Journal of Universal Computer Science, vol. 15, no. 14 (2009), 2808-2835
submitted: 26/1/09, accepted: 28/7/09, appeared: 1/8/09 © J.UCS

Petri Net Controlled Grammars: the Case of Special Petri Nets

Jürgen Dassow
(Otto-von-Guericke-Universität Magdeburg, Magdeburg, Germany
dassow@iws.cs.uni-magdeburg.de)

Sherzod Turaev
(Universiti Putra Malaysia, Serdang, Selangor, Malaysia
sherzod@fsktm.upm.edu.my
and
Universitat Rovira i Virgili, Tarragona, Spain
sherzod.turaev@urv.cat)

Abstract: A *Petri net controlled grammar* is a context-free grammar equipped with a Petri net, whose transitions are labeled with rules of the grammar or the empty string, and the associated language consists of all terminal strings which can be derived in the grammar and the 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. We present some results on the generative capacity of such grammars so that the associated Petri nets are restricted to some known special classes of Petri nets.

Key Words: grammars, Petri nets, grammars with regulated rewriting, Petri net controlled grammars
Category: F.4.2, F.4.3

1 Introduction

It is 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 applications of sequential grammars they cannot describe all aspects. On the other hand, 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 in order to generate languages with larger generative capacity than context-free ones. The monograph [Dassow and Păun 1989] gives a summary of this approach. The regularly controlled grammars are a well-known class of such grammars, where a finite automaton is associated with a grammar and the sequence of applied rules has to be accepted by the automaton.

Control by automata can be generalized if we associate a Petri net with a context-free grammar and require that the sequence of applied rules corresponds to an occurrence sequence of transitions of the Petri net; thus, we define a *Petri net controlled grammar*. The study of such kind of grammars is interesting as control by Petri nets makes possible to investigate *concurrent control mechanisms* in the theory of regulated rewriting systems. In [Dassow and Turaev 2008] we investigated grammars controlled by arbitrary (place/transition) Petri nets depending on the type of labeling (a bijection, coding and weak coding) and on the use of final markings (a finite set of final markings and the set of all reachable markings), and showed that Petri net controlled grammars have the same power as some other regulating mechanisms such as matrices, finite automata. If we consider these matrices and finite automata in terms of control mechanisms, special types of matrices and special regular languages are widely investigated in literature, for instance, as control, simple matrices ([Ibarra 1970]) or some subclasses of regular languages ([Dassow 1988, Dassow and Truthe 2008]) are considered. Thus, it is also natural to investigate grammars controlled by some special classes of Petri nets. In this paper we consider grammars controlled by (generalized) state machines, (generalized) marked graphs, causal nets, (extended) free-choice nets, asymmetric choice nets and ordinary nets. Similarly to the general case we investigate the effects of labeling policies and the definitions of final marking sets to the computational power. We prove that the family of languages generated by (arbitrary) Petri net controlled grammars coincide with the family of languages generated by grammars controlled by free-choice nets.

The paper is organized as follows. In Section 2 we give some notions and definitions from the theories of formal languages and Petri nets needed in sequel. We introduce the concept of a control of derivations in context-free grammars by Petri nets in Section 3. In Section 4 we investigate the effect of labeling on the generative power of grammars controlled by special Petri nets. We discuss the effect of different types of final markings in Petri nets on the generative power and give some characterizations by other regulated grammars in Section 5.

(This paper is a strongly extended version of the paper presented at the Third International Conference on Language and Automata Theory and Applications – LATA 2009, April 2-8, 2009, Tarragona, Spain [Dassow and Turaev 2009].)

2 Definitions

Throughout the paper, we assume that the reader is familiar with the basic concepts of the theories of formal languages and Petri nets; for details we refer to [Rozenberg and Salomaa 1997] (formal languages), [Dassow and Păun 1989] (grammars with regulated rewriting), [Desel and Esparsa 1995] (special Petri nets) and [Baumgarten 1990, Reisig and Rozenberg 1998] (Petri nets).

2.1 Strings, Grammars and Languages

Let Σ be an alphabet. A *string* over Σ is a sequence of symbols from the alphabet. A *shuffle* of strings $u = u_1u_2 \cdots u_n$ and $v = v_1v_2 \cdots v_n$ for some $u_i, v_i \in \Sigma^*$, $1 \leq i \leq n$, is defined as $u_1v_1u_2v_2 \cdots u_nv_n$. A shuffle of strings $u_1, u_2, \ldots, u_n$, $n \geq 3$, is a shuffle of a shuffle of $u_1, u_2, \ldots, u_{n-1}$ and u_n. A *semi-shuffle* of strings $u_1, u_2, \ldots, u_n$ is a shuffle of strings from $\bigcup_{i=1}^{n} u_i^*$. The *length* of a string w is denoted by $\#(w)$, and the number of occurrences of a symbol a in a string w by $\#_a(w)$. 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*.

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*. The string $x \in (V \cup \Sigma)^+$ *directly derives* $y \in (V \cup \Sigma)^*$, 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 $L(G) = \{w \in \Sigma^* \mid S \Rightarrow^* w\}$. The family of context-free languages is denoted by **CF**.

A *regularly controlled grammar* is a quintuple $G = (V, \Sigma, S, R, K)$ where V, Σ, S, R are specified as in a context-free grammar and K is a regular set over R. The language generated by G consists of all words $w \in \Sigma^*$ such that there is a derivation $S \xRightarrow{r_1r_2\cdots r_n} w$ where $r_1r_2 \cdots r_n \in K$.

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*, i.e., finite strings over a set R of context-free rules. The language generated by the grammar G consists of all strings $w \in \Sigma^*$ such that there is a derivation $S \xRightarrow{r_1r_2\cdots r_n} w$ where $r_1r_2 \cdots r_n$ is a concatenation of some matrices $m_{i_1}, m_{i_2}, \ldots, m_{i_k} \in M$, $k \geq 1$.

A *vector grammar* is a quadruple $G = (V, \Sigma, S, M)$ whose components are defined as for a matrix grammar. The language generated by the grammar G consists of all strings $w \in \Sigma^*$ such that there is a derivation $S \xRightarrow{r_1r_2\cdots r_n} w$ where $r_1r_2 \cdots r_n$ is a shuffle of some matrices $m_{i_1}, m_{i_2}, \ldots, m_{i_k} \in M$, $k \geq 1$.

A *semi-matrix grammar* is a quadruple $G = (V, \Sigma, S, M)$ whose components are defined as for a matrix grammar. The language generated by the grammar G consists of all strings $w \in \Sigma^*$ such that there is a derivation $S \xRightarrow{r_1r_2\cdots r_n} w$ where $r_1r_2 \cdots r_n$ is a semi-shuffle of some matrices $m_{i_1}, m_{i_2}, \ldots, m_{i_k} \in M$, $k \geq 1$.

A matrix (semi-matrix, vector) grammar G is called *without repetitions*, if for each rule $r \in R$, $|m_1m_2 \cdots m_n|_r = 1$. For each matrix (semi-matrix, vector) grammar, by adding chain rules, one can construct an equivalent matrix grammar without repetitions.

The families of languages generated by regularly controlled, matrix, vector and semi-matrix grammars (with erasing rules) are denoted by **rC**, **MAT**, **V** and **sMAT** ($\mathbf{rC}^\lambda$, $\mathbf{MAT}^\lambda$, $\mathbf{V}^\lambda$ and $\mathbf{sMAT}^\lambda$), respectively. It is known that

$$\mathbf{CF} \subset \mathbf{rC} = \mathbf{MAT} \subseteq \mathbf{V} \subseteq \mathbf{rC}^\lambda = \mathbf{MAT}^\lambda = \mathbf{V}^\lambda = \mathbf{sMAT}^\lambda$$

(see [Dassow and Păun 1989] and [Turaev 2006]).

We use bracket notation $\mathbf{X}^{[\lambda]}$ for a language family $\mathbf{X}$ in order to say that a statement holds in both cases: with and without erasing rules.

2.2 Petri Nets

A *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 the 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 non-negative integer labels. A Petri net N is called an *ordinary net* (ON) if $\varphi(x, y) = 1$ for all $(x, y) \in F$. We omit φ from the definition of an ordinary net. A mapping $\mu : P \to \{0, 1, 2, \dots\}$ is called a *marking*. For each place $p \in P$, $\mu(p)$ gives the number of *tokens* in p. ${}^\bullet x = \{y \mid (y, x) \in F\}$ and $x^\bullet = \{y \mid (x, y) \in F\}$ are called the sets of *input* and *output* elements of $x \in P \cup T$, respectively.

A sequence of places and transitions $\rho = x_1 x_2 \cdots x_n$ is called a *path* if and only if no place or transition except x_1 and x_n appears more than once, and $x_{i+1} \in x_i^\bullet$ for all $1 \le i \le n - 1$. A path $\rho = x_1 x_2 \cdots x_n$ is a *chain* (*cycle*) if $x_1 \ne x_n$ ($x_1 = x_n$). We denote the sets of places, transitions and arcs of a path ρ by P_ρ, T_ρ, F_ρ, respectively. The sequence of transitions in a path ρ is denoted by $tr(\rho)$.

A transition $t \in T$ is *enabled* by marking μ iff $\mu(p) \ge \varphi(p, t)$ for all $p \in P$. In this case the transition t can *occur*. Its occurrence transforms the marking μ into the marking μ' defined for each place $p \in P$ by $\mu'(p) = \mu(p) - \varphi(p, t) + \varphi(t, p)$. This transformation is denoted by $\mu \xrightarrow{t} \mu'$. 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 $\mu_1, \mu_2, \dots, \mu_k$ such that $\mu \xrightarrow{t_1} \mu_1 \xrightarrow{t_2} \dots \xrightarrow{t_k} \mu_k$. For each $1 \le i \le k$, marking μ_i is called *reachable* from marking μ. $\mathcal{R}(N, \mu)$ denotes the set of all reachable markings from a marking μ.

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 τ of M.

We consider the following main structural subclasses of Petri nets.

- 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 $t \in T$.
- A *generalized state machine* (GSM) is an ordinary Petri net such that $|{}^\bullet t| \leq 1$ and $|t^\bullet| \leq 1$ for all $t \in T$.
- A *marked graph* (MG) is an ordinary Petri net such that each place has exactly one input transition and exactly one output transition, i.e., $|{}^\bullet p| = |p^\bullet| = 1$ for all $p \in P$.
- A *generalized marked graph* (GMG) is an ordinary Petri net such that $|{}^\bullet p| \leq 1$ and $|p^\bullet| \leq 1$ for all $p \in P$.
- A *casual net* (CN) is a generalized marked graph each subnet of which is not a cycle.
- A *free-choice* net (FC) is an ordinary Petri net such that every arc is either the only arc going from the place, or it is the only arc going to a transition, i.e., if $p_1^\bullet \cap p_2^\bullet \neq \emptyset$ then $|p_1^\bullet| = |p_2^\bullet| = 1$ for all $p_1, p_2 \in P$.
- An *extended free-choice* net (EFC) is an ordinary Petri net such that if $p_1^\bullet \cap p_2^\bullet \neq \emptyset$ then $p_1^\bullet = p_2^\bullet$ for all $p_1, p_2 \in P$.
- An *asymmetric choice net* (AC) is an ordinary Petri net such that if $p_1^\bullet \cap p_2^\bullet \neq \emptyset$ then $p_1^\bullet \subseteq p_2^\bullet$ or $p_1^\bullet \supseteq p_2^\bullet$ for all $p_1, p_2 \in P$.

The hierarchy of the introduced subclasses of Petri nets is shown in Figure 1 where the arrows denote proper inclusions of the left families into the right families.

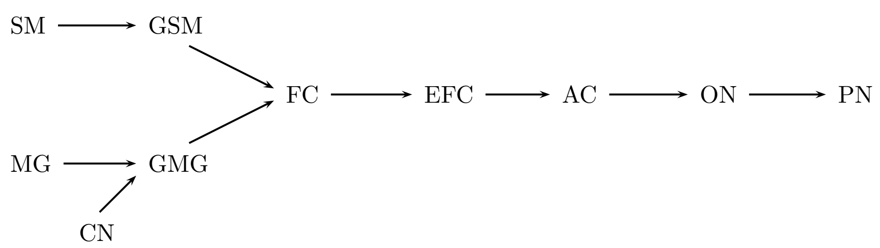

Figure 1: The hierarchy of subclasses of Petri nets

3 Grammars and Their Languages

We introduce the concept of control by Petri nets.

Definition 1. A *PN controlled grammar* is a tuple $\mathcal{G} = (V, \Sigma, S, R, N, \gamma, M)$ 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.

The grammar $\mathcal{G}$ is called a (generalized) state machine, (generalized) marked graph, causal net, (extended) free-choice net, asymmetric choice net or ordinary net controlled grammar if the net N is a (generalized) state machine, (generalized) marked graph, causal net, (extended) free-choice net, asymmetric choice net or ordinary net, respectively. We also use the common name of *special Petri net* (in short, sPN) when we refer to each special class.

Definition 2. The *language* generated by a Petri net controlled grammar $\mathcal{G}$, denoted by $L(\mathcal{G})$, 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 $\gamma : T^* \to R^*$, which is done in the usual manner. Obviously, if γ maps any transition to a rule, then $k = s$.

Different labeling strategies and different definitions of the set of final markings result various types of Petri net controlled grammars. We use a notation an (x, y)-Petri net ((generalized) state machine, (generalized) marked graph, causal net, (extended) free-choice net, asymmetric choice net and ordinary net) controlled grammar where

- $x \in \{f, -\lambda, \lambda\}$ shows the type of a labeling function γ:
  - *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 function γ,
- $y \in \{r, t, g\}$ shows the type of a set of final markings:
  - *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 denote the families of languages generated by grammars controlled by state machines, generalized state machines, marked graphs, generalize marked graphs, causal nets, free-choice nets, extended free-choice nets, asymmetric nets, ordinary nets and Petri nets, respectively, by

$$\mathbf{SM}^{[\lambda]}(x,y),\ \mathbf{GSM}^{[\lambda]}(x,y),\ \mathbf{MG}^{[\lambda]}(x,y),\ \mathbf{GMG}^{[\lambda]}(x,y),\ \mathbf{CN}^{[\lambda]}(x,y),$$
$$\mathbf{FC}^{[\lambda]}(x,y),\ \ \mathbf{EFC}^{[\lambda]}(x,y),\ \ \mathbf{AC}^{[\lambda]}(x,y),\ \ \mathbf{ON}^{[\lambda]}(x,y),\quad \mathbf{PN}^{[\lambda]}(x,y)$$

where $x \in \{f, -\lambda, \lambda\}$ and $y \in \{r, t, g\}$.

The inclusion $\mathbf{X}(x,y) \subseteq \mathbf{X}^{\lambda}(x,y)$ immediately follows from the definition where $\mathbf{X} \in \{\mathbf{SM}, \mathbf{GSM}, \mathbf{MG}, \mathbf{GMG}, \mathbf{CN}, \mathbf{FC}, \mathbf{EFC}, \mathbf{AC}, \mathbf{ON}\}$, $x \in \{f, -\lambda, \lambda\}$ and $y \in \{r, t, g\}$.

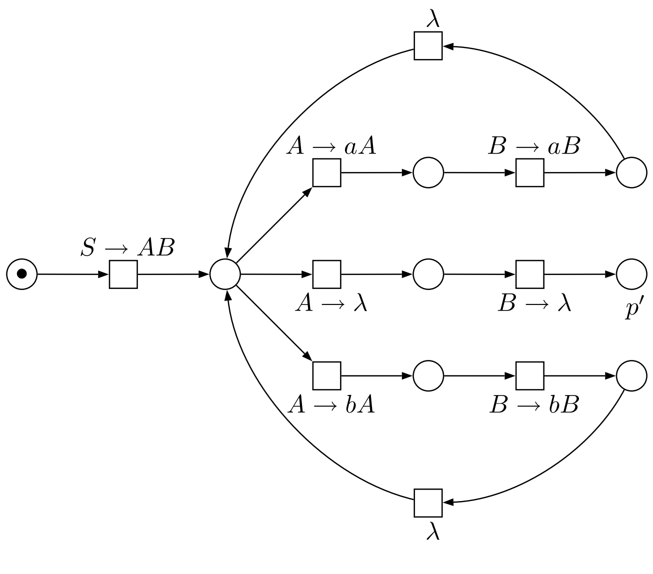

Figure 2: A state machine N_1

Example 1. Let $\mathcal{G}_1 = (\{S, A, B\}, \{a, b\}, S, R, N_1, \gamma_1, M_1)$ be a state machine controlled grammar where R consists of

$$S \to AB, A \to aA, A \to bA, A \to \lambda, B \to aB, B \to bB, B \to \lambda,$$

N_1 which is illustrated in Figure 2 is a labeled state machine and $M_1 = \{\mu\}$ where $\mu(p') = 1$ and $\mu(p) = 0$ for all $p \in P - \{p'\}$, then

$$L(\mathcal{G}_1) = \{ww \mid w \in \{a,b\}^*\} \in \mathbf{SM}^\lambda(\lambda, t).$$

Example 2. Let $\mathcal{G}_2 = (\{S, A, B\}, \{a, b\}, S, R, N_2, \gamma_2, M_2)$ be a MG controlled grammar where R is as for the grammar $\mathcal{G}_1$ in Example 1, a labeled marked graph N_2 is illustrated in Figure 3 and $M_1 = \{\mu\}$ where $\mu(p) = 0$ for all $p \in P$. Then

$$L(\mathcal{G}_2) = \{ww' \mid w \in \{a,b\}^* \text{ and } w' \in \mathrm{Perm}(w)\} \in \mathbf{MG}^\lambda(\lambda, t).$$

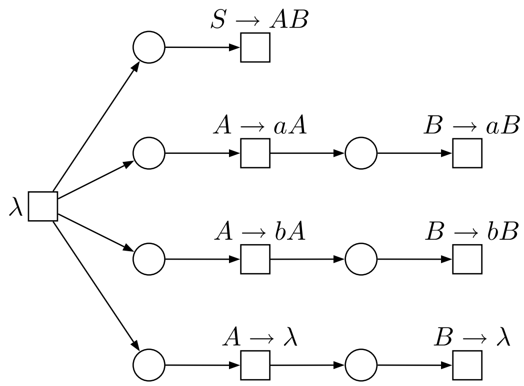

Figure 3: A marked graph N_2

4 Results: Labeling Strategies

In this section we investigate the effect of the labeling of transitions on the generative capacity of the introduced families of languages. From the definition, the next statement follows immediately.

Lemma 3. *For* $\mathbf{X} \in \{\mathbf{SM}, \mathbf{GSM}, \mathbf{MG}, \mathbf{GMG}, \mathbf{CN}, \mathbf{FC}, \mathbf{EFC}, \mathbf{AC}, \mathbf{ON}\}$ *and* $y \in \{r, t, g\}$,

$$\mathbf{X}^{[\lambda]}(f, y) \subseteq \mathbf{X}^{[\lambda]}(-\lambda, y) \subseteq \mathbf{X}^{[\lambda]}(\lambda, y).$$

Further, we show that the reverse inclusions also hold.

For each sPN, one can easily construct a net of the same type in which the transitions have different labels, by "splitting" each transition into two, i.e., by replacing a transition t with label $A \to \alpha$ by new transitions t', t'' with labels $A \to A'$, $A' \to \alpha$, respectively, where t' receives all incoming arcs of t and t'' receives all outgoing arcs of t, and a new place p_t from transition t' and to transition t''.

Lemma 4. *For* $\mathbf{X} \in \{\mathbf{SM}, \mathbf{GSM}, \mathbf{MG}, \mathbf{GMG}, \mathbf{CN}, \mathbf{FC}, \mathbf{EFC}, \mathbf{AC}, \mathbf{ON}\}$ *and* $y \in \{r, t, g\}$,

$$\mathbf{X}^{[\lambda]}(-\lambda, y) \subseteq \mathbf{X}^{[\lambda]}(f, y).$$

Proof. Let $\mathcal{G} = (V, \Sigma, S, R, N, \gamma, M)$ be a $(-\lambda, y)$-sPN controlled grammar (with or without erasing rules) where $y \in \{r, t, g\}$ and $N = (P, T, F, \iota)$. We construct a new sPN N' by replacing each transition $t \in T$ with label $A \to \alpha$ by two new transitions l_t, l'_t with labels $A \to A_t$, $A_t \to \alpha$, respectively, and

$$^\bullet l_t = {}^\bullet t, (l'_t)^\bullet = t^\bullet, l_t^\bullet = \{p_t\} = {}^\bullet l'_t,$$

where A_t, $t \in T$, is a new nonterminal symbol and p_t, $t \in T$, is a new place.

Formally, $N' = (P', T', F', \iota')$ where $P' = P \cup P_t$, $T' = \{l_t, l'_t \mid t \in T\}$ and

$$\begin{aligned} F' = &\{(p, l_t) \mid p \in {}^\bullet t, t \in T\} \cup \{(l'_t, p) \mid p \in t^\bullet, t \in T\} \\ &\cup \{(l_t, p_t), (p_t, l'_t) \mid t \in T\}. \end{aligned}$$

The initial marking ι' is defined by $\iota'(p) = \iota(p)$ if $p \in P$ and $\iota'(p) = 0$ if $p \in P_t$. We should mention that this kind of replacement of transitions of an sPN preserve its structural property.

Let $\mathcal{G}' = (V', \Sigma, S, R', N', \gamma', M')$ be an (f, y)-sPN controlled grammar where $V' = V \cup V_t$ with $V_t = \{A_t \mid t \in T\}$ and

$$R' = \{A \to A_t, A_t \to \alpha \mid A \in R \text{ and } t \in T\}.$$

The labeling function γ' is defined by $\gamma'(l_t) = A \to A_t$ and $\gamma'(l'_t) = A_t \to \alpha$ for all $l_t, l'_t \in T'$ where $\gamma(t) = A \to \alpha \in R$.

For each $\tau' \in M'$, $\tau'(p) = \tau(p)$ if $p \in P$ and $\tau'(p) = 0$ if $p \in P_t$.

In a derivation $S \xRightarrow{r_1 r_2 \cdots r_n} w \in \Sigma^*$ of $\mathcal{G}$, we replace each rule $r_i : A_i \to \alpha_i$ by the pair $r_{t_i} : A_i \to A_{t_i}$, $r'_{t_i} : A_{t_i} \to \alpha_i$, $1 \le i \le n$, and the occurrence sequence of transitions $\nu = t_1 t_2 \cdots t_n$ where $\gamma(\nu) = r_1 r_2 \cdots r_n$ by the occurrence sequence $l_{t_1} l'_{t_1} l_{t_2} l'_{t_2} \cdots l_{t_n} l'_{t_n}$ in the grammar $\mathcal{G}'$. It is difficult to see that $S \xRightarrow{r_{t_1} r'_{t_1} r_{t_2} r'_{t_2} \cdots r_{t_n} r'_{t_n}} w$ is a derivation in $\mathcal{G}'$ where $l_{t_1} l'_{t_1} l_{t_2} l'_{t_2} \cdots l_{t_n} l'_{t_n}$ with

$$\gamma'(l_{t_1} l'_{t_1} l_{t_2} l'_{t_2} \cdots l_{t_n} l'_{t_n}) = r_{t_1} r'_{t_1} r_{t_2} r'_{t_2} \cdots r_{t_n} r'_{t_n}$$

is a successful occurrence sequence in N'. Thus, $w \in L(\mathcal{G}')$.

By construction of N', if a transition l_t for some $t \in T$ in a successful occurrence sequence of transitions σ then l'_t is also in σ, similarly, if $A \to A_t$ in $D : S \overset{\pi}{\Longrightarrow} w \in \Sigma^*$ then $A_t \to \alpha$ is also in D. Without loss of generality we can assume that $\pi = \cdots (A \to A_t)(A_t \to \alpha) \cdots$ and $\sigma = \cdots l_t l'_t \cdots$ (If $\pi = \cdots (A \to A_t)\pi'(A_t \to \alpha) \cdots$ for some $\pi' \in R'^*$ and $\sigma = \cdots l_t \sigma' l'_t \cdots$ for some $\sigma' \in T'^*$ where $\gamma'(\sigma') = \pi'$, then we can change the order of the application of rules and the firing of transitions so that $D' : S \overset{\pi''}{\Longrightarrow} w \in \Sigma^*$ where $\pi'' = \cdots (A \to A_t)(A_t \to \alpha)\pi' \cdots$ with $\gamma'(\pi'') = \sigma''$, $\sigma'' = \cdots l_t l'_t \sigma' \cdots$). We replace each $(A \to A_t)(A_t \to \alpha)$ by $A \to \alpha$ and $l_t l'_t$ by t. Thus, $w \in L(\mathcal{G})$. □

For each (λ, y)-sPN controlled grammar, if we label each λ-transition with $X \to X$, start each derivation with $S' \to SX$ and erase X with rule $X \to \lambda$ at the end of the derivation, then we get the same derivation in a $(-\lambda, y)$-sPN controlled grammar, i.e.,

Lemma 5. *For* $\mathbf{X} \in \{\mathbf{SM}, \mathbf{GSM}, \mathbf{MG}, \mathbf{GMG}, \mathbf{CN}, \mathbf{FC}, \mathbf{EFC}, \mathbf{AC}, \mathbf{ON}\}$ *and* $y \in \{r, t, g\}$,

$$\mathbf{X}^\lambda(\lambda, y) \subseteq \mathbf{X}^\lambda(-\lambda, y).$$

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

A $(-\lambda, y)$-sPN controlled grammar $\mathcal{G}' = (V', \Sigma, S', R', N', \gamma', M')$ is constructed as follows. We set $V' = V \cup \{S', X\}$ and

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

where S' and X are new nonterminals. $N' = (P \cup P', T \cup T', F \cup F', \iota')$ is an sPN where

- $P' = \{p', p''\}$, $T' = \{t', t''\}$, $F' = \{(p', t'), (t', p'), (p'', t''), (t'', p'')\}$ are the sets of new places, transitions and arcs, respectively,
- $\iota'(p) = \iota(p)$ for all $p \in P$ and $\iota'(p') = \iota'(p'') = 1$.

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

$$\gamma'(t) = \begin{cases} \gamma(t) & \text{if } t \in T - T_\lambda \\ X \to X & \text{if } t \in T_\lambda \\ S' \to SX & \text{if } t = t' \\ X \to \lambda & \text{if } t = t''. \end{cases}$$

For each $\tau' \in M'$, $\tau'(p) = \tau(p)$ for all $p \in P$ and $\tau'(p') = \tau'(p'') = 1$.

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

We construct a derivation D' in $\mathcal{G}'$ 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 $X \to \lambda$:

$$S' \Rightarrow SX \xRightarrow{\overbrace{X \to X}^{|\nu_1|} \cdot r_1} w_1X \xRightarrow{\overbrace{X \to X}^{|\nu_2|} \cdot r_2} \cdots \xRightarrow{\overbrace{X \to X}^{|\nu_k|} \cdot r_k} w_kX \xRightarrow{X \to \lambda} w_k \in \Sigma^*$$

and $t'\nu_1t_1\nu_2t_2\cdots\nu_kt_kt''$, $t', t'' \in T'$, is a successful occurrence sequence in N'.

On the other hand, for each derivation

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

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

Lemma 6. *For $y \in \{r, t, g\}$ and $\mathbf{X} \in \{\mathbf{SM}, \mathbf{GSM}\}$, $\mathbf{X}(\lambda, y) \subseteq \mathbf{X}(-\lambda, y)$.*

Proof. Let $\mathcal{G} = (V, \Sigma, S, R, N, \gamma, M)$ be a (λ, y)-state machine controlled grammar where $y \in \{r, t, g\}$ and $N = (P, T, F, \iota)$. Let $T_\lambda \subseteq T$ be the set of all λ-transitions of N and $F_\lambda \subseteq F$ be the set of all incoming and outgoing arcs of the λ-transitions, i.e., $F_\lambda = \{(p,t), (t,p) \mid p \in P \text{ and } t \in T_\lambda\}$.

Before proving the lemma, we introduce some necessary notions. The *i-adjacency set* of $t \in T$ is defined by

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

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 $t' \in \mathrm{Adj}^i(t)$, $i \geq 1$ ($t' \in \mathrm{Adj}^*(t)$) is called an *i-adjacent* (*adjacent*) transition of t. $\mathrm{Adj}^+(t)$ denotes the set of non λ adjacent transitions of $t \in T$, i.e., $\mathrm{Adj}^+(t) = \mathrm{Adj}^*(t) - T_\lambda$.

We construct a new state machine N' without λ-transitions by removing each λ-transition $t_\lambda \in T_\lambda$ with the incoming and outgoing arcs, and adding a new transition t' for each adjacent transition $t \in \mathrm{Adj}(t_\lambda)^+$, and the new arcs from the input place of t_λ to t' and from t' to the output place of t.

For each $t \in \mathrm{Adj}^+(t_\lambda)$, $t_\lambda \in T_\lambda$, we introduce a new "copy" transition l_t which has the same label as t. Let

$$T_c(t_\lambda) = \{l_{t,t_\lambda} \mid t \in \mathrm{Adj}^+(t_\lambda)\} \text{ and } T_c = \bigcup_{t_\lambda \in T_\lambda} T_c(t_\lambda).$$

Let $N' = (P, T', F', \iota)$ be a state machine where $T' = (T - T_\lambda) \cup T_c$, and

$$F' = (F - F_\lambda) \cup \{(p, l_{t,t_\lambda}) \mid p \in {}^\bullet t_\lambda \text{ and } l_{t,t_\lambda} \in T_c(t_\lambda), t_\lambda \in T_\lambda\} \\ \cup \{(l_{t,t_\lambda}, p) \mid l_{t,t_\lambda} \in T_c(t_\lambda) \text{ and } p \in (\mathrm{Adj}^+(t_\lambda))^\bullet, t_\lambda \in T_\lambda\}.$$

We define a $(-\lambda, y)$-SM controlled grammar $\mathcal{G}' = (V, \Sigma, S, R, N', \gamma', M)$ where V, Σ, S, R are as for the grammar $\mathcal{G}$, and $\gamma'(t) = \gamma(t)$ if $t \in T$ and $\gamma'(l_{t,t_\lambda}) = \gamma(t)$ if $l_{t,t_\lambda} \in T_c$ is the copy of $t \in \mathrm{Adj}^+(t_\lambda)$, $t_\lambda \in T_\lambda$.

Let $S \stackrel{\pi}{\Longrightarrow} w \in \Sigma^*$, $\pi = r_1 r_2 \cdots r_n$, be a derivation in $\mathcal{G}$. Then there is an occurrence sequence $\sigma = \sigma_1^\lambda t_1 \sigma_2^\lambda t_2 \cdots \sigma_n^\lambda t_n \sigma_{n+1}^\lambda$, $\gamma(\sigma) = \pi$, which is successful for M where $\sigma_i^\lambda = t_{\lambda,i,1} t_{\lambda,i,2} \cdots t_{\lambda,i,k(i)} \in T^*$, $1 \leq j \leq k(i)$, and $\gamma(\sigma_i^\lambda) = \lambda$, $1 \leq i \leq n{+}1$. Without loss of generality we can assume that $t_i \in \mathrm{Adj}^+(t_{\lambda,i,j})$, $1 \leq j \leq k(i)$, for each $\sigma_i^\lambda \in T^+$, $1 \leq i \leq n$. Then each sequence $t_{\lambda,i,1} t_{\lambda,i,2} \cdots t_{\lambda,i,k(i)} t_i$, $k(i) \geq 1$, is replaced by $l_{t_i,t_\lambda} \in T_c(t_{\lambda,i,1})$ where $t_i \in \mathrm{Adj}^{k(i)}(t_{\lambda,i,1})$, and we get the occurrence sequence σ' in N' and $r_1 r_2 \cdots r_n = \gamma'(\sigma')$ since the transitions l_{t_i,t_λ} and t_i have the same label. Therefore, $L(\mathcal{G}) \subseteq L(\mathcal{G}')$. The inverse case can be shown by backtracking the arguments in this paragraph. □

Lemma 7. *For* $x \in \{f, -\lambda, \lambda\}$ *and* $y \in \{r, t, g\}$,

$$\mathbf{GMG}^{[\lambda]}(x, y) \subseteq \mathbf{MG}^{[\lambda]}(x, y).$$

Proof. Let $\mathcal{G} = (V, \Sigma, S, R, N, \gamma, M)$ be an (x, y)-GMG controlled grammar (with or without erasing rules) where $N = (P, T, F, \iota)$ is a generalized marked graph. Let

$$P_\emptyset^- = \{p \in P \mid {}^\bullet p = \emptyset\} \text{ and } P_\emptyset^+ = \{p \in P \mid p^\bullet = \emptyset\}.$$

Without loss of generality we can assume that $P_\emptyset^- \cap P_\emptyset^+ = \emptyset$ (if place $p \in P$ is isolated, i.e., $|{}^\bullet p| = |p^\bullet| = 0$, it can be eliminated since isolated places do not effect any derivation of the grammar).

Let

$$Q^- = \{q_p \mid p \in P_\emptyset^-\} \text{ and } Q^+ = \{q_p \mid p \in P_\emptyset^+\}$$

be the sets of new places,

$$T^- = \{t_p \mid p \in P_\emptyset^-\} \text{ and } T^+ = \{t_p \mid p \in P_\emptyset^+\}$$

be the sets of new transitions and

$$F^- = \{(t_p, q_p), (q_p, t_p), (t_p, p) \mid p \in P_\emptyset^-\}$$

and

$$F^+ = \{(p, t_p), (t_p, q_p), (q_p, t_p) \mid p \in P_\emptyset^+\}$$

be the sets of new arcs.

We construct a marked graph

$$N' = (P \cup Q^- \cup Q^+, T \cup T^- \cup T^+, F \cup F^- \cup F^+, \iota')$$

where $\iota'(p) = \iota(p)$ if $p \in P$ and $\iota'(p) = 0$ if $p \in Q^- \cup Q^+$.

We set $V' = V \cup \{B\}$ and $R' = R \cup \{B \to B\}$ where B is a new nonterminal symbol, and define a MG controlled grammar $\mathcal{G}' = (V', \Sigma, S, R', N', \gamma', M')$ where the labeling function γ' is defined by $\gamma'(t) = \gamma(t)$ if $t \in T$ and $\gamma'(t) = B \to B$ if $t \in T^- \cup T^+$. For each $\tau' \in M'$, $\tau'(p) = \tau(p)$ if $p \in P$ and $\tau'(p) = 0$ if $p \in Q^- \cup Q^+$.

By construction of N', any transition $t \in T^- \cup T^+$ never occurs and the production rule $B \to B$ is never applied in any derivation of $\mathcal{G}'$. Thus it is not difficult to see that $L(\mathcal{G}) = L(\mathcal{G}')$. $\square$

Lemma 8. *For $y \in \{r, t, g\}$ and $\mathbf{X} \in \{\mathbf{MG}, \mathbf{CN}\}$, $\mathbf{X}(\lambda, y) \subseteq \mathbf{X}(-\lambda, y)$.*

Proof. Let $\mathcal{G} = (V, \Sigma, S, R, N, \gamma, M)$ be a (λ, y)-MG controlled grammar where $N = (P, T, F, \iota)$ is a marked graph. Let $T_\lambda = \{t \mid \gamma(t) = \lambda\}$ and

$$\mathrm{Adj}^+(T_\lambda) = \{t \mid t \in \mathrm{Adj}^+(t_\lambda) \text{ for some } t_\lambda \in T_\lambda\}.$$

We assume that for each $t \in T_\lambda$, t is not a transition of some cycle ρ where $\iota(p) = 0$ for all $p \in P_\rho$ or $(t^\bullet)^\bullet = \emptyset$ (in the former case, the transition t and its incoming and outgoing arcs can be removed without effecting any firing of transitions since t never occurs; in the latter case, the transition t, its outgoing arcs and the places of $t^\bullet$ can be removed as the firing of t does not effect any derivation of the grammar $\mathcal{G}$).

Before proving the lemma, we introduce some necessary notions. Transitions $t_\lambda, t'_\lambda \in T_\lambda$ are called *neighbors* if $t_\lambda \in \mathrm{Adj}^*(t'_\lambda)$ or $t'_\lambda \in \mathrm{Adj}^*(t_\lambda)$. A subset $T'_\lambda \subseteq T_\lambda$ is called a *neighborhood set* if all transitions of T'_λ are pairwise neighbors. A neighborhood set $T'_\lambda \subseteq T_\lambda$ is *maximal* if for any $t_\lambda \in T_\lambda - T'_\lambda$ there is a transition $t'_\lambda \in T'_\lambda$ such that t_λ and t'_λ are not neighbors. Let $\mathrm{Nbr}(T_\lambda)$ be the set of all maximal neighborhood subsets of T_λ. Let $T_\lambda(t)$ denote a maximal neighborhood subset of T_λ such that $t \in \mathrm{Adj}^+(t_\lambda)$ for all $t_\lambda \in T_\lambda(t)$.

We construct a generalized marked graph N' without λ-transitions by

- removing
  - all transitions of T_λ,
  - all places of ${}^\bullet t_\lambda$ for each $t_\lambda \in T_\lambda$,
  - all incoming and outgoing arcs of each place $p \in {}^\bullet t_\lambda$, $t_\lambda \in T_\lambda$, let

$$F_\lambda = \{(p, t), (t, p) \mid p \in {}^\bullet T_\lambda\},$$

- all incoming arcs of each place $p \in {}^\bullet t \cap T_\lambda^\bullet$ where $t \in \mathrm{Adj}^+(T_\lambda)$, let

$$F_A = \{(t, p) \mid p \in {}^\bullet(\mathrm{Adj}^+(T_\lambda)) \cap T_\lambda^\bullet\},$$

– adding
  - a new transition $[t]_\lambda$ for each $t_\lambda \in T_\lambda(t)$, $T_\lambda(t) \in \mathrm{Nbr}(T_\lambda)$, $t \in T$, let

$$[T]_\lambda = \bigcup_{t \in T} \bigcup_{T_\lambda(t) \in \mathrm{Nbr}(T_\lambda)} \{[t]_\lambda \mid t_\lambda \in T_\lambda(t)\},$$

  - a new place $p[t]_\lambda$ for each $p \in {}^\bullet t_\lambda$, $t_\lambda \in T_\lambda(t)$, $T_\lambda(t) \in \mathrm{Nbr}(T_\lambda)$, $t \in T$, let

$$[P]_\lambda = \bigcup_{t \in T} \bigcup_{T_\lambda(t) \in \mathrm{Nbr}(T_\lambda)} \bigcup_{t_\lambda \in T_\lambda} \{p[t]_\lambda \mid p \in {}^\bullet t_\lambda\},$$

  - for each $p \in {}^\bullet t_\lambda$, $t_\lambda \in T_\lambda(t)$, $T_\lambda(t) \in \mathrm{Nbr}(T_\lambda)$, $t \in T$, add new arcs $(t', p[t]_\lambda)$ where $t' \in {}^\bullet p$ and $t' \notin T_\lambda$, $([t']_\lambda, p[t]_\lambda)$ where $t'_\lambda \in T_\lambda \cap {}^\bullet p$, $(p[t]_\lambda, [t]_\lambda)$, let

$$\begin{aligned}[F]_\lambda = \bigcup_{t \in T} \bigcup_{T_\lambda(t) \in \mathrm{Nbr}(T_\lambda)} \bigcup_{t_\lambda \in T_\lambda} (&\{(t', p[t]_\lambda) \mid t' \in {}^\bullet p, t' \notin T_\lambda, p \in {}^\bullet t_\lambda\} \\ &\cup \{([t']_\lambda, p[t]_\lambda) \mid t'_\lambda \in T_\lambda \cap {}^\bullet p, p \in {}^\bullet t_\lambda\} \\ &\cup \{(p[t]_\lambda, [t]_\lambda) \mid p \in {}^\bullet t_\lambda\}),\end{aligned}$$

    and for each $p \in {}^\bullet t$, $t \in \mathrm{Adj}^+(T_\lambda)$, add a new arc $([t]_\lambda, p)$ where $t_\lambda \in {}^\bullet p \cap T_\lambda$, let

$$[F]_A = \bigcup_{t \in \mathrm{Adj}^+(T_\lambda)} \{([t]_\lambda, p) \mid p \in {}^\bullet t, t_\lambda \in {}^\bullet p \cap T_\lambda\}.$$

Formally, $N' = (P', T', F', \iota')$ is a generalized marked graph where

$$\begin{aligned} P' &= (P - {}^\bullet T_\lambda) \cup [P]_\lambda, \\ T' &= (T - T_\lambda) \cup [T]_\lambda, \\ F' &= (F - (F_\lambda \cup F_A)) \cup [F]_\lambda \cup [F]_A, \end{aligned}$$

and the initial marking ι' is defined by $\iota'(p) = \iota(p)$ for all $p \in P - {}^\bullet T_\lambda$ and $\iota'(p[t]_\lambda) = \iota(p)$ for all $p[t]_\lambda \in [P]_\lambda$ where $p \in {}^\bullet t_\lambda$.

We define a generalized marked graph $\mathcal{G}' = (V, \Sigma, S, R', N', \gamma', M')$ where

– V, Σ, S are defined as for $\mathcal{G}$, $R' = R \cup \{A \to A \mid A \to \alpha \in R\}$ and N' is constructed above;

– the labeling function γ' is defined by $\gamma'(t) = \gamma(t)$ if $t \in T - T_\lambda$ and $\gamma'([t]_\lambda) = A \to A$ if $[t]_\lambda \in T_\lambda(t')$ where $t' = \gamma^{-1}(A \to \alpha) \in \mathrm{Adj}^+(t_\lambda)$;

– for each final marking $\tau' \in M'$ (if M' is a finite set of final markings), $\tau'(p) = \tau(p)$ if $p \in P$ and $\tau'(p[t]_\lambda) = \tau(p)$ for all $p[t]_\lambda \in [P]_\lambda$ where $p \in {}^\bullet t_\lambda$.

Let

$$S \xRightarrow{\pi} w \in \Sigma^*, \ \pi = r_1 r_2 \cdots r_n \tag{1}$$

be a derivation in $\mathcal{G}$. Then there is a successful occurrence sequence of transitions $\sigma = t_1 t_2 \cdots t_m$, $m \geq n \geq 1$, such that $\gamma(\sigma) = \pi$. Let

$$\iota \xrightarrow{t_1} \mu_1 \xrightarrow{t_2} \mu_2 \xrightarrow{t_3} \cdots \xrightarrow{t_m} \mu_m \tag{2}$$

We construct a successful occurrence sequence σ' of transitions of N' on the base of (2) as follows: all transitions in (2) which are from $T - T_\lambda$ also remain in σ'. If in order to fire a transition $t \in \mathrm{Adj}^+(T_\lambda)$ which is in (2), some transitions $t_{\lambda,i_1}, t_{\lambda,i_2}, \ldots, t_{\lambda,i_l} \in T_\lambda(t)$, $l \geq 1$, where $t_{\lambda,i_{j+1}} \in \mathrm{Adj}^1(t_{\lambda,i_j})$, $2 \leq j \leq l-1$, and $t \in \mathrm{Adj}^1(t_{\lambda,i_l})$ are to be fired then $t_{\lambda,i_1}, t_{\lambda,i_2}, \ldots, t_{\lambda,i_l}$ are replaced by transitions $[t]_{\lambda,i_1}, [t]_{\lambda,i_2}, \ldots, [t]_{\lambda,i_l}$, otherwise, i.e., if the firing of a transition $t_\lambda \in T_\lambda$ does not effect the firing of $t \in \mathrm{Adj}^+(t_\lambda)$, it is removed. Correspondingly, a derivation in the grammar $\mathcal{G}'$ is constructed from (1) by adding a rule $A \to A$ for each $[t']_\lambda$ where $\gamma(t) = A \to \alpha$ and $t \in \mathrm{Adj}^+(t'_\lambda)$. It is clear that the result of the derivation does not change. Therefore, $L(\mathcal{G}) \subseteq L(\mathcal{G}')$.

The inverse case can be easily shown: each $[t']_\lambda$ with $\gamma(t) = A \to \alpha$ and $t \in \mathrm{Adj}^+(t'_\lambda)$, is replaced by t_λ and its label $A \to A$ is removed in the derivation and the same string is generated. □

Lemma 9. *For* $\mathbf{X} \in \{\mathbf{EFC}, \mathbf{AC}, \mathbf{ON}\}$ *and* $y \in \{r, t, g\}$,

$$\mathbf{X}(\lambda, y) \subseteq \mathbf{X}(-\lambda, y).$$

Proof. Let $\mathcal{G} = (V, \Sigma, S, R, N, \gamma, M)$ be a (λ, y)-extended free-choice (asymmetric choice, ordinary) net controlled grammar with $N = (P, T, F, \iota)$. Let

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

For each $t_\lambda \in T_\lambda$, we define the set of new transitions

$$T(t_\lambda) = \{[t] \mid t \in \mathrm{Adj}^+(t_\lambda)\}.$$

We introduce the set $R(t_\lambda)$ of new rules with respect to each $t_\lambda \in T_\lambda$

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

We define a $(-\lambda, y)$-extended free-choice (asymmetric-choice, ordinary) net controlled grammar $\mathcal{G}' = (V, \Sigma, S, R', N', \gamma', M')$ where

$$R' = R \cup \bigcup_{t \in T_\lambda} R(t)$$

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

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

This method of the addition of new arcs preserves the structural properties of an extended free-choice, asymmetric choice and ordinary nets.

The function $\gamma' : T' \to R'$ is defined by $\gamma'(t) = \gamma(t)$ for all $t \in T$ and $\gamma'([t]) = A \to A \in R(t_\lambda)$ where $[t] \in T(t_\lambda)$ and $t \in \mathrm{Adj}^+(t_\lambda)$ where $\gamma(t) = A \to \alpha \in R$.

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

$$t'_{1,1} \cdots t'_{1,k(1)} t_1 t'_{2,1} \cdots t'_{2,k(2)} t_2 \cdots t_n t'_{n+1,1} \cdots t'_{n+1,k(n+1)} \tag{3}$$

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

Each λ-transition $t'_{i,j}$, $1 \le i \le n$, $1 \le j \le k(i)$ in (3) can be replaced by the transition $t''_{i,j}$ in N', $1 \le i \le n$, $1 \le j \le k(i)$ with the label $A_i \to A_i$ where A_i is the left side of the rule $r_i = \gamma(t_i)$, $1 \le i \le n$. Then

$$t''_{1,1} \cdots t''_{1,k(1)} t_1 t''_{2,1} \cdots t''_{2,k(2)} t_2 \cdots t''_{n,1} \cdots t''_{n,k(n)} t_n \tag{4}$$

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 $\mathcal{G}'$ where

$$\sigma_i = r''_{i,1} r''_{i,2} \cdots r''_{i,k(i)}, r''_{i,j} = \gamma'(t''_{i,j}), 1 \le i \le n, 1 \le j \le k(i).$$

Using the same idea, we can show the inverse inclusion. □

Lemma 10. *For $y \in \{r, t, g\}$, $\mathbf{PN}^{[\lambda]}(\lambda, y) \subseteq \mathbf{FC}^{[\lambda]}(\lambda, y)$.*

Proof. Let $\mathcal{G} = (V, \Sigma, S, R, N, \gamma, M)$ be a Petri net controlled grammar (with or without erasing rules) where $N = (P, T, F, \varphi, \iota)$. For each arc $(p, t) \in F$, we introduce new places $p_i[p, t]$, new transitions $t_i[p, t]$ and new arcs $(p, t_i[p, t])$, $(t_i[p, t], p_i[p, t])$, $(p_i[p, t], t)$ whose weights are 1's, $1 \le i \le \varphi(p, t)$, and for each arc $(t, p) \in F$, we introduce new places $p_j[t, p]$, new transitions $t_j[t, p]$ and new arcs $(t, p_j[t, p])$, $(p_j[t, p], t_j[t, p])$, $(t_j[t, p], p)$ whose weights are 1's, $1 \le j \le \varphi(t, p)$. Let

$$\begin{aligned}
P_F =& \{p_i[p,t] \mid (p,t) \in F, 1 \le i \le \varphi(p,t)\} \\
&\cup \{p_j[t,p] \mid (t,p) \in F, 1 \le j \le \varphi(t,p)\}, \\
T_F =& \{t_i[p,t] \mid (p,t) \in F, 1 \le i \le \varphi(p,t)\} \\
&\cup \{t_j[t,p] \mid (t,p) \in F, 1 \le j \le \varphi(t,p)\}, \\
F' =& \{(p, t_i[p,t]), (t_i[p,t], p_i[p,t]) \mid (p,t) \in F, 1 \le i \le \varphi(p,t)\} \\
&\cup \{(p_i[p,t], t) \mid (p,t) \in F, 1 \le i \le \varphi(p,t)\} \\
&\cup \{(t, p_j[t,p]), (p_j[t,p], t_j[t,p]) \mid (t,p) \in F, 1 \le j \le \varphi(t,p)\} \\
&\cup \{(t_j[t,p], p) \mid (t,p) \in F, 1 \le j \le \varphi(t,p)\}.
\end{aligned}$$

We construct a net $N' = (P \cup P_F, T \cup T_F, F', \iota')$ where the initial marking ι' is defined by $\iota'(p) = \iota(p)$ for all $p \in P$ and $\iota'(p) = 0$ for all $p \in P_F$.

Let ${}^\bullet t = \{p_1, p_2, \dots, p_k\}$ for a transition $t \in T$ in N. Then for this transition in N' we have ${}^\bullet t = \bigcup_{i=1}^{k} \{p_j[p_i, t] \mid 1 \le j \le \varphi(p_i, t)\}$ and $(p_j[p_i, t])^\bullet = \{t\}$ for all $1 \le i \le k$ and $1 \le j \le \varphi(p_i, t)$. It follows that N' is a free-choice net.

We define an FC controlled grammar $\mathcal{G}' = (V, \Sigma, S, R, N', \gamma', M')$ where the components V, Σ, S, R are defined as for the grammar $\mathcal{G}$, the free-choice net N' is constructed above. We set $\gamma'(t) = \gamma(t)$ if $t \in T$ and $\gamma'(t) = \lambda$ if $t \in T_F$; for each $\tau' \in M'$, $\tau'(p) = \tau(p)$ if $p \in P$, and for $p \in P_F$, $\tau'(p) = 0$ if $y \in \{g, t\}$, otherwise $0 \le \tau'(p) \le \tau'(p')$ where $p' \in {}^\bullet({}^\bullet p)$.

Let $D : S \xRightarrow{r_1 r_2 \cdots r_m} w \in \Sigma^*$ be a derivation in $\mathcal{G}$. Then there is a successful occurrence sequence of transitions $\nu = t_1 t_2 \cdots t_n$ for M in N such that $\gamma(\nu) = r_1 r_2 \cdots r_m$. We replace ν by $\nu' = \nu_1 t_1 \nu_2 \cdots \nu_n t_n$ in N' where for all $1 \le i \le n$, $\nu_i \in \mathrm{Perm}({}^\bullet({}^\bullet t_i))$ where

$${}^\bullet({}^\bullet t_i) = \{t_j[p_{i_l}, t_i] \mid 1 \le j \le \varphi(p_{i_l}, t_i), 1 \le i \le n, 1 \le l \le s\}.$$

In order to fire each t_i, $1 \le i \le n$, in N', we need to fire all transitions of ${}^\bullet({}^\bullet t_i)$ at least once, therefore, ν' is successful for M' and $r_1 r_2 \cdots r_m = \gamma'(\nu')$, i.e., D is a derivation in $\mathcal{G}'$.

Let $t_1 t_2 \cdots t_n$ be a successful occurrence sequence for M'. By construction, each occurrence of t_i, $1 \le i \le n$, needs at least one occurrence of all transitions of ${}^\bullet({}^\bullet t_i)$. Without loss of generality we can assume that $\nu = \nu_1^\lambda \sigma_1^\lambda t_1 \nu_2^\lambda \cdots \sigma_n^\lambda t_n \nu_{n+1}^\lambda$

where

$$\sigma_i^\lambda = \prod_{l=1}^{s} \prod_{j=1}^{\varphi(p_{i_l},t_i)} t_j[p_{i_l}, t_i], 1 \le i \le n,$$

and $\nu_i^\lambda \in T_F^*$, $1 \le i \le n+1$.

We replace $\prod_{l=1}^{s} \prod_{j=1}^{\varphi(p_{i_l},t_i)} t_j[p_{i_l}, t_i]t_i$ by t_i, $1 \le i \le n$, and erase ν_i^λ, $1 \le i \le n+1$. The obtained occurrence sequence $\nu' = t_1 t_2 \cdots t_n$ is successful for M in N. Then a derivation $S \xRightarrow{r_1 r_2 \cdots r_n} w \in \Sigma^*$ in $\mathcal{G}'$ where $r_1 r_2 \cdots r_n = \gamma'(\nu)$ is also a derivation in $\mathcal{G}$ and $r_1 r_2 \cdots r_n = \gamma'(\nu')$. □

The immediate consequence of this lemma is

Corollary 11. *For* $\mathbf{X} \in \{\mathbf{EFC}, \mathbf{AC}, \mathbf{ON}\}$ *and* $y \in \{r, g, t\}$,

$$\mathbf{X}^{[\lambda]}(\lambda, y) \subseteq \mathbf{FC}^{[\lambda]}(\lambda, y).$$

Lemma 12. *For* $y \in \{r, t, g\}$, $\mathbf{FC}(\lambda, y) \subseteq \mathbf{FC}(-\lambda, y)$.

Proof. The proof is based on the following idea: for a FC controlled grammar $\mathcal{G}$, $L(\mathcal{G}) \in \mathbf{FC}(\lambda, y)$, we construct an equivalent EFC controlled grammar $\mathcal{G}'$, $L(\mathcal{G}') \in \mathbf{EFC}(-\lambda, y)$ (with an extended free-choice net N' without λ-transitions), according to Lemma 6, next we again transform the grammar $\mathcal{G}'$ into an equivalent FC controlled grammar $\mathcal{G}''$, $L(\mathcal{G}'' \in \mathbf{FC}(-\lambda, y)$, which is equivalent to $\mathcal{G}$.

Let $\mathcal{G} = (V, \Sigma, S, R, N, \gamma, M)$ be a free-choice net controlled grammar with $N = (P, T, F, \iota)$. Let

$$T_\lambda = \{t \in T \mid \gamma'(t) = \lambda\} \text{ and } T(t_\lambda) = \{[t] \mid t \in \mathrm{Adj}^+(t_\lambda)\},$$

$t_\lambda \in T_\lambda$, be the set of new transitions. By Lemma 6, we define an EFC net controlled grammar $\mathcal{G}' = (V, \Sigma, S, R', N', \gamma', M')$ (with the notions of the proof of the lemma), which is equivalent to the grammar $\mathcal{G}$, where $N' = (P, T', F', \iota)$.

By construction of N', for all $t \in T - T_\lambda$, $p_1^\bullet \cap p_2^\bullet \cap \cdots \cap p_k^\bullet = \{t\}$ where ${}^\bullet t = \{p_1, p_2, \ldots, p_k\}$ (the property of "free-choiceness"). On the other hand, for each transition $t_\lambda \in T_\lambda$, all transitions of $T(t_\lambda)$ have the same set of input places, i.e., for all $t_1, t_2 \in T(t_\lambda)$, ${}^\bullet t_1 = {}^\bullet t_2$ (the property of "extended free-choiceness").

Let

$$F_\lambda^- = \bigcup_{t_\lambda \in T_\lambda} \{(p, t) \in F' \mid t \in T(t_\lambda)\}.$$

For each $t \in T(t_\lambda)$, $t_\lambda \in T_\lambda$, we replace each incoming arc (p, t) by a new place $p[p, t]$, a new transition $t[p, t]$ and new arcs $(p, t[p, t])$, $(t[p, t], p[p, t])$, $(p[p, t], t)$.

Let

$$P_{P\times T} = \bigcup_{t_\lambda\in T_\lambda} \bigcup_{t\in T(t_\lambda)} \{p[p,t] \mid (p,t)\in F'\},$$
$$T_{P\times T} = \bigcup_{t_\lambda\in T_\lambda} \bigcup_{t\in T(t_\lambda)} \{t[p,t] \mid (p,t)\in F'\},$$
$$F_{P\times T} = \bigcup_{t_\lambda\in T_\lambda} \bigcup_{t\in T(t_\lambda)} \{(p,t[p,t]),(t[p,t],p[p,t]),(p[p,t],t) \mid (p,t)\in F'\}.$$

We construct a net $N'' = (P'', T'', F'', \iota'')$ where

$$P'' = P \cup P_{P\times T}, T'' = T' \cup T_{P\times T}, F'' = (F - F_\lambda^-) \cup F_{P\times T},$$

and the initial marking is defined by $\iota''(p) = \iota(p)$ for all $p \in P$ and $\iota''(p) = 0$ for all $p \in P_{P\times T}$.

If ${}^\bullet t = \{p_1, p_2, \ldots, p_k\}$ for a transition $t \in T(t_\lambda)$ in N' then for this transition in N'' we get ${}^\bullet t = \{p[p_1,t], p[p_2,t], \ldots, p[p_k,t]\}$ and

$$(p[p_1,t])^\bullet \cap (p[p_2,t])^\bullet \cap \cdots \cap (p[p_k,t])^\bullet = \{t\}.$$

It follows that N'' is a free-choice net.

We define a FC controlled grammar $\mathcal{G}'' = (V, \Sigma, S, R', N'', \gamma'', M'')$ where V, Σ, S, R' are defined as for $\mathcal{G}'$ and the net N'' is constructed above.

The labeling function γ'' is defined by $\gamma''(t) = \gamma'(t)$ for all $t \in T'$ and for $t[p,t] \in T_{P\times T}$, $\gamma''(t[p,t]) = \gamma'(t)$, $t \in T(t_\lambda)$ (the label of each $t \in T(t_\lambda)$ is a chain rule of he form $A \to A$, see the proof of Lemma 6).

For each $\tau'' \in M''$, $\tau''(p) = \tau'(p)$ if $p \in P$ and for $p[p,t] \in P_{P\times T}$, $\tau''(p[p,t]) = 0$ if $y \in \{g,t\}$, and if $y = r$ then $0 \le \tau''(p[p,t]) \le \tau''(p)$ where $p \in {}^\bullet({}^\bullet p[p,t])$.

Further we can repeat the arguments of the proof of Lemma 6. □

From the presented lemmas above, we can conclude that the labeling strategies of transitions of special Petri nets do not effect on the generative powers of the families of languages generated by grammars controlled by these nets.

Theorem 13. *For* $\mathbf{X} \in \{\mathbf{SM}, \mathbf{GSM}, \mathbf{MG}, \mathbf{GMG}, \mathbf{CN}, \mathbf{FC}, \mathbf{EFC}, \mathbf{AC}, \mathbf{ON}\}$, *and* $y \in \{r,t,g\}$,

$$\mathbf{X}^{[\lambda]}(f,y) = \mathbf{X}^{[\lambda]}(-\lambda,y) = \mathbf{X}^{[\lambda]}(\lambda,y).$$

5 Results: Final Markings

In this section, we give some characterizations of the classes of languages generated by sPN controlled grammars by other classes of regulated languages.

From the structural properties of special Petri nets and Lemmas 7, 10, the next statement follows immediately

Theorem 14. *For* $\mathbf{X} \in \{\mathbf{FC}, \mathbf{EFC}, \mathbf{AC}, \mathbf{ON}\}$ *and* $x \in \{f, -\lambda, \lambda\}$, $y \in \{r, g, t\}$,

$$\mathbf{SM}(x,y) \subseteq \mathbf{GSM}(x,y) \subseteq \mathbf{X}(x,y) \subseteq \mathbf{X}^{\lambda}(x,y),$$
$$\mathbf{CN}(x,y) \subseteq \mathbf{MG}(x,y) = \mathbf{GMG}(x,y) \subseteq \mathbf{X}(x,y) \subseteq \mathbf{X}^{\lambda}(x,y).$$

Lemma 15. $\mathbf{SM}^{[\lambda]}(\lambda, r) \subseteq \mathbf{SM}^{[\lambda]}(\lambda, t)$.

Proof. Let $\mathcal{G} = (V, \Sigma, S, R, N, \gamma, M)$ be a state machine controlled grammar (with or without erasing rules) where $N = (P, T, F, \iota)$.

Since the firing of a transition in a state machine moves one token from the input place to the output place, the number of tokens in the net remains the same in any firing of a transition. It follows that the set M of all reachable markings is finite, i.e.,

$$|M| \leq \binom{n+k-1}{k-1}$$

where $n = \sum_{p \in P} \iota(p)$ and $k = |P|$ ($\binom{n+k-1}{k-1}$ is the number of solutions in non-negative integers to the equation $x_1 + x_2 + \cdots + x_k = n$, see [Graham et al. 1996]). $\square$

From Lemma 15 the next statements follow

Corollary 16. $\mathbf{SM}^{[\lambda]}(\lambda, r) \subseteq \mathbf{SM}^{[\lambda]}(\lambda, g)$ *and* $\mathbf{SM}^{[\lambda]}(\lambda, g) \subseteq \mathbf{SM}^{[\lambda]}(\lambda, t)$.

Proof. 1. If a finite set of final marking is defined as the set of all reachable markings, Lemma 15 also holds for "g"-case.

2. Let $\mathcal{G} = (V, \Sigma, S, R, N, \gamma, M)$ be a state machine controlled grammar (with or without erasing rules) where $N = (P, T, F, \iota)$.

Let $M = \{\tau \mid \tau(p) \geq \tau'(p) \text{ for all } p \in P \text{ and for some } \tau' \in M'\}$ where $M' = \{\tau'_1, \tau'_2, \ldots, \tau'_k\}$ is a finite set of final markings.

Since

$$\sum_{p \in P} \tau(p) = \sum_{p \in P} \iota(p)$$

for each marking $\tau \in M$, $\tau(p) \leq \sum_{p \in P} \iota(p)$ for all $p \in P$. We define a set of final markings as

$$M'' = \{\tau \mid \tau'(p) \leq \tau(p) \leq K \text{ for all } p \in P \text{ and for some } \tau' \in M'\}$$

where $K = \sum_{p \in P} \iota(p)$. $\square$

Lemma 17. $\mathbf{SM}^{[\lambda]}(\lambda, t) \subseteq \mathbf{SM}^{[\lambda]}(\lambda, r)$.

Proof. Let $\mathcal{G} = (V, \Sigma, S, R, N, \gamma, M)$ be a state machine controlled grammar where $N = (P, T, F, \iota)$ and $M = \{\tau_1, \tau_2, \ldots, \tau_k\}$. The proof of the lemma consists of the following steps.

Step I. *First, we construct $|M|$ "copies" of the grammar $\mathcal{G}$.*

For each $\tau_i \in M$, $1 \le i \le k$, we define a state machine controlled grammar $\mathcal{G}_i = (V_i, \Sigma, S_i, R_i, N_i, \gamma_i, M_i)$ where

– the context-free components V_i and R_i are defined by

$$V_i = \{\psi_i(A) \mid A \in V\} \text{ and } R_i = \{\psi_i(A) \to \psi(\alpha) \mid A \to \alpha \in R\}$$

where $\psi_i : V \cup \Sigma \to V_i \cup \Sigma$, $1 \le i \le k$, are bijections and $\psi(a) = a$ for all $a \in \Sigma$;

– the sets of places, transitions and arcs of $N_i = (P_i, T_i, F_i, \iota_i)$ are, respectively, defined by

$$\begin{aligned} P_i =& \{\phi_i(p) \mid p \in P\}, \\ T_i =& \{\phi_i(t) \mid t \in T\}, \\ F_i =& \{(\phi_i(x), \phi_i(y)) \mid (x, y) \in F\} \end{aligned}$$

where $\phi_i : P \cup T \to P_i \cup T_i$, $1 \le i \le k$, are bijections;

– the initial marking ι_i is defined by $\iota_i(\phi_i(p) = \iota(p)$ for all $p \in P$;

– the labeling function γ_i is defined by $\gamma_i(\phi_i(t)) = \psi_i(A) \to \alpha \in R_i$ if $\gamma(t) = A \to \alpha \in R$ and $\gamma_i(\phi_i(t)) = \lambda$ if $\gamma(t) = \lambda$;

– the set of final markings $M_i = \{\mu_i\}$ where $\mu_i(\phi_i(p)) = \tau_i(p)$, $\tau_i \in M$.

Step II. *In order to control not to generate strings of the language of a grammar $\mathcal{G}_i$, $1 \le i \le k$, before reaching the final marking μ_i, we change each terminal symbol $a \in \Sigma$ to new nonterminal symbols $\overline{a}$, and add new places, transitions and arcs to N_i such a way that if and only if the final marking is reached, the nonterminal symbols $\overline{a}$ are changed back to a.*

Let $\{p_1, p_2, \ldots, p_n\} \subseteq P_i$ where $\mu_i(p_l) > 0$, $1 \le l \le n$ and let $k_l = \mu_i(p_l)$, $1 \le l \le n$. We introduce the following sets of new places, transitions and arcs:

$$\begin{aligned} \overline{P}_i =& \{q_{l,j} \mid 1 \le l \le n, 1 \le j \le k_l\}, \\ \overline{T}_i =& \{t'_{l,j} \mid 1 \le l \le n, 1 \le j \le k_l\} \\ & \cup \bigcup_{a \in \Sigma} \{t_{l,j,a} \mid 1 \le l \le n, 1 \le j \le k_l, a \in \Sigma\}, \\ \overline{F}_i =& \{(p_l, t'_{l,j}), (t'_{l,j}, q_{l,j}) \mid 1 \le l \le n, 1 \le j \le k_l\} \\ & \cup \bigcup_{a \in \Sigma} \{(q_{l,j}, t_{l,j,a}), (t_{l,j,a}, q_{l,j}) \mid 1 \le l \le n, 1 \le j \le k_l\}. \end{aligned}$$

We construct a free-choice net $N'_i = (P'_i, T'_i, F'_i, \iota'_i)$ where

$$P'_i = P_i \cup \overline{P}_i, T'_i = T_i \cup \overline{T}_i, F'_i = F_i \cup \overline{F}_i$$

and $\iota'_i(p) = \iota_i(p)$ for all $p \in P_i$ and $\iota'_i(p) = 0$ for all $p \in \overline{P}_i$.

A free-choice net controlled grammar $\mathcal{G}'_i = (V'_i, \Sigma, S_i, R'_i, N'_i, \gamma'_i, M'_i)$ is defined as follows:

- $V'_i = V_i \cup \bigcup_{a\in\Sigma} V_{l,j,a}$ where $V_{l,j,a} = \{a_{l,j} \mid 1 \le l \le n, 1 \le j \le k_l\}$ is the set of new nonterminal symbols;
- Let $\overline{R}_i = \{A \to \varphi_i(\alpha) \mid A \to \alpha \in R_i\}$ where the weght function $\varphi_i : V_i \cup \Sigma \to V_i \cup \{a_{1,1} \mid a \in \Sigma\}$ is bijection, defined by $\varphi(x) = x$ if $x \in V_i$ and $\varphi(x) = x_{1,1}$ if $x \in \Sigma$. We set for each $a \in \Sigma$,

$$\begin{aligned} R_{i,a} =& \{a_{l,j} \to a_{l,j+1} \mid 1 \le l \le n, 1 \le j \le k_l - 1\} \\ & \cup \{a_{l,k_l} \to a_{l+1,1} \mid 1 \le l \le n-2\} \\ & \cup \{a_{n,k_n} \to a\} \end{aligned}$$

  and define $R'_i = \overline{R}_i \cup \bigcup_{a\in\Sigma} R_{i,a}$;
- the labeling function γ' is defined by
  - $\gamma'_i(t) = A \to \varphi_i(\alpha) \in \overline{R}_i$ if $t \in T_i$ and $\gamma_i(t) = A \to \alpha \in R_i$,
  - $\gamma'_i(t'_{l,j}) = \lambda$ for $1 \le l \le n, 1 \le j \le k_l$,
  - $\gamma'_i(t_{l,j,a}) = a_{l,j} \to a_{l,j+1}$ for $1 \le l \le n, 1 \le j \le k_l - 1$,
  - $\gamma'_i(t_{l,k_l,a}) = a_{l,k_l} \to a_{l+1,1}$ for $1 \le l \le n-2$,
  - $\gamma'_i(t_{n,k_n,a}) = a_{n,k_n} \to a$;
- the set of final markings $M'_i = \{\mu'_i\}$ where $\mu'_i(p) = 0$ for all $p \in P_i$ and $\mu'_i(p) = 1$ for all $p \in \overline{P}_i$.

One can generate strings of the form $w_{1,1} \in \{a_{1,1} \mid a \in \Sigma\}^*$ under control of "N_i-part" of the net N'_i. In order to change nonterminal symbols of $\{a_{1,1} \mid a \in \Sigma\}$ to terminal symbols of Σ, $\sum_{p\in P_i} \mu'_i(p) = \sum_{p\in P_i} \iota'_i(p)$ number of tokens, i.e., all tokens have to been moved from the places of P_i to the places of $\overline{P}_i$.

Step III. *We define such a free-choice net controlled grammar $\mathcal{G}''$ that the language generated by this grammar is the union of languages generated by the grammars $\mathcal{G}'_i$, $1 \le i \le k$, constructed in Step II.*

We define a FC controlled grammar $\mathcal{G}'' = (V'', \Sigma, S'', R'', N'', \gamma'', M'')$ where

– the context-free components V'' and R'' are defined by

$$V'' = \bigcup_{i=1}^{k} V'_i \cup \{S''\}$$

where S'' is a new nonterminal symbol and

$$R'' = \bigcup_{i=1}^{k} R'_i \cup \{S'' \to S_i \mid 1 \le i \le k\};$$

– the free-choice net $N'' = (P'', T'', F'', \iota'')$ is defined by

$$P'' = \bigcup_{i=1}^{k} P'_i \cup \{p'\} \cup \{p'_i \mid 1 \le i \le k\},$$
$$T'' = \bigcup_{i=1}^{k} T'_i \cup \{t'_i \mid 1 \le i \le k\},$$
$$F'' = \bigcup_{i=1}^{k} F'_i \cup \{(p', t'_i), (t'_i, p'_i) \mid 1 \le i \le k\},$$

and the initial marking $\iota''(p) = \iota'_i(p)$ if $p \in P'_i$ and $\iota''(p') = 1$, $\iota''(p'_i) = 0$, $1 \le i \le k$;

– the labeling function γ'' is defined by $\gamma''(t) = \gamma'_i(t)$ if $t \in T'_i$ and $\gamma''(t'_i) = S'' \to S_i$, $1 \le i \le k$;

– for each final marking $\tau'' \in M''$, $\tau''(p) = \tau'_i(p)$ if $p \in P'_i$, $\iota''(p') = 0$, and $\iota''(p'_j) = 1$ for some $1 \le j \le k$ and $\iota''(p'_i) = 0$ for all $1 \le i \ne j \le k$;

It is not difficult to see that after one of the rules of $\{S'' \to S_i \mid 1 \le i \le k\}$ is applied, rules of only one of the grammars $\mathcal{G}'_i$, $1 \le i \le k$, can be used in a derivation of $\mathcal{G}''$, i.e., a string w is in $L(\mathcal{G}'')$ iff there is a derivation $S'' \Rightarrow S_i \Rightarrow^* w \in L(\mathcal{G}'_i)$, $1 \le i \le k$. On the other hand, we can initialize any derivation $S_i \Rightarrow^* w \in L(\mathcal{G}'_i)$ with the rule $S'' \to S_i$, $1 \le i \le k$, i.e., $w \in L(\mathcal{G}'')$. □

Lemma 18. $\mathbf{MAT}^{[\lambda]} \subseteq \mathbf{SM}^{[\lambda]}(f,t)$.

Proof. Let $G = (V, \Sigma, S, M)$ be a matrix grammar (with or without erasing rules) and

$$M = \{m_1, m_2, \ldots, m_n\} \text{ where } m_i = r_{i,1} r_{i,2} \cdots r_{i,k(i)}, 1 \le i \le n.$$

Without loss of generality we can assume that G is without repetitions. Let $R = \{r_{i,j} \mid 1 \le i \le n, 1 \le j \le k(i)\}$. We define an (f,t)-SM controlled grammar

$\mathcal{G} = (V, \Sigma, S, R, N, \gamma, \{\mu\})$ where the sets of places, transitions and arcs of the a SM $N = (P, T, F, \iota)$ are defined by

$$\begin{aligned} P =& \{p_0\} \cup \{p_{i,j} \mid 1 \le i \le n, 1 \le j \le k(i) - 1\}, \\ T =& \{t_{i,j} \mid 1 \le i \le n, 1 \le j \le k(i)\}, \\ F =& \{(p_0, t_{i,1}), (t_{i,k(i)}, p_0) \mid 1 \le i \le n\} \cup \{(p_{i,k(i)-1}, t_{i,k(i)}) \mid 1 \le i \le n\} \\ & \cup \{(t_{i,j}, p_{i,j}) \mid 1 \le i \le n, 1 \le j \le k(i) - 1\}. \end{aligned}$$

The initial marking is defined by $\iota(p_0) = 1$, and $\iota(p) = 0$ for all $P - \{p_0\}$.

The bijection $\gamma : T \to R$ is defined by

$$\gamma(t_{i,j}) = r_{i,j}, 1 \le i \le n, 1 \le j \le k(i)$$

and the final marking μ is the same as the initial marking ι.

Let

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

be a derivation in $\mathcal{G}$, where $m_{i_j} \in M$, $1 \le j \le l$, and

$$w_{j-1} \xRightarrow{m_{i_j}} w_j : \; w_{j-1} \xRightarrow{r_{i_j,1} r_{i_j,2} \cdots r_{i_j,k(i_j)}} w_j.$$

By the definition of γ, $\gamma(\sigma_j) = m_{i_j}$ where $\sigma_j = t_{i_j,1} t_{i_j,2} \cdots t_{i_j,k(i_j)}$ for all $1 \le j \le l$. Then the occurrence sequence of transitions $\iota \xrightarrow{\sigma_1 \sigma_2 \cdots \sigma_l} \iota$ is a successful for $\{\mu\}$. Therefore, $S \xRightarrow{m_{i_1} m_{i_2} \cdots m_{i_l}} w_l \in \Sigma^*$ is a derivation in $\mathcal{G}$.

The inverse inclusion can also be shown using the same arguments. □

Lemma 19. *For* $y \in \{r, g, t\}$, $\mathbf{SM}^{[\lambda]}(\lambda, y) \subseteq \mathbf{rC}^{[\lambda]}$.

Proof. Let $\mathcal{G} = (V, \Sigma, S, R, N, \gamma, M)$ be a SM controlled grammar (with or without erasing rules) where $N = (P, T, F, \iota)$. We construct a (deterministic) finite automaton $\mathcal{A}$ whose states are the markings of the net N (since the set of all reachable markings of a state machine is finite, it can be considered as a set of states) and there is an arc from state μ to state μ' with label t iff marking μ' is obtained from marking μ by firing transition t. The initial marking is considered as the initial state and the set of final markings M as a set of final states.

Formally, $\mathcal{A} = (M', T, \iota, \delta, M)$ where M' is the set of all reachable markings of the net N and the state-transition function $\delta : M' \times T \to M'$ is defined by $\delta(\mu, t) = \mu'$ iff $\mu \xrightarrow{t} \mu'$. It is not difficult to see that $\sigma = t_1 t_2 \cdots t_n \in L(\mathcal{A})$ iff σ is a successful occurrence sequence of transitions of N. Let $K = \{\gamma(\sigma) \mid \sigma \in L(\mathcal{A})\}$. Therefore, $L(\mathcal{G}) = L(G)$ where $G = (V, \Sigma, S, R, K)$ is a regularly controlled grammar. □

From Theorem 13 and Lemmas 18, 19, we have

Corollary 20. *For* $x \in \{f, -\lambda, \lambda\}$ *and* $y \in \{r, g, t\}$, $\mathbf{MAT}^{[\lambda]} = \mathbf{SM}^{[\lambda]}(x, y)$.

Lemma 21. $\mathbf{V}^{[\lambda]} \subseteq \mathbf{MG}^{[\lambda]}(f, t) \cap \mathbf{CN}^{[\lambda]}(f, t) \cap \mathbf{GSM}^{[\lambda]}(f, t)$.

Proof. Let $G = (V, \Sigma, S, M)$ be a vector grammar (with or without erasing rules) where

$$M = \{m_1, m_2, \ldots, m_n\} \text{ with } m_i = r_{i,1}r_{i,2} \cdots r_{i,k_i}, 1 \leq i \leq n.$$

Without loss of generality we can assume that G is without repetition. Let R be the set of all rules of M, i.e., $R = \{r_{i,j} \mid 1 \leq i \leq n, 1 \leq j \leq k_i\}$.

We define an (f, t)-ON controlled grammar $\mathcal{G} = (V, \Sigma, S, R, N, \gamma, M)$ with $N = (P, T, F, \iota)$ where

- the sets of places, transitions and arcs are, respectively,

$$\begin{aligned} P =& \{p_{i,j} \mid 1 \leq i \leq n, 1 \leq j \leq k_i - 1\}, \\ T =& \{t_{i,j} \mid 1 \leq i \leq n, 1 \leq j \leq k_i\}, \\ F =& \{(t_{i,j}, p_{i,j}) \mid 1 \leq i \leq n, 1 \leq j \leq k_i - 1\} \\ & \cup \{(p_{i,j}, t_{i,j+1}) \mid 1 \leq i \leq n, 1 \leq j \leq k_i - 1\}; \end{aligned}$$

- the initial marking is defined by $\iota(p) = 0$ for all $p \in P$;
- the labeling function $\gamma : T \to R$ is a bijection defined by $\gamma(t_{i,j}) = r_{i,j}$, $1 \leq i \leq n, 1 \leq j \leq k_i$ and the set of final markings $M = \{\mu\}$ where $\mu = \iota$.

By construction, N satisfies the structural properties of a marked graph, a casual net and generalized state machine, which consists of disjoint paths $\rho_i = t_{i,1}p_{i,1}t_{i,2}p_{i,2} \cdots p_{i,k_i-1}t_{i,k_i}$ where $\gamma(tr(\rho_i)) = m_i$, $1 \leq i \leq n$, and the firing of the transitions of a path ρ_i simulates the application of the rules of the matrix m_i. Moreover, a derivation $S \xRightarrow{r_1r_2\cdots r_l} w \in \Sigma^*$ in the context-free grammar (V, Σ, S, R) is a derivation in G if $r_1r_2 \cdots r_l$ is a shuffle of some matrices $m_{i_1}, m_{i_2}, \ldots, m_{i_k} \in M$, and a derivation in $\mathcal{G}$ if $t_1t_2 \cdots t_l = \gamma^{-1}(r_1r_2 \cdots r_l)$ is a shuffle of $tr(\rho_{i_1}), tr(\rho_{i_2}), \ldots, tr(\rho_{i_k})$ where $\gamma(tr(\rho_{i_j})) = m_{i_j}$, $1 \leq j \leq k$. Thus, it is easy to see that each derivation in G can be simulated by a derivation in $\mathcal{G}$ and vise versa. □

Lemma 22. $\mathbf{sMAT}^{[\lambda]} \subseteq \mathbf{SM}^{[\lambda]}(f, t) \cap \mathbf{MG}^{[\lambda]}(f, t)$.

Proof. For each semi-matrix grammar we construct an (f, t)-ordinary net controlled grammar where the net consists of disjoint cycles which correspond to the matrices of the semi-matrix grammar and the firing of the transitions of a cycle in the net simulates the application of the rules of the corresponding matrix in each derivation in the grammar.

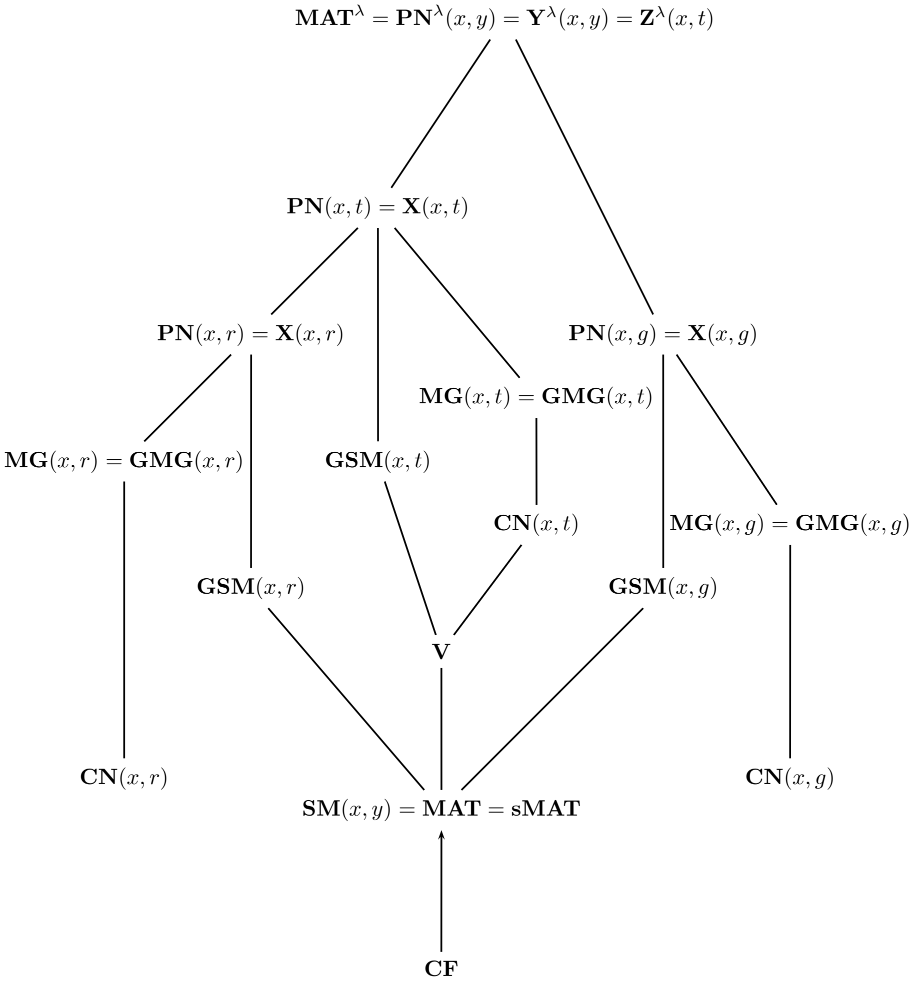

Figure 4: The hierarchy of language families generated by Petri net controlled grammars

Let $G = (V, \Sigma, S, M)$ be a semi-matrix grammar (with or without erasing rules) where

$$M = \{m_1, m_2, \ldots, m_n\}, m_i = r_{i,1} r_{i,2} \cdots r_{i,k_i}, 1 \le i \le n.$$

Without loss of generality we can assume that G is without repetition. Let R be the set of all rules of M, i.e., $R = \{r_{i,j} \mid 1 \le i \le n, 1 \le j \le k_i\}$.

We define an (f,t)-ON controlled grammar $\mathcal{G} = (V, \Sigma, S, R, N, \gamma, M)$ with $N = (P, T, F, \iota)$ where

– the sets of places, transitions and arcs are, respectively,

$$P = \{p_{i,j} \mid 1 \leq i \leq n, 1 \leq j \leq k_i\},$$
$$T = \{t_{i,j} \mid 1 \leq i \leq n, 1 \leq j \leq k_i\},$$
$$F = \{(t_{i,j}, p_{i,j}) \mid 1 \leq i \leq n, 1 \leq j \leq k_i\} \cup \{(p_{i,k_i}, t_{i,1}) \mid 1 \leq i \leq n\};$$

– the initial marking is $\iota(p_{i,k_i}) = 1$, $1 \leq i \leq n$ and $\iota(p) = 0$ for all $p \in P - \{p_{i,k_i} \mid 1 \leq i \leq n\}$;

– the labeling function $\gamma : T \to R$ is a bijection where $\gamma(t_{i,j}) = r_{i,j}$, $1 \leq i \leq n, 1 \leq j \leq k_i$;

– a set of final markings is $M = \{\mu\}$ where $\mu = \iota$.

By construction, N is a state machine and also a marked graph which consists of disjoint cycles $\rho_i = p_{i,1}t_{i,1}p_{i,2}t_{i,2} \cdots p_{i,k_i}t_{i,k_i}p_{i,1}$ and $\gamma(tr(\rho_i)) = m_i$, $1 \leq i \leq n$. Using the same arguments of the proof of Lemma 21, one can easily show that $L(G) = L(\mathcal{G})$. □

Now we summarize our results in the following theorem.

Theorem 23. *The relations in Figure 4 hold where* $x \in \{f, -\lambda, \lambda\}$, $y \in \{r, g, t\}$, $\mathbf{X} \in \{\mathbf{FC}, \mathbf{EFC}, \mathbf{AC}, \mathbf{ON}\}$, $\mathbf{Y} \in \{\mathbf{SM}, \mathbf{GSM}, \mathbf{FC}, \mathbf{EFC}, \mathbf{AC}, \mathbf{ON}\}$ *and* $\mathbf{Z} \in \{\mathbf{MG}, \mathbf{GMG}, \mathbf{CN}\}$*; the lines (arrow) denote (proper) inclusions of the lower families into the upper families.*

References

[Baumgarten 1990] Baumgarten, B.: "Petri-Netze. Grundlagen und Anwendungen"; Wissensschaftverlag, Mannheim.

[Dassow 1988] Dassow, J.: "Subregularly Controlled Derivations: Context-Free Case"; Rostock. Math. Kolloq. **34** (1988) 61–70.

[Dassow and Păun 1989] Dassow, J., Păun, Gh.: "Regulated rewriting in formal language theory"; Springer-Verlag, Berlin.

[Dassow and Truthe 2008] Dassow, J., Truthe, B.: "Subregularly Tree Controlled Grammars and Languages"; Proc. the 12^{th} International Conference AFL 2008, E. Csuhaj-Varjú, Z. Esik (eds.), Balatonfured, Hungary (2008) 158–169.

[Dassow and Turaev 2008] Dassow, J., Turaev, S.: "Arbitrary Petri Net Controlled Grammars"; Proc. the 2^{nd} International Workshop "Non-Classical Formal Languages in Linguistics"–ForLing 2008, G.Bel-Enguix, M.D. Jiménez-López (eds.), Tarragona, Spain (2008) 27–39.

[Dassow and Turaev 2009] Dassow, J., Turaev, S.: "Grammars controlled by special Petri nets"; Language and Automata Theory and Applications. Third International Conference LATA 2009 Carlos Martín-Vide, Adrian Horia Dediu, Mihai Ionescu (eds.), LNCS **5457**, Springer (2009) 326–337.

[Desel and Esparsa 1995] Desel, J., Esparsa, J.: "Free-Choice Petri Nets"; Cambridge Tracts in Theoretical Computer Science **10**, Cambridge University Press.

[Graham et al. 1996] Graham, R.L., Groetschel M., Lovász, L. (eds.): "Handbook of Combinatorics"; Vol. **I–II**, Elsevier, North-Holland, Amsterdam and MIT Press, Cambridge, Mass.

[Rozenberg and Salomaa 1997] Rozenberg, G., Salomaa, A. (eds.): "Handbook of Formal Languages"; Vol. **I–III**, Springer, Berlin.

[Ibarra 1970] Ibarra, O.H.: "Simple matrix grammars"; Inform. Control **17** (1970) 359–394.

[Reisig and Rozenberg 1998] Reisig, W., Rozenberg, G. (eds.): "Lectures on Petri nets I: Basic models"; Lect. Notes Comp. Sci. **1491**, Springer, Berlin.

[Turaev 2006] Turaev, S.: "Semi-matrix grammars"; Proc. 2^{nd} Doctoral Workshop on Mathematical and Engineering Methods in Computer Science–MEMICS-2006 (2006) 245–252.