Triangle: Language, Literature, Computation, n. 8, 2012
Publicacions Universitat Rovira i Virgili · ISSN: 2013-939X
https://revistes.urv.cat/index.php/triangle

Towards a Unified Theory of Regulated Grammars

Sherzod Turaev

Research Group on Mathematical Linguistics
Universitat Rovira i Virgili
Tarragona, Spain
E-mail: sherzod.turaev@urv.cat

Summary. This paper shows that Petri net control mechanisms in grammars can be considered to be a unifying approach to different controls and characterizations which are used in well-known families of languages such as matrix and vector languages.

1 Introduction

It is well-known that context-free languages are not sufficient to model phenomena which are known in natural and programming languages. On the other hand, context-sensitive languages are much more powerful and have negative features (e.g. among decision problems only membership is decidable and all existing algorithms for this problem have exponential complexities). Therefore, the languages of interest are those that use context-free production rules and have larger generative capacities. Many grammars have a context-free base and a mechanism that controls the sequences of rules in a derivation [1, 3, 4, 5, 8, 10, 14, 15].

A context-free grammar and its derivation process can be described by a Petri net, where places correspond to nonterminals and terminals, transitions are the counterpart of productions, and tokens reflect the occurrences of symbols in the

sentential form, and there is a one-to-one correspondence between the application of (sequences of) rules and the firing of (sequences of) transitions. Therefore, controlling the sequences of rules in a derivation by adding some features to the associated Petri net is natural and straightforward.

In [2] it has been shown that by adding some places and arcs which satisfy some structural requirements, *random context* languages can be generated. In this paper we show that *vector*, *semi-matrix* and *matrix* languages can also be generated by grammars controlled by Petri nets. Thus, control by Petri nets can be considered as a unifying approach to different types of controls and characterizations used in well-known language families.

The paper is organized as follows. In Section 2 we give some notions and definitions from the theories of formal languages and the Petri nets needed in the sequel. We also introduce the Petri net associated with a context-free grammar. In Section 3 we construct the Petri net control mechanisms, define the corresponding grammars, and give some examples. Section 4 contains the results on the generative powers of families of languages generated by Petri net controlled grammars.

2 Definitions

The reader is assumed to be familiar with the basic notions of formal language theory and Petri net theory contained in [4, 6, 11, 7, 9, 12, 13].

2.1 Grammars

Let $\Sigma = \{a_1, a_2, \ldots, a_k\}$ be an alphabet. A *string* over Σ is a sequence of symbols from the alphabet. 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. A word $u = u_1u_2\cdots u_n,\ u_1, u_2, \ldots, u_n \in \Sigma$ is a *scattered substring* of $v \in \Sigma^*$, denoted by $u \ll v$, if $v = v_1u_1v_2\cdots u_nv_{n+1}$ for some $v_1, v_2, \ldots, v_{n+1} \in \Sigma^*$ with $n \geq 1$. We write $u \not\ll v$ if u is not a scattered substring of v. The *length* of a string w is denoted by $|w|$, and the number of occurrences of a symbol a in a string w by $|w|_a$. 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*.

For $u, v \in \Sigma^*$ their *shuffle* is defined by $\mathrm{sh}(u, v) = u_1v_1u_2v_2\cdots u_nv_n$ where $u = u_1u_2\cdots u_n$ and $v = v_1v_2\cdots v_n,\ u_i, v_i \in \Sigma^*,\ 1 \leq i \leq n$. A shuffle of u and v is *proper* if neither $\mathrm{sh}(u, v) = uv$ nor $\mathrm{sh}(u, v) = vu$. A proper shuffle is denoted by $\mathrm{sh}^+(u, v)$. The operation shuffle can be generalized for more than two strings by $\mathrm{sh}(u_1, u_2, \ldots, u_n) = \mathrm{sh}(\mathrm{sh}(u_1, u_2, \ldots, u_{n-1}), u_n)$, $n \geq 3$. A shuffle of $u_1, u_2, \ldots, u_n$, $n \geq 2$, is *semi*, denoted by $\mathrm{ssh}(u_1, u_2, \ldots, u_n)$, if $u_i = u_j$, for some $1 \leq i < j \leq n$, then $\mathrm{sh}^+(u_i, u_j) \not\ll \mathrm{sh}(u_1, u_2, \ldots, u_n)$. In words, a semi-shuffle string does not contain self-shuffled scattered substrings.

A *context-free grammar* is a quadruple $G = (V, \Sigma, S, R)$ where V and Σ are the disjoint finite sets of *nonterminal* and *terminal* symbols, respectively, $S \in V$ is the *start* symbol and the finite set $R \subset V \times (V \cup \Sigma)^*$ is the 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 word $x \in (V \cup \Sigma)^+$ *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 $x = x_1 A x_2$ and $y = x_1 \alpha x_2$. The reflexive and transitive closure of $\Rightarrow$ is denoted by $\Rightarrow^*$. A derivation using the sequence of rules $\pi = r_1 r_2 \cdots r_n$ is denoted by $\overset{\pi}{\Longrightarrow}$ or $\overset{r_1 r_2 \cdots r_n}{\Longrightarrow}$. The *language* generated by G is defined by $L(G) = \{w \in \Sigma^* \mid S \Rightarrow^* w\}$.

A *matrix grammar* is a quintuple $G = (V,\ \Sigma,\ S,\ M)$ 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. The language generated by the grammar G consists of all strings $w \in \Sigma^*$ such that there is a derivation $S \overset{r_1 r_2 \cdots r_n}{\Longrightarrow} w$ where $r_1 r_2 \cdots r_n = m_{j_1} m_{j_2} \cdots m_{j_k}$ for some $m_{j_1}, m_{j_2}, \ldots, m_{j_k} \in M$.

A *vector grammar* is a quintuple $G = (V,\ \Sigma,\ S,\ M)$ 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 $S \overset{r_1 r_2 \cdots r_n}{\Longrightarrow} w$ where $r_1 r_2 \cdots r_n = \mathrm{sh}(m_{j_1}, m_{j_2}, \ldots, m_{j_k})$ for some $m_{j_1}, m_{j_2}, \ldots, m_{j_k} \in M$.

A *semi-matrix grammar* is a quintuple $G = (V,\ \Sigma,\ S,\ M)$ 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 $S \overset{r_1 r_2 \cdots r_n}{\Longrightarrow} w$ where $r_1 r_2 \cdots r_n = \mathrm{ssh}(m_{j_1}, m_{j_2}, \ldots, m_{j_k})$ for some $m_{j_1}, m_{j_2}, \ldots, m_{j_k} \in M$.

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

For each matrix grammar, an equivalent matrix grammar without repetitions can be constructed by adding chain rules.

The families of languages generated by matrix, vector and semi-matrix grammars (with erasing rules) are denoted by **MAT**, **V** and **sMAT** (**MAT**$^\lambda$, **V**$^\lambda$ and **sMAT**$^\lambda$), 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 the set of *directed arcs*, $\varphi : F \to \{1, 2, \ldots\}$ is a *weight function*.

A Petri net can be represented by a bipartite directed graph with the node set $P \cup T$ where places are drawn as *circles*, transitions as *boxes* and arcs as *arrows* 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. ${}^\bullet x = \{y : (y,x) \in F\}$ and $x^\bullet = \{y : (x,y) \in F\}$ are called *pre-* and *post-sets* of $x \in P \cup T$, respectively. The elements of ${}^\bullet t\,({}^\bullet p)$ are called *input* places (transitions) and the elements of $t^\bullet\,(p^\bullet)$ are called *output* places (transitions) of $t\,(p)$.

A transition $t \in T$ is *enabled* by 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' defined for each place $p \in P$ by $\mu'(p) = \mu(p) - \varphi(p,t) + \varphi(t,p)$. A finite sequence $t_1, t_2, \ldots, t_k$ 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 initial marking ι.

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 $1 \leq i \leq n-1$. $tr(\rho)$ denotes the sequence of transitions in a path ρ. A path $\rho = x_1 x_2 \cdots x_n$ is a *cycle* if $n = 1$ or $x_1 = x_n$ and $n \geq 2$. A path $\rho = x_1 x_2 \cdots x_n$ is a *chain* if $n = 1$ or $x_1 \neq x_n$ and $n \geq 2$. We denote by P_ρ, T_ρ, F_ρ the sets of places, transitions, and arcs of a path ρ. Pathes ρ_1 and ρ_2 are called *disjoint* if $P_{\rho_1} \cap P_{\rho_2} = \emptyset$ and $T_{\rho_1} \cap T_{\rho_2} = \emptyset$.

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* if it is enabled at the initial marking ι and finished at a final marking τ of M.

2.3 cf Petri nets

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

Definition 1. *Let $G = (V, \Sigma, S, R)$ be a context-free grammar. The cf Petri net with respect to the grammar G is a (labeled marked) Petri net $N = (P, T, F, \varphi, \iota, \beta, \gamma)$ where*

- *labeling functions $\beta : P \to (V \cup \Sigma)$ and $\gamma : T \to R$ are bijections;*
- *$(p,t) \in F$ iff $\gamma(t) = A \to w \in R$ where $\beta(p) = A$ and $\varphi(p,t) = 1$;*
- *$(t,p) \in F$ iff $\gamma(t) = A \to w \in R$ where $\beta(p) = x$, $|w|_x > 0$ and $\varphi(t,p) = |w|_x$;*
- *$\iota(\beta^{-1}(S)) = 1$ and $\iota(\beta^{-1}(x)) = 0$ for all $x \in (V \cup \Sigma) \setminus \{S\}$.*

Example 1. Let G_1 be a context-free grammar with the rules: $S \to AB, A \to aA|bA|a|b, B \to aB|bB|a|b$. (the other components of the grammar can be seen

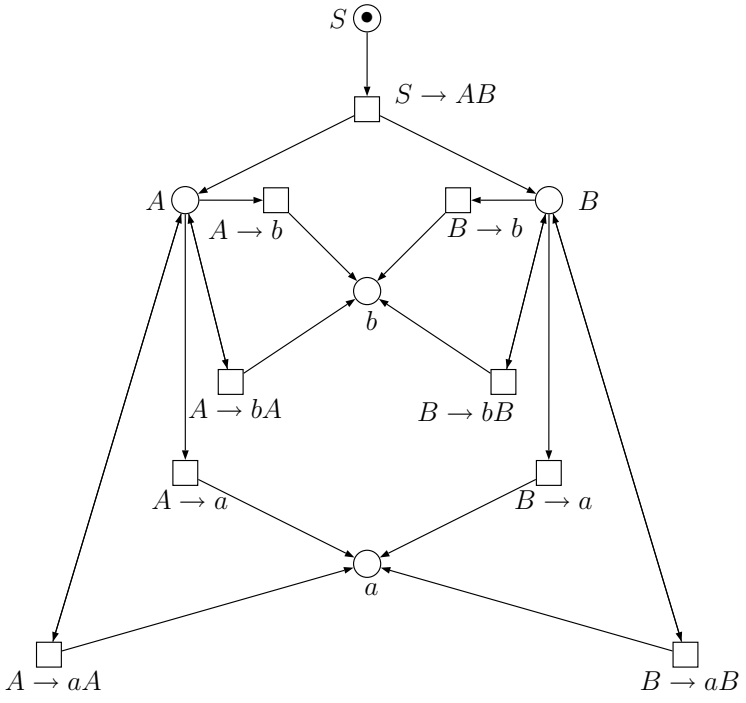

Fig. 1. A cf Petri net N.

from these rules). Figure 1 illustrates the corresponding cf Petri net N. Obviously, $L(G_1) = \{a, b\}^+$.

The following proposition, which directly follows from the definition, shows the similarity between terminal derivations in a context-free grammar and successful occurrences in the corresponding cf Petri net.

Proposition 1. *Let $N = (P, T, F, \varphi, \iota, \beta, \gamma)$ be the cf Petri net with respect to a context-free grammar $G = (V, \Sigma, S, R)$. Then $S \overset{r_1 r_2 \cdots r_n}{\Longrightarrow} w \in \Sigma^*$ is a derivation in G iff $\iota \overset{t_1 t_2 \cdots t_n}{\longrightarrow} \tau$ is an occurrence sequence of transitions in N such that $\gamma(t_1 t_2 \cdots t_n) = r_1 r_2 \cdots r_n$ and $\tau(\beta^{-1}(x)) = 0$ for all $x \in V$.*

3 Petri Net Controlled Grammars

In this section we construct Petri net control mechanisms and define Petri net controlled grammars.

3.1 Chain, cyclic and supervised cyclic controls

We add new places and arcs, called *control places* and *arcs*, to a cf Petri net in such a way that the new places with input and output transitions make up chains or cycles. Let $N = (P, T, F, \varphi, \iota, \beta, \gamma)$ be a cf Petri net with respect to a context-free grammar $G = (V, \Sigma, S, R)$. Let $P_V = \{p \mid p = \beta^{-1}(x), x \in V\}$ and $P_\Sigma = \{p \mid p = \beta^{-1}(x), x \in \Sigma\}$. We use these notations in all definitions hereinafter.

Chain control. Let $\Pi = \{\rho_1, \rho_2, \ldots, \rho_n\}$ be a set of disjoint chains where for all $\rho \in \Pi$, $\rho = t_1 p_1 t_2 p_2 \cdots p_{k-1} t_k$ where $T_\rho = \{t_1, t_2, \ldots, t_k\} \subseteq T$ and $P_\rho = \{p_1, p_2, \ldots, p_{k-1}\}$, $F_\rho = \{(t_i, p_i) \mid 1 \le i \le k-1\} \cup \{(p_i, t_{i+1}) \mid 1 \le i \le k-1\}$ are the set of new places and arcs, respectively. Let

$$P_\Pi = \bigcup_{\rho \in \Pi} P_\rho, \quad T_\Pi = \bigcup_{\rho \in \Pi} T_\rho, \quad F_\Pi = \bigcup_{\rho \in \Pi} F_\rho.$$

Without loss of generality we assume that $T = T_\Pi$ (since each transition of T can be considered as a chain or a cycle).

Definition 2. A chain-cf Petri net *is a system* $\hat{N} = (\hat{P}, T, \hat{F}, \hat{\varphi}, \hat{\iota}, \hat{\beta}, \gamma, M)$ *where*

- $\hat{P} = P \cup P_\Pi$ *and* $\hat{F} = F \cup F_\Pi$,
- *the weight function* $\hat{\varphi}$ *is defined by*

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

- *the initial marking* $\hat{\iota}$ *is defined by*

$$\hat{\iota}(p) = \begin{cases} \iota(p) & \text{if } p \in P \\ 0 & \text{if } p \in P_\Pi, \end{cases}$$

- *the labeling function* $\hat{\beta} : \hat{P} \to V \cup \Sigma \cup \{\lambda\}$ *is defined by*

$$\hat{\beta}(p) = \begin{cases} \beta(p) & \text{if } p \in P \\ \lambda & \text{if } p \in P_\Pi, \end{cases}$$

- M *is a set of final markings where for each* $\tau \in M$, $\tau(p) = 0$ *for all* $p \in \hat{P} - P_\Sigma$.

Example 2. Figure 2 illustrates a chain-cf Petri net $\hat{N}$ which is based on the context-free grammar introduced in Example 1.

According to the construction of a chain-cf Petri net $\hat{N}$, the firing of its transitions follows the rules below:

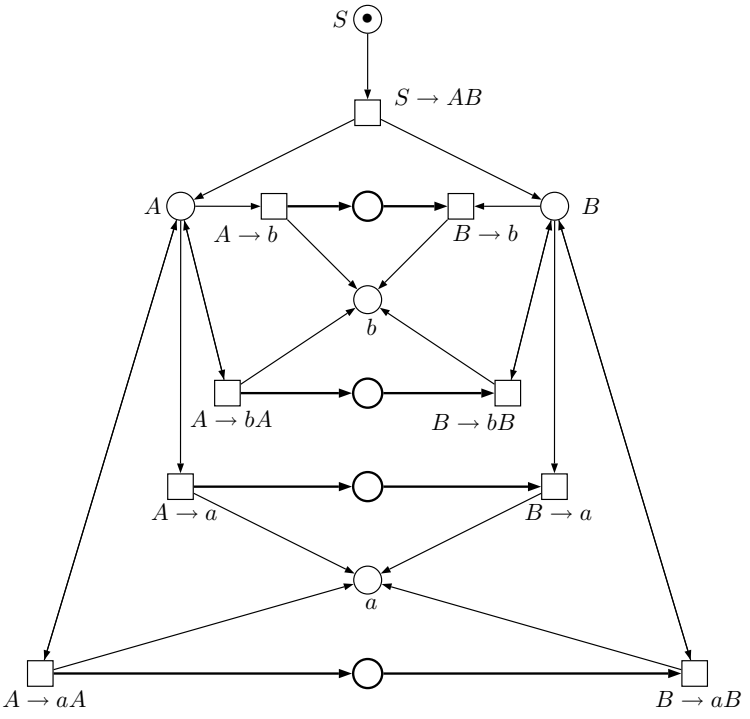

Fig. 2. A chain-cf Petri net $\hat{N}$.

- for each chain $\rho = t_1 p_1 t_2 p_2 \cdots t_{n-1} p_{n-1} t_n \in \Pi$, the firing of its transitions starts with the firing of transition t_1, and t_1 can occur any time (when it is enabled), it does not depend on the next transitions in the sequence $tr(\rho)$;
- transition t_i can occur if t_{i-1}, $1 < i \leq n$, has occurred, and for each occurrence of sequence σ of transitions of $\hat{N}$, $|\sigma|_{t_{i-1}} \geq |\sigma|_{t_i}$. Moreover, if σ is successful then $|\sigma|_{t_1} = |\sigma|_{t_2} = \cdots = |\sigma|_{t_n}$;
- the transitions of disjoint chains can occur concurrently since the places of a chain merely control the firing of the transitions of the chain.

Consequently,

Proposition 2. *For each successful occurrence sequence transitions $\sigma = t_1 t_2 \cdots t_n$ of a chain-cf Petri net $\hat{N}$, $\sigma = \mathrm{sh}(tr(\rho_{i_1}), tr(\rho_{i_2}), \ldots, tr(\rho_{i_s}))$ for some (not necessarily different) chains $\rho_{i_1}, \rho_{i_2}, \ldots, \rho_{i_s} \in \Pi$, $s \geq 1$.*

Cyclic control. Let $\Pi = \{\rho_1, \rho_2, \ldots, \rho_n\}$ be a set of disjoint cycles where for all $\rho \in \Pi$, $\rho = p_1 t_1 p_2 t_2 \cdots p_k t_k p_1$ where $T_\rho = \{t_1, t_2, \ldots, t_k\} \subseteq T$ and $P_\rho = \{p_1, p_2, \ldots, p_k\}$, $F_\rho = \{(p_i, t_i) \mid 1 \leq i \leq k\} \cup \{(t_i, p_{i+1}) \mid 1 \leq i \leq k-1\} \cup \{(t_k, p_1)\}$ are the set of new places and arcs, respectively. Let

$$P_\Pi = \bigcup_{\rho \in \Pi} P_\rho, \quad T_\Pi = \bigcup_{\rho \in \Pi} T_\rho, \quad F_\Pi = \bigcup_{\rho \in \Pi} F_\rho.$$

Definition 3. A cyclic-cf Petri net *is a system* $\check{N} = (\check{P}, T, \check{F}, \check{\varphi}, \check{\iota}, \check{\beta}, \gamma, M)$ *where*

- $\hat{P} = P \cup P_\Pi$ *and* $\hat{F} = F \cup F_\Pi$,
- *the weight function* $\check{\varphi}$ *is defined by*

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

- *the initial marking* $\hat{\iota}$ *is defined by*

$$\check{\iota}(p) = \begin{cases} \iota(p) & \text{if } p \in P \\ 1 & \text{if } p = p_1 \in P_\rho, \rho \in \Pi \\ 0 & \text{if } p \in P_\rho - \{p_1\}, \rho \in \Pi, \end{cases}$$

- *the labeling function* $\check{\beta} : \check{P} \to V \cup \Sigma \cup \{\lambda\}$ *is defined by*

$$\check{\beta}(p) = \begin{cases} \beta(p) & \text{if } p \in P \\ \lambda & \text{if } p \in P_\Pi, \end{cases}$$

- M *is a set of final markings where for each* $\tau \in M$,

$$\check{\tau}(p) = \begin{cases} 0 & \text{if } p \in P_V \\ 1 & \text{if } p = p_1 \in P_\rho, \rho \in \Pi \\ 0 & \text{if } p \in P_\rho - \{p_1\}, \rho \in \Pi. \end{cases}$$

Example 3. Figure 3 illustrates a cyclic-cf Petri net $\check{N}$ which is based on the context-free grammar given in Example 1.

According to the construction of a cyclic-cf Petri net $\check{N}$, the firing of its transitions follows the rules below:

- for each cycle $\rho = p_1 t_1 p_2 t_2 \cdots p_n t_n p_1 \in \Pi$, the firing of its transitions starts with the firing of transition t_1, and t_1 can occur next when transitions $t_2, t_3, \ldots, t_n$ have occurred in the defined order;
- for each occurrence of sequence σ of transitions of $\check{N}$, $|\sigma|_{t_{i-1}} = |\sigma|_{t_i}$ or $|\sigma|_{t_{i-1}} = |\sigma|_{t_i} + 1$, and if σ is successful then $|\sigma|_{t_1} = |\sigma|_{t_2} = \cdots = |\sigma|_{t_n}$;

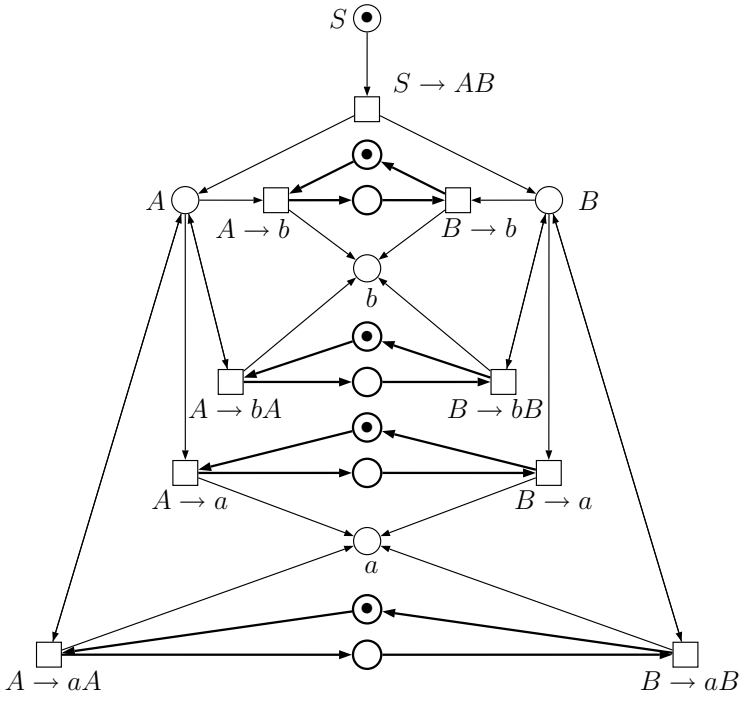

Fig. 3. A cyclic-cf Petri net $\check{N}$.

- the transitions of disjoint cycles can occur concurrently since the places of a cycle merely control the firing of the transitions of the cycle while the occurrence of the transitions of the same cycle a second time can start when the first occurrence of the transitions finishes.

Consequently,

Proposition 3. *For each successful occurrence sequence transitions* $\sigma = t_1 t_2 \cdots t_n$ *of a cyclic-cf Petri net* $\check{N}$, $\sigma = \mathrm{ssh}(tr(\rho_{i_1}), tr(\rho_{i_2}), \ldots, tr(\rho_{i_s}))$ *for some (not necessarily different) chains* $\rho_{i_1}, \rho_{i_2}, \ldots, \rho_{i_s} \in \Pi$, $s \geq 1$.

Supervised cyclic control. Let $\Pi = \{\rho_1, \rho_2, \ldots, \rho_n\}$ be a set of cycles such that $P_{\rho_1} \cap P_{\rho_2} \cap \cdots \cap P_{\rho_n} = \{p_0\}$ where for all $\rho \in \Pi$, $\rho = p_0 t_1 p_1 t_2 \cdots p_{k-1} t_k p_0$ where $T_\rho = \{t_1, t_2, \ldots, t_k\} \subseteq T$ and $P_\rho = \{p_0, p_1, \ldots, p_{k-1}\}$, $F_\rho = \{(p_i, t_{i+1}) \mid 0 \leq i \leq k-1\} \cup \{(t_i, p_i) \mid 1 \leq i \leq k-1\} \cup \{(t_k, p_0)\}$ are the set of new places and arcs, respectively. Let

$$P_{\Pi} = \bigcup_{\rho \in \Pi} P_{\rho}, \quad T_{\Pi} = \bigcup_{\rho \in \Pi} T_{\rho}, \quad F_{\Pi} = \bigcup_{\rho \in \Pi} F_{\rho}.$$

Definition 4. *A system* $\tilde{N} = (\tilde{P}, T, \tilde{F}, \tilde{\varphi}, \tilde{\iota}, \tilde{\beta}, \gamma, M)$ *is called* a supervised-cyclic-cf Petri net *where*

- $\tilde{P} = P \cup P_{\Pi}$ *and* $\tilde{F} = F \cup F_{\Pi}$,
- *the weight function* $\tilde{\varphi}$ *is defined by*

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

- *the initial marking* $\tilde{\iota}$ *is defined by*

$$\tilde{\iota}(p) = \begin{cases} \iota(p) & \text{if } p \in P \\ 1 & \text{if } p = p_0 \\ 0 & \text{if } p \in P_{\rho} - \{p_0\}, \rho \in \Pi, \end{cases}$$

- *the labeling function* $\tilde{\beta} : \tilde{P} \to V \cup \Sigma \cup \{\lambda\}$ *is defined by*

$$\tilde{\beta}(p) = \begin{cases} \beta(p) & \text{if } p \in P \\ \lambda & \text{if } p \in P_{\Pi}, \end{cases}$$

- M *is a set of final markings where for each* $\tau \in M$,

$$\tilde{\tau}(p) = \begin{cases} 0 & \text{if } p \in P_V \\ 1 & \text{if } p = p_0 \\ 0 & \text{if } p \in P_{\rho} - \{p_0\}, \rho \in \Pi. \end{cases}$$

Example 4. Figure 4 illustrates a supervised-cyclic-cf Petri net $\tilde{N}$ which is based on the context-free grammar given in Example 1.

According to the construction of a supervised-cyclic-cf Petri net $\tilde{N}$, the firing of its transitions follows the rules below:

- for each cycle $\rho = p_0 t_1 p_1 t_2 \cdots p_{n_1} t_n p_0 \in \Pi$, the firing of its transitions starts with the firing of transition t_1, and t_1 can occur next when transitions $t_2, t_3, \ldots, t_n$ have occurred in the defined order;
- for each successful occurrence of sequence σ of transitions of $\tilde{N}$, $|\sigma|_{t_1} = |\sigma|_{t_2} = \cdots = |\sigma|_{t_n}$;
- if the occurrence of the transitions of a cycle ρ_1 has started then the occurrence of the transitions of a second (not necessarily different) cycle ρ_2 can start when all transitions of ρ_1 have fired.

Consequently,

Proposition 4. *For each successful occurrence sequence transitions* $\sigma = t_1 t_2 \cdots t_n$ *of a supervised-cyclic-cf Petri net* $\tilde{N}$, $\sigma = tr(\rho_{i_1}) tr(\rho_{i_2}) \cdots tr(\rho_{i_s})$ *for some (not necessarily different) cycles* $\rho_{i_1}, \rho_{i_2}, \ldots, \rho_{i_s} \in \Pi$, $s \geq 1$.

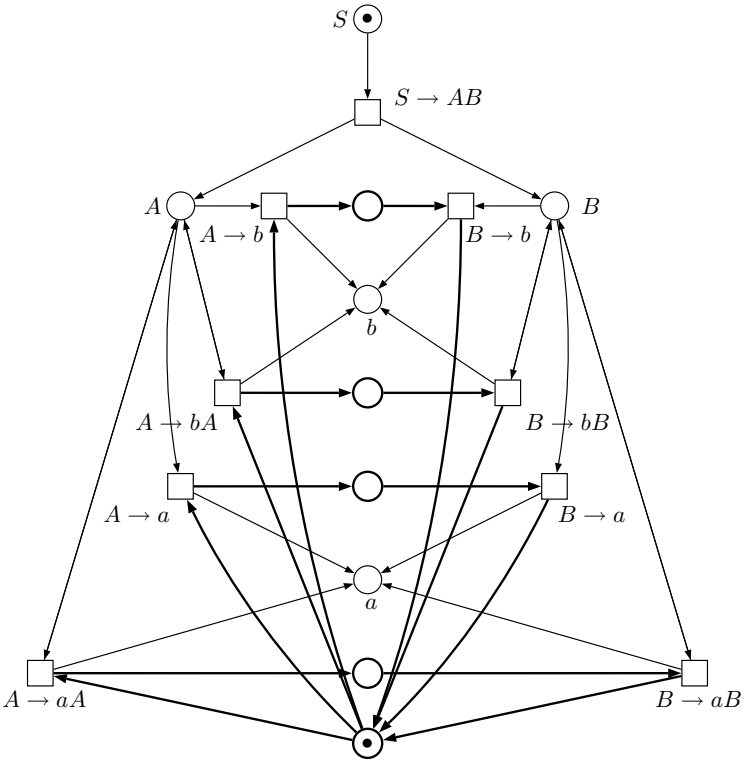

Fig. 4. A supervised-cyclic-cf Petri net $\tilde{N}$.

3.2 Grammars and examples

In this section we define the grammars controlled by the Petri nets introduced in the section above.

Definition 5. *A* chain- (respectively, cyclic-, supervised-cyclic-) cf Petri net controlled grammar *(in short a ch- (c-, sc-) PN controlled grammar) is a tuple* $G = (V, \Sigma, S, R, \hat{N})$ *(*$G = (V, \Sigma, S, R, \tilde{N})$*,* $G = (V, \Sigma, S, R, \tilde{N})$*) where* V*,* Σ*,* S*,* R *are defined as for a context-free grammar and* $\hat{N}$ *(*$\tilde{N}$*,* $\tilde{N}$*) is a chain- (cyclic-, supervised-cyclic-) cf Petri net with respect to the context-free grammar* (V, Σ, S, R)*.*

Definition 6. *The language generated by a ch- (c-, sc-) Petri net controlled grammar* G*, denoted by* $L(G)$*, consists of all strings* $w \in \Sigma^*$ *such that there is a derivation* $S \overset{r_1 r_2 \cdots r_k}{\Longrightarrow} w \in \Sigma^*$ *and a successful occurrence sequence of transitions* $\nu = t_1 t_2 \cdots t_k$

of $\hat{N}$ *(*$\check{N}$*,* $\tilde{N}$*) such that* $r_1 r_2 \cdots r_k = \hat{\gamma}(t_1 t_2 \cdots t_k)$ *(*$r_1 r_2 \cdots r_k = \check{\gamma}(t_1 t_2 \cdots t_k)$*,* $r_1 r_2 \cdots r_k = \tilde{\gamma}(t_1 t_2 \cdots t_k)$*, respectively).*

Example 5. Let $G_2 = (V, \Sigma, S, R, \hat{N})$ be a ch-Petri net controlled grammar where components V, Σ, S, R are defined as for the context-free grammar G_2 in Example 1, and $\hat{N}$ is the chain-cf Petri net in Figure 2.

After transition $t_0 = \hat{\beta}^{-1}(S \to AB)$ fires, transitions $t_1 = \hat{\beta}^{-1}(A \to aA)$, $t_3 = \hat{\beta}^{-1}(A \to bA)$, $t_2 = \hat{\beta}^{-1}(A \to a)$, and $t_4 = \hat{\beta}^{-1}(A \to b)$ are enabled. Transitions t_1 and (or) t_3 can occur several times and in any order, and the corresponding control places receive as many tokens as the numbers of occurrences of these transitions. Then transitions $t_5 = \hat{\beta}^{-1}(B \to aB)$ and $t_7 = \hat{\beta}^{-1}(B \to bB)$ occur as many times as t_1 and (or) t_3 occur. To go a final marking, transition $t_2 = \hat{\beta}^{-1}(A \to a)$ or $t_4 = \hat{\beta}^{-1}(A \to b)$ occurs, then, respectively, transition $t_6 = \hat{\beta}^{-1}(B \to a)$ or $t_8 = \hat{\beta}^{-1}(B \to b)$ occurs. One can see that G_2 generates a vector language

$$L(G_2) = \{wxw'x \mid x \in \{a, b\}, w \in \{a, b\}^*, w' \in \mathrm{Perm}(w)\}.$$

Example 6. We consider the same context-free grammar as that in Example 1 with the cyclic-cf Petri net $\check{N}$ in Figure 3 (the transitions of $\check{N}$ have the same labels as those of $\hat{N}$), i.e., $G_3 = (V, \Sigma, S, R, \check{N})$.

After transition t_0 occurs transitions t_1, t_2, t_3, t_4 are enabled. The following cases may arise: (1) the sequence $t_1 t_5$ or (and) $t_3 t_7$ occurs many times (in any order); (2) the sequence $t_1 t_3$ or $t_3 t_1$ occurs, then $t_5 t_7$ or $t_7 t_5$ occurs; (3) cases (1) and (2) repeat in any order; (4) to go to a final marking the sequence $t_2 t_6$ or $t_4 t_8$ occurs. It is easy to see that G_4 generates a semi-matrix language

$$L(G_3) = \{w_1\{\lambda, ab, ba\}w_2\{\lambda, ab, ba\} \cdots x \cdot w_1\{\lambda, ab, ba\}w_2\{\lambda, ab, ba\} \cdots x \mid x \in \{a, b\}, w_1, w_2, \ldots \in \{a, b\}^*\}.$$

Example 7. Let G_4 consist of context-free components of the grammar G_1 (Example 1) and the supervised-cyclic-cf Petri net $\tilde{N}$ in Figure 4 (it can be seen that the transitions preserve the labels).

The execution of $\tilde{N}$ starts with the occurrence of t_0, and transitions t_1, t_2, t_3, and t_4 are enabled. Next, only one of them can occur, enabling, respectively, t_5, t_6, t_7, and t_8. The sequence $t_1 t_5$ or $t_3 t_7$ occurs many times (in any order). To go to a final marking one of the sequences $t_2 t_6$, $t_4 t_8$ occurs. We can see that G_3 generates a matrix language

$$L(G_3) = \{ww \mid w \in \{a, b\}^+\}.$$

We denote the families of languages generated by ch- {c-, sc-}Petri net controlled grammars (with erasing rules) by **CHPN**, **CPN**, **SCPN** (**CHPN**$^\lambda$, **CPN**$^\lambda$, **SCPN**$^\lambda$), respectively.

4 Generative Capacity

In this section we show that the introduced Petri net controlled grammars simulate some well-known regulated grammars.

Lemma 1.

(1) $\mathbf{V} \subseteq \mathbf{CHPN}$ *and* $\mathbf{V}^{\lambda} \subseteq \mathbf{CHPN}^{\lambda}$,
(2) $\mathbf{sMAT} \subseteq \mathbf{CPN}$ *and* $\mathbf{sMAT}^{\lambda} \subseteq \mathbf{CPN}^{\lambda}$,
(3) $\mathbf{MAT} \subseteq \mathbf{SCPN}$ *and* $\mathbf{MAT}^{\lambda} \subseteq \mathbf{SCPN}^{\lambda}$.

Proof. Let $G = (V, \Sigma, S, M)$ be a vector (semi-matrix, matrix) grammar where $M = \{m_1, m_2, \ldots, m_k\}$. Without loss of generality we can assume that G is a grammar without repetition. Let R be the set of all rules of M. We construct a cf Petri net $N = (P, T, F, \varphi, \iota, \beta, \gamma)$ (with the notions of Def. 1).

(CHAIN): For each matrix $m = r_1 r_2 \cdots r_n \in M$ we define a chain $\rho_m = t_1 p_1 t_2 p_2 \cdots t_{n-1} p_{n-1} t_n$ where $p_1, p_2, \ldots, p_{n-1}$ are new places and $\gamma(t_i) = r_i$, $1 \leq i \leq n$. Then $\gamma(tr(\rho)) = m$ since γ is a bijection. Let $\Pi = \{\rho_{m_1}, \rho_{m_2}, \ldots, \rho_{m_k}\}$. As chains of Π are disjoint, we construct a chain-cf Petri net $\hat{N}$ (with notions of Def. 2), and define ch-Petri net controlled grammar $\hat{G} = (V, \Sigma, S, R, \hat{N})$. Let $S \overset{r_1 r_2 \cdots r_s}{\Longrightarrow} w \in \Sigma^*$ be a derivation in the vector grammar G. Then $r_1 r_2 \cdots r_s = \mathrm{sh}(m'_1, m'_2, \ldots, m'_l)$ for some $m'_1, m'_2, \ldots, m'_l \in M$. By bijection γ,

$$\begin{aligned} t_1 t_2 \cdots t_s &= \gamma^{-1}(r_1 r_2 \cdots r_s) \\ &= \gamma^{-1}(\mathrm{sh}(m'_1, m'_2, \ldots, m'_l)) \\ &= \mathrm{sh}(\gamma^{-1}(m'_1), \gamma^{-1}(m'_2), \ldots, \gamma^{-1}(m'_l)) \\ &= \mathrm{sh}(tr(\rho'_1), tr(\rho'_2), \ldots, tr(\rho'_l)). \end{aligned} \tag{1}$$

Hence, $t_1 t_2 \cdots t_s$ is a successful occurrence sequence in $\hat{N}$, and $S \overset{r_1 r_2 \cdots r_s}{\Longrightarrow} w \in \Sigma^*$ is also derivation in $\hat{G}$, i.e., $L(G) \subseteq L(\hat{G})$. The inverse inclusion can also be shown by backtracking (1).

(CYCLE): For each matrix $m = r_1 r_2 \cdots r_n \in M$ we define a cycle $\rho_m = p_1 t_1 p_2 \cdots p_n t_n p_1$ where $p_1, p_2, \ldots, p_n$ are new places and $\gamma(t_i) = r_i$, $1 \leq i \leq n$. Since γ is a bijection, $\gamma(tr(\rho)) = m$. Let $\Pi = \{\rho_{m_1}, \rho_{m_2}, \ldots, \rho_{m_k}\}$. As Π consists of disjoint cycles, we construct a cyclic-cf Petri net $\check{N}$ (with notions of Def. 3) and define c-Petri net controlled grammar $\check{G} = (V, \Sigma, S, R, \check{N})$.
Let $S \overset{r_1 r_2 \cdots r_s}{\Longrightarrow} w \in \Sigma^*$ be a derivation in the semi-matrix grammar G. Then $r_1 r_2 \cdots r_s = \mathrm{ssh}(m'_1, m'_2, \ldots, m'_l)$ for some $m'_1, m'_2, \ldots, m'_l \in M$. By bijection γ,

$$\begin{aligned} t_1t_2\cdots t_s &= \gamma^{-1}(r_1r_2\cdots r_s) \\ &= \gamma^{-1}(\mathrm{ssh}(m'_1, m'_2, \ldots, m'_l)) \\ &= \mathrm{ssh}(\gamma^{-1}(m'_1), \gamma^{-1}(m'_2), \ldots, \gamma^{-1}(m'_l)) \\ &= \mathrm{ssh}(tr(\rho'_1), tr(\rho'_2), \ldots, tr(\rho'_l)). \end{aligned} \tag{2}$$

It follows that $t_1t_2\cdots t_s$ is a successful occurrence sequence in $\check{N}$. Therefore, $L(G) \subseteq L(\check{G})$. The inverse inclusion can be shown in similar way.

(SUPERVISED CYCLE): For each matrix $m = r_1r_2\cdots r_n \in M$ we define a cycle $\rho_m = p_0t_1p_1\cdots p_{n-1}t_np_0$ where $p_0, p_1, \ldots, p_{n-1}$ are new places and $\gamma(t_i) = r_i$, $1 \leq i \leq n$. Since γ is a bijection, $\gamma(tr(\rho)) = m$. Let $\Pi = \{\rho_{m_1}, \rho_{m_2}, \ldots, \rho_{m_k}\}$. As the cycles of Π have the unique common place p_0, we construct a supervised-cyclic-cf Petri net $\tilde{N}$ (with notions of Def. 4) and define sc-Petri net controlled grammar $\tilde{G} = (V, \Sigma, S, R, \tilde{N})$.
Let $S \overset{r_1r_2\cdots r_s}{\Longrightarrow} w \in \Sigma^*$ be a derivation in the matrix grammar G. Then $r_1r_2\cdots r_s = m'_1m'_2\cdots m'_l)$ for some $m'_1, m'_2, \ldots, m'_l \in M$. By bijection γ,

$$\begin{aligned} t_1t_2\cdots t_s &= \gamma^{-1}(r_1r_2\cdots r_s) \\ &= \gamma^{-1}(m'_1m'_2\cdots m'_l) \\ &= \gamma^{-1}(m'_1)\gamma^{-1}(m'_2)\cdots\gamma^{-1}(m'_l) \\ &= tr(\rho'_1)tr(\rho'_2)\cdots tr(\rho'_l). \end{aligned} \tag{3}$$

Thus $t_1t_2\cdots t_s$ is a successful occurrence sequence in $\check{N}$ and $L(G) \subseteq L(\check{G})$. The inverse inclusion can be shown using similar arguments.

By analogous considerations, it can be shown that:

Lemma 2.

(1) $\mathbf{CHPN} \subseteq \mathbf{V}$ *and* $\mathbf{CHPN}^\lambda \subseteq \mathbf{V}^\lambda$,
(2) $\mathbf{CPN} \subseteq \mathbf{sMAT}$ *and* $\mathbf{CPN}^\lambda \subseteq \mathbf{sMAT}^\lambda$,
(3) $\mathbf{SCPN} \subseteq \mathbf{MAT}$ *and* $\mathbf{SCPN}^\lambda \subseteq \mathbf{MAT}^\lambda$.

The results of Lemmas 1, 2, Theorem 2.1.2 ([4]) and Theorem 7 ([14]) can be summarized in the following theorem.

Theorem 1. *The inclusions presented in Figure 5 hold.*

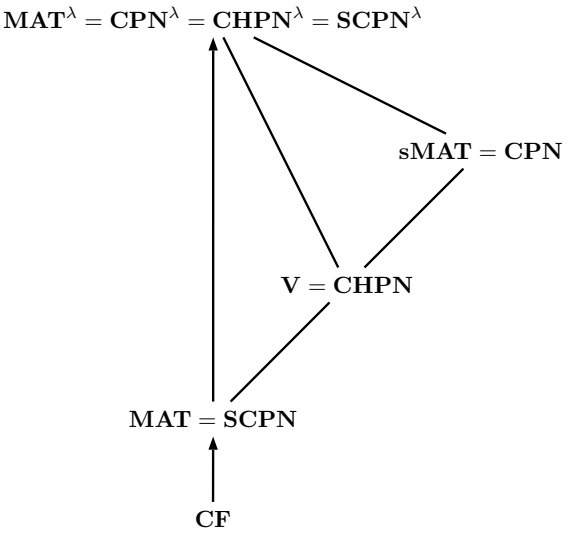

Fig. 5. If two families are connected by a line (an arrow), then the upper family includes (includes properly) the lower family.

References

1. Abraham, A. (1965). Some questions of phrase-structure grammars. *Computational Linguistics*, 4: 61–70.
2. Ceska, M. and V. Marek (2001). Petri nets and random-context grammars. In *Proceedings of the 35th Spring Conference: Modeling and Simulation of Systems (MOSIS'01)*, pp. 145-152. MARQ Ostrava Hardec and Moravici.
3. Cremers, A.B. and O. Mayer (1973). On matrix languages. *Information Control*, 23: 86–96.
4. Dassow, J. and Gh. Păun (1989). Regulated rewriting in formal language theory. Berlin: Springer.
5. Ginsburg, S. and E.H. Spanier (1968). Control sets on grammars. *Math. Syst. Th.*, 2: 159–177.
6. Hack, M. (1976). *Petri net languages.* MIT Laboratory of Computer Science Technical report 159. Cambridge
7. Hopcroft, J.E. and J.D. Ullman (1990). *Introduction to automata theory, languages, and computation.* Reading: Addison-Wesley.
8. Păun, Gh. (1980). A new generative device: valence grammars. *Rev. Roum. Math. Pures Appl.*, 25: 911–924.
9. Reisig, W. and G. Rozenberg (eds.) (1998). Lectures on Petri nets I: Basic models. *Lecture Notes in Computer science*, 1491. Berlin: Springer.

10. Rosenkratz, D.J. (1969). Programmed grammars and classes of formal languages. *Journal of the ACM*, 16: 107–131.
11. Rozenberg, G. and A. Salomaa (1997). *Handbook of formal languages*. Berlin: Springer.
12. Starke, P.H. (1978). Free Petri net languages. *Lecture Notes in Computer Science*, 64: 506–515.
13. Starke, P.H. (1980). *Petri-Netze*. Deutscher Verlag der Wissenschaften.
14. Turaev, S. (2006). Semi-matrix grammars. In *Proceedings 2nd Doctoral Workshop on Mathematical and Engineering Methods in Computer Science MEMICS-2006*, pp. 245–252.
15. Van Der Walt, A.P.J. (1970). Random context grammars. In *Proceedings of the Symposium on Formal Languages*. Oberwolfach.