Jurnal Teknologi

Full paper

Languages of Watson-Crick Petri Net

Nurhidaya Mohamad Jan[a*], Fong Wan Heng[b], Nor Haniza Sarmin[a], Sherzod Turaev[c]

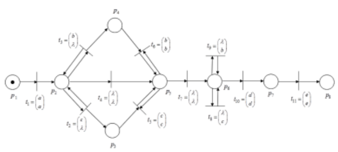
[a]*Department of Mathematical Sciences, Faculty of Sciences, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor, Malaysia*
[b]*Ibnu Sina Institute for Fundamental Science Studies, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor, Malaysia*
[c]*Department of Computer Science, Kulliyah of Information and Communication Technology, International Islamic University Malaysia, 53100 Kuala Lumpur, Malaysia*

*Corresponding author: nurhidayamj@gmail.com

Article history

Received :1 January 2014
Received in revised form :
1 June 2014
Accepted :10 September 2014

Graphical abstract



Abstract

Watson-Crick automata are one of the automata types that are used for transitions with two reading heads. This automata work on double stranded sequences related by a complementarity relation similar with the Watson-Crick complementarity of DNA molecules. Watson-Crick automata can be related to a Petri net, which is a graphical and mathematical formalism suitable for the modeling and analysis of concurrent, asynchronous and distributed systems. From the relation between Watson-Crick automata and Petri net, a new model namely Watson-Crick Petri net has been developed. In this paper, a new variant of Watson-Crick automata with Petri net as a control unit is introduced. We also presented various types of Watson-Crick Petri nets that are generated using transitions labelled with various labelling policies and finite sets of final markings.

Keywords: Automata; Watson-Crick; Petri net

Abstrak

Automata Watson-Crick adalah salah satu jenis automata yang menggunakan peralihan dengan dua kepala pembaca. Automata ini bertindak pada urutan dua terkandas yang berkaitan dengan hubungan saling melengkapi seperti pelengkapan Watson-Crick pada molekul DNA. Automata Watson-Crick boleh dikaitkan dengan jaring Petri, yang merupakan formalisme grafik dan matematik yang sesuai untuk model dan analisis serentak, segerak dan sistem teragih. Dari hubungan antara automata Watson-Crick dan jaring Petri, suatu model baru iaitu Watson-Crick jaring Petri telah dibangunkan. Dalam kertas kerja ini, satu varian baru automata Watson-Crick yang menggunakan jaring Petri sebagai unit kawalan diperkenalkan. Kami juga menyampaikan pelbagai jenis Watson-Crick jaring Petri yang dijanakan dengan menggunakan peralihan yang dilabelkan dengan pelbagai dasar pelabelan dan set terhingga tanda akhir.

Kata kunci: Automata; Watson-Crick; jaring Petri

1.0 INTRODUCTION

Deoxyribonucleic acid (DNA) is the genetic material in an organism. There are four bases in DNA which are adenine (A), thymine (T), guanine (G), and cytosine (C). Thermodynamically stable hydrogen bonding occurs between thymine and adenine, and between cytosine and guanine according to the Watson-Crick complementarity.[1] There are two hydrogen bonds between adenine and thymine, and three hydrogen bonds between cytosine and guanine. Watson-Crick base pairs allow the DNA helix to maintain a regular helical structure that is independent of its nucleotide sequence.

Watson-Crick automaton is an automaton counterpart of the computability model which is an abstraction of the way of using the Watson-Crick complementarity as in Adleman's DNA computing experiment known as the sticker system.[2] The automata work on tapes which are double stranded sequences of symbols related by Watson-Crick complementarity that are similar to the DNA molecules. Watson-Crick automata cannot exploit the other fundamental features of DNA molecules such as the massive parallelism. Therefore, we are considering Petri net as a control unit instead of Watson-Crick automata.

In 1962, Carl Adam Petri introduced petri net as a model of information flow in systems.[3] This model provided an elegant and useful mathematical formalism for modeling concurrent systems and their behaviors. This model can be represented by a graph or a net. In this paper, a new model that relates between Watson-Crick and Petri net called as Watson-Crick Petri net is introduced.

2.0 PRELIMINARIES

Some basic concepts and definitions in the Petri net and Watson-Crick automata[2-9] used throughout this paper are listed in the following.

Watson-Crick automata are finite state automata working on double-stranded tapes which are introduced to investigate the potential use of DNA molecules in computing. The definition of Watson-Crick automata is presented in the following.

Definition 1: Watson-Crick Automata[4]

A Watson-Crick automata is a construct $M=(V,\rho,K,s_0,F,\delta)$ where V is a finite set of alphabets, ρ is a symmetric relation, $\rho\subseteq V\times V$, K is the finite set of internal states, $s_0\in K$ is an initial state, $F\subseteq K$ is final state, δ is a transition function where $\delta: K\times\begin{pmatrix}V^*\\V^*\end{pmatrix}\to P(K)$ is a mapping such that $\delta\left(s,\begin{pmatrix}x\\y\end{pmatrix}\right)\neq\phi$ only for finitely many triples $(s,x,y)\in K\times V^*\times V^*$.

Next, an example of Watson-Crick automata is shown.

Example 1: Watson-Crick Automaton[5]

Figure 1 shows the transition diagram of a Watson-Crick automaton $M=(\{a,b,c\},\rho,\{q_0,q_1,q_2,q_f\},q_0,\{q_f\},\delta)$ where ρ and δ are given by

$$\rho=\left\{\begin{pmatrix}a\\a\end{pmatrix},\begin{pmatrix}b\\b\end{pmatrix},\begin{pmatrix}c\\c\end{pmatrix}\right\},$$

$$\delta\left(q_0,\begin{pmatrix}a\\\lambda\end{pmatrix}\right)=q_0,\qquad \delta\left(q_0,\begin{pmatrix}b\\a\end{pmatrix}\right)=q_1,$$

$$\delta\left(q_1,\begin{pmatrix}b\\a\end{pmatrix}\right)=q_1,\qquad \delta\left(q_1,\begin{pmatrix}c\\b\end{pmatrix}\right)=q_2,$$

$$\delta\left(q_2,\begin{pmatrix}c\\b\end{pmatrix}\right)=q_2,\qquad \delta\left(q_2,\begin{pmatrix}\lambda\\c\end{pmatrix}\right)=q_f,$$

$$\delta\left(q_f,\begin{pmatrix}\lambda\\c\end{pmatrix}\right)=q_f.$$

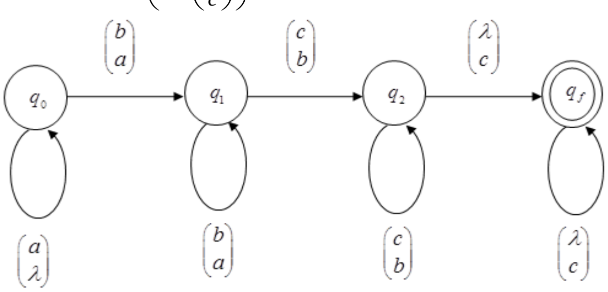


Figure 1 Watson Crick automaton, *M*

It is not difficult to see that the language accepted by Watson-Crick automaton M is $L(M)=\{a^nb^nc^n : n\geq 1\}$.

There are few types of Watson-Crick automata as stated in the following.

Definition 2: Types of Watson-Crick Automata[6]

There are several variants of Watson-Crick automata such as:

- stateless if $K=F=\{s_0\}$,
- all-final, if $F=K$,
- simple, if for all $s\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\\x_2\end{pmatrix}s'\in P$ where $x_1=\lambda$ or $x_2=\lambda$,
- 1-limited, if for all $s\begin{pmatrix}x_1\\x_2\end{pmatrix}\to\begin{pmatrix}x_1\\x_2\end{pmatrix}s'\in P$ where $|x_1x_2|=1$.

Petri net is a model of information flow in systems based on the concepts of asynchronous and concurrent operation by the parts of a system. Some definitions related to Petri net are listed in the following.

Definition 3: Petri Net[3]

A Petri net is a construct $N=(P,T,F,\phi)$ where P is the finite set of places, T is the finite set of transitions, $F\subseteq(P\times T)\cup(T\times P)$ is the set of directed arcs, $\phi: F\to\mathbb{N}$ is a weight function on the arcs.

A Petri net can be represented by a bipartite directed graph with the node set $P\cup T$ where places are drawn as circles, transitions as boxes or lines and arcs as arrows. The arrow representing an arc $(x,y)\in F$ is labeled with $\phi(x,y)$. If $\phi(x,y)=1$, then the label is omitted.

An example of Petri net is presented next.

Example 2: Petri net, *N*

The graph in Figure 2 represents a Petri net $N=(P,T,F,\phi)$ where P,T,F and ϕ are given by $P=\{p_1,p_2\}$, $T=\{t_1\}$, $F=\{(p_1,t_1),(t_1,p_2)\}$ and $\phi(p_1,t_1)=1$, $\phi(t_1,p_2)=2$.

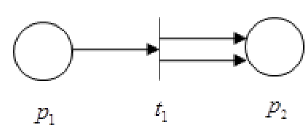


Figure 2 Petri net, *N*

Definition 4: Marking[3]

A mapping $\mu: P\to\mathbb{N}_0$ is called a marking. For each place $p\in P$, $\mu(p)$ gives the number of tokens in p. Tokens are drawn as small black dots in the places, e.g. •.

Pre- and post-sets of $x\in P\cup T$, are denoted as ${}^{\bullet}x=\{y\mid(y,x)\in F\}$ and $x^{\bullet}=\{y\mid(x,y)\in F\}$, respectively. The elements of ${}^{\bullet}p\,({}^{\bullet}t)$ are called input places (transitions) and the elements of $p^{\bullet}\,(t^{\bullet})$ are called output places (transitions) of $t(p)$ for $t\in T\,(p\in P)$.

Definition 5: Marked Petri Net[3]
A Petri net with tokens is called a marked petri net. A marked Petri net is a system $N=(P,T,F,\phi,i)$ where (P,T,F,ϕ) is a Petri net and i is the initial marking.

A transition $t\in T$ is enabled at a marking μ if and only if $\forall p\in P, \phi(p,t)\le \mu(p)$. If a transition t is enabled, transition t can fire by transforming the marking μ into the marking μ' defined by $\mu'(p)=\mu(p)-\phi(p,t)+\phi(t,p)$, $\forall p\in P$.

Definition 6: Petri Net with Final Marking[3]
A Petri net with final marking is a system $N=(P,T,F,\phi,i,M)$ where (P,T,F,ϕ,i) is a marked Petri net and $M\subseteq \Re(N,i)$ is set of markings which are called final markings. An occurance sequence ν of transitions is called succesful for M if it is enabled at the initial marking i and finished at a final marking τ of M.

Definition 7: Labelled Petri Net[3]
A labeled Petri net is a construct $A=(P,T,F,\phi,i,M,\Sigma,\ell)$ where $N=(P,T,F,\phi,i,M)$ is a Petri net with final markings, Σ is an alphabet and $\ell: T\to \Sigma\cup\{\lambda\}$ is a labelling function. The labelling function ℓ is extended to occurance sequences in natural way,i.e., if $\sigma t\in T^*$ is an occurance sequence then $\ell(\sigma t)=\ell(\sigma)\ell(t)$ and $\ell(\lambda)=\lambda$. For an sequence occurance $\sigma\in T^*, \ell(\sigma)$ is called a labell sequence.

Definition 8: Petri Net Language[7]
A language generated by a Petri net is a set of labell sequences corresponding to the occurrence sequences of the Petri net.

Definition 9: Types of Petri Net Languages[7]
A Petri net language generated by a labeled Petri net A is called

- free, f if a different labell is associated to each transition, and no transition is labelled with the empty strings,
- $\lambda-$free, $-\lambda$ if no transition is labelled with the empty string,
- arbitrary, λ if no restriction is posed on the labelling function ℓ.

In Petri net, there are four types of Petri net languages.

- P-type if M is the set of all reachable markings from the initial markings I, i.e. $M=\Re(N,i)$, $\Re(N,\mu)$ denotes the set of all reachable markings from a marking μ.
- L-type if $M\subseteq \Re(N,i)$ is a finite set,
- G-type if for a given set $M_0\subseteq \Re(N,i)$, each marking $\mu\in M$ is greater or equal to any marking M_0,
- T-type if M is the set of all terminal markings of N.

Next, the relationship of the families of Petri net language is shown in Figure 3. In the figure, the symbol $X\in\{P,L,G,T\}$ represents the family of X-type of Petri net languages whereas the symbol $y\in\{f,-\lambda,\lambda\}$ represents y-labeling policy. All arrows between two classes of languages in the figure indicate proper containment.

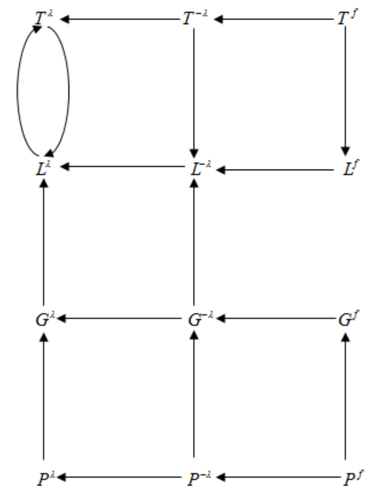


Figure 3 The relationship of the families of Petri net languages

Petri nets have several main structural subclasses and one of the subclasses is the state machine. A state machine (SM) is an ordinary Petri net such that each transition has exactly one input place and exactly one output place, i.e., $|{}^\bullet t|=|t^\bullet|=1$ for all $t\in T$. Figure 4 shows a state machine diagram.

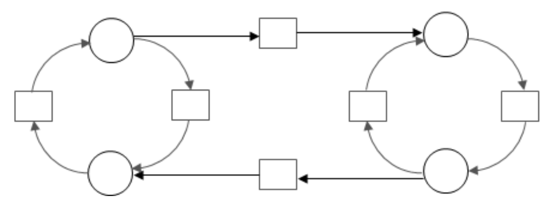


Figure 4 A state machine

In the study of languages, Petri nets have widely been used. Petri net languages are the set of strings generated by all possible firing sequences.

3.0 MAIN RESULTS

In this section, a new mathematical model that relate between Watson-Crick and Petri net known as Watson-Crick Petri net is introduced. Some definitions related to Watson-Crick Petri net are given in this section.

Definition 10: Watson-Crick Petri Net
A Watson-Crick Petri net is defined as $W=(N,\Sigma,\rho,\ell)$ where $N=(P,T,F,\phi,i,M)$ is a Petri net with final markings where P is the finite set of places, T is the finite set of transitions, $F\subseteq(P\times T)\cup(T\times P)$ is the set of directed arcs, $\phi: F\to\mathbb{N}$ is a weight function on the arcs, i is the initial marking, $M\subseteq\Re(N,i)$ is a set of markings which are called final markings, Σ is an alphabet,

$\rho \subseteq \Sigma \times \Sigma$ is a symmetric relation, and $\ell : T \to \begin{pmatrix} \Sigma \cup \{\lambda\} \\ \Sigma \cup \{\lambda\} \end{pmatrix}$ is a labelling function.

Definition 11: Watson-Crick Petri Net Language
A Watson-Crick Petri net language is a set of labell sequences corresponding to the occurrence sequences of the Watson-Crick Petri net.

Definition 12: Watson-Crick Petri Net Languages using the Class of Labelling Functions
A Watson-Crick Petri net language determined using the class of labelling functions is called

- strong free (denoted by *s*), if for any $t_1, t_2 \in T$ with $\ell(t_1) = \begin{pmatrix} a_1 \\ b_1 \end{pmatrix}$, $\ell(t_2) = \begin{pmatrix} a_2 \\ b_2 \end{pmatrix}$ where $a_1 \neq a_2$ and $b_1 \neq b_2$. Besides that, no transition is labelled with the empty strings, i.e. for any $t \in T$ with $\ell(t) = \begin{pmatrix} a \\ b \end{pmatrix}$ where $a \neq \lambda$ and $b \neq \lambda$.
- weak free (denoted by *w*), if for any $t_1, t_2 \in T$ with $\ell(t_1) = \begin{pmatrix} a_1 \\ b_1 \end{pmatrix}, \ell(t_2) = \begin{pmatrix} a_2 \\ b_2 \end{pmatrix}$ where $a_1 \neq a_2$ or $b_1 \neq b_2$. Besides, no transition is labelled with the empty strings, i.e. for any $t \in T$ with $\ell(t) = \begin{pmatrix} a \\ b \end{pmatrix}$, either $a \neq \lambda$ or $b \neq \lambda$.
- strong $\lambda-$free (denoted by $-s\lambda$), if no transition is labelled with the empty string, i.e. for any $t \in T$ with $\ell(t) = \begin{pmatrix} a \\ b \end{pmatrix}$ where $a \neq \lambda$ and $b \neq \lambda$.
- weak $\lambda-$free (denoted by $-w\lambda$), if no transition is labelled with the empty string, i.e. for any $t \in T$ with $\ell(t) = \begin{pmatrix} a \\ b \end{pmatrix}$, either $a \neq \lambda$ or $b \neq \lambda$.
- arbitrary (denoted by λ), if no restriction is posed on the labelling function ℓ.

Definition 13: Watson-Crick Petri Net Languages using the Set of Final States
A Watson-Crick Petri net language generated by a Watson-Crick Petri net *W* determined using the definition of the set of final states is called

- *P*-type if *M* is the set of all reachable markings from the initial markings *i*, i.e. $M = \Re(N, \iota)$, where $\Re(N, \iota)$ denotes the set of all reachable markings from a marking μ.
- *L*-type if $M \subseteq \Re(N, \iota)$ is a finite set.
- *G*-type if for a given set $M_0 \subseteq \Re(N, \iota)$, each marking $\mu \in M$ is greater or equal to any marking M_0.
- *T*-type if *M* is the set of all terminal markings of *N*.

Next, an example of Watson-Crick Petri net is shown.

Example 3:
Figure 5 represents the Watson-Crick Petri net $W = (N, \Sigma, \rho, \ell)$ where $P = \{p_1, p_2, ..., p_8\}$, $T = \{t_1, t_2, ..., t_{11}\}$ where $t_1 = \begin{pmatrix} a \\ a \end{pmatrix}$, $t_2 = \begin{pmatrix} c \\ \lambda \end{pmatrix}, t_3 = \begin{pmatrix} b \\ \lambda \end{pmatrix}$, $t_4 = t_7 = \begin{pmatrix} \lambda \\ \lambda \end{pmatrix}, t_5 = \begin{pmatrix} c \\ c \end{pmatrix}, t_6 = \begin{pmatrix} b \\ b \end{pmatrix}, t_8 = \begin{pmatrix} \lambda \\ c \end{pmatrix}$, $t_9 = \begin{pmatrix} \lambda \\ b \end{pmatrix}, t_{10} = \begin{pmatrix} d \\ d \end{pmatrix}, t_{11} = \begin{pmatrix} e \\ e \end{pmatrix}$, $F = \{(p_1, t_1), (t_1, p_2), (p_2, t_2)$, $(t_2, p_2), (p_2, t_3), (t_3, p_2), (p_2, t_4), (t_2, p_3), (t_3, p_4), (t_4, p_5)$, $(p_3, t_5), (t_5, p_5), (p_5, t_5), (p_4, t_6), (t_6, p_5), (p_5, t_6), (p_5, t_7)$, $(t_7, p_6), (p_6, t_8), (t_8, p_6), (p_6, t_9), (t_9, p_6), (p_6, t_{10}), (t_{10}, p_7)$, $(p_7, t_{11}), (t_{11}, p_8)\}, \phi(x, y) = 1$ for all $(x, y) \in P \times T \cup T \times P$, $i = [1, 0, ..., 0]$, $M = [0, 0, ..., 0, 1]$, $\Sigma = \{a, b, c, d, e\}$ and $\rho = \left\{ \begin{pmatrix} a \\ a \end{pmatrix}, \begin{pmatrix} b \\ b \end{pmatrix}, \begin{pmatrix} c \\ c \end{pmatrix}, \begin{pmatrix} d \\ d \end{pmatrix}, \begin{pmatrix} e \\ e \end{pmatrix} \right\}$.

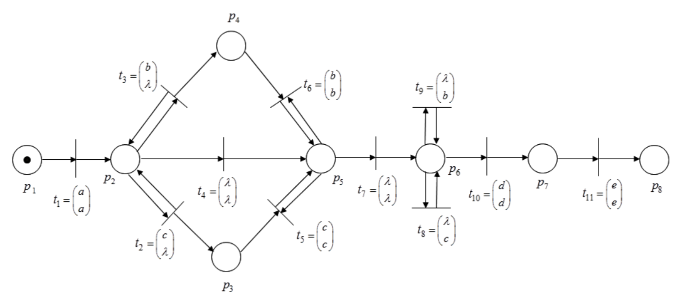


Figure 5 Watson-Crick Petri net, *W*

From the Watson-Crick Petri net, *W*, the language generated by Watson-Crick Petri net, $L(W)$ is $L(W) = \{aw : w \in \{b, c\}\} \cup \{aww' : w \in \{b, c\}$ and $w' = \text{prefix}(w)\} \cup \{aww'd : w \in \{b, c\}$ and $w' = \text{prefix}(w)\} \cup \{aww'e : w \in \{b, c\}$ and $w' = \text{prefix}(w)\}$.

Next, a theorem is obtained after analyzing Watson-Crick Petri net languages as follows:

Theorem 1: $WKT^{[\lambda]} = WKL^{[\lambda]}$
The families of Watson-Crick Petri net *T*-type and *L*-type languages are denoted as $WKT^{[\lambda]}$ and $WKL^{[\lambda]}$. Let Watson-Crick Petri net $W = (N, \Sigma, \rho, \ell)$ generate *T*-type languages with arbitrary labelling, *L*(*W*). So, $L(W) \in WKT^{[\lambda]}$. We need to show $L(W) = L(W')$ where $L(W')$ are *L*-type languages with arbitrary labelling.

Case 1: $WKT^{[\lambda]} \subseteq WKL^{[\lambda]}$

We need to show $L(W) \subseteq L(W')$. Let $P' = P$ where $P = \{p_1, p_2, ..., p_n\}$, $T' = T \cup T''$ where $T = \{t_1, t_2, ..., t_n\}$ and $T'' = \{t_{p_i} : p_i \in P\}$, $F' = F \cup F''$ where $F \subseteq (P \times T) \cup (T \times P)$ and $F'' = \{(p, t_{p_i}) : p \in P\}$, $\phi'(x,y) = \phi(x,y) \cup \phi''(p, t_p)$ where $\phi(x,y) = 1$ for $(x,y) \in F$ and $\phi''(p, t_p) = 1$, $i' = i$ where i is an initial marking, $M' = M$ where M is a final marking, $\Sigma' = \Sigma$ where Σ is an alphabet, $\rho' = \rho$ where $\rho \subseteq \Sigma \times \Sigma$ is a symmetric relation, $\ell' = \ell \cup \ell''$ where $\ell(t_i) = \binom{x}{y} \cup \binom{\lambda}{\lambda}$ for $\{x,y\} \in \Sigma$ and $\ell''(t_p) = \binom{\lambda}{\lambda}$.

Let L be the language generated by Watson-Crick Petri net. Then, L can be generated by W, the sample firing sequence is $t_1, t_2, ..., t_n, t_{p_1}, t_{p_2}, ..., t_{p_n}$. Hence, $L = L(W')$. Therefore, $L(W) \subseteq L(W')$.

Case 2: $WKL^{[\lambda]} \subseteq WKT^{[\lambda]}$

We need to show $L(W') \subseteq L(W)$. Let $P = P'$ where $P = \{p_1, p_2, ..., p_n\}$, $T = T' - T''$ where $T = \{t_1, t_2, ..., t_n\}$ and $T'' = \{t_{p_i} : p_i \in P\}$, $F = F' - F''$ where $F'' = \{(p, t_{p_i}) : p \in P\}$, $\phi(x,y) = \phi'(x,y)$ where $\phi(x,y) = 1$ for $(x,y) \in F$, $i = i'$ where i is an initial marking, $M = M''$ where M'' is the set of all terminal markings, $\Sigma = \Sigma'$ where Σ is an alphabet, $\rho = \rho'$ where $\rho \subseteq \Sigma \times \Sigma$ is a symmetric relation, $\ell = \ell' - \ell''$ where $\ell(t_i) = \binom{x}{y} \cup \binom{\lambda}{\lambda}$ for $\{x,y\} \in \Sigma$ and $\ell''(t_p) = \binom{\lambda}{\lambda}$.

Let L be the language generated by Watson-Crick Petri net. Then, W' can generate L if L is generated following the final marking M'. The sample firing sequence is $t_1, t_2, ..., t_n$. Hence, $L = L(W)$. Therefore, $L(W') \subseteq L(W)$.

4.0 DISCUSSION

Petri net is a model based on the concepts of asynchronous and concurrent operation by parts of a system; whereas, Watson-Crick automaton is inspired by the Watson-Crick complementarity of nucleotides in the double stranded DNA molecules. However, this automaton cannot exploit the other fundamental features of DNA molecules namely the massive parallelism. Therefore, a new mathematical model named Watson-Crick Petri net is introduced in this paper. This model relates between Watson-Crick automaton and a Petri net where the control unit of the Watson-Crick automaton is replaced by the Petri net. The language generated by a Watson-Crick Petri net can be determined either by using the class of labelling functions or the set of final markings.

5.0 CONCLUSION

In this paper, we introduced a new mathematical model that relates between Watson-Crick automata and Petri net namely Watson-Crick Petri net. Various types of Watson-Crick Petri net namely strong free, weak free, strong λ−free, weak λ−free, arbitrary, *P*-type, *L*-type, *G*-type and *T*-type are also presented. Besides that, an example and theorem of Watson-Crick Petri net language are also given.

Acknowledgement

The first author would like to thank UTM for the UTM Zamalah Scholarship. The second and third authors also would like to thank the Ministry of Education (MOE) and Research Management Centre (RMC), UTM for the financial funding through Research University Fund Vote No. 07J41.

References

[1] D. Freifelder. 1978. *The DNA Molecule Structure and Properties.* United States of America: W. H. Freeman and Company.
[2] L. Kari, G. Paun, G. Rozenberg, A. Salomaa and S. Yu. 1998. *Acta Informatica.* 35: 401.
[3] J. L. Peterson. 1976. *Journal Computer and System Sciences*. 13: 1.
[4] G. Paun, G. Rozenberg, and A. Salomaa. 1998. *DNA Computing: New Computing Paradigms.* Germany: Springer-Verlag Berlin Heidelberg New York.
[5] E. Czeizler and E. Czeizler. 2006. *Bull. EATCS.* 356: 190.
[6] R. Freund, G. Paun, G. Rozenberg and A. Salomaa. 1997. *Proceeding 3rd DIMACS Workshop on DNA Based Computers.* 297.
[7] S. Turaev. 2010. Petri Net Controlled Grammars. Ph.D. Thesis. Universitat Rovira I Virgili.
[8] J. L. Peterson. 1977. *Journal Computer and System Sciences*. 9: 223.
[9] H. Yen. 2006. *Studies in Computational Intelligence (SCI).* 25: 343.