of $\delta(q_0, q_1,$

$\begin{pmatrix} x_{12} \\ y_{12} \end{pmatrix}$). We then proceed to the next diagonal and so on. The first plane is the array denoted by p_1 and the second plane is the array denoted by p_2.

The language accepted by Watson-Crick online tessellation automaton M is defined as $L_{WC-OTA}(M) = \{p_1 \in \Sigma^{**} / \delta(q_0, \begin{pmatrix} p_1 \\ p_2 \end{pmatrix}) = q_f$ for some $q_f \in F$ where p_1 is the array from the 1st plane and p_2 is the array from the 2nd plane}.

4. Timed automaton

Definition 5. [2] A non-deterministic finite automaton is a 5-tuple of the form
$M = (Q, \Sigma, s_0, F, \delta)$ where
Q is the set of states
Σ is the alphabet
$s_0 \in Q$ is the initial state
$F \subseteq Q$ is the set of final states
$\delta : Q \times \Sigma \to 2^Q$ is the transition mapping written as $\delta(s, a) = s'$ where $s, s' \in Q$ and $a \in \Sigma$.
If $card(\delta(s, a)) \leq 1$ for all $s \in Q$ and $a \in \Sigma$, then we say that the automaton is deterministic.
A relation $\Rightarrow$ is defined in the following way on the set $Q \times \Sigma^*$ for $s, s' \in Q$, $a \in \Sigma$, $x \in \Sigma^*$, we write $(s, ax) \Rightarrow (s', x)$ if $s' \in \delta(s, a)$, by definition $(s, \lambda) \Rightarrow (s, \lambda)$. If $\Rightarrow^*$ is the reflexive and transitive closure of the relation $\Rightarrow$, then the language of the strings recognized by the automaton M is defined by

$$L(M) = \{x \in \Sigma^* / (s_0, x) \Rightarrow^* (s, \lambda), s \in F\}.$$

The language accepted by an automaton is called a regular language.

Let $x = a_1 a_2 \ldots \in \Sigma^\omega$. We say that $r : s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \xrightarrow{a_3} \ldots$ is a run of M over x if $s_0 = q_0$ and $\delta(s_{i-1}, a_i) = s_i$ for $i = 1, 2, \ldots$ and $s_i \in Q$. For a run r the set Inf(r) consists of the states $s \in Q$ such that $s = s_i$ for infinitely many $i \geq 0$.

The acceptance condition on Inf(r) are given below:

1. Büchi condition: $Inf(r) \cap F \neq \phi$.
2. Muller condition: A family $\mathcal{F} \subseteq 2^Q$ of final state sets is considered and $Inf(r) \in \mathcal{F}$.

Then $L^\omega(M)$ is the ω-language consisting of the words $x \in \Sigma^\omega$ such that M has an accepting run over x. An ω-language L is called ω-regular if there exists an automaton M such that $L = L^\omega(M)$.

We now recall the definition of timed automata and timed languages [1]. We define timed words by coupling a real-valued time with each symbol in a word. The set of nonnegative real numbers $\mathbb{R}$ is chosen as the time domain.

Definition 6. [1] A timed sequence $\tau = \tau_1 \tau_2 \ldots$ is an infinite sequence of time values $\tau_i \in \mathbb{R}$ (set of real numbers) with $\tau_i > 0$ satisfying the following conditions.

1. monotonicity: τ increases strictly monotonically i.e., $\tau_i < \tau_{i+1}$ for all $i \geq 1$.
2. for every $t \in \mathbb{R}$, there is some $i \geq 1$ such that $\tau_i > t$.

A timed word over Σ is a pair (σ, τ) where $\sigma = \sigma_1 \sigma_2 \ldots$ is an infinite word over Σ and τ is a time sequence. A timed language over Σ is a sequence of timed words over Σ. If a timed word (σ, τ) is viewed as an input of an automaton, it presents the symbol σ_i at time τ_i. If each symbol σ_i is interpreted to denote an event occurrence then the corresponding component τ_i is interpreted as the time of occurrence of σ_i. In an automaton the choice of the next state depends upon the input symbol read. Similarly in the timed automaton defined in [1], the choice also depends upon the time of the input symbol relative to the times of the previously read symbols. For this purpose a finite set of real valued clocks are associated with each transition table. A clock can be set to zero simultaneously with any transition. At any instant, the reading of a clock equals the time elapsed since the last time it was reset. With each transition a clock constraint is associated such that the transition is considered only if the current values of the clocks satisfy this constraint.

Definition 7. [1] A timed Büchi automaton
$M = (Q, \Sigma, Q_0, C, E, F)$ where
Q is a finite set of states
Σ is a finite alphabet
Q_0 is a set of initial states
C is a finite set of clocks
F is a set of final states with Büchi acceptance condition
$E \subset Q \times Q \times \Sigma \times 2^c \times \phi(c)$ gives the set of transitions and $\phi(c)$ are clock constraints over C.

An edge (s, s_1, a, μ, δ) represents a transition from state s to state s_1 on input symbol a. The set $\mu \subset C$ gives the clocks to be reset with this transition and δ is a clock constraint over C defined inductively by $\delta = x \leq c \,/\, c \leq x \,/\, \neg\, \delta \,/\, \delta_1 \vee \delta_2$ where x is a clock in C and c is a constant in the set of rationals.

A clock interpretation v for a set C of clocks assigns a real value to each clock i.e., it is a mapping from C to $\mathbb{R}$. We say that a clock interpretation v for C satisfies a clock constraint δ over C iff δ evaluates to true using the values given by v.
Given a timed word (σ, τ), the transition of M starts in its start states at time 0 with all its clock initialized to 0. As time advances the values of all clocks change reflecting the elapsed time. At time τ_i, M changes state from s to s_1 using the transition of the form $(s, s_1, \sigma_i, \mu, \delta)$ reading input σ_i if the current values of clocks satisfy δ. With this transition the clocks in μ are reset to 0 and thus start counting time with respect to the time of occurrence of its transition. A run of timed automaton is defined in the following way. A run records the state and the values of all the clocks at the transition points.

A run r denoted by $(\overline{s}, \overline{v})$ of a timed automaton over a timed word (σ, τ) is an infinite sequences of the form $r : < s_0, v_0 > \xrightarrow[\tau_1]{\sigma_1} < s_1, v_1 > \xrightarrow[\tau_2]{\sigma_2} < s_2, v_2 > \xrightarrow[\tau_3]{\sigma_3} \ldots$ where $s_i \in Q$ and $v_i \in (C \to \mathbb{R}]$ for all $i \geq 0$. A run $r = (\overline{s}, \overline{v})$ of a timed Büchi automaton over a timed word (σ, τ) is called an accepting run if $Inf(r) \cap F \neq \phi$. For a timed Büchi automaton M, the language L(M) of timed word it accepts is defined to be the set $\{(\sigma, \tau) /$

M has an accepting run over $(\sigma, \tau)\}$. A timed language L is a timed regular language if L = L(M) for some timed Büchi automaton M.

5. Timed Watson-Crick online tessellation automaton

Watson-Crick online tessellation automaton can be extended to define infinite arrays and thereby we define timed Watson-Crick online tessellation automaton on ω-arrays.

Definition 8. A timed Watson-Crick online tessellation automaton is given as $M = (\Sigma, \rho, Q, q_0, F, \delta, C, \phi(C))$ where Σ is the input alphabet, $\rho \subseteq \Sigma \times \Sigma$ is a symmetric relation, Q is a set of states, $q_0 \in Q$ is the initial state, $C = \{C_1, C_2, ..., C_n\}$ is a finite set of clocks $\delta : Q \times Q \times \begin{pmatrix}\Sigma' \\ \Sigma'\end{pmatrix} \to 2^Q \times C$ is a transition function where $\Sigma' = \Sigma \cup \{\lambda\}$.

i.e., $\delta\left(q_1, q_2, \begin{pmatrix}x_{ij} \\ y_{ij}\end{pmatrix}\right) = (q_3, C_k)$

$\phi(C) = \{v_i : \begin{pmatrix}\Sigma' \\ \Sigma'\end{pmatrix} \times Q \times \{C_i\} \to \mathbb{R}_+,\ i = 1,...,n\}$

$\mathbb{R}_+$ is the set of positive real numbers and $v_k\left(\begin{pmatrix}x_{ij} \\ y_{ij}\end{pmatrix}, q_1, C_k\right) = k_{ij}$ and

$F \subseteq Q$ is set of final states.

The language accepted by the timed Watson-Crick online tessellation automaton is defined as $L^{\omega\omega}_{TWC-OTA}(M) = \{ (p_1, \tau) / p_1$ is accepted by Büchi acceptance condition and τ is a time sequence and $\tau_k \le \Sigma_{i+j=k+1} k_{ij}\}$.

Example 1. Let $M = (\Sigma, \rho, Q, q_0, F, \delta, C, \phi(C))$ where $\Sigma = \{a, b, c\}$, $\rho = ((a, a), (b, b), (c, c))$, $Q = \{q_0, q_1, q_2, q_3, q_4\}$, $F = \{q_4\}$, $C = \{C_1, C_2, C_3, C_4\}$, $\phi(C) : \{v_i : \begin{pmatrix}\Sigma' \\ \Sigma'\end{pmatrix} \times Q \times C_i \to \mathbb{R}_+$, $i = 1, \ldots, n\}$ and $\delta : Q \times Q \times \begin{pmatrix}\Sigma' \\ \Sigma'\end{pmatrix} \to 2^Q \times C$ given by

$\delta(q_0, q_0, (\begin{smallmatrix}a \\ \lambda\end{smallmatrix})) = (q_1, C_1)$, $\delta(q_0, q_2, (\begin{smallmatrix}c \\ b\end{smallmatrix})) = (q_3, C_3)$, $\delta(q_0, q_1, (\begin{smallmatrix}a \\ \lambda\end{smallmatrix})) = (q_1, C_1)$,

$\delta(q_2, q_2, (\begin{smallmatrix}b \\ a\end{smallmatrix})) = (q_2, C_2)$, $\delta(q_1, q_0, (\begin{smallmatrix}a \\ \lambda\end{smallmatrix})) = (q_1, C_1)$, $\delta(q_0, q_3, (\begin{smallmatrix}c \\ b\end{smallmatrix})) = (q_3, C_3)$,

$\delta(q_0, q_1, (\begin{smallmatrix}b \\ a\end{smallmatrix})) = (q_2, C_2)$, $\delta(q_3, q_2, (\begin{smallmatrix}c \\ b\end{smallmatrix})) = (q_3, C_3)$, $\delta(q_1, q_1, (\begin{smallmatrix}a \\ \lambda\end{smallmatrix})) = (q_1, C_1)$,

$\delta(q_0, q_3, (\begin{smallmatrix}\lambda \\ c\end{smallmatrix})) = (q_4, C_4)$, $\delta(q_0, q_2, (\begin{smallmatrix}b \\ a\end{smallmatrix})) = (q_2, C_2)$, $\delta(q_3, q_3, (\begin{smallmatrix}c \\ b\end{smallmatrix})) = (q_3, C_3)$

$\delta(q_2, q_1, (\begin{smallmatrix}b \\ a\end{smallmatrix})) = (q_2, C_2)$, $\delta(q_0, q_4, (\begin{smallmatrix}\lambda \\ c\end{smallmatrix})) = (q_4, C_4)$, $\delta(q_4, q_4, (\begin{smallmatrix}\lambda \\ c\end{smallmatrix})) = (q_4, C_4)$,

$\delta(q_4, q_3, (\begin{smallmatrix}\lambda \\ c\end{smallmatrix})) = (q_4, C_4)$

$v_1((\begin{smallmatrix} a \\ \lambda \end{smallmatrix}), q_0, C_1) = 1$ and 0 elsewhere, $v_2((\begin{smallmatrix} b \\ a \end{smallmatrix}), q_1, C_2) = 2$ and 0 elsewhere

$v_2((\begin{smallmatrix} b \\ a \end{smallmatrix}), q_1, C_2) = 2$ and 0 elsewhere, $v_3((\begin{smallmatrix} c \\ b \end{smallmatrix}), q_2, C_3) = 3$ and 0 elsewhere

$v_3((\begin{smallmatrix} c \\ b \end{smallmatrix}), q_2, C_3) = 3$ and 0 elsewhere, $v_4((\begin{smallmatrix} \lambda \\ c \end{smallmatrix}), q_4, C_4) = 4$ and 0 elsewhere.

$L^{\omega\omega}_{TWC-OTA}(M) = \{(p, \tau) \in \Sigma^{\omega}$ / number of consecutive columns containing the letter `a' = number of consecutive columns containing the letter `b' followed by infinite finite number of columns containing the letter `c' in p and for $n \neq 1$,

$$t_i = 2i \text{ for } i = 3n, \quad t_i \geq 1 \text{ for } i \neq 3n.$$

Definition 9. A timed online tessellation Muller automaton is a tuple $M = (\Sigma, \rho, Q, q_0, \delta, C, \phi(C), \mathcal{F})$ where $\Sigma, \rho, Q, q_0, \delta, C, \phi(C)$ are as in Definition 8 and $\mathcal{F} \subseteq 2^Q$ is a family of final states. The collection of all timed arrays (p, τ) accepted by M is denoted as $L^{\omega\omega}_{TWC-OTA}(M)$.

Theorem 1. For a timed Watson-Crick online tessellation automaton, the acceptance conditions Büchi and Muller are equivalent.

Proof: Let $M = (\Sigma, \rho, Q, q_0, F, \delta, C, \phi(C))$ be a timed Watson-Crick online tessellation automaton accepted by the Büchi acceptance mode. Consider the timed online tessellation automaton M' where the acceptance is in the Muller mode with the same timed transition table as that of M and with the acceptance family $\mathcal{F} = \{Q' \subseteq Q, Q' \cap F \neq \phi\}$. Then $L^{\omega\omega}(M) = L^{\omega\omega}(M')$.

Conversely given the timed Watson-Crick OTA in Muller mode, we construct a timed Watson-Crick OTA in Büchi mode accepting the same array language.

Let the timed Watson-Crick OTA be given by $M = (\Sigma, \rho, Q, q_0, \delta, C, \phi(C), \mathcal{F})$. We see that $L^{\omega\omega}_{TWC-OTA\,Muller}(M) = \bigcup_{F \in \mathcal{F}} L^{\omega\omega}_{TWC-OTA\,Muller}(A_F)$ where $A_F = (\Sigma, \rho, Q, q_0, \delta, C, \phi(C), \{F\})$.

Hence it is enough to construct for each acceptance set F, a timed Watson-Crick OTA in Büchi mode as A'_F accepting $L^{\omega\omega}_{TWC-OTA\,Büchi}(A_F)$.

Assume $F = \{q_1, q_2, ..., q_k\}$. States of A'_F are of the form q_i where $q \in Q$ and $i \in \{0, 1, 2, ..., k\}$. The initial state is $\{q^0_0\}$. The automaton simulates the transition of A.

For every transition in A of the form $\delta\left(q_1, q_2, \begin{pmatrix} x_{ij} \\ y_{ij} \end{pmatrix}\right) = q_{ij}$ and $v_k\left(\begin{pmatrix} x_{ij} \\ y_{ij} \end{pmatrix}, q_{ij}, C_k\right) = k_{ij}$ the automaton A'_F has a transition $\delta'\left(q_1^{i_1}, q_2^{j_1}, \begin{pmatrix} x_{ij} \\ y_{ij} \end{pmatrix}\right) = q_{ij}^{i_1}$, $i_1 = 0, 1, 2, ..., k$ and $v_k\left(\begin{pmatrix} x_{ij} \\ y_{ij} \end{pmatrix}, q_{ij}, C_k\right) = k_{ij}$.

For every transition $\delta\left(q_1, q_2, \begin{pmatrix} x_{ij} \\ y_{ij} \end{pmatrix}\right)$ where q_1, $q_2 \in$ F for each i = 1, ..., k there is an A'_F transition $\delta\left(q_1^{i_1}, q_2^{j_1}, \begin{pmatrix} x_{ij} \\ y_{ij} \end{pmatrix}\right)$ where

$$j_1 = (i_1 + 1) \quad \text{mod } k \text{ if } q_0 = q_i^{i_1}$$
$$j_1 \neq (i_1 + 1) \quad \text{mod } k \text{ if } j_1 = i_1.$$

The only final state is q_k^k. Hence we have seen the equivalence of Büchi and Muller of timed Watson-Crick OTA.

6. Conclusion

Watson-Crick online tessellation automaton has been defined. Timed Watson-Crick online tessellation automaton is defined and the equivalence of Büchi and Muller timed Watson-Crick online tessellation automaton has been proved.

REFERENCES

1. R.Alur and D.L.Dill, A theory of timed automata, *Theoretical Computer Science,* 126 (1994) 183–235.
2. E.Petre, Watson-Crick ω-automata, *Journal of Automata, Languages and Combinatorics*, 8 (2003) 59–70.
3. R. Freund, Gh. Păun, G. Rozenberg and A. Salomaa, Watson-Crick finite automata, Proceedings of 3rd DIMACS Workshop on DNA based computers, Philadelphia (1997) 297–328.
4. K. Inoue and A. Nakamura, Some properties of two dimensional on-line tessellation acceptor, *Information Sciences,* 13 (1977) 95–121.
5. M.J.Samuel, V.R.Dare and D.G.Thomas, A study on timed online tessellation automata, Proceedings of the 2nd *National Conference on Mathematical and Computational Models* (NCMCM 2003), (2003) 267–272.
6. A.Păun and M.Păun, State and transition complexity of watson-crick finite automata, *Proceedings of Fundamentals of Computation Theory,* FCT '99, LNCS, 1684, Springer-Verlag (1999), 409–420.
7. W.Thomas, Languages, Automata and Logic, In G. Rozenberg and A. Salomaa, editors, Handbook of Formal Language, 3 (1997) 389–455.