ROMANIAN JOURNAL OF INFORMATION SCIENCE AND TECHNOLOGY
Volume **12**, Number 2, 2009, 191–207

Petri Net Controlled Grammars: the Power of Labeling and Final Markings*

Jürgen DASSOW[1], Sherzod TURAEV[2]

[1] Facultät für Informatik, Otto-von-Guericke-Universität Magdeburg
PSF 4120, D-39016 Magdeburg, Germany
E-mail: dassow@iws.cs.uni-magdeburg.de

[2] GRLMC, Universitat Rovira i Virgili
Pl. Imperial Tàrraco 1, 43005 Tarragona, Spain
E-mail: sherzod.turaev@urv.cat

Abstract. Essentially, a Petri net controlled grammar is a context-free grammar equipped with a Petri net and a function which maps transitions of the net to rules of the grammar. The language consists of all terminal words that can be obtained by applying of a sequence of productions which is the image of an occurrence sequence of the Petri net under the function. We study the generative power of such grammars on the type of the function which can be a bijection or a coding or a weak coding and with respect to three types of admitted sets of occurrence sequences. We show that the generative power does not depend on the type of the function. Moreover, the restriction to occurrence sequences, which transform the initial marking to a marking in a given finite set of markings, leads to a more powerful class of grammars than the allowance of all occurrence sequences. Furthermore, we present some new characterizations of the family of matrix languages in terms of Petri net controlled grammars.

1. Introduction

It is a well-known fact that context-free grammars are not able to cover all phenomena of natural and programming languages, and also with respect to other application of sequential grammars they cannot describe all aspects. On the other hand,

*This paper is an extended version of the paper presented at the Second International Workshop on Non-classical Formal Languages in Linguistics, Tarragona, Spain, September 19-20, 2008 [5]. In particular, we consider another definition for the set of final markings. In order to give the complete information to the reader, the proofs of some statements of [5] are recalled.

context-sensitive grammars are powerful enough but have bad features with respect to decidability problems which are undecidable or at least very hard. Therefore it is a natural idea to introduce grammars which use context-free rules and have a device which controls the application of the rules. We refer to the monograph [3] for a summary of this approach. The regularly controlled grammars are a well-known class of such grammars; here a finite automaton is associated with a grammar and the sequence of applied rules has to be accepted by the automaton.

In this paper we consider a generalization of regularly controlled grammars. Instead of a finite automaton we associate a Petri net with a context-free grammar and require that the sequence of applied rules corresponds to an occurrence sequence of the Petri net, i.e., to sequences of transitions which can be fired in succession. However, one has to decide what type of correspondence is used and what concept is taken as an equivalent of acceptance. Since the sets of occurrence sequences form the language of a Petri net, we choose the correspondence and the equivalent for acceptance according to the variations which are used in the theory of Petri net languages.

Therefore as correspondence we choose a bijection (between transitions and rules) or a coding (any transition is mapped to a rule) or a weak coding (any transition is mapped to a rule or the empty word) which agree with the classical three variants of Petri net languages (see e.g. [15], [6], [14]).

In the theory of Petri net languages two types of acceptance are considered: only those occurrence sequences belonging to the languages which transform the initial marking into a marking from a given finite set of markings or all occurrence sequences are taken (independent of the obtained marking). If we use only the occurrence sequence leading to a marking in a given finite set of markings we say that the Petri net controlled grammar is of t-type; if we consider all occurrence sequences, then the grammar is of r-type. We add a further type which can be considered as a complement of the t-type. Obviously, if we choose a finite set M of markings and require that the marking obtained after the application of the occurrence sequence is smaller than at least one marking of M (the order is componentwise), then we can choose another finite set M' of markings and require that the obtained marking belongs to M'. The complementary approach requires that the obtained marking is larger than at least one marking of the given set M. The corresponding class of Petri net controlled grammars is called of g-type.

Therefore we obtain nine classes of Petri net controlled grammars since we have three different types of correspondence and three types of the set of admitted occurrence sequences. In this paper we investigate the generative power of these classes of Petri net controlled languages. Thus the paper is an extension of [5] where not all nine classes have been considered.

We mention that the paper is a continuation of the papers [2], [16] and [4], too, where instead of arbitrary Petri nets only such Petri nets have been considered where the places and transitions correspond in a one-to-one manner to nonterminals and rules, respectively.

By the above remarks Petri net controlled grammars are motivated by reasons inside the theory of formal grammars describing natural and programming languages. However, there is also a motivation which comes from the modeling of automated manufacturing systems, metabolic pathways and related structures. An automated manufacturing systems is generally a set of *activities* which interact with a set of *resources* and result in a product. For modeling such systems Petri nets are very often used (see e.g. [9] and [13]). However, Petri nets are only good tools for the description of the communication among the components, of the process control and of the behavioral properties of the systems; but they do not fit very well for the manufacturing itself. Therefore a better model can be obtained by Petri net controlled grammars. The grammar is able to cover the generating processes and these processes are controlled by the Petri net. In this context the types of acceptance correspond to no requirement, an upper and an lower bound for the size of some resources.

An analogous situation holds for the modeling of metabolic pathways (see e.g. [7] and [1]).

The paper is organized as follows. In Section 2 we recall some concepts from the theories of formal languages and Petri nets. In Section 3 we introduce our concept of control of derivations in context-free grammars by Petri nets. Section 4 contains the results on the influence of the labeling function on the generative power. In Section 5 we discuss the effect of different types of final markings on the generative power.

2. Preliminaries

We assume that the reader is familiar with the basic concepts of formal language theory and Petri nets. In this section we only recall some notions and notation. For details we refer to [8], [12], [15] and [11].

2.1. Grammars

Let Σ be an alphabet. A *string* over Σ is a sequence of symbols from the alphabet. The *empty* string is denoted by λ which is of length 0. The set of all strings over the alphabet Σ is denoted by Σ^*. A subset L of Σ^* is called a *language*. If $w = w_1w_2w_3$ 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 $|w|_a$.

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*. A string $x \in (V \cup \Sigma)^+$ *directly derives* a string $y \in (V \cup \Sigma)^*$, written as $x \Rightarrow y$, iff there is a rule $r = A \to \alpha \in R$ such that $x = x_1Ax_2$ 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 $\pi = r_1r_2 \cdots r_n$ is denoted by $\stackrel{\pi}{\Rightarrow}$ or $\xrightarrow{r_1r_2\cdots r_n}$. The *language* generated by G is defined by $L(G) = \{w \in \Sigma^* \mid S \Rightarrow^* w\}$.

A *matrix grammar* is a quadruple $G = (V, \Sigma, S, M)$ where V, Σ, S are defined as for a context-free grammar, M is a finite set of *matrices* which are finite strings over a set of context-free rules (or finite sequences of context-free rules). The language generated by G is $L(G) = \{w \in \Sigma^* \mid S \stackrel{\pi}{\Rightarrow} w \text{ and } \pi \in M^*\}$.

A matrix grammar G is called *without repetitions*, if each rule r of G occurs in $M = \{m_1, m_2, \ldots, m_n\}$ exactly once, i.e., $|m_1 m_2 \cdots m_n|_r = 1$.

For each matrix grammar, by adding chain rules, one can construct an equivalent matrix grammar without repetitions.

The families of languages generated by matrix grammars without erasing rules and by matrix grammars with erasing rules are denoted by **MAT** and $\mathbf{MAT}^\lambda$, respectively.

2.2. Petri Nets

A (place/transition) *Petri net* (PN) is a construct $N = (P, T, F, \varphi)$ where P and T are disjoint finite sets of *places* and *transitions*, respectively, $F \subseteq (P \times T) \cup (T \times P)$ is a set of *directed arcs*, $\varphi : (P \times T) \cup (T \times P) \to \{0, 1, 2, \dots\}$ is a *weight function*, where $\varphi(x, y) = 0$ for all $(x, y) \in ((P \times T) \cup (T \times P)) - F$.

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* with labels $\varphi(p, t)$ or $\varphi(t, p)$. If $\varphi(p, t) = 1$ or $\varphi(t, p) = 1$, 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 the *pre-* and *post-sets* of $x \in P \cup T$, respectively. The elements of ${}^\bullet t$ (${}^\bullet p$) are called the *input* places (transitions) and the elements of $t^\bullet$ ($p^\bullet$) are called the *output* places (transitions) of t (p).

A transition $t \in T$ is *enabled* by a marking μ iff $\mu(p) \geq \varphi(p, t)$ for all $p \in P$. In this case t can *occur* (*fire*). Its occurrence transforms the marking μ into the marking μ', written as $\mu \xrightarrow{t} \mu'$, defined for each place $p \in P$ by $\mu'(p) = \mu(p) - \varphi(p, t) + \varphi(t, p)$. A finite sequence $t_1 t_2 \cdots t_k$ of transitions is called *an occurrence sequence* enabled at a marking μ if there are markings $\mu_1, \mu_2, \ldots, \mu_k$ such that $\mu \xrightarrow{t_1} \mu_1 \xrightarrow{t_2} \ldots \xrightarrow{t_k} \mu_k$. In short this sequence can be written as $\mu \xrightarrow{t_1 t_2 \cdots t_k} \mu_k$ or $\mu \xrightarrow{\nu} \mu_k$ where $\nu = t_1 t_2 \cdots t_k$. For each $1 \leq i \leq k$, the marking μ_i is called *reachable* from the marking μ. The set of all reachable markings of a Petri net N from a marking μ is denoted by $\mathcal{R}(N, \mu)$.

A *marked* Petri net is a system $N = (P, T, F, \varphi, \iota)$ where (P, T, F, φ) is a Petri net, ι is the *initial marking.* Let M be a set of markings, which will be called *final* markings. An occurrence sequence ν of transitions is called *successful for* M if it is enabled at the initial marking ι and finished at a final marking $\tau \in M$. If M is understood from the context, we say that ν is a *successful occurrence sequence.*

3. Petri Net Controlled Grammars and their Languages

We now introduce our concept of control.

Definition 1. A *Petri net controlled grammar* is a tuple $G = (V, \Sigma, S, R, N, \gamma, M)$ where V, Σ, 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 labeling function and M is a set of final markings.

Definition 2. The *language* generated by a Petri net controlled grammar G, denoted by $L(G)$, consists of all strings $w \in \Sigma^*$ such that there is a derivation $S \xRightarrow{r_1 r_2 \cdots r_k} w \in \Sigma^*$ and an occurrence sequence $\nu = t_1 t_2 \cdots t_s$ which is successful for M such that $r_1 r_2 \cdots r_k = \gamma(t_1 t_2 \cdots t_s)$.

Definition 2 uses the extended form of the labeling function $\gamma : T^* \to R^*$; this extension is done in the usual manner.

Obviously, if γ maps any transition to a rule, then $k = s$ in Definition 2.

Example 1. Let $G_1 = (\{S, A, B, C\}, \{a, b, c\}, S, R, N_1, \gamma_1, M_1)$ be a Petri net controlled grammar where $R = \{S \to ABC,\ A \to aA,\ B \to bB,\ C \to cC,\ A \to a,\ B \to b,\ C \to c\}$ and N_1 is illustrated in Fig. **??**. If M_1 is the set of all reachable markings, then G_1 generates the language

$$L(G_1) = \{a^n b^m c^k \mid n \geq m \geq k \geq 1\}.$$

If $M_1 = \{\mu\}$ with $\mu(p) = 0$ for all $p \in P$, then it generates the language

$$L(G_1) = \{a^n b^n c^n \mid n \geq 1\}.$$

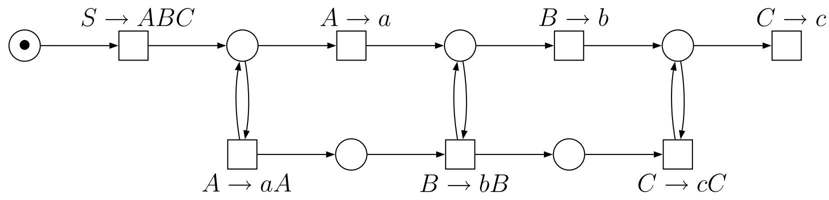

Fig. 1. A Petri net N_1.

Different labeling strategies and different definitions of the set of final markings result in various types of Petri net controlled grammars. In this paper we consider the following types of Petri net controlled grammars.

Definition 3. A Petri net controlled grammar $G = (V, \Sigma, S, R, N, \gamma, M)$ is called

- *free* (abbreviated by f) if a different label is associated to each transition, and no transition is labeled with the empty string;
- λ-*free* (abbreviated by $-\lambda$) if no transition is labeled with the empty string;
- *extended* (abbreviated by λ) if no restriction is posed on the labeling function γ.

Definition 4. A Petri net controlled grammar $G = (V, \Sigma, S, R, N, \gamma, M)$ 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;
- *g-type* if for a given finite set $M_0 \subseteq \mathcal{R}(N, \iota)$, M is the set of all markings such that for every marking $\mu \in M$ there is a marking $\mu' \in M_0$ such that $\mu \geq \mu'$.

We use the notation (x, y)-*PN controlled grammar* where $x \in \{f, -\lambda, \lambda\}$ shows the type of a labeling function and $y \in \{r, t, g\}$ shows the type of a set of final markings.

We denote by $\mathbf{PN}(x, y)$ and $\mathbf{PN}^\lambda(x, y)$ the families of languages generated by (x, y)-PN controlled grammars without and with erasing rules, respectively, where $x \in \{f, -\lambda, \lambda\}$ and $y \in \{r, t, g\}$.

We also use bracket notation $\mathbf{PN}^{[\lambda]}(x, y)$, $x \in \{f, -\lambda, \lambda\}$, $y \in \{r, t, g\}$, in order to say that a statement holds both in case with erasing rules and in case without erasing rules.

The following inclusions are obvious.

Lemma 1. *For $x \in \{f, -\lambda, \lambda\}$ and $y \in \{r, t, g\}$,* $\mathbf{PN}(x, y) \subseteq \mathbf{PN}^\lambda(x, y)$.

4. The Effect of Labeling on the Generative Power

The following lemma follows immediately from the definition of the labeling functions.

Lemma 2. *For $y \in \{r, t, g\}$,* $\mathbf{PN}^{[\lambda]}(f, y) \subseteq \mathbf{PN}^{[\lambda]}(-\lambda, y) \subseteq \mathbf{PN}^{[\lambda]}(\lambda, y)$. □

We now prove that the reverse inclusions also hold.

Lemma 3. *For $y \in \{r, t, g\}$,* $\mathbf{PN}^{[\lambda]}(-\lambda, y) \subseteq \mathbf{PN}^{[\lambda]}(f, y)$.

Proof. Let $G = (V, \Sigma, S, R, N, \gamma, M)$ be a $(-\lambda, y)$-Petri net controlled grammar (with or without erasing rules) where $y \in \{r, t, g\}$ and $N = (P, T, F, \varphi, \iota)$. Let

$$\begin{aligned} R^{>1} &= \{r : A \to \alpha \in R \mid |\gamma^{-1}(r)| > 1\}, \\ T^{>1} &= \{t \in T \mid \gamma(t) = r, r \in R^{>1}\}, \\ F^{>1} &= \{(p, t) \in F \mid t \in T^{>1}\} \cup \{(t, p) \in F \mid t \in T^{>1}\}. \end{aligned}$$

For each rule $r : A \to \alpha \in R^{>1}$, we define the set $V_r = \{A_t \mid \gamma(t) = r\}$ of new nonterminal symbols, and with the rule r, we associate the set

$$R_r = \{A \to A_t, A_t \to \alpha \mid r : A \to \alpha \in R^{>1} \text{ and } \gamma(t) = r\}$$

of new rules. Correspondingly, we set

$$T_r = \{c_t^1, c_t^2 \mid r : A \to \alpha \in R^{>1} \text{ and } \gamma(t) = r\}$$

where c_t^1 and c_t^2 are new transitions labeled by the rules $A \to A_t$ and $A_t \to \alpha$ for each t with $\gamma(t) = r$, respectively. We define the following sets of new places

$$P_r = \{p_t \mid r : A \to \alpha \in R^{>1} \text{ and } \gamma(t) = r\}$$

and arcs

$$\begin{aligned} F_r = &\{(p, c_t^1) \mid c_t^1 \in T_r \text{ and } p \in {}^\bullet t\} \cup \{c_t^2, p) \mid c_t^2 \in T_r \text{ and } p \in t^\bullet\} \\ &\cup \{(c_t^1, p_t), (p_t, c_t^2) \mid c_t^1, c_t^2 \in T_r \text{ and } p_t \in P_r\}. \end{aligned}$$

Let $X^\diamond = \bigcup_{r\in R^{>1}} X_r$ where $X \in \{V, R, P, T, F\}$. We consider an (f, y)-Petri net controlled grammar $G' = (V', \Sigma, S, R', N', \gamma', M')$ where $V' = V \cup V^\diamond$ and $R' = (R - R^{>1}) \cup R^\diamond$ and $N' = (P', T', F', \varphi', \iota')$ is a Petri net where the set of places, transitions and arcs are defined by

$$P' = P \cup P^\diamond,\ T' = (T - T^{>1}) \cup T^\diamond,\ F' = (F - F^{>1}) \cup F^\diamond;$$

the weight function φ' is defined by

$$\varphi'(x, y) = \begin{cases} \varphi(x, y) & \text{if } (x, y) \in F, \\ \varphi(p, t) & \text{if } x = p \in {}^\bullet t \text{ and } y = c_t^1,\ t \in T^{>1}, \\ \varphi(t, p) & \text{if } x = c_t^2 \text{ and } p \in t^\bullet,\ t \in T^{>1}, \\ 1 & \text{otherwise;} \end{cases}$$

the initial marking ι' is defined by

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

the bijection γ' is defined by $\gamma'(t) = \gamma(t)$ if $t \in T - T_\lambda$ and for all $c_t^1, c_t^2 \in T_r$, $r \in R^{>1}$, $\gamma'(c_t^1) = A \to A_t$ and $\gamma'(c_t^2) = A_t \to \alpha$;

for each $\tau' \in M'$,

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

Let $S \xRightarrow{r_1\cdots r_j} w_j \xRightarrow{r} w_j' \xRightarrow{r_{j+1}\cdots r_k} w_k \in \Sigma^*$ be a derivation in G where $r : A \to \alpha \in R^{>1}$. Then the rule $r : A \to \alpha$ can be replaced by the pair $A \to A_t$, $A_t \to \alpha$ for some $t \in T^{>1}$ in one-to-one correspondence with the transition t of N where $\gamma(t) = r$, by the transitions c_t^1 and c_t^2 of N', and vice versa. Hence $L(G) = L(G')$. □

Lemma 4. *For* $y \in \{r, t, g\}$*,* $\mathbf{PN}(\lambda, y) \subseteq \mathbf{PN}(-\lambda, y)$*.*

Proof. Let $G = (V, \Sigma, S, R, N, \gamma, M)$ be a (λ, y)-Petri net controlled grammar with $N = (P, T, F, \iota)$. Let $T_\lambda = \{t \in T \mid \gamma(t) = \lambda\}$ and

$$F_\lambda = \{(p, t) \mid p \in P \text{ and } t \in T_\lambda\} \cup \{(t, p) \mid t \in T_\lambda \text{ and } p \in P\}.$$

We define the *i-adjacency set* of $t \in T$ by

$$\mathrm{Adj}^i(t) = \{t'' \mid t'' \in (\mathrm{Adj}^1(t')) \text{ for some } t' \in \mathrm{Adj}^{i-1}(t) \cap T_\lambda\} \text{ for } i \geq 2$$

where $\mathrm{Adj}^1(t) = (t^\bullet)^\bullet$ and the *complete adjacency* set by

$$\mathrm{Adj}^*(t) = \bigcup_{i\geq 1} \mathrm{Adj}^i(t).$$

A transition $t' \in \mathrm{Adj}^*(t)$ is called an *adjacent transition* of t. $\mathrm{Adj}^+(t)$ denotes the set of non λ adjacent transitions of $t \in T$, i.e., $\mathrm{Adj}^+(t) = \mathrm{Adj}^*(t) - T_\lambda$.

Let $T_\lambda = \{t_1, t_2, \ldots, t_n\}$. For each $t_i \in T_\lambda$, $1 \leq i \leq n$, we define the set of new transitions $T(t_i) = \{[t]_i \mid t \in \mathrm{Adj}^+(t_i)\}$. We introduce the set $R(t_i)$ of new rules with respect to each $t_i \in T_\lambda$, $1 \leq i \leq n$,

$$R(t_i) = \{A \to A \mid A \to \alpha = \gamma(t) \in R \text{ and } t \in \mathrm{Adj}^+(t_i)\}.$$

We define a $(-\lambda, y)$-PN controlled grammar $G' = (V, \Sigma, S, R', N', \gamma', M')$ where $R' = R \cup \bigcup_{t_i \in T_\lambda} R(t_i)$ and $N' = (P, T', F', \iota)$ where

$$\begin{aligned} T' &= (T - T_\lambda) \cup \bigcup_{t_i \in T_\lambda} T(t_i), \\ F' &= (F - F_\lambda) \cup \bigcup_{t_i \in T_\lambda} \{(p, [t]_i) \mid p \in {}^\bullet t_i \text{ and } [t]_i \in T(t_i)\} \\ &\quad \cup \bigcup_{t_i \in T_\lambda} \{([t]_i, p) \mid [t]_i \in T(t_i) \text{ and } p \in t_i^\bullet\}. \end{aligned}$$

The weight function φ' is defined by

- $\varphi'(x, y) = \varphi(x, y)$ if $(x, y) \in F - F_\lambda$,
- $\varphi'(p, [t]_i) = \varphi(p, t_i)$ if $p \in {}^\bullet t_i$ and $[t]_i \in T(t_i)$, $t_i \in T_\lambda$,
- $\varphi'([t]_i, p) = \varphi(t_i, p)$ if $p \in t_i^\bullet$ and $[t]_i \in T(t_i)$, $t_i \in T_\lambda$.

The labeling function $\gamma' : T' \to R'$ is defined by

- $\gamma'(t) = \gamma(t)$ for all $t \in T$,
- $\gamma'([t]_i) = A \to A \in R(t_i)$ where $[t]_i \in T(t_i)$, $t_i \in T_\lambda$ and $t \in \mathrm{Adj}^+(t_i)$ with $\gamma(t) = A \to \alpha \in R$.

Let $S \xRightarrow{r_1 r_2 \cdots r_n} w_n \in \Sigma^*$ be a derivation in G. Then

$$t'_{11} \cdots t'_{1k(1)} t_1 t'_{21} \cdots t'_{2k(2)} t_2 \cdots t_n t'_{n+11} \cdots t'_{n+1k(n+1)} \tag{1}$$

is a successful occurrence sequence in N where $\gamma(t_i) = r_i$, $1 \leq i \leq n$ and $t'_{ij} \in T_\lambda$ for all $1 \leq i \leq n+1$, $1 \leq j \leq k(i)$ such that $t_i \in \mathrm{Adj}^+(t'_{ij})$ for all $1 \leq i \leq n$, $1 \leq j \leq k(i)$.

Each λ-transition t'_{ij}, $1 \leq i \leq n$, $1 \leq j \leq k(i)$ in (1) can be replaced by the transition t''_{ij} in N', $1 \leq i \leq n$, $1 \leq j \leq k(i)$ with the label $A_i \to A_i$ where A_i is the left side of the rule r_i, $\gamma(r_i) = t_i$, $1 \leq i \leq n$. Then

$$t''_{11} \cdots t''_{1k(1)} t_1 t''_{21} \cdots t''_{2k(2)} t_2 \cdots t''_{n1} \cdots t''_{nk(n)} t_n \tag{2}$$

is a successful occurrence sequence in N' and correspondingly

$$S \xRightarrow{\sigma_1 r_1 \sigma_2 r_2 \cdots \sigma_n r_n} w_n \in \Sigma^*$$

is a derivation in G' where $\sigma_i = r''_{i1} r''_{i2} \cdots r''_{ik(i)}$, $\gamma'(r''_{ij}) = t''_{ij}$, $1 \leq i \leq n$, $1 \leq j \leq k(i)$. Using the same idea, we can show the inverse inclusion. □

It is easy to see that the proof of Lemma 4 holds for grammars with erasing rules, too. We present another proof in the following lemma since its construction has a smaller increase of the number of places, transitions and edges.

Lemma 5. *For* $y \in \{r, t, g\}$, $\mathbf{PN}^{\lambda}(\lambda, y) \subseteq \mathbf{PN}^{\lambda}(-\lambda, y)$.

Proof. Let $G = (V, \Sigma, S, R, N, \gamma, M)$ be a (λ, y)-PN controlled grammar where $y \in \{r, t, g\}$ and $N = (P, T, F, \varphi, \iota)$. Let T_λ be the set of all λ-transitions of T. We construct the $(-\lambda, y)$-PN controlled grammar $G' = (V', \Sigma, S', R', N', \gamma', M')$ as follows.

We set $V' = V \cup \{S', X\}$ where S' and X are new symbols and

$$R' = R \cup \{S' \to SX, X \to X, X \to \lambda\}$$

and construct $N' = (P', T', F', \varphi', \iota')$ where

- the sets of places, transitions and arcs of N' are defined by

$$\begin{aligned} P' &= P \cup \{p', p''\}, \\ T' &= T \cup \{t', t''\}, \\ F' &= F \cup \{(p', t'), (t', p''), (p'', t'')\}, \end{aligned}$$

- the weight function is defined by

$$\varphi'(x, y) = \begin{cases} \varphi(x, y) & \text{if } (x, y) \in F, \\ 1 & \text{otherwise,} \end{cases}$$

- the initial marking is defined by $\iota'(p) = \iota(p)$ for all $p \in P$ and $\iota'(p') = 1, \iota'(p'') = 0$,
- for every $\tau' \in M'$, $\tau'(p) = \tau(p)$ for all $p \in P$ and $\tau'(p') = \tau'(p'') = 0$,
- and the total function $\gamma' : T' \to R'$ is defined by $\gamma'(t) = \gamma(t)$ if $t \in T - T_\lambda$, $\gamma'(t) = X \to X$ if $t \in T_\lambda$, $\gamma'(t') = S' \to SX$, and $\gamma'(t'') = X \to \lambda$.

Let $D : S \xRightarrow{r_1 r_2 \cdots r_k} w_k \in \Sigma^*$ be a derivation in G where $\nu = \nu_1 t_1 \nu_2 t_2 \cdots \nu_k t_k \nu_{k+1}$, $\gamma(t_i) = r_i$ for all for all $1 \leq i \leq k$ and $\gamma(\nu_i) = \lambda$ for all $1 \leq i \leq k+1$ is an occurrence sequence in N enabled at the initial marking ι and finishing at a marking $\mu_k \in M$.

We construct a derivation D' in G' from the derivation D as follows. We initialize the derivation D with the rule $S' \to SX$. For any λ-transition t in the occurrence sequence ν we apply the rule $X \to X$ and terminate the derivation with the rule $X \to \lambda$:

$$S' \Rightarrow SX \xRightarrow{\overbrace{X \to X}^{|\nu_1|} \cdot r_1} w_1 X \xRightarrow{\overbrace{X \to X}^{|\nu_2|} \cdot r_2} \cdots \xRightarrow{\overbrace{X \to X}^{|\nu_k|} \cdot r_k} w_k X \xRightarrow{X \to \lambda} w_k \in \Sigma^*$$

and $t' \nu_1 t_1 \nu_2 t_2 \cdots \nu_k t_k \nu_{k+1} t''$ is a successful occurrence sequence in N' where $\mu(p') = \mu(p'') = 0$ for any $\mu \in M'$.

On the other hand, for each derivation

$$S' \Rightarrow SX \xrightarrow{r_1 \cdots r_j} w_j X \xrightarrow{X \to \lambda} w_j \xrightarrow{r_{j+1} \cdots r_m} w_m \in \Sigma^*$$

in G' by removing the first step, $(j+1)$-th step and the nonterminal symbol X from the derivation, we get a derivation in G where the corresponding occurrence in N' sequence is obtained by removing the transitions t', t'' and changing the labels $X \to X$ of transitions to λ. □

The following theorem is a combination of the lemmas given above.

Theorem 6. *For* $y \in \{r, t, g\}$,

$$\mathbf{PN}(f, y) = \mathbf{PN}(-\lambda, y) = \mathbf{PN}(\lambda, y) \subseteq \mathbf{PN}^\lambda(f, y) = \mathbf{PN}^\lambda(-\lambda, y) = \mathbf{PN}^\lambda(\lambda, y).$$

5. The Effect of Final Markings on the Generative Power

We start with a lemma which shows that the use of final markings increases the generative power.

Lemma 7. $\mathbf{PN}^{[\lambda]}(\lambda, r) \subseteq \mathbf{PN}^{[\lambda]}(\lambda, t)$.

Proof. Let $G = (V, \Sigma, S, R, N, \gamma, M)$ be a (λ, r)-PN controlled grammar (with or without erasing rules) where $N = (P, T, F, \varphi, \iota)$. We set

$$T_p = \{t_p \mid p \in P\} \text{ and } F_p = \{(p, t_p) \mid p \in P\}$$

where t_p and (p, t_p) for all $p \in P$ are new transitions and arcs, respectively. We construct a (λ, t)-PN controlled grammar $G' = (V, \Sigma, S, R, N', \gamma', M_0)$ with the Petri net $N' = (P, T \cup T_p, F \cup F_p, \varphi', \iota)$ where

- the weight function φ' is defined by $\varphi'(x, y) = \varphi(x, y)$ if $(x, y) \in F$ and $\varphi'(x, y) = 1$ if $(x, y) \in F_p$,
- the labeling function γ' is defined by $\gamma'(t) = \gamma(t)$ if $t \in T$ and $\gamma'(t) = \lambda$ if $t \in T_p$,
- the set M_0 of final markings is defined by $M_0 = \{(0, 0, \ldots, 0)\}$.

Let $S \xrightarrow{r_1 r_2 \cdots r_k} w_k \in \Sigma^*$ be a derivation in G where $\nu = t_1 t_2 \cdots t_s$, $\gamma(\nu) = r_1 r_2 \cdots r_k$, is an occurrence sequence in N enabled at ι and finished at some $\mu_s \in M$. We continue the occurrence sequence ν by firing the transition t_p $\mu_s(p)$ times, for each place $p \in P$, and after $\sum_{p \in P} \mu_s(p)$ steps we get the marking μ' where $\mu'(p) = 0$ for all $p \in P$. Thus $L(G) \subseteq L(G')$.

Moreover, it is easy to see that an earlier use of a transition t_p either leads to a blocking of the derivation (since an input place p of a transition t has not enough tokens and therefore, the corresponding rule $\gamma(t)$ cannot be applied) or it has no influence on the derivation, i.e., the use of t_p can be shifted after the finishing of the derivation. Therefore $L(G) = L(G')$ holds. □

Corollary 8. $\mathbf{PN}^{[\lambda]}(\lambda, r) \subseteq \mathbf{PN}^{[\lambda]}(\lambda, g)$.

Proof. Let $G = (V, \Sigma, S, R, N, \gamma, M)$ be a (λ, r)-PN controlled grammar (with or without erasing rules) where $N = (P, T, F, \varphi, \iota)$. We construct a (λ, g)-PN controlled grammar $G'' = (V, \Sigma, S, R, N', \gamma', M')$ where V, Σ, S, R, N' and γ' are defined as for the grammar G' in the proof of Lemma 7. If we define M' as the set of any marking $\mu \in \mathcal{R}(N', \iota)$ which is greater than or equal to $\mu' = (0, 0, \ldots, 0)$, then the inclusion follows immediately. □

Lemma 9. $\mathbf{PN}^{[\lambda]}(\lambda, g) \subseteq \mathbf{PN}^{[\lambda]}(\lambda, t)$.

Proof. Let $G = (V, \Sigma, S, R, N, \gamma, M)$ be a (λ, g)-PN controlled grammar (with or without erasing rules) where $N = (P, T, F, \varphi, \iota)$ and M is the set of all markings such that for every marking $\mu \in M$ there is a marking μ' of a given finite set $M_0 \subseteq \mathcal{R}(N, \iota)$ such that $\mu \geq \mu'$. Let p_0 be a new place. We define

- the sets $T_{M_0} = \{t_\mu \mid \mu \in M_0\}$ and $T_P = \{t_p \mid p \in P\}$ of new transitions;
- the sets

$$F_{M_0}^- = \{(p, t_\mu) \mid \mu \in M_0 \text{ and } p \in P \text{ where } \mu(p) \neq 0\},$$
$$F_{M_0}^+ = \{(t_\mu, p_0) \mid \mu \in M_0\},$$
$$F_P = \{(p, t_p) \mid p \in P\}$$

  of new arcs.

We construct the Petri net $N' = (P \cup \{p_0\}, T \cup T_{M_0} \cup T_P, F \cup F_{M_0}^- \cup F_{M_0}^+ \cup F_P, \varphi', \iota')$ where

- the weight function φ' is defined by $\varphi'(x, y) = \varphi(x, y)$ for all $(x, y) \in F$, $\varphi'(p, t_\mu) = \mu(p)$ for each $(p, t_\mu) \in F_{M_0}^-$, and $\varphi'(t_\mu, p_0) = 1$ for each $(t_\mu, p_0) \in F_{M_0}^+$;
- the initial marking ι' is defined by $\iota'(p) = \iota(p)$ for all $p \in P$ and $\iota'(p_0) = 0$.

We define a (λ, t)-PN controlled grammar $G' = (V, \Sigma, S, R, N', \gamma', M')$ where

- $\gamma'(t) = \gamma(t)$ if $t \in T$ and $\gamma'(t) = \lambda$ otherwise;
- $M' = \{\mu'\}$ where $\mu'(p) = 0$ for all $p \in P$ and $\mu'(p_0) = 1$.

Let $D : S \stackrel{\pi}{\Rightarrow} w \in \Sigma^*$, $\pi = r_1 r_2 \cdots r_n$, be a derivation in G, then there is an occurrence sequence $\nu = t_1 t_2 \cdots t_s$ such that $\iota \stackrel{\nu}{\to} \mu$ where $\gamma(\nu) = \pi$ and $\mu \in M$. By definition, there is a marking $\mu' \in M_0$ such that $\mu \geq \mu'$. It follows that the transition t'_μ can occur and the place p_0 receives a token, and the rest tokens in places of P can be removed by firing transitions t_p. It is not difficult to see that D is also a derivation in G'.

If $D' : S \stackrel{\pi}{\Rightarrow} w \in \Sigma^*$, $\pi = r_1 r_2 \cdots r_n$, is a derivation in G' with a successful occurrence sequence $\nu = t_1 t_2 \cdots t_s$ where $\gamma'(\pi) = \nu$, then $\iota' \stackrel{\nu}{\to} \mu'$ where $\mu'(p) = 0$

for all $p \in P$ and $\mu'(p_0) = 1$. Since $\mu'(p_0) = 1$, $|\nu|_{t_\mu} = 1$ for some $\mu \in M_0$. Without loss of generality we can assume that $\nu = \nu' \cdot \nu''$ where ν' contains only transitions of T and ν'' contains only transitions of T_P and the transition t_μ. Then, $\gamma'(\nu') = \pi$, $\gamma'(\nu'') = \lambda$ and $\iota' \xrightarrow{\nu'} \mu''$ where $\mu'' \geq \mu$. It follows that D' is also a derivation in G.□

Lemma 10. $\mathbf{PN}^{[\lambda]}(\lambda, g) \subseteq \mathbf{PN}^{[\lambda]}(\lambda, r)$

Proof. Let $G = (V, \Sigma, S, R, N, \gamma, M)$ be a (λ, g)-Petri net controlled grammar where $N = (P, T, F, \varphi, \iota)$ and M is the set of all markings such that for every marking $\mu \in M$ there is a marking μ' in a given finite set $M_0 \subseteq \mathcal{R}(N, \iota)$ such that $\mu \geq \mu'$.

We set $\bar{\Sigma} = \{\bar{a} \mid a \in \Sigma\}$ where $\bar{a}$, $a \in \Sigma$, is a new nonterminal symbol and define a bijection $\phi : V \cup \Sigma \to V \cup \bar{\Sigma}$ as

$$\phi(x) = \begin{cases} x & \text{if } x \in V, \\ \bar{x} & \text{if } x \in \Sigma \end{cases}$$

Let

$$\bar{R} = \{A \to \phi(\alpha) \mid A \to \alpha \in R\} \text{ and } R_\Sigma = \{\bar{a} \to a \mid a \in \Sigma\}.$$

We define a (λ, r)-PN controlled grammar $G' = (V \cup \bar{\Sigma}, \Sigma, S, \bar{R} \cup R_\Sigma, N', \gamma', M')$ where $N' = (P', T', F', \varphi', \iota')$ and

$$P' = P \cup \{p', p''\},\ T' = T \cup T_{M_0} \cup T_\Sigma,\ F' = F \cup F_{M_0} \cup F_\Sigma \cup F_{p'} \cup F_{p''}$$

where p', p'' are new places,

$$T_{M_0} = \{t_\mu \mid \mu \in M_0\} \text{ and } T_\Sigma = \{t_a \mid a \in \Sigma\}$$

are sets of new transitions,

$$\begin{aligned} F_{M_0} &= \{(p, t_\mu) \mid p \in P \text{ and } t_\mu \in T_{M_0}\}, \\ F_{p'} &= \{(p', t_\mu) \mid t_\mu \in T_{M_0}\}, \\ F_{p''} &= \{(t_\mu, p'') \mid t_\mu \in T_{M_0}\}, \\ F_\Sigma &= \{(p'', t_a), (t_a, p'') \mid t_a \in T_\Sigma\} \end{aligned}$$

are sets of new arcs.

The weight function φ' is defined by $\varphi'(x, y) = \varphi(x, y)$ if $(x, y) \in F$, $\varphi'(p, t_\mu) = \mu(p)$ if $\mu \in M_0$ and $\varphi'(x, y) = 1$ if $(x, y) \in F_{p'} \cup F_{p''} \cup F_\Sigma$.

The initial marking ι' is defined by $\iota'(p) = \iota(p)$ if $p \in P$ and $\iota'(p') = 1$, $\iota'(p'') = 0$.

The bijection γ' is defined by $\gamma'(t) = A \to \phi(\alpha)$ if $t \in T$ and $\gamma(t) = A \to \alpha$, $\gamma'(t) = \lambda$ if $t \in T_{M_0}$, and $\gamma'(t_a) = \bar{a} \to a$ for all $a \in \Sigma$.

For each $\tau' \in M'$, $\tau'(p') = 0$, $\tau'(p'') = 1$.

Let $D : S \xRightarrow{r_1 r_2 \cdots r_n} w \in \Sigma^*$ be a derivation in G and $\nu = t_1 t_2 \cdots t_m$, $\iota \xrightarrow{\nu} \mu_m$, is a successful occurrence sequence of transitions of N where $\gamma(\nu) = r_1 r_2 \cdots r_n$. By definition, $\mu_m \geq \mu$ for some $\mu \in M_0$. Let $w = a_{i_1} a_{i_2} \cdots a_{i_k}$, $a_{i_1}, a_{i_2}, \ldots, a_{i_k} \in \Sigma$, $k \geq 1$.

We construct a derivation D' in G' with respect to D as follows:

$$D' : S \xRightarrow{\bar{r}_1\bar{r}_2\cdots\bar{r}_n} \bar{w} \xRightarrow{r_{a_{i_1}} r_{a_{i_2}} \cdots r_{a_{i_k}}} w$$

where $\bar{r}_i \in \bar{R}$, $1 \leq i \leq n$ and $\bar{r} : A \to \phi(\alpha)$ for each $r : A \to \alpha \in R$, $\bar{w} = \bar{a}_{i_1}\bar{a}_{i_2}\cdots\bar{a}_{i_k}$, and $r_{a_{i_j}} : \bar{a}_{i_j} \to a_{i_j}$, $1 \leq j \leq k$. One can easily see that $\nu' = \nu \cdot t_\mu \cdot t_{a_{i_1}} t_{a_{i_2}} \cdots t_{a_{i_k}}$ is a successful occurrence sequence of transitions of N' and $\gamma'(\nu') = \bar{r}_1\bar{r}_2\cdots\bar{r}_n \cdot r_{a_{i_1}} r_{a_{i_2}} \cdots r_{a_{i_k}}$. Therefore, $L(G) \subseteq L(G')$.

Let $S \overset{\pi}{\Rightarrow} w \in \Sigma^*$ be a derivation in G'. Then, $\pi \in (\bar{R} \cup R_\Sigma)^*$ and the corresponding successful occurrence of sequence ν of transitions of N' is of the form $\nu = \nu' \cdot t_\mu \cdot \nu''$ for some $\nu', \nu'' \in (T \cup T_\Sigma)^*$ and for some $t_\mu \in T_{M_0}$.

Without loss of generality we can change of the order of application of rules in π such that $\pi = \pi' \cdot \pi''$ where $\pi' \in \bar{R}^*$ and $\pi'' \in R_\Sigma^*$. Correspondingly, for ν we have $\nu = \nu' \cdot t_\mu \cdot \nu''$ where $\gamma'(\nu') = \pi'$ and $\gamma'(\nu'') = \pi''$. It follows that $S \xRightarrow{r_1 r_2 \cdots r_n} w$ is a derivation in G where $r_1 r_2 \cdots r_n$ corresponds to $\pi' = \bar{r}_1\bar{r}_2\cdots\bar{r}_n$ and $\gamma(t_1 t_2 \cdots t_m) = r_1 r_2 \cdots r_n$ where $\gamma'(t_1 t_2 \cdots t_m) = \pi'$. Hence, $t_1 t_2 \cdots t_m$ is a successful occurrence sequence for M. It follows that $L(G') \subseteq L(G)$. □

In the remaining part we discuss the relation between Petri net controlled languages and matrix languages.

Lemma 11. *For* $x \in \{f, -\lambda, \lambda\}$ *and* $y \in \{r, t, g\}$, $\mathbf{PN}^\lambda(x, y) \subseteq \mathbf{MAT}^\lambda$.

Proof. Let $G = (V, \Sigma, S, R, N, \gamma, M')$ be a (x, y)-Petri net controlled grammar with $N = (P, T, F, \varphi, \iota)$ where $x \in \{f, -\lambda, \lambda\}$ and $y \in \{r, t, g\}$. Let $P = \{p_1, p_2, \ldots, p_n\}$.

We set $V' = V \cup \bar{P} \cup \{S', B\}$ where $\bar{P} = \{\bar{p} \mid p \in P\}$ is a set of new nonterminal symbols and S', B are new nonterminal symbols. Let

$${}^\bullet t = \{p_{i_1}, p_{i_2}, \ldots, p_{i_k}\} \text{ and } t^\bullet = \{p_{j_1}, p_{j_2}, \ldots, p_{j_m}\},\ t \in T.$$

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

$$\sigma_{i_1} : \underbrace{\bar{p}_{i_1} \to \lambda, \bar{p}_{i_1} \to \lambda, \ldots, \bar{p}_{i_1} \to \lambda}_{\varphi(p_{i_1}, t)} \tag{3}$$

$$\sigma_{i_2} : \underbrace{\bar{p}_{i_2} \to \lambda, \bar{p}_{i_2} \to \lambda, \ldots, \bar{p}_{i_2} \to \lambda}_{\varphi(p_{i_2}, t)} \tag{4}$$

$$\ldots$$

$$\sigma_{i_k} : \underbrace{\bar{p}_{i_k} \to \lambda, \bar{p}_{i_k} \to \lambda, \ldots, \bar{p}_{i_k} \to \lambda}_{\varphi(p_{i_k}, t)} \tag{5}$$

$$\sigma_B : B \to B\bar{p}_{j_1}^{\varphi(t, p_{j_1})} \bar{p}_{j_2}^{\varphi(t, p_{j_2})} \cdots \bar{p}_{j_m}^{\varphi(t, p_{j_m})} \tag{6}$$

and define the matrix

$$m_r = (\sigma_{i_1}, \sigma_{i_2}, \ldots, \sigma_{i_k}, \sigma_B, r) \tag{7}$$

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

$$m_0 = (S' \to SB \cdot \prod_{p \in P} \bar{p}^{|\iota(p)|}) \tag{8}$$

According to types of the sets of final markings we consider three cases of erasing rules:

Case $y = t$. For each $\tau \in M'$,

$$m_{\tau,\lambda} = (B \to \lambda, \underbrace{\bar{p}_1 \to \lambda, \ldots, \bar{p}_1 \to \lambda}_{\tau(p_1)}, \ldots, \underbrace{\bar{p}_n \to \lambda, \ldots, \bar{p}_n \to \lambda}_{\tau(p_n)}). \tag{9}$$

Case $y = r$.

$$m_{p,\lambda} = (\bar{p} \to \lambda) \text{ for each } p \in P \text{ and } m_{B,\lambda} = (B \to \lambda) \tag{10}$$

Case $y = g$. Here we consider matrices (9) together with matrices (10).

We consider the matrix grammar $G' = (V', \Sigma, S', M)$ where M consists of all matrices of (7)-(8) and matrices (9) (Case $y = t$), matrices (10) (Case $y = r$) or matrices (9)-(10) (Case $y = g$).

Let $D : S \xRightarrow{r_1 r_2 \cdots r_n} w \in \Sigma^*$ be a derivation in G. Then $\nu = t_1 t_2 \cdots t_s$ where $\gamma(\nu) = r_1 r_2 \cdots r_n$ is an occurrence sequence of transitions of N enabled at the initial marking ι.

We construct the derivation D' in G' which simulates the derivation D. The derivation D' starts with $S' \Rightarrow SB \cdot \prod_{p \in P} \bar{p}^{|\iota(p)|}$ applying the matrix (8), then for each pair of a transition t in ν and the corresponding rule $r = \gamma(t)$, we choose a matrix of the form (7). When the terminal string $w \in \Sigma^*$ is generated, in order to erase the remaining symbols from $\bar{P}$ and the symbol B we use matrices of the form (9), (10) or (9) and (10) depending on $y \in \{r, t, g\}$.

Let $D' : S' \xRightarrow{m_0} SB \cdot \prod_{p \in P} \bar{p}^{|\iota(p)|} \xRightarrow{m_{i_1} m_{i_2} \cdots m_{i_n}} w_n = w \in \Sigma^*$ be a derivation in G'. Since $V \cap \bar{P} = \emptyset$, we can write a derivation $D'' : S \xRightarrow{r_{j_1} r_{j_2} \cdots r_{j_k}} w_{j_k} = w \in \Sigma^*$ where r_{j_i} is the rule of the non-erasing matrix $m_{r_{j_i}}$, $1 \le i \le k$ 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 (7) in D' shows that there are at least $\varphi(p_{i_1}, t)$ pieces of $\bar{p}_{i_1}$, etc., and at least $\varphi(p_{i_k}, t)$ pieces of $\bar{p}_{i_k}$ in the sentential form, i.e., the input places $p_{i_1}, p_{i_2}, p_{i_k}$ of t have at least $\varphi(p_{i_1}, t)$, $\varphi(p_{i_2}, t)$, ..., $\varphi(p_{i_k}, t)$ tokens, respectively. Thus, the transition t, $\gamma(t) = r$ is enabled in N. We can construct the successful occurrence sequence $\iota \xrightarrow{t_{j_1} t_{j_2} \cdots t_{j_k}} \mu_k$ where $\gamma(t_{j_i}) = r_{j_i}$, $1 \le i \le k$. Hence, D'' is a derivation in G. Thus $L(G') \subseteq L(G)$.

Now let $E : S \xRightarrow{r_{j_1} r_{j_2} \cdots r_{j_k}} w_{j_k} = w \in \Sigma^*$ be a derivation in G. Then we also have the derivation $E' : S' \xRightarrow{m_0} SB \xRightarrow{m_{j_1} m_{j_2} \cdots m_{j_k}} w'_{j_k} B$ in G' where w'_{j_k} differs from w_{j_k}

only in letters $\bar{p}$ with $p \in P$. These letters and B can be erased with matrices (9), (10) or (9) and (10) depending on $y \in \{r, t, g\}$. Thus $L(G) \subseteq L(G')$. $\square$

Lemma 12. $\mathbf{MAT}^{[\lambda]} \subseteq \mathbf{PN}^{[\lambda]}(-\lambda, r)$.

Proof. Let $G = (V, \Sigma, S, M)$ be a matrix grammar (with or without erasing rules) and $M = \{m_1, m_2, \ldots, m_n\}$ where $m_i = (r_{i1}, r_{i2}, \ldots, r_{ik(i)})$, $1 \leq i \leq n$. Without loss of generality we can assume that G is without repetitions. Let $R = \{r_{ij} \mid 1 \leq i \leq n, 1 \leq j \leq k(i)\}$.

We set $\bar{\Sigma} = \{\bar{a} \mid a \in \Sigma\}$ where, for $a \in \Sigma$, $\bar{a}$ is a new nonterminal symbol. We define the bijection $\psi : V \cup \Sigma \to V \cup \bar{\Sigma}$ by

$$\psi(x) = \begin{cases} x & \text{if } x \in V, \\ \bar{x} & \text{if } x \in \Sigma, \end{cases}$$

and for each rule $r = A \to x_1 x_2 \cdots x_l \in R$, we introduce the new rule $\bar{r} = A \to \psi(x_1)\psi(x_2)\cdots\psi(x_l)$.

Let $\bar{M} = \{\bar{m}_1, \bar{m}_2, \ldots, \bar{m}_n\}$ where $\bar{m}_i = (\bar{r}_{i1}, \bar{r}_{i2}, \ldots, \bar{r}_{ik(i)})$, $1 \leq i \leq n$, and $M_\Sigma = \{(\bar{a} \to a) \mid a \in \Sigma\}$.

We construct the matrix grammar $G' = (V \cup \bar{\Sigma}, \Sigma, S, \bar{M} \cup M_\Sigma)$. Obviously, $L(G) = L(G')$.

We define a $(-\lambda, r)$-PN controlled grammar $G'' = (V \cup \bar{\Sigma}, \Sigma, S, R', N, \gamma, M')$, where $R' = R \cup \{\bar{a} \to a \mid a \in \Sigma\}$ and $N = (P, T, F, \varphi, \iota)$ is a control Petri net where the sets of places, transitions and arcs are respectively defined by

$$\begin{aligned} P =& \{p_0\} \cup \{p_{ij} \mid 1 \leq i \leq n, 1 \leq j \leq k(i) - 1\}, \\ T =& \{t_a \mid a \in \Sigma\} \cup \{t_{ij} \mid 1 \leq i \leq n, 1 \leq j \leq k(i)\}, \\ F =& \{(p_0, t_a), (t_a, p_0) \mid a \in \Sigma\} \cup \{(p_0, t_{i1}), (t_{ik(i)}, p_0) \mid 1 \leq i \leq n\} \\ & \cup \{(t_{ij}, p_{ij}) \mid 1 \leq i \leq n, 1 \leq j \leq k(i) - 1\} \\ & \cup \{(p_{ik(i)-1}, t_{ik(i)}) \mid 1 \leq i \leq n\}. \end{aligned}$$

The weight function is defined by $\varphi(x, y) = 1$ for all $(x, y) \in F$, and the initial marking is defined by $\iota(p_0) = 1$, and $\iota(p) = 0$ for all $P' - \{p_0\}$. The labeling function $\gamma : T \to R'$ is defined by $\gamma(t_a) = \bar{a} \to a$ for all $a \in \Sigma$ and $\gamma(t_{ij}) = r_{ij}$, $1 \leq i \leq n, 1 \leq j \leq k(i)$.

Let

$$S = w_0 \xRightarrow{m_{i_1}} w_1 \xRightarrow{m_{i_2}} \cdots \xRightarrow{m_{i_l}} w_l = w \in \Sigma^* \tag{11}$$

be a derivation in G', where m_{i_j}, $1 \leq j \leq l$ is an element of $\bar{M}$ or M_Σ, and

$$w_{j-1} \xRightarrow{m_{i_j}} w_j : \; w_{j-1} \xRightarrow{\bar{r}_{i_j 1}\bar{r}_{i_j 2}\cdots\bar{r}_{i_j k(i_j)}} w_j \text{ or } w_{j-1} \xRightarrow{\bar{a}\to a} w_j$$

for some $a \in \Sigma$. Then by definition of γ, $\mu_{j-1} \xrightarrow{\nu_j} \mu_j$ where $\nu_j = t_{i_j 1} t_{i_j 2} \cdots t_{i_j k(i_j)}$ or $\nu_j = t_a$ and $\mu_j = \iota$ for all $1 \leq j \leq l$. Hence, according to (11), we can construct the successful occurrence sequence $\iota \xrightarrow{\nu_1 \nu_2 \cdots \nu_l} \iota$ of transitions of N. Therefore, $S \xRightarrow{\pi_1 \pi_2 \cdots \pi_l} w_l \in \Sigma^*$ is a derivation in G'', where, for each $1 \leq j \leq l$, $\pi_j = \bar{r}_{i_j 1}\bar{r}_{i_j 2}\cdots\bar{r}_{i_j k(i_j)}$ or $\pi_j = \bar{a} \to a$ for some $a \in \Sigma$.

Let $D : S \xRightarrow{r_1 r_2 \cdots r_l} w \in \Sigma^*$ be a derivation in G'' where $\nu = t_1 t_2 \cdots t_l$, $\gamma(t_i) = r_i$, $1 \leq i \leq l$, is a successful occurrence sequence of transitions of N.

If t_{i1} with $1 \leq i \leq l$ starts in ν, then in the next steps $t_{i2}, t_{i3}, \ldots, t_{ik(i)}$ can only fire in this order. Another t_{j1}, $1 \leq j \leq l$ or t_a for some $a \in \Sigma$ can fire after $t_{ik(i)}$ occurs. By definition of γ, the corresponding label rules $\bar{r}_{i1}, \bar{r}_{i2}, \ldots, \bar{r}_{ik(i)}$ are the elements of one matrix $\bar{m}_i \in \bar{M}$, i.e., $\bar{m}_i = (\bar{r}_{i1}, \bar{r}_{i2}, \ldots, \bar{r}_{ik(i)})$. Thus the application of matrices of G' can be simulated by occurrence sequence of transitions of N. It follows that D is also a derivation in G'. □

Now we summarize our results in the following theorem.

Theorem 13. *The relations in Figure 2 hold where* $x \in \{f, -\lambda, \lambda\}$ *and the lines denote inclusions of the lower families into the upper families.*

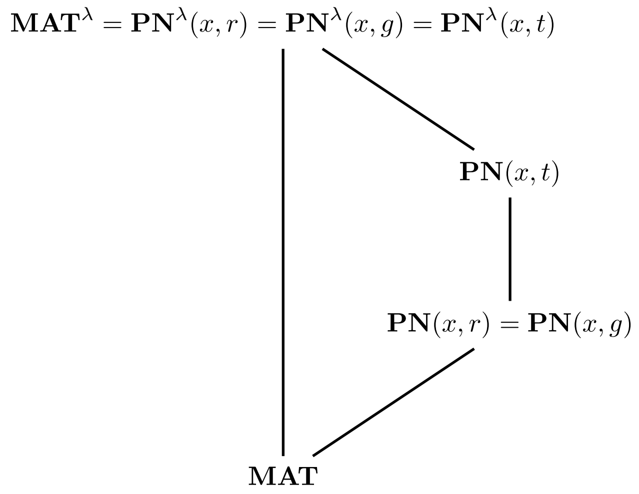

Fig. 2. The hierarchy of language families generated by arbitrary Petri net controlled grammars.

Acknowledgments. We would like to thank the anonymous reviewers for their valuable comments and useful remarks about this paper.

References

[1] ALTMAN R.B., PELEG M., YEH I., *Modelling biological processes using workflow and Petri net models*, Bioinformatics **18**(6), 2002, pp. 825–837.

[2] CESKA M., MAREK V., *Petri nets and random-context grammars*, *Proc. 35th Spring Conference: Modeling and Simulation of Systems (MOSIS'01)*, MARQ Ostrava Hardec nad Moravici, 2001, pp. 145–152.

[3] DASSOW J., PĂUN G., *Regulated rewriting in formal language theory*, Springer-Verlag, Berlin, 1989.

[4] DASSOW J., TURAEV S., *k-Petri net controlled grammars*, in: Martín-Vide, C., Otto, F., Fernau, H. (eds.) *2nd International Conference on Language and Automata Theory and Applications – LATA 2008*, LNCS **5196**, Springer, Berlin, 2008, pp. 209–220.

[5] DASSOW J., TURAEV S., *Arbitrary Petri net controlled grammars*, in: *Proc. 2nd International Workshop ForLing 2008* (Eds.: G. Bel-Enguix, M.D. Jiménez-López), Tarragona, Spain, pp. 27–39, 2008.

[6] HACK M., *Petri net languages*, MIT, Lab. Comp. Sci., Techn. report **159**, Cambridge, Mass., 1976.

[7] HOFESTÄDT R.A., *Petri net application of metabolic processes*, J. Systems Analysis, Modeling and Simulation **16**, 1994, pp. 113–122.

[8] HOPCROFT J.E., ULLMAN J.D., *Introduction to automata theory, languages, and computation*, Addison-Wesley Longman Publishing Co., Inc., 1990.

[9] GUPTA S.M., MOORE K.E., *Petri Net Models of Flexible and Automated Manufacturing Systems: A Survey*, International Journal of Production Research, **34**(11), 1996, pp. 3001–3035.

[10] LEIBMAN M.N., MAVROVOUNIOTIS M.L., REDDY V.N., *Petri net representation in metabolic pathways*, *Proc. First ISMB*, 1993, pp. 328–336.

[11] REISIG W., ROZENBERG G. (eds.), *Lectures on Petri nets I: Basic models*, LNCS **1491**, Springer-Verlag, Berlin, 1998.

[12] ROZENBERG G., SALOMAA A. (eds.), *Handbook of Formal Languages*, Vol. **I–III**, Springer-Verlag, Berlin, 1997.

[13] SILVA M., VALETTE R., *Petri Nets and Flexible Manufacturing*, LNCS, 1989, pp. 374–417.

[14] STARKE P.H., *Free Petri net languages*, In: LNCS **64**, Springer-Verlag, Berlin, 1978, pp. 506–515.

[15] STARKE P.H., *Petri-Netze*, Deutscher Verlag der Wissenschaften, 1980.

[16] TURAEV S., *Petri Net Controlled Grammars*, in: *Proc. 3rd Doctoral Workshop on MEMICS–2007*, Znojmo, Czech Republic, 2007, pp. 233–240.