Journal of Fundamental Sciences Vol. 6, No. 2 (2010) 98-103.

ISSN 1823-626X

**Journal of Fundamental Sciences**

**available online at http://jfs.ibnusina.utm.my**

# Grammars controlled by petri nets with inhibitor arcs

Sherzod Turaev* and Mohd Hasan Selamat

*Faculty of Computer Science and Information Technology, University Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia.*

Received 24 May 2010, Revised 25 June 2010, Accepted 17 July 2010, Available online 29 October 2010

**ABSTRACT**

A *Petri net controlled grammar* is a grammar equipped with a Petri net whose transitions are labeled with production rules of the grammar, and the associated language consists of all terminal strings which can be derived in the grammar and the sequence of rules in every terminal derivation corresponds to some occurrence sequence of transitions of the Petri net which is enabled at the initial marking and finished at a final marking of the net. In this paper we define *grammars controlled by Petri nets with inhibitor arcs* and investigate their computational capacities.

| Formal grammars and languages | Grammars with regulated rewriting | Petri nets | Petri net controlled grammars |

http://dx.doi.org/10.11113/mjfas.v6n2.190

## 1. INTRODUCTION

The symbolic representation of a modeled system in the form of strings makes its processes by information processing tools very easy. Moreover, Coding Theory, Cryptography, Computation Theory, Computational Linguistics, Natural Computing, and many other fields directly use strings for the description and analysis of modeled systems. In the modeling we usually have to deal with infinite sets of strings. Thus, it is very important to define some finite devices which generate all the strings of these sets. If we consider strings as *words*, then a set of strings can be considered as a *language* and a generative device as a *grammar*. In general, a grammar consists of finite sets of *terminal* symbols, *nonterminal* symbols and *production rules* as well as the *axiom* (the start symbol). To generate a word in the language, one begins with the start symbol. Then production rules are applied in any order, until a word that contains neither the start symbol nor nonterminal symbols is produced.

With respect to the forms of production rules, grammars are divided into two major classes: *context-free* where the left-hand side of each production rule consists of a single nonterminal symbol and *context-sensitive* where the left-hand side of each production rule consists of a nonterminal symbol surrounded by a context of terminal and nonterminal symbols. Though context-free grammars have beautiful mathematical properties, and they are easily applicable in practical problems, they can't cover all aspect of modeled phenomena. On the other hand, context-sensitive grammars are too powerful to be used in applications. In order to overcome this problem, we need some "context-free like" generative devices, which have as many context-free-like properties as possible, but are also able to describe the non-context-free features of the specific languages in the question. One of the solutions is that a context-free grammar should be considered with some additional (control) mechanism which restricts the application of the rules in order to avoid some derivations and obtain a subset of the language generated in usual way. The computational power of some grammars with control mechanism turns out to be greater than the power of context-free grammars. The consideration of different types of control mechanisms leads to the definition of different types of grammars with controlled (regulated) rewriting. For instance, in the monograph [5], we can find the detailed information on various types of grammars with regulated rewriting such as matrix, programmed, valence, random context, and tree controlled grammars, etc.

However, the rapid developments in present day industry, biology, and other areas challenge to deal with various tasks which need suitable tools for their modeling and investigation. We introduce *Petri net controlled grammars* as models for representing and analyzing of the systems such as automated manufacturing systems in industry and metabolic pathways in systems biology, where Petri nets are responsible for the structure and communication of the systems, and grammars represent generative processes in these systems. In our recent papers [1, 2, 3, 4, 7, 8, 9] we have defined various types of Petri net controlled grammars (considering different classes of place/transition Petri nets) motivated by different theoretical

*Corresponding author at: Faculty of Computer Science and Information Technology, University Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia*
*E-mail address: sherzod@fsktm.upm.edu.my (Sherzod Turaev)*

and practical problems, and studied their mathematical properties.

Our study has shown that the use of place/transition Petri nets as control mechanisms for context-free grammars allows only generating families of languages, which are strict subclasses of the family of context-sensitive languages. Thus, it is interesting to consider grammars controlled by high level Petri nets (Petri nets with inhibitor arcs, colored Petri nets, etc.), which have a larger modeling and computational power. In this paper we define grammars controlled by Petri nets with inhibitor arcs and investigate their computational power.

## 2. PRELIMINARIES

**Grammars.** Let $\Sigma$ be an *alphabet*, which is a finite nonempty set of symbols. A *word* over the alphabet $\Sigma$ is a finite sequence of symbols from the alphabet $\Sigma$. The *empty* word is denoted by $\lambda$. The set of all words over the alphabet $\Sigma$ is denoted by $\Sigma^*$. A subset of $\Sigma^*$ is called a *language*.

A *phrase structure* grammar is a quadruple $G = (V, \Sigma, S, R)$ where $V$ and $\Sigma$ are two disjoint alphabets of *nonterminal* and *terminal* symbols, respectively, $S\in V$ is the *start symbol* and $R \subseteq (V\cup\Sigma)^*V(V\cup\Sigma)^*\times(V\cup\Sigma)^*$ is a finite set of (*production*) *rules*. Usually, a rule $(u,v)\in R$ is written in the form $u\to v$. A rule of the form $u\to\lambda$ is called an *erasing rule*. A string $x\in(V\cup\Sigma)^*$ *directly derives* a string $y\in(V\cup\Sigma)^*$ in $G$, written as $x[\rangle y$, if and only if there is a rule $u\to v\in R$ such that $x = x_1ux_2$ and $y=x_1vx_2$ for some $x_1, x_2\in(V\cup\Sigma)^*$. The reflexive and transitive closure of the relation $[\rangle$ is denoted by $[\rangle^*$. A derivation using the sequence of rules $\pi = r_1r_2\cdots r_k$, $r_i\in R$, $1\leq i\leq k$, is denoted by $[\pi\rangle$ or $[r_1r_2\cdots r_k\rangle$. A string $w\in(V\cup\Sigma)^*$ such that $S[\rangle^*w$ is called a *sentential form*. The *language* generated by $G$, denoted by $L(G)$, is defined by $L(G) = \{w\in\Sigma^* \mid S[\rangle^*w\}$. The families of languages generated by phrase structure grammars are denoted by **RE**.

A phrase-structure grammar $G = (V, \Sigma, S, R)$ is called *context-sensitive* if each rule $u\to v\in R$ has $u = u_1Au_2$, $v = u_1xu_2$ for $u_1, u_2\in(V\cup\Sigma)^*$, $A\in V$ and $x\in(V\cup\Sigma)^+$ (in context sensitive grammars $S\to\lambda$ is allowed, provided that $S$ does not appear in the right-hand members of rules in $R$), and it is called *context-free* if each rule $u\to v\in R$ has $u\in V$.

The families of languages generated by context-sensitive and context-free grammars are denoted by **CS** and **CF**, respectively. The next strict inclusions hold (for details, see [6]).

### Theorem 2.1 CF $\subset$ CS $\subset$ RE

A *matrix grammar* is a quintuple $G = (V, \Sigma, M, S, B)$ where $V$, $\Sigma$ and $S$ are defined as in phrase structure grammars, $M$ is a finite set of matrices each of which is a finite sequence $m$: $(A_1\to w_1, A_2\to w_2, \ldots, A_n\to w_n)$, $n\geq1$, of context-free rules over $(V\cup\Sigma)$, and $B$ is a finite set of occurrences of such rules on $M$. For some words $x$ and $y$ in $(V\cup\Sigma)^*$ and a matrix $m$: $(A_1\to w_1, A_2\to w_2, \ldots, A_n\to w_n)\in M$, we write $x[m\rangle y$ iff there are words $x_0, x_1, \ldots, x_n$ such that $x_0=x$, $x_n=y$, and for $1\leq i\leq n$, either
$x_{i-1} = z_{i-1}A_iz'_{i-1}$, $x_i = z_{i-1}w_iz'_{i-1}$ for some $z_{i-1}, z'_{i-1}\in(V\cup\Sigma)^*$
or $x_{i-1} = x_i$, the rule $A_i\to w_i$ is not applicable to $x_{i-1}$, and this occurrence of $A_i\to w_i$ appears in $B$. One says that the rules whose occurrences appear in $B$ are used in *appearance checking mode*, and that a matrix grammar is defined with (without) appearance checking if $B \neq \emptyset$ ($B = \emptyset$). The language generated by $G$ is defined as $L(G) = \{w\in\Sigma^* \mid S[\rangle^*w\}$. The family of languages generated by matrix grammars with appearance checking (without appearance checking) is denoted by $\mathbf{MAT}^{[\lambda]}{}_{\text{ac}}$ ($\mathbf{MAT}^{[\lambda]}$). The following theorem shows the relationships of the families of matrix languages (for details, see [5]).

### Theorem 2.2 CF $\subset$ MAT $\subset$ MAT$_{\text{ac}}$ $\subset$ CS *and* MAT $\subseteq$ MAT$^{\lambda}$ $\subset$ MAT$^{\lambda}{}_{\text{ac}}$ = RE

**Petri nets.** A *place/transition Petri net* (simply *Petri net*) is a construct $N = (P, T, F, \phi)$ 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*, $\phi$: $F\to\{1,2, \ldots\}$ 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 $\phi(x,y)=1$, the label is omitted. A mapping $\mu$: $P\to\{0,1,2, \ldots\}$ 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. ${}^\bullet x=\{y \mid (y,x)\in F\}$ and $x^\bullet=\{y \mid (x,y)\in F\}$ are called *pre-* and *post-sets* of $x\in P\cup T$, respectively. For $t\in T$ ($p\in P$), the elements of ${}^\bullet t$ (${}^\bullet p$) are called *input* places (transitions) and the elements of $t^\bullet$ ($p^\bullet$) are called *output* places (transitions) of the transition $t$ (the place $p$).

A transition $t\in T$ is *enabled* by marking $\mu$ 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 $\mu$ into the marking $\mu'$ defined for each place $p\in P$ by $\mu'(p)=\mu(p)-\phi(p,t)+\phi(t,p)$. We write $\mu[t\rangle\mu'$ to indicate that the firing of $t$ in $\mu$ leads to $\mu'$. A finite sequence $t_1t_2\cdots t_k$, $t_i\in T$, $1\leq i\leq k$, is called *an occurrence sequence* enabled at a marking $\mu$ and finished at a marking $\mu'$ if there are markings $\mu_1, \mu_2, \ldots, \mu_{k-1}$ such that $\mu[t_1\rangle\mu_1[t_2\rangle\mu_2[t_3\rangle\cdots[t_{k-1}\rangle\mu_{k-1}[t_k\rangle\mu'$. In short this sequence can be written as $\mu[t_1t_2\cdots t_k\rangle\mu'$ or $\mu[\nu\rangle\mu'$ where $\nu= t_1t_2\cdots t_k$.

A *marked* Petri net is a system $N = (P, T, F, \phi, \iota)$ where $(P, T, F, \phi)$ is a Petri net, $\iota$ is the *initial marking*.

**Example 2.3.** Figure 1 depicts a Petri net $N$=$(P, T, F, \phi, \iota)$ with $P$=$\{p_1, p_2, p_3, p_4, p_5\}$ and $T$=$\{t_1, t_2, \text{t}_3, t_4\}$. We can see that $\phi(p_3,t_3)$=2. The initial marking $\iota$ is defined by $\iota(p_1)$=$\iota(p_4)$=1 and $\iota(p)$=0 for all $P-\{p_1, p_4\}$.

A Petri net *with final markings* is a construct $N = (P, T, F, \phi, \iota, M)$ where $(P, T, F, \phi, \iota)$ is a marked Petri net and $M$ is a set of markings which are called *final* markings. An occurrence sequence $\nu$ of transitions is called *successful* for

$M$ if it is enabled at the initial marking $\iota$ and finished at a final marking $\tau$ of $M$.

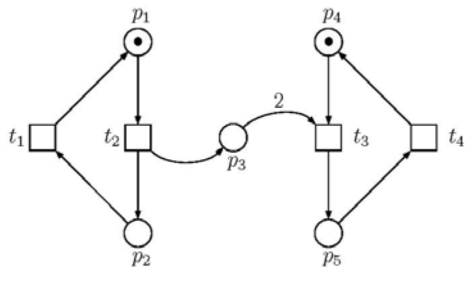

**Figure 1:** A Petri net

A *Petri net with inhibitor arcs* is a 6-tuple $N = (P, T, F, I, \phi, \iota)$ where $(P, T, F, \phi, \iota)$ is a marked net and $I \subseteq P\times T$ is a set of *inhibitor arcs*. Graphically, an inhibitor arc is represented by a special arrow head. Figure 2 shows a Petri net with inhibitor arcs, which is obtained from the Petri net in Figure 1 by adding an inhibitor arc $(p_2,t_3)$. Let $^\circ t = \{p\in P \mid (p,t)\in I\}$. A transition $t\in T$ can occur at a marking $\mu$ if and only if $\mu(p) = 0$ for all $p\in {}^\circ t$ and $\mu(p) \geq \phi(p,t)$ for all $p\in P-{}^\circ t$.

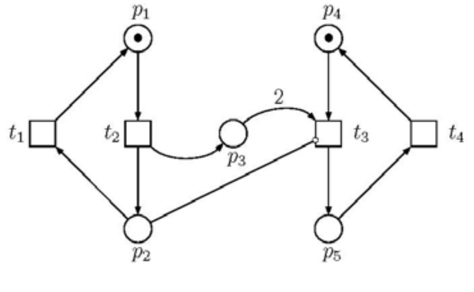

**Figure 2:** A Petri net with inhibitor arcs

### Petri net controlled grammars.

The construction of the following type of Petri nets is based on the idea of using similarity between the firing of a transition and the application of a production rule in a derivation in which places are nonterminals and tokens are different occurrences of nonterminals.

A *context-free Petri net* (a *cf Petri net* for short) with respect to a context-free grammar $G = (V, \Sigma, S, R)$ is a tuple $N = (P, T, F, \phi, \beta, \gamma, \iota)$ where

- $(P, T, F, \phi)$ is a Petri net;
- the labelling functions $\beta$: $P\rightarrow V$ and $\gamma$: $T\rightarrow R$ are bijections;
- there is an arc from place $p$ to transition $t$ if and only if $\gamma(t) = A\rightarrow\alpha$ and $\beta(p) = A$. The weight of the arc $(p,t)$ is 1;
- there is an arc from transition $t$ to place $p$ if and only if $\gamma(t) = A\rightarrow\alpha$ and $\beta(p) = X$ where $|\alpha|_X>0$. The weight of the arc $(t,p)$ is $|\alpha|_X$;
- the initial marking $\iota$ is defined by $\iota(\beta^{-1}(S)) = 1$and $\iota(p) = 0$ for all $p\in P-\{\beta^{-1}(S)\}$.

**Example 2.4.** Figure 3 illustrates the cf Petri net $N_1$ with respect to the grammar $G_1 = (\{S, A, B\}, \{a,b\}, S, R)$ where $R$ consists of $r_0$: $S\rightarrow AB$, $r_1$: $A\rightarrow aAb$, $r_2$: $A\rightarrow\lambda$, $r_3$: $B\rightarrow cB$, $r_4$: $B\rightarrow\lambda$.

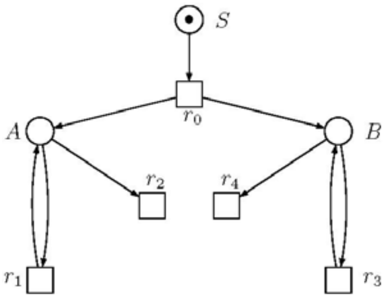

**Figure 3:** A context-free Petri net $N_1$

We recall the definition of an $s$-Petri net, i.e. context-free Petri nets with such additional places and arcs that these new places with input and output transitions of a net compose cycles with the common place (for details, see [1, 8, 9]).

Let $G=(V, \Sigma, S, R)$ be a context-free grammar with its corresponding context-free Petri net $N = (P, T, F, \phi, \beta, \gamma, \iota)$. Let $T_1, T_2, \ldots, T_n$ be a partition of $T$.

An $s$-Petri net is a Petri net with final markings $N_s = (P\cup Q, T, F\cup E, \varphi, \xi, \gamma, \mu_0, \tau)$ where

- $Q = \cup_{\rho\in\Pi}P_\rho$, $E = \cup_{\rho\in\Pi}F_\rho$;
- the weight function $\varphi$ is defined by $\varphi(x,y) = \phi(x,y)$ if $(x,y)\in F$ and $\varphi(x,y) = 1$ if $(x,y)\in E$;
- the labelling function $\xi$: $P\cup Q\rightarrow V\cup\{\lambda\}$ is defined by $\xi(p) = \beta(p)$ if $p\in P$ and $\xi(p) = \lambda$ if $p\in Q$;
- $\mu_0$ is the initial marking where $\mu_0(p) = \iota(p)$ if $p\in P$, $\mu_0(p_0) = 1$ and $\mu_0(p) = 0$ if $p\in Q-\{p_0\}$;
- $\tau$ is the final marking where $\tau(p_0) = 1$ and $\tau(p) = 0$ if $p\in(P\cup Q)-\{p_0\}$.

**Example 2.5.** Figure 4 depicts an $s$-Petri net which are constructed with respect to the grammar $G_2 = (\{S, A, B\}, \{a, b\}, S, R)$ where $R$ consists of $r_0$: $S\rightarrow AB$, $r_1$: $A\rightarrow\lambda$, $r_2$: $B\rightarrow\lambda$, $r_3$: $A\rightarrow aA$, $r_4$: $B\rightarrow aB$, $r_5$: $A\rightarrow bA$, $r_6$: $B\rightarrow bB$.

An *s-PN controlled grammar* is a system $G = (V, \Sigma, S, R, N_s)$ where $G' = (V, \Sigma, S, R)$ is a context-free grammar and $N_s$ is an $s$-Petri net with respect to the context-free grammar $G'$. The *language* generated by an $s$-Petri net controlled grammar $G$ consists of all strings $w\in\Sigma^*$ such that there is a derivation $S[r_1r_2\cdots r_k\rangle w\in\Sigma^*$ and a successful occurrence sequence of transitions $\nu=t_1t_2\cdots t_k$ of $N_s$ such that $r_1r_2\cdots r_k =\gamma( t_1t_2\cdots t_k)$. $\mathbf{PN}^{[\lambda]}_s$ denotes the family of languages generated by $s$-PN controlled grammars. It was shown in [10] that

**Theorem 2.2.** $\mathbf{MAT}^{[\lambda]} = \mathbf{PN}^{[\lambda]}_s$.

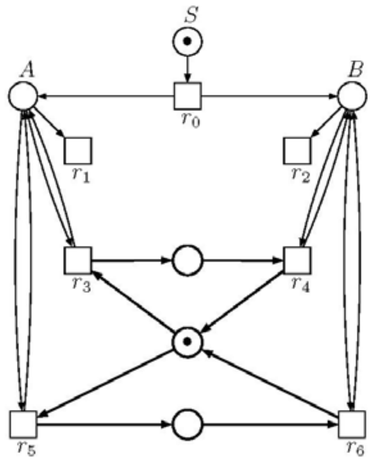

**Figure 4:** An *s*-Petri net $N_2$ (additional places are represented without labels)

Further, we recall the main definitions and main results concerning to arbitrary (place/transition) Petri net controlled grammars (for details, see [2, 3, 4]).

A *Petri net controlled grammar* is a tuple $G = (V, \Sigma, S, R, N, \gamma, M)$ where $V, \Sigma, S, R$ are defined as for a context-free grammar and $N = (P, T, F, \varphi, \iota)$ is a (marked) Petri net, $\gamma$: $T \to R \cup \{\lambda\}$ is a transition labelling function and $M$ is a finite set of final markings (this definition uses the extended form of the transition labelling function $\gamma$: $T^* \to R^*$; this extension is done in the usual manner).

The *language* generated by a Petri net controlled grammar $G$ consists of all strings $w \in \Sigma^*$ such that there is a derivation $S[r_1r_2\cdots r_k\rangle w \in \Sigma^*$and an occurrence sequence $\nu = t_1t_2\cdots t_s$ which is successful for $M$ such that $r_1r_2\cdots r_k = \gamma( t_1t_2\cdots t_s)$.

**Example 2.5.** Let $G_3 = (\{S, A, B, C\}, \{a, b, c\}, S, R, N_3, \gamma_3, M_3)$ be a Petri net controlled grammar where $R$ consists of rules $S \to ABC$, $A \to aA$, $B \to bB$, $C \to cC$, $A \to a$, $B \to b$, $C \to c$; the net $N_3$ is illustrated in Figure 5 and $M_3 = \{\mu\}$ with $\mu(p) = 0$ for all $p \in P$, then $G_3$ generates the language $L(G_3) = \{a^nb^nc^n \mid n \geq 1\} \notin$ **CF**.

**PN** and **PN**$^\lambda$ denote the families of languages generated by PN controlled grammars without and with erasing rules, respectively. The computational power of Petri net controlled grammars has been investigated in [4, 11]. It was shown that

**Theorem 2.2.** **MAT** $\subseteq$ **PN** = **PN**$^{[\lambda]}$ = **MAT**$^{[\lambda]}$.

## 3. THE EFFECT OF INHIBITOR ARCS

**Definition 3.1.** A Petri net controlled grammar $G = (V, \Sigma, S, R, N, \gamma, M)$ is called a *grammar controlled by a Petri net with inhibitor arcs* (in short, a *PNI controlled grammar*) if $N$ is a Petri net with inhibitor arcs.

The language $L(G)$ generated by a PNI controlled grammar $G$ is defined as usual manner. **PNI** and **PNI**$^\lambda$ denote the families of languages generated by PNI controlled grammars without and with erasing rules, respectively.

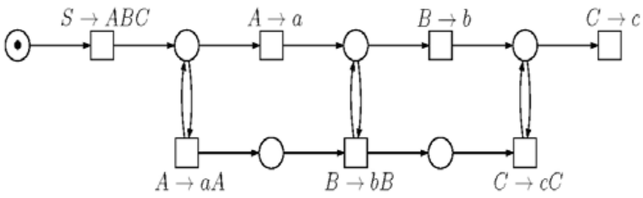

**Figure 5:** A Petri net $N_3$

**Example 2.5.** We consider the PNI controlled grammar $G_4 = (\{S, S', A\}, \{a\}, S, \{S \to SS|S'|A, S' \to S, A \to a\}, N_4, \gamma_4, M_4)$ where $N_4$ and its transition labeling are illustrated in Figure 6, and $M_4 = \{\mu\}$ with $\mu(p) = 0$ for all $p \in P$. We show that $G_4$ generates the language $L(G_4) = \{ a^{2^n} \mid n \geq 1\}$.

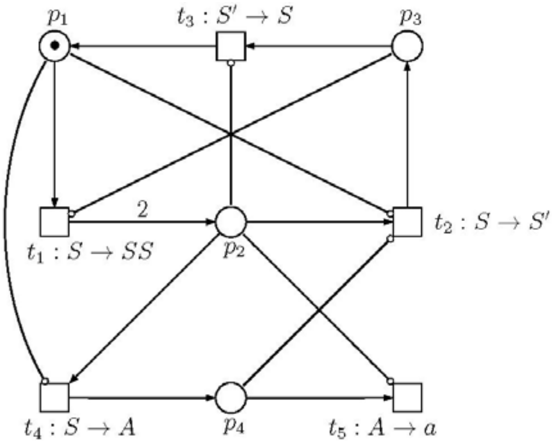

**Figure 6:** A Petri net with inhibitor arcs $N_4$

***Proof***. Let place $p_1$ has $k \geq 1$ tokens. Inhibitor arcs $(p_1,t_2)$ and $(p_1,t_4)$ controls that all tokens are moved from place $p_1$ to place $p_2$ before transition $t_2$ or transition $t_4$ fires. Then place $p_2$ receives $2k$ tokens. In the next step transitions $t_2$ and $t_4$ are enabled. If transition $t_2$ ($t_4$) occurs, then transition $t_4$ ($t_2$) cannot occur until all tokens are removed from place $p_3$ ($p_4$). Thus only one of transitions $t_2$ and $t_4$ can occur. Inhibitor arc $(p_2,t_3)$ ($(p_2,t_5)$) controls that all tokens are moved from place $p_2$ to place $p_3$ ($p_4$). Inhibitor arc $(p_3,t_1)$ controls that all tokens are moved from place $p_3$ to place $p_1$ until transition $t_1$ fires. Thus $2k$ tokens (i.e. the doubled number of the tokens) return to place $p_1$. The occurrences of transition $t_5$ $2k$ times remove all tokens from place $p_4$ and the execution of the Petri net reaches to the final marking. The corresponding derivation for this occurrence of transitions has the form

$$D: S[\rangle^* S^k[\rangle^* S^{2k}[\rangle^* S'^{2k} [\rangle^* S^{2k} [\rangle^* S^{4k} [\rangle^* \cdots [\rangle^* S^{2\cdot 2\cdots 2k} [\rangle^* \cdots$$

Since in the initial marking place $p_1$ has only one token, Derivation $D$ generates sentential forms $S^{2^n}$, $n\geq 1$. Then in order to terminate the derivation we replace all $S$'s with $A$'s, i.e.

$$D': S[\rangle^* S^{2\cdot 2\cdots 2k} [\rangle^* A^{2\cdot 2\cdots 2k} [\rangle^* a^{2^n} .$$

Next we prove the main results of the paper.

**Lemma 3.3. MAT$^{[\lambda]}_{ac} \subseteq$ PNI$^{[\lambda]}$.**

***Proof***. Let $G = (V, \Sigma, S, M, B)$ be a matrix grammar with or without erasing rules. Without loss of generality we assume that $G$ is without repetitions. Let $R$ be the set of all rules occurred in $M$. By Theorem 2.6, we can construct an $s$-PN controlled grammar $G' = (V, \Sigma, S, R, N_s)$ equivalent to the matrix grammar $G'' = (V, \Sigma, S, M)$.

We modify the net $N_s = (P\cup Q, T, F\cup E, \varphi, \xi, \gamma, \mu_0, \tau)$ as follows. Let $r$: $A\rightarrow\alpha\in B$ with $\gamma(t) = r$, $\xi(p) = A$. Let $r'$: $A'\rightarrow\alpha'$ is the next rule after the rule $r$ in the matrix, i.e. $m$: $(\ldots, A\rightarrow\alpha, A'\rightarrow\alpha', \ldots)$. Let $\gamma(t') = r'$ and $\xi(p') = A'$. We add a new transition $t_A$ labeled with $r'$, an inhibitor arc $(p,t_A)$, and an arc $(t_A,q_{r'})$ where $q_{r'}\in Q\cap{}^\bullet t'$. If the rule $r$: $A\rightarrow\alpha$ is not applicable in a derivation, we can skip this rule, and we can apply the next rule $r'$: $A'\rightarrow\alpha'$ in the matrix since $r\in B$. Correspondingly, if $r$ is not applicable, place $p$ does not have any token, and transition $t_A$ can fire, correspondingly, the rule $r'$: $A'\rightarrow\alpha'$ can be applied. This modification of $N_s$ allows us to take into consideration of the checking of occurrences of nonterminals in sentential forms. Let $N'_s$ be the Petri net obtained from $N_s$ by the modification above. Then $G''' = (V, \Sigma, S, R, N'_s)$ is a PNI controlled grammar where $L(G) = L(G''')$.

We show that the inverse inclusion also holds for the λ-case.

**Lemma 3.4. PNI$^\lambda \subseteq$ MAT$^\lambda_{ac}$.**

***Proof***. Let $G = (V, \Sigma, S, R, N, \gamma, M')$ be a PNI controlled grammar with $N = (P, T, F, I, \phi, \iota)$. Let $P = \{p_1, p_2, \ldots, p_n\}$. We set $V' = V\cup P'\cup\{S', C, X\}$ where $P' = \{p' \mid p\in P\}$ is a set of new nonterminal symbols and $S'$, $C$, $X$ are new nonterminal symbols. Let for $t\in T$,

${}^\bullet t = \{p_{i,1}, \ldots, p_{i,k}\}$, ${}^\circ t = \{p_{i,k+1}, \ldots, p_{i,l}\}$ and $t^\bullet = \{p_{j,1}, \ldots, p_{j,m}\}$.

We associate the following sequences of rules with each transition $t\in T$

- $\sigma_{i,1}$: $p'_{i,1}\rightarrow\lambda, p'_{i,1}\rightarrow\lambda, \ldots, p'_{i,1}\rightarrow\lambda$
  (the number of the erasing rules is $\phi(p_{i,1},t)$), (3.1)
- $\sigma_{i,2}$: $p'_{i,2}\rightarrow\lambda, p'_{i,2}\rightarrow\lambda, \ldots, p'_{i,2}\rightarrow\lambda$
  (the number of the erasing rules is $\phi(p_{i,2},t)$), (3.2)
- $\sigma_{i,k}$: $p'_{i,k}\rightarrow\lambda, p'_{i,k}\rightarrow\lambda, \ldots, p'_{i,k}\rightarrow\lambda$
  (the number of the erasing rules is $\phi(p_{i,k},t)$), (3.3)
- $\sigma_X$: $p'_{i,k+1}\rightarrow X, p'_{i,k+2}\rightarrow X, \ldots, p'_{i,l}\rightarrow X$ (3.4)
- $\sigma_C$: $C\rightarrow C\, p'^{\phi(t,p_{j,1})}_{j,1}\, p'^{\phi(t,p_{j,2})}_{j,2} \cdots p'^{\phi(t,p_{j,m})}_{j,m}$ (3.5)

and define the matrix

- $m_r = (\sigma_{i,1}, \sigma_{i,2}, \ldots, \sigma_{i,k}, \sigma_C, \sigma_X, r)$ (3.6)

where $r = A\rightarrow\alpha = \gamma(t)\in R$. Furthermore, we add the starting matrix

- $m_0 = (S' \rightarrow SC\prod_{p\in P} p'^{|\iota(p)|})$ (3.7)

and for each $\tau\in M'$, the “erasing” matrix

- $m_{r,\lambda} = (C\rightarrow\lambda, p'_1\rightarrow\lambda, \ldots, p'_1\rightarrow\lambda, p'_2\rightarrow\lambda, \ldots, p'_2\rightarrow\lambda, \ldots, p'_n\rightarrow\lambda, \ldots, p'_n\rightarrow\lambda)$ (3.8)

where $m_{r,\lambda}$ has $\tau(p_i)$ occurrences of each rule $p'_i\rightarrow\lambda$, $1\leq i\leq n$. We consider the matrix grammar $G' = (V', \Sigma, S', M, B)$ where $M$ consists of all matrices of the form (3.6)-(3.8) and $B$ consists of the rules of the form $p'_{i,k+1}\rightarrow X$, $p'_{i,k+2}\rightarrow X$, $\ldots$, $p'_{i,l}\rightarrow X$.

Let $D$: $S[r_1r_2\cdots r_n\rangle w\in\Sigma^*$ be a derivation in $G$. Then $\nu = t_1t_2\cdots t_s$, $\gamma(\nu) = r_1r_2\cdots r_n$ is an occurrence sequence of transitions of $N$ enabled at the initial marking $\iota$. We construct the derivation $D'$ in $G'$ which simulates the derivation $D$. The derivation $D'$ starts with

$$S[\rangle SC\prod_{p\in P} p'^{|\iota(p)|}$$

applying the matrix (3.7), then for each pair of a transition $t$ in $\nu$ and the corresponding rule $r=\gamma(t)$, we choose a matrix of the form (3.6). When the terminal string $w\in\Sigma^*$ is generated, in order to erase the remaining symbols from $P'$ and the symbol $C$ we use matrices of the form (3.8).

Let,

$$D': S'[m_0\rangle SC\prod_{p\in P} p'^{|\iota(p)|}[m_{i,1}m_{i,2}\cdots m_{i,n}\rangle w_n = w\in\Sigma^*$$

be a derivation in $G'$. Since $V\cap P'=\emptyset$, we can write a derivation $D''$: $S[r_{j,1}\, r_{j,2}\cdots r_{j,k}\rangle w_{j,1} = w\in\Sigma^*$ where $r_{j,i}$ is the rule of the non-erasing matrix $m_{r,j,i}$, $1\leq i\leq k$ in $D'$ and we omit those steps in $D'$ in which erasing matrices are used.

The application of a matrix $m_r$ of the form (3.6) in $D'$ shows that there are at least $\varphi(p_{i,1},t)$ pieces of $p'_{i,1}$, etc., and at least $\varphi(p_{i,k},t)$ pieces of $p'_{i,k}$ in the sentential form, i.e. the input places $p_{i,1}$, $p_{i,2}$, $\ldots$, $p_{i,k}$ of $t$ have at least $\varphi(p_{i,1},t)$, $\varphi(p_{i,2},t)$, $\ldots$, $\varphi(p_{i,k},t)$ tokens, respectively. Moreover, the rules of the form $p'_{i,k+1}\rightarrow X$, $p'_{i,k+2}\rightarrow X$, $\ldots$, $p'_{i,l}\rightarrow X$ control that places from ${}^\circ t$ do not have any tokens. Thus, the transition $t$, $\gamma(t) = r$, is enabled in $N$. We can construct the successful occurrence sequence $\iota[t_{j,1}t_{j,2}\cdots t_{j,k}\rangle\mu_k$ where $\gamma(\, t_{j,i}) = r_{j,i}$, $1\leq i\leq k$. Hence, $D''$ is a derivation in $G$. Thus $L(G')\subseteq L(G)$.

Now let $E$: $S[r_{j,1}r_{j,2}\cdots r_{j,k}\rangle w_{j,k} = w\in\Sigma^*$ be a derivation in $G$. Then we also have the derivation $E'$: $S'[m_0\rangle SC[m_{j,1}m_{j,2}\cdots m_{j,k}\rangle w'_{j,k}C$ in $G'$ where $w'_{j,k}$ differs from $w_{j,k}$ only in letters $p'$, $p\in P$. These letters and $C$ can be erased with matrices (3.8). Thus $L(G) \subseteq L(G')$.

We summarize our results in the following theorem

**Theorem 3.5.** $\mathbf{MAT}^{\lambda}_{ac} \subseteq \mathbf{PNI} \subseteq \mathbf{MAT}^{\lambda}_{ac} = \mathbf{PNI}^{\lambda}$.

## 4. CONCLUSION

In this paper we have defined and investigated grammars controlled by Petri nets with inhibitor arcs, where we have considered only arbitrary transition labeling and the finite set of final markings. The paper clearly establishes some basic facts. A topic is worth being investigated in this area is the consideration of different types of transition labeling strategies and the different definitions of final marking sets. Another interesting topic is the study of the hierarchy problems with respect to the number of inhibitor arcs.

## ACKNOWLEDGEMENT

We acknowledge financial support from University Putra Malaysia via RUGS grant number 05-01-10-0896RU.

## REFERENCES

[1] J. Dassow and S. Turaev, LNCS 5196 (2008) 209-220.

[2] J. Dassow and S. Turaev, LNCS 5457 (2009) 326-337.

[3] J. Dassow and S. Turaev, J. of Universal Computer Science, 15(14) (2009) 2808-2835.

[4] J. Dassow and S. Turaev, Romanian J. of Information Science and Technology, 12(2) (2009) 191–207.

[5] J. Dassow and Gh. Paun, Springer, Berlin, 1989.

[6] G. Rozenberg and A. Salomaa, Springer, 1997.

[7] M.H. Selamat and S. Turaev, ICCRD 2010 (2010) 51-55.

[8] R. Stiebe and S. Turaev, EPTCS, 3 (2009) 193-203.

[9] S. Turaev, MEMICS 2007 (2007) 233-240.

[10] S. Turaev, PhD Thesis, 2010.

[11] G. Zetzsche, DLT 2009, (2009) 490-501.