MJFAS MALAYSIAN JOURNAL OF FUNDAMENTAL AND APPLIED SCIENCES
PRINT ISSN: 2289-5981 | ONLINE ISSN: 2289-599X

RESEARCH ARTICLE

State machine of place-labelled petri net controlled grammars

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

[a] Kolej PERMATA Insan, Kompleks PERMATA Insan, Universiti Sains Islam Malaysia, 71800 Nilai, Negeri Sembilan, Malaysia
[b] Department of Mathematical Sciences, Faculty of Science, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, 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 28 August 2017
Accepted 17 Novemberr 2017

Abstract

A place-labelled Petri net controlled grammar is, in general, a context-free grammar equipped with a Petri net and a function which maps places of the net to productions of the grammar. The languages of place-labelled Petri net controlled grammar consist of all terminal strings that can be obtained by parallel application of the rules of multisets which are the images of the sets of input places in a successful occurrence sequence of the Petri net. In this paper, we investigate the structural subclass of place-labelled Petri net controlled grammar which focus on the state machine. We also establish the generative capacity of state machine of place-labelled Petri net controlled grammars.

***Keywords*:** Petri net, context-free, grammar, state machine, structural subclass

INTRODUCTION

In 1962, Petri net was introduced by Petri as a model of the flow of information in systems based on the concepts of concurrent and asynchronous operation (Petri, 1962). This model provided an elegant and useful mathematical formalism for modeling concurrent systems and their behaviors.

The behavior of a concurrent system modelled by Petri net can be characterised by the set of all executable transition sequences. Moreover, a language over an alphabet of symbols can be corresponded to the executable transition sequences of the Petri net. Since Petri nets successfully describe and analyse the information flow and the control of action in such systems, they are useful tools for studying the properties of formal languages. Petri nets are initially used as language generating/accepting tools (for instance, see (Ginzburg *et al.*, 1980; Jantzen *et al.*, 1994; Valk *et al.*, 1981; Yen, 1996)). In recent studies, they have been widely applied as regulation mechanisms for grammar systems (ter Beek *et al.*, 2002), automata (Farwer *et al.*, 2008; Farwer *et al.*, 2007; Jantzen *et al.*, 2008; Zetzsche, 2009), and grammars (Dassow *et al.*, 2008; Dassow *et al.*, 2009; Stiebe, *et al.*, 2009).

A Petri net controlled grammar is a context-free grammar associated with a transition of Petri net and a function between transitions of the Petri net to grammar productions (Turaev, 2010). All the sets of terminal strings obtained by applying the sequence of productions, which is the image of an occurrence sequence of the Petri net under the function is called as a language.

Several variants of Petri net controlled grammars have been developed: A generalization of regularly controlled grammars has been considered rather than a finite automaton (Dassow *et al.*, 2008; Dassow *et al.*, 2009). A Petri net is associated with a context-free grammar and the sequence of applied rules which corresponds to an occurrence sequence of the Petri net is needed. Besides that, a *k*-Petri net controlled grammar has been developed and its properties have been investigated (Dassow *et al.*, 2008).

Another variant of Petri net controlled grammar is a Petri net with place capacities. When a Petri net regulates the defining grammar by permitting only the derivations, with each nonterminal number in each sentential form, it is shown that the Petri net is bounded by its capacity (Stiebe *et al.*, 2009).

However, for all variation of Petri net controlled grammars, the production rules of a grammar are associated only with transitions of a Petri net. Thus, it is also interesting to consider the place labelling strategies with Petri net controlled grammars. Theoretically, it would complete the node labelling cases, i.e., we study the cases where the production rules are associated with places of a Petri net, not only with its transitions. Moreover, the place labelling makes possible the consideration of parallel application of production rules in Petri net controlled grammars, which allows formal language based models to be developed for synchronized/parallel discrete event systems.

Informally, a place-labelled Petri net controlled grammar is a context-free grammar equipped with a Petri net and a function which maps places of the Petri net to productions of the context-free grammar (Mohamad Jan *et al.*, 2015). The structural subclass of place-labelled Petri net controlled grammar namely state machine has been investigated in this paper. This paper is organized as follows. *Methodology* contains the necessary definitions and results from the formal language theory, Petri nets and Petri net controlled grammars that are used in sequel. Finally, state machine of place-labelled Petri net controlled grammars is explored in *Main Results*.

METHODOLOGY

In this paper, the generative power of place-labelled Petri net controlled grammar is presented by the investigation of subclass of place-labelled Petri net controlled grammar, namely state machine. Firstly, a new definition of state machine for place-labelled Petri net controlled grammars is defined. Next, an example of state machine of place-labelled Petri net controlled grammar is given, where each transition of place-labelled Petri net controlled grammar has exactly one input place and exactly one output place. Besides that, a proposition and a theorem are presented, which shows the increase in the generative power of place-labelled Petri net controlled grammars by using the state machine. The theorem states that the state machine of Petri net is shown as a subset of state machine of place-labelled Petri net controlled grammars.

In this section, some basic concepts of the theories of formal languages, Petri nets and place-labelled Petri net controlled grammars (Mohamad Jan *et al.*, 2015; Reisig *et al.*, 1998; Linz, 2001; Rozenberg *et al.*, 1997; Peterson, 1976) that will be used throughout this paper are listed. We only recall some notions, notations and results directly related to the current work.

Some definitions of grammars, Petri nets and place-labelled Petri net controlled grammars are presented in the following.

Grammars

Some definitions used in describing grammars are listed below.

Definition 1 (Linz, 2001) An alphabet is a finite set of elements called terminal symbols or characters. Let $\Sigma = \{a_1, a_2, ..., a_k\}$ be an alphabet with k elements, i.e. its cardinality is $|\Sigma| = k$. The set of all strings over the alphabet Σ is denoted by Σ^*.

Definition 2 (Linz, 2001) A string is a finite sequence of symbols from the alphabet. Empty string λ is the only string satisfying the identity $x\lambda = \lambda x = x$ for every string x. A subset L of Σ^* is called a language. If $w = w_1 w_2 w_3$ for some $w_1, w_2, w_3 \in \Sigma^*$, then w_2 is called a substring of w. The length of a string w is denoted by $|w|$, and the number of occurrences of a symbol a in a string w by $|w|_a$.

Definition 3 (Linz, 2001) A context-free grammar is a quadruple $G = (V, \Sigma, S, R)$ where V and Σ are disjoint finite sets of nonterminal and terminal symbols, respectively, $S \in V$ is the start symbol and a finite set $R \subseteq V \times (V \cup \Sigma)^*$ is a set of production rules. Usually, a rule (A, x) is written as $A \to x$. A rule of the form $A \to \lambda$ is called an erasing rule.

Definition 4 (Rozenberg *et al.*, 1997) A string $x \in (V \cup \Sigma)^+$ directly derives a string $y \in (V \cup \Sigma)^*$, written as $x \Rightarrow y$, if and only if there is a rule $r = A \to \alpha \in R$ such that $x = x_1 A x_2$ and $y = x_1 \alpha x_2$. The reflexive and transitive closure of $\Rightarrow$ is denoted by $\Rightarrow^*$. A derivation using the sequence of rules $\pi = r_1 r_2 \cdots r_n$ is denoted by $\overset{\pi}{\Rightarrow}$ or $\overset{r_1 r_2 \cdots r_n}{\Rightarrow}$.

Definition 5 (Rozenberg *et al.*, 1997) The language generated by G is defined by $L(G) = \{w \in \Sigma^* \mid S \Rightarrow^* w\}$.

Petri Nets

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

Definition 6 (Peterson, 1976) A Petri net (PN) is a quadruple $N = (P, T, F, \phi)$ where P and T are disjoint finite sets of places and transitions, respectively, $F \subseteq (P \times T) \cup (T \times P)$ is the set of directed arcs, $\phi : F \to N$ is a weight function. 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 and arcs as arrows. The arrow representing an arc $(x, y) \in F$ is labelled with $\phi(x, y)$; if $\phi(x, y) = 1$, then the label is omitted.

Definition 7 (Reisig *et al.*, 1998) A mapping $\mu : P \to N_0$ is called a marking. For each place $p \in P, \mu(p)$ gives the number of tokens in p. Graphically, tokens are drawn as small solid dots inside circles. The symbols ${}^\bullet x = \{y \mid (y, x) \in F\}$ and $x^\bullet = \{y \mid (x, y) \in F\}$ are called pre- and post-sets of $x \in P \cup T$, respectively. For $X \subseteq P \cup T$, define ${}^\bullet X = \bigcup_{x \in X} {}^\bullet x$ and $X^\bullet = \bigcup_{x \in X} x^\bullet$. For $t \in T (p \in P)$, the elements of ${}^\bullet t ({}^\bullet p)$ are called input places (transitions) and the elements of $t^\bullet \left(p^\bullet\right)$ are called output places (transitions) of ${}^\bullet t \left({}^\bullet p\right)$.

Definition 8 (Reisig *et al.*, 1998) A transition $t \in T$ is enabled by marking μ if and only if $\mu(p) \geq \phi(p, t)$ for all $p \in {}^\bullet t$. In this case t can occur (fire). Its occurrence transforms the marking μ into the marking μ' defined for each place $p \in P$ by $\mu'(p) = \mu(p) - \phi(p, t) + \phi(t, p)$. We write $\mu \overset{t}{\Rightarrow}$ to denote that t may fire in μ, and $\mu \overset{t}{\Rightarrow} \mu'$ to indicate that the firing of t in μ leads to μ'.

Definition 9 (Reisig *et al.*, 1998) A marking μ is called terminal in which no transition is enabled. A finite sequence $t_1 t_2 \cdots t_k \in T^*$ is called an occurrence sequence enabled at a marking μ and finished at a marking μ' if there are markings $\mu_1, \mu_2, \ldots, \mu_{k-1}$ such that $\mu \overset{t_1}{\Rightarrow} \mu_1 \overset{t_2}{\Rightarrow} \ldots \overset{t_{k-1}}{\Rightarrow} \mu_{k-1} \overset{t_k}{\Rightarrow} \mu'$.

In short this sequence can be written as $\mu \overset{t_1 t_2 \cdots t_k}{\Rightarrow} \mu'$ or $\mu \overset{\nu}{\Rightarrow} \mu'$ where $\nu = t_1 t_2 \cdots t_k$. For each $1 \leq i \leq k$, marking μ_i is called reachable from marking μ. $\mathcal{R}(N, \mu)$ denotes the set of all reachable markings from a marking μ.

Definition 10 (Reisig *et al.*, 1998) A marked Petri net is a system $N = (P, T, F, \phi, \iota)$ where (P, T, F, ϕ) is a Petri net, ι is the initial marking. A Petri net with final markings is a construct $N = (P, T, F, \phi, \iota, M)$ where (P, T, F, ϕ, ι) is a marked Petri net and $M \subseteq \mathcal{R}(N, \iota)$ is the set of markings which are called final markings.

In order to improve Petri net analyzability, several structural subclasses are introduced by restriction on their structure namely state machine, generalized state machine, marked graph, generalized marked graph, casual net, free-choice net, extended free-choice net and assymmetric choice net.

In this paper, we only consider the structural subclass of state machine.

Definition 11 (Peterson, 1976) 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 $t \in T$.

Place-Labelled Petri Net Controlled Grammars

In the following, some concepts of Petri net controlled grammars are presented.

Definition 12 (Mohamad Jan *et al.*, 2015) A place-labelled Petri net controlled grammar (a *p*PN controlled grammar for short) is a 7-tuple $G = (V, \Sigma, R, S, N, \beta, M)$ where (V, Σ, R, S) is a context-free

grammar, N is a (marked) Petri net, $\beta: P \to R \cup \{\lambda\}$ is a place labeling function and M is a set of final markings.

Let $A \subseteq P$, we use the notations $\beta(A)$ and $\beta_{-\lambda}(A)$ to denote the multisets $[\beta(p)\rangle\ p \in A]$ and $[\beta(p)\rangle\ p \in A$ and $\beta(p) \neq \lambda]$, respectively.

Definition 13 (Mohamad Jan *et al.*, 2015) The symbol $x \in (V \cup \Sigma)^*$ directly derives $y \in (V \cup \Sigma)^*$ with a multiset $\pi = \lfloor A_{i_1} \to \alpha_{i_1}, A_{i_2} \to \alpha_{i_2}, \ldots, A_{i_k} \to \alpha_{i_k} \rfloor \subseteq R^{\oplus}$ written as $x \overset{\pi}{\Rightarrow} y$, if and only if $x = x_1 A_{i_1} x_2 A_{i_2} \cdots x_k A_{i_k} x_{k+1}$ and $y = x_1 \alpha_{i_1} x_2 \alpha_{i_2} \cdots x_k \alpha_{i_k} x_{k+1}$ where $x_j \in (V \cup \Sigma)^*$ for $1 \le j \le k+1$, and $\pi = \beta_{-\lambda}(\bullet t)$ for some $t \in T$ enabled at a marking $\mu \in R(N, \iota)$.

Definition 14 (Mohamad Jan *et al.*, 2015) A derivation

$$S \overset{\pi_1}{\Rightarrow} w_1 \overset{\pi_2}{\Rightarrow} w_2 \overset{\pi_3}{\Rightarrow} \cdots \overset{\pi_n}{\Rightarrow} w_n = w \in \Sigma^* \qquad (1)$$

where $\pi_i \subseteq R^{\oplus}, 1 \le i \le n,$ is called successful if and only if $\pi_i = \beta_{-\lambda}(\bullet t_i)$ for some $t_i \in T, 1 \le i \le n,$ and $t_1 t_2 \cdots t_n \in T^*$ is a successful occurrence sequence in N. For short, (1) can be written as $S \overset{\pi_1 \pi_2 \cdots \pi_n}{\Rightarrow} w$.

Definition 15 (Mohamad Jan *et al.*, 2015) The language $L(G)$ generated by a grammar G consists of strings $w \in \Sigma^*$ such that there is a successful derivation $S \overset{\pi_1 \pi_2 \cdots \pi_n}{\Rightarrow} w$ in G.

A language of place-labelled Petri net controlled grammar can be determined by the class of labelling strategies and the set of final markings. Some definitions used in describing various variants of languages generated by place-labelled Petri net controlled grammars are defined below.

Definition 16 (Mohamad Jan *et al.*, 2015) A place-labelled Petri net controlled grammar $G = (V, \Sigma, S, R, N, \beta, M)$ determined by the class of labelling strategies is called:

- free (denoted by f) if a different label is associated to each place, and no place is labelled with the empty string;
- λ-free (denoted by -λ) if no place is labelled with the empty string;
- arbitrary (denoted by λ) if no restriction is posed on the labelling function β.

Definition 17 (Mohamad Jan *et al.*, 2015) A place-labelled Petri net controlled grammar $G = (V, \Sigma, S, R, N, \beta, M)$ determined by the set of final markings is called:

- r-type if M is the set of all reachable markings from the initial marking ι i.e. $M = \mathcal{R}(N, \iota)$,
- t-type if $M \subseteq \mathcal{R}(N, \iota)$ is a finite set.

The notation $(x, y) - p\text{PN}$ controlled grammar where $x \in \{f, -\lambda, \lambda\}$ shows the type of labelling function and $y \in \{r, t\}$ shows the type of a set of final markings. The notation $p\text{PN}^{\lambda}(x, y)$ and $p\text{PN}(x, y)$ denote the families of languages generated by $(x, y) - p\text{PN}$ controlled grammars with and without erasing rules respectively, where $x \in \{f, -\lambda, \lambda\}$ and $y \in \{r, t, g\}$. Also, the bracket notation $p\text{PN}^{[\lambda]}(x, y), x \in \{f, -\lambda, \lambda\}, y \in \{r, t\}$ is used to say that a statement holds both in case with erasing rules and in case without erasing rules.

The effects of labelling strategies of place-labelled Petri net controlled grammars are presented in (Mohamad Jan *et al.*, 2015). The following proposition on the effects of labelling strategies of place-labelled Petri net controlled grammars is obtained immediately from Definition 16.

Proposition 1 (Mohamad Jan *et al.*, 2015)
For $y \in \{r, t\}$, $p\text{PN}^{[\lambda]}(f, y) \subseteq p\text{PN}^{[\lambda]}(-\lambda, y) \subseteq p\text{PN}^{[\lambda]}(\lambda, y)$.

MAIN RESULTS

In this section, we establish the generative capacity of state machine structural subclass of place-labelled Petri net controlled grammars. This new concept is given in the following definition.

Definition 18
A place-labelled Petri net controlled grammar is said to be a state machine (pSM) if each transition has exactly one input place and exactly one output place, i.e., $|\bullet t| = |t\bullet| = 1$ for all $t \in T$.

From Definition 18, Proposition 2 shows the inclusion properties of free, λ – free and arbitrary place-labelled Petri net controlled grammar.

Proposition 2
For $y \in \{r, t\}$, $p\text{SM}^{[\lambda]}(f, y) \subseteq p\text{SM}^{[\lambda]}(-\lambda, y) \subseteq p\text{SM}^{[\lambda]}(\lambda, y)$.

Proof
From Proposition 1, free place-labelled Petri net controlled grammar (with or without erasing rules) is a subset of λ – free place-labelled Petri net controlled grammar (with or without erasing rules), thus subset to arbitrary place-labelled Petri net controlled grammar (with or without erasing rules). Thus this completes the proof.

Next, an example of state machine of place-labelled Petri net controlled grammar is given. Example 1 illustrates the place-labelled Petri net controlled grammar that generates $L = \{ww \mid w \in \{a, b\}^*\} \in p\text{PN}(f, t)$.

Example 1
Let $G = \{\{S, A, B\}, \{a, b\}, R, S, N, \beta, M\}$ be a place-labelled Petri net controlled grammar with the production rules R:

$$S \to AB, A \to aA, A \to a, A \to b, A \to bA, B \to B',$$
$$B' \to a, B' \to aB, B' \to b, B' \to bB.$$

A marked Petri net $N = \{P, T, F, \phi, \iota\}$ is labelled with a set of places $P = \{p_1, p_2, \cdots, p_{10}\}$, a set of transitions $T = \{t_1, t_2, \cdots, t_{15}\}$, a set of directed arcs:

$$F = \{(p_1, t_1), (t_1, p_2), (p_2, t_2), (p_2, t_3), (p_2, t_4), (p_2, t_5), (t_2, p_3), (t_3, p_5),$$
$$(t_4, p_7), (t_5, p_9), (t_2, p_3), (t_3, p_5), (t_4, p_7), (t_5, p_9), (p_3, t_6), (p_5, t_7), (p_7, t_8),$$
$$(p_9, t_9), (t_6, p_4), (t_7, p_6), (t_8, p_8), (t_9, p_{10}), (p_6, t_{10}), (t_{10}, p_6), (p_8, t_{11}),$$
$$(t_{11}, p_8), (p_4, t_{12}), (t_{12}, p_2), (p_{10}, t_{13}), (t_{13}, p_2), (p_6, t_{14}), (p_8, t_{15})\}.$$

The weight function $\phi(x, y) = 1$ for all $(x, y) \in P$ and the initial marking $\iota = \{1, 0, \cdots, 0\}$. Also, the labelling function of the places are labelled with

$$\beta(p_1) = S \to ABC, \beta(p_2) = B \to B', \beta(p_3) = A \to aA,$$
$$\beta(p_4) = B' \to aB, \beta(p_5) = A \to a, \beta(p_6) = B' \to a,$$
$$\beta(p_7) = A \to b, \beta(p_8) = B' \to b, \beta(p_9) = A \to bA, \beta(p_{10}) = B' \to bBA,$$

and the final marking $M = \{0, \cdots, 0, 0\}$.

Figure 1 represents a Petri net N with respect to G. Each transition in N has exactly one input place and exactly one output place, $|\bullet t| = |t\bullet| = 1$ for all $t \in T$. Then, the language is $L = \{ww \mid w \in \{a, b\}^*\} \in p\text{PN}(f, t)$.

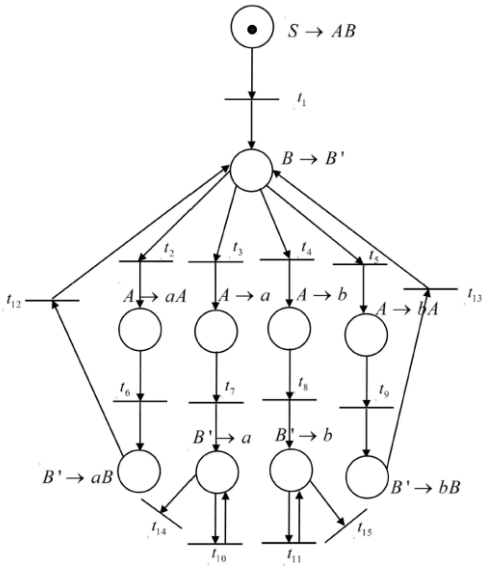

Fig. 1 A state machine of place-labelled Petri net controlled grammar G.

Furthermore, the state machine of Petri net is shown as a subset of state machine of place-labelled Petri net controlled grammars in the following theorem.

Theorem 1

For $x \in \{f, \lambda, -\lambda\}$ and $y \in \{r, t\}$, $\mathrm{SM}(x,y) \subseteq p\mathrm{SM}(x,y)$.

Proof

Let $G = (V, \Sigma, S, R, N, \beta, M)$ be a (x,y) state machine Petri net controlled grammar where $x \in \{f, -\lambda, \lambda\}$ and $y \in \{r,t\}$. For the place $p \in P$, where $\iota(p) \neq 0$, we introduce a new place p_0, a set of transitions $T_0 = \{t_{0_i} \mid t_{0_i} \in T, 1 \le i \le \lceil p \rceil\}$, and a set of arcs, $F_0 = \{(p_0, t_{0_i}) \mid 1 \le i \le \lceil p \rceil\} \cup \{(t_{0_i}, p_t) \mid t \in p^{\bullet}\}$.

We define a set of transitions, $T_f = \{t_f \in T \mid ((t_f)^{\bullet})^{\bullet} = \phi\}$, a set of places $P_f = \{p_{t_f} = p_f \mid t_f \in T_f\}$, and a set of arcs $F_f = \{(p, t_f) \mid p \in P \wedge p^{\bullet} = \phi, t_f \in T_f\} \cup \{(t_f, p_f) \mid t_f \in T_f, p_f \in P_f, \beta(p') = \gamma(p)\}$.

Let $G = (V, \Sigma, S, R, N', \gamma, M')$ with the Petri net $N' = (P', T', F', \phi', \iota')$ where the set of places, transitions and arcs are constructed as $P' = P \cup \{p_0\} \cup P_f, T' = T \cup T_0 \cup T_f$, and $F' = F \cup F_0 \cup F_f$ respectively, the weight function $\phi' : F' \to N$ is set as $\phi'(x,y) = 1$ for all $(x,y) \in F'$, and the initial marking is defined as

$$\iota'(p_0) = \iota(p) \text{ if } \mu(p) \neq \phi,$$

$$\iota'(p) = \begin{cases} \iota(p) & \text{if } p \in P, \\ 0 & \text{otherwise .} \end{cases}$$

Further, we set the place-labelling function $\beta' : P' \to R$ by setting

$$\beta'(p) = \gamma(p) \text{ for all } p \in P,$$

$$\beta'(p') = \beta(p) \text{ for all } p^{\bullet} = \phi,$$

and for every $\mu \in M$, we set $\nu_{\mu} \in M'$ as $\nu_{\mu}(p) = \mu(p)$ for all $p \in P$.

We now consider the successful derivation in G:

$$S \overset{r_1}{\Rightarrow} w_1 \overset{r_2}{\Rightarrow} \ldots \overset{r_n}{\Rightarrow} w_n = w \in \Sigma^*. \quad (2)$$

We can also construct the derivation

$$S \overset{[r_1]}{\Rightarrow} w_1 \overset{[r_2]}{\Rightarrow} \ldots \overset{[r_n]}{\Rightarrow} w_n = w \in \Sigma^*$$

which simulates (2) and it is not difficult to see that the occurrence sequence corresponding to the derivation $t_{0_1} t_{0_2} \ldots t_{0_{|p|}} t_1 \ldots t_n t_f$, which starts at the initial marking ι and ends at a final marking $\mu \in M'$. Thus, $L(G) = L(G')$.

CONCLUSION

We investigated the structural subclasses of place-labelled Petri net controlled grammar namely the state machine. The first lemma shows the inclusion properties of *free*, $\lambda - free$ and *arbitrary* place-labelled Petri net controlled grammar. We also showed that state machine of Petri net is a subset of state machine of place-labelled Petri net controlled grammars.

ACKNOWLEDGEMENT

The first author would like to thank Universiti Teknologi Malaysia (UTM) for the UTM Zamalah Scholarship. The second author and third authors would also like to thank the Ministry of Higher Education (MOHE) and Research Management Centre (RMC), UTM for the financial funding through Research University Grant Vote No. 13H18. Also, the fourth author would like to thank International Islamic University Malaysia (IIUM) for the financial funding through Research Initiative Grant Scheme RIGS16-368-0532."

REFERENCES

Petri, C. A. (1962). *Kommunication mit Automaten*. University of Bonn. Ph.D. Thesis.

Ginzburg, A., Yoeli, M. 1980. Vector Addition Systems and Regular Languages. *J. Comput. System Sci.* 20, 227-284.

Jantzen, M., Petersen, H. 1994. Cancellation in Context-Free Languages: Enrichment by Induction. *Theoret. Comput. Sci.* 127, 149-170.

Valk, R., Vidal-Naquet, G. 1981. Petri Nets and Regular Languages. *J. Comput. System Sci.* 23, 229-325.

Yen, H. C. 1996. On the Regularity of Petri Net Languages. *Inf. and Comput.* 124, 168–181.

ter Beek, M. H., Kleijn, H. C. M. 2002. Petri Net Control for Grammar Systems. *Form. and Natur. Comput.* 2300, 220–243.

Farwer, B., Jantzen, M., Kudlek, M., Rolke, H., Zetzsche, G. 2008. Petri Net Controlled Finite Automata. *Fund. Inform.* 85, 111–121.

Farwer, B., Kudlek, M., Rolke, H. 2007. Concurrent Turing Machines. *Fund. Inform.* 79, 303–317.

Jantzen, M., Kudlek, M., Zetzsche, G. 2008. Language Classes Defined by Concurrent Finite Automata. *Fund. Inform.* 85, 267–280.

Zetzsche, G 2009, 'Grid Erasing in Petri Net Languages and Matrix Grammars', in 13th International Conference on Developments in Language Theory: proceedings of 13th ICDLT, Universität Stuttgart, Stuttgart, Germany, 30 June-3 July 2009, pp. 490-501.

Dassow, J., Turaev, S. 2008. Arbitrary Petri Net Controlled Grammars', in Language and Automata Theory and Application: proceedings of LATA, Rovira i Virgili University, Tarragona, Spain, 13-19 March 2008, pp. 27-39.

Dassow, J., Turaev, S. 2009. Grammars Controlled by Special Petri Nets', Language and Automata Theory and Application: proceedings of LATA, Rovira i Virgili University, Tarragona, Spain, 2-8 April 2009, pp. 326-337.

Stiebe, R., Turaev, S. 2009. Capacity Bounded Grammars. *J. Autom. Lang. Comb.* 15, 175-194.

Turaev, S. (2010). *Petri Net Controlled Grammars*. Universitat Rovira I Virgili. Ph.D. Thesis.

Dassow, J., Turaev, S. 2009. Petri Net Controlled Grammars: The Power of Labeling and Final Markings. *Romanian J. of Inform. Sci. and Tech.* 12, 191-207.

Dassow, J., Turaev, S. 2010. Petri Net Controlled Grammars with a Bounded Number of Additional Places. *Acta Cybernet.* 19, 609-634.

Mohamad Jan, N., Turaev, S., Fong, W. H., Sarmin, N. H. 2015. A New Variant of Petri Net Controlled Grammars. 22th National Symposium on Mathematical Science: proceedings of SKSM22, Grand Blue Wave Hotel, Shah Alam, Selangor, Malaysia, 24-26 November 2014, pp. 24-26.

Reisig, W., Rozenberg, G. 1998, Lectures on Petri Nets I: Basic Models, Berlin: Springer-Verlag.
Linz, P. 2001. An Introduction to Formal Languages and Automata 3rd. Edition. USA: Jones and Bartlett Publishers.
Rozenberg, G., Salomaa, A. 1997, Handbook of Formal Languages. Berlin: Springer-Verlag.
Peterson, J. L., 1976. Computation Sequence Sets. *J. Comput. System Sci.* 13, 1-24.