UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010

UNIVERSITAT ROVIRA I VIRGILI

Facultat de Lletres
Departament de Filologies Romàniques

# PETRI NET CONTROLLED GRAMMARS

PhD Dissertation

Presented by
Sherzod TURAEV

Supervised by
Jürgen DASSOW

Tarragona, 2010

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010

Supervisor:

Prof. Dr. Jürgen Dassow

Fakultät für Informatik
Institut für Wissens- und Sprachverarbeitung
Otto-von-Guericke-Universität Magdeburg
D-39106 Magdeburg, Germany

Tutor:

Dr. Gemma Bel Enguix

Grup de Recerca en Lingüística Matemàtica
Facultat de Lletres
Universitat Rovira i Virgili
Av. Catalunya, 35
43002 Tarragona, Spain

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010

# Acknowledgements

It is a pleasure to thank many people who helped me to made this thesis possible.

First of all I would like to express my sincere gratitude to my supervisor, Professor Jürgen Dassow. He helped me with his enthusiasm, his inspiration, and his efforts. During the course of this work, he provided with many helpful suggestions, constant encouragement, important advice, good teaching, good company, and lots of good ideas. It has been an honor to work with him.

I would like to thank the head of our research group, Professor Carlos Martín-Vide, for his great effort in establishing the wonderful circumstances of PhD School on Formal Languages and Applications, and for his help in receiving financial support throughout my PhD. This work was made thanks to the financial support of research grant 2006FI-01030 from AGAUR of Catalonia Government.

I would like to thank all professors in our PhD school, especially Professors Tom Head, Erzsébet-Chuhaj Varjú, Victor Mitrana, Markus Holzer and Claudio Moraga, for providing solid knowledge and interest in formal languages and automata theory. I wish to express my cordial appreciation to Professor Victor Mitrana who encouraged me to work on this topic, gave me constructive advice and recommended me to work with Professor Dassow. Special thanks are due to my co-author, Dr. Ralf Stiebe, Otto-von-Guericke-University Magdeburg, for his ideas, efforts, useful discussions and time spent on our papers.

I would like to express my thanks to my colleagues at GRLMC, and my friends (Alexander Krassovitskiy, Areti Panou, Cătălin Tîrnăucă, Alexander Perekrestenko and Abhik Ghosh) for useful discussions and for making my staying in Tarragona enjoyable. I specially thank Dr. Gemma Bel Enguix, Lilica Voicu and Alexander Krassovitskiy for their assistance with all types of documentation and organization matters.

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010

I am also grateful to all members of the Department of Knowledge and Language Engineering, Faculty of Computer Science, Otto-von-Guericke-University Magdeburg (in particular, to Dr. Bianca Truthe) for their kind and generous hospitality during my stays, and for their fruitful discussions, suggestions, ideas.

I should also thank Professor Abdimajid Pulatov, National University of Uzbekistan, who has been a significant presence in my life. He taught me not only to conduct scientific research but also to think about life, politics, philosophy. I am always thankful for his wisdom and knowledge. Many thanks to my teacher Mr. Sayfiddin Qalandarov who taught me mathematics and always inspired me to progress in science.

My heartiest thanks go to my best friends Jessica and Carlo who never let me feel that I was away from my family, and helped me in everything. I express my deep love for my wife Matluba, and our children Jaloliddin and Zahiriddin. Their patience, love and encouragement always upheld me. Lastly, and most importantly, I wish to thank my parents, Rustam Turaev and Qurbonoy Tursunova for their support and love. They always helped me and encouraged me to concentrate on my study.

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010

# Contents

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010

# List of Figures

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010

# 1 Introduction

Formal language theory, introduced by Noam Chomsky in the 1950s as a tool for a description of natural languages [12, 13, 14], has also been widely involved in modeling and investigating phenomena such as generative (production) processes appearing in computer science, artificial intelligence and other related fields. In formal language theory a model for a phenomenon is usually constructed by representing it as a set of words, i.e., a *language* over a certain alphabet, and defining a generative mechanism, i.e., a *grammar* which identifies exactly the words of this set. With respect to the forms of their rules, grammars and their languages are divided into four classes of *Chomsky hierarchy*: *recursively enumerable* or *type 0*, *context-sensitive* or *type 1*, *context-free* or *type 2* and *regular* or *type 3*.

Context-free grammars are the most investigated type of Chomsky hierarchy which, in addition, have good mathematical properties and are extensively used in many applications of formal languages. However, they cannot cover all aspects which occur in modeling of phenomena. On the other hand, context-sensitive grammars, the next level in Chomsky hierarchy, are too powerful to be used in applications of formal languages, and have bad features, for instance, for context-sensitive grammars, the emptiness problem is undecidable and the existing algorithms for the membership problem, thus for the parsing, have exponential complexities. Moreover, such concepts as a derivation tree, which is an important tool for the anal-

ISBN:978-84-693-1536-1/DL:T-644-2010

ysis of context-free languages, cannot be transformed to context-sensitive grammars. Therefore, it is of interest to consider "intermediate" grammars which are more powerful than context-free grammars and have similar properties. One type of such grammars, called *grammars with regulated rewriting* (*controlled* or *regulated grammars* for short), is defined by considering grammars with some additional mechanisms which extract some subset of the generated language in order to cover some aspects of modeled phenomena. Due to the variety of investigated practical and theoretical problems, different additional mechanisms to grammars can be considered. Since Abraham [1] first defined matrix grammars in 1965, several grammars with restrictions such as programmed, random context, valence grammars, and etc., have been introduced (see [24]). However, the rapid developments in present day technology, industry, medicine and other areas challenge to deal with more and more new and complex problems, and to look for new suitable tools for the modeling and investigation of these problems.

In our thesis we propose to use Petri nets as regulation mechanisms to context-free grammars and define *Petri net controlled grammars*. This idea can be justified with the following facts. On the one hand, control by Petri nets extends possibilities to investigate concurrent control mechanisms in formal language theory. [48] can be considered as the first paper in this direction where the regulation in matrix grammars is simulated by Petri nets in order to solve some open problems in matrix languages. In [67] it was also shown that the additional requirement in random context grammars can be simulated by Petri nets. Control by Petri nets has also been introduced and studied in automata theory [37, 38, 36, 57] and grammar systems theory [9].

On the other hand, grammars controlled by Petri nets can be very appropriate tools for modeling and analyzing phenomena in automated manufacturing systems and systems biology, where Petri nets are responsible for the structure and communication in systems and grammars represent gen-

ISBN:978-84-693-1536-1/DL:T-644-2010

erative processes. For instance, the following circumstance demonstrates that biochemical processes – *metabolic pathways* – in living cells can be accurately modeled and investigated using formal grammars with Petri nets.

One of the main goals of systems biology is to understand the processes in a living cell. Living cells are composed of a number of compounds (metabolites, enzymes, co-factors, ions, and etc.) and chemical reactions that occur simultaneously. A complete understanding of the behavior of these reactions is possible only through a complete analysis in both qualitative and quantitative terms. A qualitative analysis of the behavior of these reactions constitutes the qualitative study of metabolic pathways. A metabolic pathway is a series of chemical reactions occurring within a cell, catalyzed by enzymes, resulting in either the formation of a metabolic product to be used or stored by the cell, or the initiation of another metabolic pathway (Figure 1.1 illustrates a metabolic pathway of the citric acid cycle).

Often metabolites participate in more than one metabolic pathway, forming a complex network of reactions. Metabolic pathways may be of two general types: *catabolic* and *anabolic*. Catabolic pathways involve the breakdown or digestion of large, complex molecules. Anabolic pathways involve the synthesis of large molecules, generally by joining smaller molecules together.

The important issues in a qualitative analysis of metabolic pathways are the selection of appropriate descriptions for whole sets of pathways, and the selection of operations that can be used to combine these sets and identify qualitative properties and recurring pathway structures from them. The descriptions of sets of pathways should limit the computational complexity and make results easier to comprehend.

In order to represent and analyze metabolic pathways several methods such as ordinary differential equations [49], Boolean logic and state machines [60], genetic grammars [51], rule based models [50], stochastic parameterized grammars [69], and graph grammar based models [15], and

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010

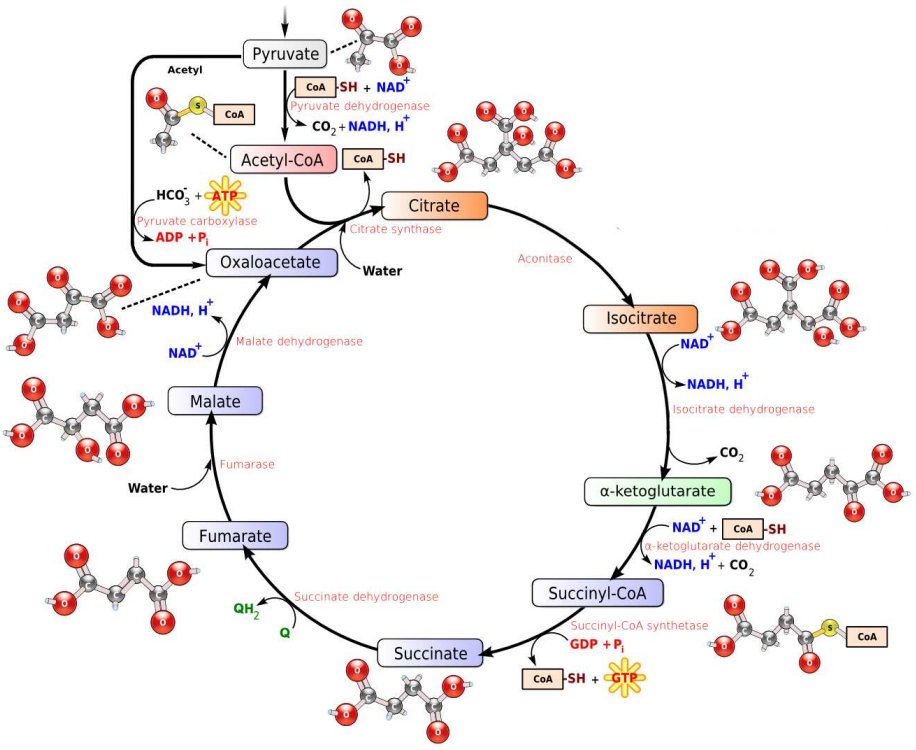


Figure 1.1: Citric acid cycle

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010

various graphical models [63] including Kauffman binary networks [61], signal flow graphs [83], bond graphs [98], different types of Petri net models [7, 11, 33, 52, 64] have been used. Among wide spectrum of models, Petri nets offer a simple and intuitive representation of a metabolic pathway structure where places represent biochemical entities, tokens in a place indicate the presence of the corresponding entity, transitions represent reactions, the arc weights encode the stoichiometry of a reaction [64] (Figure 1.2 represents a simplified Petri net model of the citric acid cycle illustrated in Figure 1.1). Occurrence sequences of transitions of a Petri net simulate sequences of biochemical reactions in the corresponding metabolic pathway.

The analysis of complex networks of metabolic pathways involving a tremendous amount of data such as metabolites, biochemical reactions and their properties, enzymes, etc., requires the use of an automated information processing, i.e., computers. The necessity of codifying all the information in comprehensible manner for the computer processing motivates to investigate easy computer-implementable and coherent symbolic methods for the representation and analysis of metabolic pathways.

At this point the similarity between the application of a production rule of a grammar and the firing of a transition of a Petri net [18], and on the other hand, the similarity between the firing of a transition of a Petri net and a biochemical reaction of a metabolic pathway prompts to consider integrated models of grammars and Petri nets for metabolic pathways.

We propose the model for metabolic pathways using grammars and Petri nets, which can be considered as an extended version of the Petri net based model introduced in [64]. Symbols of a grammar represent compounds of a biochemical reaction, and places labeled by symbols represent the status of the corresponding compounds where tokens are separate instances of the symbols, which shows the available amount of the corresponding substance. Production rules represent biochemical reactions, and transitions labeled by rules represent the status of the corresponding production rules. A tran-

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010

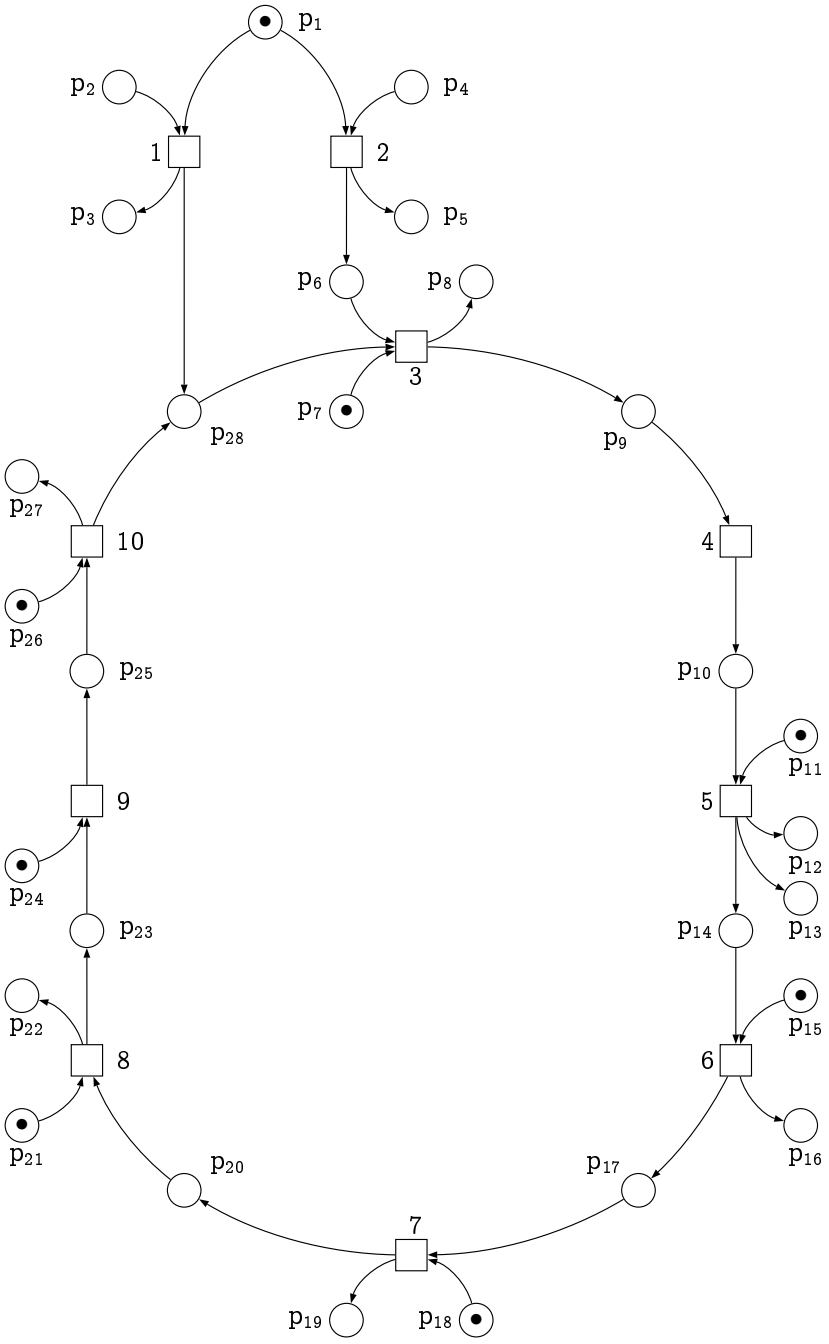


Figure 1.2: A simplified Petri net representation of citric acid cycle

ISBN:978-84-693-1536-1/DL:T-644-2010

sition is labeled by a production rule such that its input places are labeled by the symbols of the left-hand side of the rule and its output places are labeled by the symbols of the right-hand side of the rule. A rule can be applied if the transition labeled by this rule can be fired. Weights assigned to arcs indicate numbers of occurrences of symbols erased and appeared by the application of a rule when the transition labeled by this rule fires. Then a metabolic pathway can be simulated by a derivation of the grammar, where the sequence of production rules in the derivation are chosen according to the occurrence sequence of transitions of the corresponding Petri net. Thus, Petri nets can be considered as additional mechanisms to grammars in order to choose production rules for derivations of grammars.

Though biochemical reactions are represented by production rules of type 0 grammars in general case, we consider the simpler models investigating catabolic pathways which consists of only decomposition reactions. These types of biochemical reactions can be represented by context-free rules, in which enzymes participated in reactions can be represented by additional places.

The thesis introduces various variants of Petri net controlled grammars using different types of Petri nets and investigates their computational and closure properties. Thesis is organized as follows.

*Chapter 2* starts by giving, as prerequisites, some basic concepts and results from the areas formal languages and Petri nets: strings, grammars, languages, Petri nets, Petri net languages and so on, which will be used in our further investigations.

When we study the controls by extended context-free Petri nets (isomorphic Petri nets to context-free grammars enriched with additional components), we can see that controls used in grammars with regulated rewriting can be represented by some special subnets of these Petri nets, for instance, the net consisting of disjoint chains corresponds to the regulation in vector grammars while the net consisting of cycles with the single common place

ISBN:978-84-693-1536-1/DL:T-644-2010

matches to the restriction in matrix grammars. If the subnet is required to be composed of disjoint cycles, then we come to the definition of a new type of regulated grammars: we introduce a new variant of matrix grammars called *semi-matrix* grammars and investigate their generative power and closure properties in *Chapter 3*. From the definitions of matrix and vector grammars one can see that in a matrix grammar the shuffling of matrices is not allowed while in a vector grammar matrices can be shuffled. Semi-matrix grammars differ from the foregoing grammars by application of matrices in derivations: the shuffling of matrices is allowed only for different matrices. Semi-matrix grammars complete the variations of matrix grammars with respect to the start of a matrix.

The study of Petri net controlled grammars starts from *Chapter 4*. A context-free grammar and its derivation process can be described by a Petri net, called a *context-free Petri net* (a *cf Petri net* for short), where places correspond to nonterminals, transitions are the counterpart of the production rules, the tokens reflect the occurrences of symbols in the sentential form, and there is a one-to-one correspondence between the application of (sequence of) rules and the firing of (sequence of) transitions. Therefore, the control of the derivations in a context-free grammar can be implemented by adding some features to the associated Petri net.

Depending on what kind of elements (places, transitions or/and arcs) to add and how to add them, various control mechanisms can be defined. In our current research, as additional elements to a cf Petri net, we choose places and arcs. As control, first, we consider a subnet consisting of new places, transitions of the cf Petri net, and new arcs from/to these new places to/from transitions of the net; it leads to define grammars controlled by $k$-Petri nets, i.e., cf Petri nets with additional $k$ places. Next we consider more complex control mechanisms: we add new places and arcs in such a way that a control subnet consists of chains or cycles, and correspondingly, we define $z(c, s)$-Petri net controlled grammars. We investigate the computational

ISBN:978-84-693-1536-1/DL:T-644-2010

power and closure properties of families of languages generated by extended Petri net controlled grammars.

In *Chapter 5* we consider a generalization of the Petri net controlled grammars defined in the previous chapter: we associate an arbitrary place/ transition Petri net with a context-free grammar and require that the sequence of applied rules corresponds to an occurrence sequence of transitions in the Petri net. On the other hand, this type of Petri net controlled grammars is also a generalization of regularly controlled grammars, i.e., instead of a finite automaton, a Petri net is associated with a context-free grammar. With respect to different labeling strategies and different definitions of final marking sets, we define various classes of Petri net controlled grammars. Here we study the influence of the labeling functions and the effect of the final markings on the generative power.

It is known that many decision problems in formal language theory are equivalent to the reachability problem in Petri net theory, which has been shown that it is decidable, however, it has exponential time complexity. The result of this has been the definition of a number of structural subclasses of Petri nets with a smaller complexity and still adequate modeling power. Thus, it is interesting to consider grammars controlled by such kind of subclasses of Petri nets. In *Chapter 6* we continue our study of arbitrary Petri net controlled grammars by restricting Petri nets to their structural subclasses, i.e., special Petri nets such as state machines, marked graphs, and free-choice nets, and so on.

If in previous chapters we investigate Petri net controlled grammars using static properties of Petri nets, in *Chapter 7* we examine Petri net controlled grammars with respect to dynamical properties of Petri nets. Here we use (cf, extended cf, and arbitrary) Petri nets with place capacities. We also investigate capacity-bounded grammars which are counterparts of grammars controlled by Petri nets with place capacities.

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010

Finally, in *Chapter 8* we draw some general conclusions and present suggestions for further research.

The majority of the results of in the thesis has been published in scientific journals and presented at conferences. Minor improvements of presentations and proofs have been performed in many places. Chapter 3 is mainly based on the presentation at [99]. The results of Chapter 4 are based on works presented at [27, 28, 100] and published [32]. Chapter 5 describes results presented at [26] and published in [31]. Chapter 6 is formed by the presentation at [29] and the article published in [30]. Finally, the results of Chapter 7 are published in [96, 97, 95].

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010

# 2 Preliminaries

In this chapter we recall some prerequisites, by giving basic notions and notations of the theories formal languages, Petri nets and Petri net languages which are used in the thesis. The reader is referred to [53, 85, 24, 68, 82, 47, 74, 71] for further information.

## 2.1 Formal Languages

### 2.1.1 General Notations

Throughout the thesis we use the following general notations. $\in$ denotes the membership of an element to a set while the negation of set membership is denoted by $\notin$. The inclusion is denoted by $\subseteq$ and the strict (proper) inclusion is denoted by $\subset$. The union, intersection, difference and cross product of two sets are denoted by $\cup$, $\cap$, $-$, $\times$, respectively. $\emptyset$ denotes the empty set. The set of positive (non-negative) integers is denoted by $\mathbb{N}$ ($\mathbb{N}_0$). The set of integers is denoted by $\mathbb{Z}$. The power set of a set X is denoted by $2^X$, while the cardinality of a set X is denoted by $|X|$.

### 2.1.2 Strings, Languages and Operations

Let $\Sigma$ be an *alphabet* which is a finite nonempty set of symbols. A *string* (sometimes a *word*) over the alphabet $\Sigma$ is a finite sequence of symbols

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



from $\Sigma$. The *empty* string is denoted by $\lambda$. The set of all strings over the alphabet $\Sigma$ is denoted by $\Sigma^*$. The set of nonempty strings over $\Sigma$ is denoted by $\Sigma^+$, i.e., $\Sigma^+ = \Sigma^* - \{\lambda\}$. A subset of $\Sigma^*$ is called a *language*. A language $L \in \Sigma^*$ is $\lambda$-*free* if $\lambda \notin L$.

For two strings $u$ and $v$, their *concatenation* is defined as the juxtaposition $uv$. If $w = w_1w_2$, for some $w_1, w_2 \in \Sigma^*$, then $w_1$ is called a prefix of $w$ and $w_2$ is called a suffix of $w$. The sets of all prefixes and suffixes of a string $w$ are denoted by $\mathrm{Pref}(w)$ and $\mathrm{Suf}(w)$, respectively. If $w = w_1w_2w_3$ for some strings $w_1, w_2, w_3 \in \Sigma^*$, then $w_2$ is called a *substring* of $w$. A string $v$ is called a *scattered substring* of a string $u$ if there are strings $v_1, v_2, \ldots, v_n, u_1, u_2, \ldots, u_{n+1} \in \Sigma^*$, $n \geqslant 1$, such that $v = v_1v_2 \cdots v_n$ and $u = u_1v_1u_2v_2 \cdots u_nv_nu_{n+1}$.

For a string $w = a_1a_2 \cdots a_n$, $a_i \in \Sigma$, $1 \leqslant i \leqslant n$, the string $a_n \cdots a_2a_1$ is called the *mirror image* of $w$ and denoted by $w^R$. For a language $L \subseteq \Sigma^*$, its mirror image is defined as $L^R = \{w^R \mid w \in L\}$.

The *length* of a word $w$, denoted by $|w|$, is the number of occurrences of symbols in $w$. The number of occurrences of a symbol $a$ in a string $w$ is denoted by $|w|_a$. For a subset $\Delta$ of $\Sigma$, the number of occurrences of symbols of $\Delta$ in a string $w \in \Sigma^*$ is denoted by $|w|_\Delta$.

*Parikh vector* associated with $w \in \Sigma^*$ with respect to the alphabet $\Sigma = \{a_1, a_2, \ldots, a_k\}$ is defined by $p_\Sigma(w) = (|w|_{a_1}, |w|_{a_2} \ldots, |w|_{a_k})$. For a string $w \in \Sigma^*$, the set of all its permutations is defined by

$$\mathrm{Perm}(w) = \{w' \mid p_\Sigma(w') = p_\Sigma(w)\}.$$

For two languages $L_1, L_2 \subseteq \Sigma^*$ the *operation shuffle* is defined by

$$\mathrm{Shuf}(L_1, L_2) = \{u_1v_1u_2v_2 \cdots u_nv_n \mid u_1u_2 \cdots u_n \in L_1, v_1v_2 \cdots v_n \in L_2, \\ u_i, v_i \in \Sigma^*, 1 \leqslant i \leqslant n\}$$

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



and for $L \subseteq \Sigma^*$,

$$\begin{aligned} \mathrm{Shuf}^1(L) &= L, \\ \mathrm{Shuf}^k(L) &= \mathrm{Shuf}(\mathrm{Shuf}^{k-1}(L), L), k \geqslant 2, \\ \mathrm{Shuf}^*(L) &= \bigcup_{k \geqslant 1} \mathrm{Shuf}^k(L). \end{aligned}$$

Regarding languages as sets, the operations of union, intersection, difference for languages are defined in usual fashion. The *concatenation* of two languages $L_1$ and $L_2$ is

$$L_1 L_2 = \{uv \mid u \in L_1, v \in L_2\}.$$

For an integer $n \geqslant 0$ and a language $L$, $L^n$ is defined by $L^0 = \{\lambda\}$, $L^n = L^{n-1}L$, for $n > 0$. The *Kleene closure* of a language $L$, denoted by $L^*$, is defined by

$$L^* = \bigcup_{i \geqslant 0} L^i,$$

and its $\lambda$-*free Kleene closure*, denoted by $L^+$, is defined by

$$L^+ = \bigcup_{i \geqslant 1} L^i.$$

A language $L$ over $\Sigma$ is called *regular* if it can be constructed by a finite number of applications of the operations union, concatenation and Kleene closure from subsets of $\Sigma \cup \{\lambda\}$. The family of all regular languages is denoted by REG.

A mapping $s : \Sigma^* \to 2^{\Delta^*}$ is called a *substitution* if $s(\lambda) = \{\lambda\}$ and $s(uv) = s(u)s(v)$. $s$ is said to be $\lambda$-*free* if $\lambda \notin s(a)$ for all $a \in \Sigma$. A substitution is called a *homomorphism* if $|s(a)| = 1$ for all $a \in \Sigma$ (we write $s(a) = u$ instead of $s(a) = \{u\}$). A homomorphism $h : \Sigma^* \to \Delta^*$ is called a *coding* if $h(a) \in \Delta$ for each $a \in \Sigma$ and a *week coding* if $h(a) \in \Delta \cup \{\lambda\}$ for each $a \in \Sigma$. For a homomorphism $h : \Sigma^* \to \Delta^*$, the mapping $h^{-1} : \Delta^* \to 2^{\Sigma^*}$

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



defined by

$$h^{-1}(u) = \{v \in \Sigma^* \mid h(v) = u\}$$

is called an *inverse homomorphism*.

If $L \subseteq (\Sigma\{\lambda, c, c^2, \ldots, c^{k-1}\})^*$ for some $c \notin \Sigma$ and some constant $k$, and if $h$ is a homomorphism defined by $h(a) = a$ for all $a \in \Sigma$ and $h(c) = \lambda$, then $h$ is said to be $k$-*restricted* on $L$.

The *left derivative* of a language $L \subseteq \Sigma^*$ with respect to a string $x \in \Sigma^*$ is

$$\partial_x^l(L) = \{w \in \Sigma^* \mid xw \in L\}.$$

The *right derivative* of a language $L \subseteq \Sigma^*$ with respect to a string $x \in \Sigma^*$ is

$$\partial_x^r(L) = \{w \in \Sigma^* \mid wx \in L\}.$$

Let $o$ be a $k$-ary operation on languages and $\mathcal{L}$ be a family of languages. $\mathcal{L}$ is said to be closed under the operation $o$ if for all languages $L_1, L_2, \ldots L_k \in \mathcal{L}$, $o(L_1, L_2, \ldots, L_k) \in \mathcal{L}$. A language family is called an *abstract family of languages* (abbreviated AFL) if and only if it is closed under union, concatenation, $\lambda$-free Kleene closure, $\lambda$-free homomorphisms, inverse homomorphisms and intersections with regular languages. A family of languages closed under all AFL operations except concatenation and $\lambda$-free Kleene closure is called a semi-AFL.

**Theorem 2.1.** If $\mathcal{L}$ is a family of languages closed under intersections with regular sets, union with regular sets, substitution by $\lambda$-free regular sets and restricted homomorphisms, then $\mathcal{L}$ is closed under inverse homomorphisms.

**Theorem 2.2.** A semi-AFL is closed under right and left derivatives.

Two languages $L_1$ and $L_2$ are called equal if $L_1 - \{\lambda\} = L_2 - \{\lambda\}$. Two language families $\mathcal{L}_1$ and $\mathcal{L}_2$ are called equal if for each language $L_1 \in \mathcal{L}_1$ there is a language $L_2 \in \mathcal{L}_2$ which is equal to $L_1$ and vice versa.

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



A *finite automaton* is a system $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$ where $Q$ is a finite non-empty set of *states*, $\Sigma$ is the *input alphabet*, $q_0 \in Q$ is the *initial state*, $F \subseteq Q$ is the set of *final states* and $\delta : Q \times \Sigma \to 2^Q$ is the *transition function*. Such an automaton *recognizes* a string $w = a_1 a_2 \cdots a_n \in \Sigma^*$, $a_1, a_2, \ldots, a_n \in \Sigma$, if and only if there are states $q_1, q_2, \ldots, q_n \in Q$ such that $q_1 \in \delta(q_0, a_1)$, $q_{i+1} \in \delta(q_i, a_{i+1})$ for $1 \leqslant i \leqslant n-1$ and $q_n \in F$. The set of all strings recognized by $\mathcal{A}$ is denoted by $L(\mathcal{A})$.

**Theorem 2.3.** The family REG is exactly the family of languages recognizable by finite automata.

### 2.1.3 Grammars

A *phrase structure* (*Chomsky*) grammar is a quadruple $G = (V, \Sigma, S, R)$ where $V$ and $\Sigma$ are two disjoint alphabets of *nonterminal* and *terminal* symbols, respectively, $S \in V$ is the *start symbol* and

$$R \subseteq (V \cup \Sigma)^* V (V \cup \Sigma)^* \times (V \cup \Sigma)^*$$

is a finite set of (*production*) *rules*. Usually, a rule $(u, v) \in R$ is written in the form $u \to v$. A rule of the form $u \to \lambda$ is called an *erasing rule*.

A phrase structure grammar $G = (V, \Sigma, S, R)$ is called a *GS grammar* (a phrase structure grammar due to Ginsburg and Spanier [42]) if

$$R \subseteq V^+ \times (V \cup \Sigma)^*.$$

The families of languages generated by GS grammars and by phrase structure grammars are denoted by GS and RE, respectively. It is well-known that the family GS is equal to the family RE.

A string $x \in (V \cup \Sigma)^*$ *directly derives* a string $y \in (V \cup \Sigma)^*$ in $G$, written as $x \Rightarrow_G y$ if and only if there is a rule $u \to v \in R$ such that $x = x_1 u x_2$ and $y = x_1 v x_2$ for some $x_1, x_2 \in (V \cup \Sigma)^*$. If $G$ is understood, we write $x \Rightarrow y$.

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



The reflexive and transitive closure of the relation $\Rightarrow$ is denoted by $\Rightarrow^*$. A derivation using the sequence of rules $\pi = r_1 r_2 \cdots r_k$, $r_i \in R$, $1 \leqslant i \leqslant k$, is denoted by $\stackrel{\pi}{\Rightarrow}$ or $\stackrel{r_1 r_2 \cdots r_k}{\Longrightarrow}$.

A string $w \in (V \cup \Sigma)^*$ such that $S \Rightarrow_G^* w$ is called a *sentential form*. The *language* generated by G, denoted by $L(G)$, is defined by

$$L(G) = \{w \in \Sigma^* \mid S \Rightarrow^* w\}.$$

Two grammars $G_1$ and $G_2$ are called *equivalent* if $L(G_1) = L(G_2)$.

A phrase-structure grammar $G = (V, \Sigma, S, R)$ is called

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

The families of languages generated by context-sensitive, context-free and linear grammars are denoted by CS, CF, LIN. Further we denote the family of finite languages by FIN. The next strict inclusions, named *Chomsky hierarchy*, hold

**Theorem 2.4.** $FIN \subset REG \subset LIN \subset CF \subset CS \subset RE$.

### 2.1.4 Grammars with Regulated Writing

The idea of regulated rewriting consists of restricting the application of the rules in a context-free grammar in order to avoid some derivations and hence obtaining a subset of the context-free language generated in

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



usual way. The computational power of some context-free grammars with regulated rewriting turns out to be greater than the power of context-free grammars. Various regulation mechanisms used in regulated grammars can be classified into general types by their common features, for instance

- control by prescribed sequences of production rules:
  *matrix grammars* (Abraham [1], 1965) – the set of production rules is divided into sequences (called *matrices*) and if the application of a matrix is started, a second matrix can be started after finishing the application of the first one. And the rules have to been applied in the order given a matrix; *vector grammars* (Cremers and Mayer [17],1973) – in which a new matrix can be started before finishing those which have been started earlier; *regularly controlled grammars* (Ginsburg and Spanier [42], 1968) – the sequence of production rules applied in a derivation belong to a given regular language associated with the grammar;

- control by computed sequences of production rules:
  *programmed grammars* (Rosenkrantz [89], 1969) – after applying a production rule, the next production rule has to be chosen from its success field; if the left hand side of the rule does not occur in the sentential form, a rule from its failure field has to be chosen; *valence grammars* (Păun [79], 1980) – where with each sentential form an element of a monoid is associated, which is computed during the derivation and derivations where the element associated with the terminal word is the neutral element of the monoid are accepted;

- control by context conditions:
  where the applicability of a rule depends on the current sentential form; with any rule some restrictions are associated for sentential forms which have to be satisfied in order to apply the rule: *random context grammars* (Cremers, Maurer and Mayer [16], 1973) – the

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



restriction is the belonging to a regular language associated with the rule; *conditional grammars* (Fris [41], 1968) – the restriction to special regular sets; *semi-conditional grammars* (Kelemen [62], 1984) – the restriction to words of length one in the permitting and forbidden contexts; *ordered grammars* (Fris [41], 1968) – a production rule can be applied if there is no greater applicable production rule;

- control by partial parallelism:
  *Indian parallel grammars* (Siromoney and Krithivasan [92], 1974) – all occurrences of one letter are replaced (according to one rule); *Russian parallel grammars* (Levitina [65], 1972) – which combines the context-free and Indian parallel feature; *scattered context grammars* (Greibach and Hopcroft [45], 1969) – in which only a fixed number of symbols can be replaced in a step but the symbols can be different;

- control by memory:
  *indexed grammars* (Aho [2], 1968) – the application of production rules gives sentential forms where the nonterminal symbols are followed by sequences of indexes (*stack* of special symbols), and indexes can be erased only by rules contained in these indexes but erasing of the indexes is done in reverse order of their appearance.

Further we recall the definitions of those regulated grammars which are used in the proofs of some statements in this thesis.

A *regularly controlled grammar* is a quintuple $G = (V, \Sigma, S, R, K)$ where $V, \Sigma, S, R$ 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 $S \stackrel{r_1 r_2 \cdots r_n}{\Longrightarrow} w$ where $r_1 r_2 \cdots r_n \in K$.

A *matrix grammar* is a quadruple $G = (V, \Sigma, S, M)$ where $V, \Sigma, S$ are defined as for a context-free grammar, $M$ is a finite set of *matrices* which are finite strings over a set $R$ of context-free rules (or finite sequences of

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



context-free rules). For $m \in M$, we use both notations $m = r_1 r_2 \cdots r_s$ and $m : (r_1, r_2, \ldots, r_s)$. The language generated by the grammar G is

$$L(G) = \{w \in \Sigma^* \mid S \stackrel{\pi}{\Rightarrow} w \text{ and } \pi \in M^*\}.$$

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 is defined by

$$L(G) = \{w \in \Sigma^* \mid S \stackrel{\pi}{\Rightarrow} w \text{ and } \pi \in Shuf^*(M)\}.$$

A matrix (vector) grammar G is called *without repetitions* if for each rule $r \in R$, $|m_1 m_2 \cdots m_n|_r = 1$. For each matrix (vector) grammar, by adding chain rules, we can construct an equivalent matrix grammar without repetitions.

An *additive valence grammar* is a quintuple $G = (V, \Sigma, S, R, \nu)$ where $V, \Sigma, S, R$ are defined as for a context-free grammar and $\nu$ is a mapping from R into $\mathbb{Z}$. The language generated by G consists of all strings $w \in \Sigma^*$ such that there is a derivation $S \stackrel{r_1 r_2 \cdots r_n}{\Longrightarrow} w$ where $\sum_{i=1}^{n} \nu(r_i) = 0$.

A *positive valence grammar* is a quintuple $G = (V, \Sigma, S, R, \nu)$ whose components are defined as for additive valence grammars. The language generated by G consists of all strings $w \in \Sigma^*$ such that there is a derivation $S \stackrel{r_1 r_2 \cdots r_n}{\Longrightarrow} w$ where $\sum_{i=1}^{n} \nu(r_i) = 0$ and for any $1 \leqslant j < n$, $\sum_{i=1}^{j} \nu(r_i) \geqslant 0$.

A *programmed grammar* (without appearance checking) is a sextuple $G = (V, \Sigma, S, R, D, \sigma)$ where $V, \Sigma, S, R$ are defined as for a context-free grammar, D is a finite set of labels for the rules of R such that D can be also interpreted as a function which outputs a rule when being given a label, $\sigma$ is a function from D into the set of subsets D, i.e., $2^D$. For $(x, r_1), (y, r_2) \in (V \cup \Sigma)^* \times D$ and $D(r_1) = (\alpha \to \beta)$ we write $(x, r_1) \Rightarrow (y, r_2)$

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



if and only if

$$x = x_1 \alpha x_2, y = x_1 \beta x_2 \text{ and } r_2 \in \sigma(r_1).$$

The $\sigma(r)$ is called the *success field* of $r \in R$. The language generated by G is defined as

$$L(G) = \{w \in \Sigma^* \mid (S, r_1) \Rightarrow^* (w, r_2) \text{ for some } r_1, r_2 \in D\}.$$

The families of languages generated by regularly controlled, matrix, vector, additive valence and positive valence grammars (with erasing rules) are denoted by rC, MAT, VEC, Prog, aV, pV ($rC^\lambda$, $MAT^\lambda$, $VEC^\lambda$, $Prog^\lambda$, $aV^\lambda$, $pV^\lambda$), respectively.

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

**Theorem 2.5.** The following inclusions and equalities hold (for details, see [24]):

(1) $CF \subset aV = aV^\lambda \subset MAT = rC = Prog = pV$;

(2) $MAT \subseteq VEC \subset CS$;

(3) $MAT \subseteq MAT^\lambda = rC^\lambda = VEC^\lambda = Prog^\lambda = pV^\lambda \subset RE$.

## 2.2 Petri Nets

Petri nets, introduced by Carl Adam Petri [75] in 1962, provide a powerful mathematical formalism for describing and analyzing the flow of information and control in concurrent systems.

### 2.2.1 Basic Definitions

A *Petri net* (PN) is a construct $N = (P, T, F, \phi)$ where P and T are disjoint finite sets of *places* and *transitions*, respectively, $F \subseteq (P \times T) \cup (T \times P)$ is

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



the set of *directed arcs*, $\phi : F \to \mathbb{N}$ is a *weight function*.

A Petri net can be represented by a bipartite directed graph with the node set $P \cup T$ where places are drawn as *circles*, transitions as *boxes* and arcs as *arrows*. The arrow representing an arc $(x, y) \in F$ is labeled with $\phi(x, y)$; if $\phi(x, y) = 1$, then the label is omitted.

An *ordinary net* (ON) is a Petri net $N = (P, T, F, \phi)$ where $\phi(x, y) = 1$ for all $(x, y) \in F$. We omit $\phi$ from the definition of an ordinary net, i.e., $N = (P, T, F)$.

A mapping $\mu : P \to \mathbb{N}_0$ is called a *marking*. For each place $p \in P$, $\mu(p)$ gives the number of *tokens* in $p$. Graphically, tokens are drawn as small solid *dots* inside circles. ${}^\bullet x = \{y \mid (y, x) \in F\}$ and $x^\bullet = \{y \mid (x, y) \in F\}$ are called *pre-* and *post-sets* of $x \in P \cup T$, respectively. For $X \subseteq P \cup T$, define ${}^\bullet X = \bigcup_{x \in X} {}^\bullet x$ and $X^\bullet = \bigcup_{x \in X} x^\bullet$. For $t \in T$ ($p \in P$), the elements of ${}^\bullet t$ (${}^\bullet p$) are called *input* places (transitions) and the elements of $t^\bullet$ ($p^\bullet$) are called *output* places (transitions) of $t$ ($p$).

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 $1 \leqslant i \leqslant n - 1$. We denote the sets of places, transitions and arcs of a path $\rho$ by $P_\rho, T_\rho, F_\rho$, respectively. The sequence of transitions in a path $\rho$ is denoted by $tr(\rho)$. Two paths $\rho_1$, $\rho_2$ are called *disjoint* if $P_{\rho_1} \cap P_{\rho_2} = \emptyset$ and $T_{\rho_1} \cap T_{\rho_2} = \emptyset$. A path $\rho$ is called a *chain*(*cycle*) if $x_1, x_n \in T$ and $x_1 \neq x_n$ ($x_1 = x_n$).

A transition $t \in T$ is *enabled* by marking $\mu$ if and only if $\mu(p) \geqslant \phi(p, t)$ for all $p \in {}^\bullet t$. In this case $t$ can *occur* (*fire*). Its occurrence transforms the marking $\mu$ into the marking $\mu'$ defined for each place $p \in P$ by $\mu'(p) = \mu(p) - \phi(p, t) + \phi(t, p)$. We write $\mu \xrightarrow{t}$ to denote that $t$ may fire in $\mu$, and $\mu \xrightarrow{t} \mu'$ to indicate that the firing of $t$ in $\mu$ leads to $\mu'$. A marking $\mu$ is called *terminal* if in which no transition is enabled. A finite sequence $t_1 t_2 \cdots t_k$, $t_i \in T, 1 \leqslant i \leqslant k$, is called *an occurrence sequence* enabled at a marking $\mu$ and finished at a marking $\mu'$ if there are markings

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



$\mu_1, \mu_2, \ldots, \mu_{k-1}$ such that $\mu \xrightarrow{t_1} \mu_1 \xrightarrow{t_2} \ldots \xrightarrow{t_{k-1}} \mu_{k-1} \xrightarrow{t_k} \mu'$. In short this sequence can be written as $\mu \xrightarrow{t_1 t_2 \cdots t_k} \mu'$ or $\mu \xrightarrow{\nu} \mu'$ where $\nu = t_1 t_2 \cdots t_k$. For each $1 \leqslant i \leqslant k$, marking $\mu_i$ is called *reachable* from marking $\mu$. $\mathcal{R}(N, \mu)$ denotes the set of all reachable markings from a marking $\mu$.

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

*Example* 2.1. Figure 2.1 depicts a Petri net $N = (P, T, F, \phi, \iota)$ with $P = \{p_1, p_2, p_3, p_4, p_5\}$ and $T = \{t_1, t_2, t_3, t_4, t_5\}$. We can see that $\phi(t_2, p_3) = \phi(p_3, t_3) = 2$. The initial marking $\iota$ is defined by $\iota(p_1) = \iota(p_4) = 1$ and $\iota(p) = 0$ for all $P - \{p_1, p_4\}$.

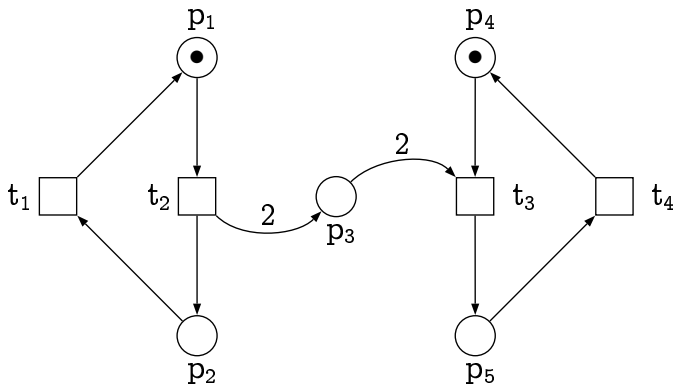


Figure 2.1: A Petri net

A Petri net *with final markings* is a construct $N = (P, T, F, \phi, \iota, M)$ where $(P, T, F, \phi, \iota)$ is a marked Petri net and $M \subseteq \mathcal{R}(N, \iota)$ is set of markings which are called *final* markings. If $M = \{\mu\}$, then instead of $\{\mu\}$ we simply write $\mu$ in the definition. An occurrence sequence $\nu$ of transitions is called *successful* for $M$ if it is enabled at the initial marking $\iota$ and finished at a final marking $\tau$ of $M$. If $M$ is understood from the context, we say that $\nu$ is a successful occurrence sequence.

A Petri net $N$ is said to be $k$-*bounded* if the number of tokens in each place does not exceed a finite number $k$ for any marking reachable from

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



the initial marking $\iota$, i.e., $\mu(p) \leqslant k$ for all $p \in P$ and for all $\mu \in \mathcal{R}(N, \iota)$. A Petri net $N$ is said to be *bounded* if it is k-bounded for some $k \geqslant 1$.

A Petri net with *place capacity* is a system $N = (P, T, F, \phi, \iota, \kappa)$ where $(P, T, F, \phi, \iota)$ is a marked Petri net and $\kappa : P \to \mathbb{N}_0$ is a function assigning to each place a number of maximal admissible tokens. A marking $\mu$ of the net $N$ is valid if $\mu(p) \leqslant \kappa(p)$, for each place $p \in P$. A transition $t \in T$ is *enabled* by a marking $\mu$ if additionally the successor marking is valid.

### 2.2.2 Special Petri Nets

It is known that many decision problems are equivalent to the reachability problem [46], which has been shown to be decidable. However, it has exponential space complexity [66], thus from a practical point of view, Petri nets may be too powerful to be analyzed. The result of this has been the definition of a number of subclasses of Petri nets in order to find a subclass with a smaller complexity and still adequate modeling power for practical purposes. These subclasses are defined by restrictions on their structure intended to improve their analyzability.

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 $t \in T$ (Figure 2.2). This means that there can not be concurrency but there can be conflict.

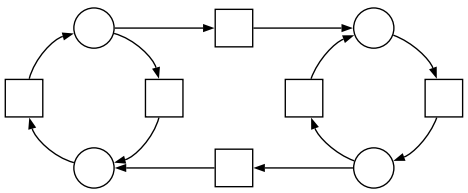

Figure 2.2: A state machine

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



- A *generalized state machine* (GSM) is an ordinary Petri net such that $|{}^\bullet t| \leqslant 1$ and $|t^\bullet| \leqslant 1$ for all $t \in T$ (Figure 2.3).

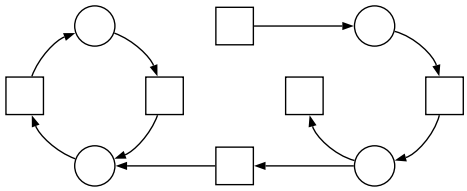

Figure 2.3: A generalized state machine

- 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 $p \in P$ (Figure 2.4). This means that there can not be conflict but there can be concurrency.

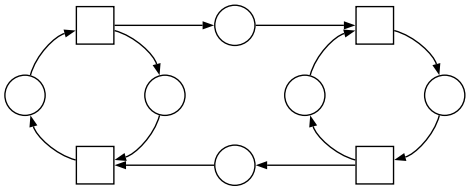

Figure 2.4: A marked graph

- A *generalized marked graph* (GMG) is an ordinary Petri net such that $|{}^\bullet p| \leqslant 1$ and $|p^\bullet| \leqslant 1$ for all $p \in P$ (Figure 2.5).

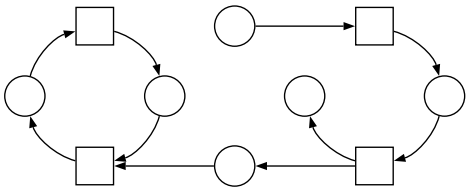

Figure 2.5: A generalized marked graph

- A *casual net* (CN) is a generalized marked graph each subgraph of which is not a a cycle (Figure 2.6).

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



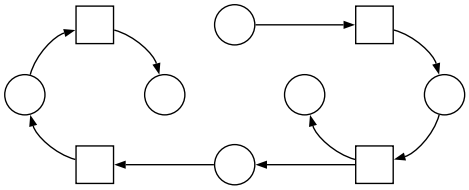

Figure 2.6: A casual net

- A *free-choice* net (FC) is an ordinary Petri net such every arc is either the only arc going from the place, or it is the only arc going to a transition, i.e., that if $p_1^\bullet \cap p_2^\bullet \neq \emptyset$ then $|p_1^\bullet| = |p_2^\bullet| = 1$ for all $p_1, p_2 \in P$ (Figure 2.7). This means that there can be both concurrency and conflict but not the same time.

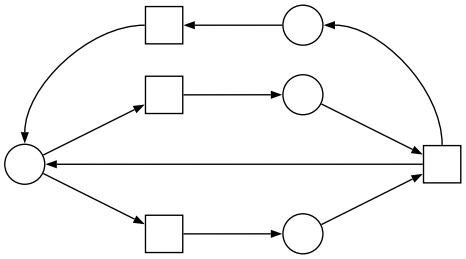

Figure 2.7: A free-choice net

- 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 $p_1, p_2 \in P$ (Figure 2.8).

- 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 $p_1, p_2 \in P$ (Figure 2.9). In asymmetric choice nets concurrency and conflict (in sum, confusion) may occur but not asymmetrically.

The hierarchy of the introduced subclasses of Petri nets is shown in Figure 2.10.

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



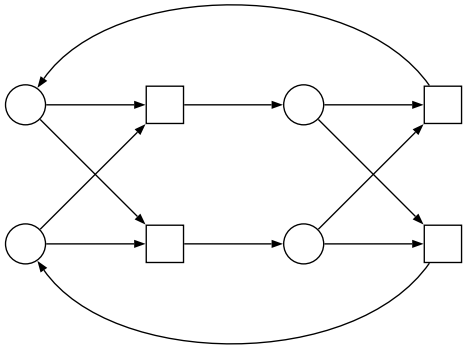

Figure 2.8: An extended free-choice net

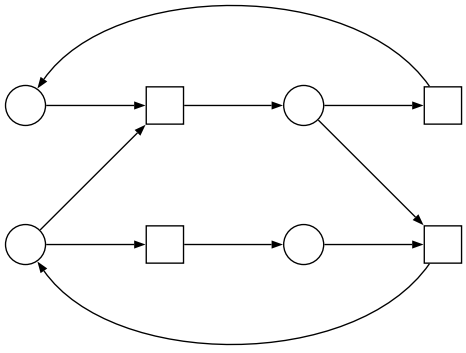

Figure 2.9: An asymmetric choice net

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



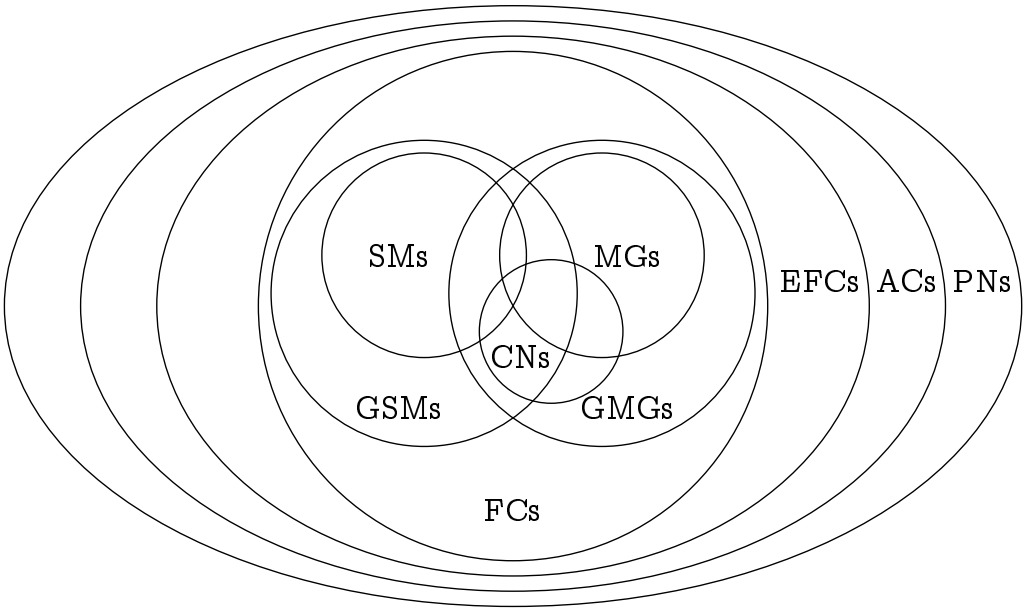


Figure 2.10: The hierarchy of Petri net classes

### 2.2.3 Petri Net Languages

Petri nets, which are graphical and mathematical modeling tools applicable to many discrete systems, have widely been used in the study of formal languages. One of the fundamental approaches in this area is to consider Petri nets as *language generators*. If the transitions in a Petri net are labeled with a set of (not necessary distinct) symbols, a sequence of transition firing generates a string of symbols. The set of strings generated by all possible firing sequences defines a language called a *Petri net language*. Petri net languages have received a lot of attention since the late seventies [55, 56, 58, 72, 74]. Comprehensive surveys on Petri net languages can be found in the work of Jantzen [56] and Peterson [74].

A *labeled Petri net* is a net $A = (P, T, F, \phi, \iota, M, \Sigma, \ell)$ where $N = (P, T, F, \phi, \iota, M)$ is a Petri net with final markings, $\Sigma$ is an alphabet and $\ell : T \to \Sigma \cup \{\lambda\}$ is a labeling function. The labeling function $\ell$ is extended to occurrence sequences in natural way, i.e., if $\sigma t \in T^*$ is an occurrence

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



sequence then $\ell(\sigma t) = \ell(\sigma)\ell(t)$ and $\ell(\lambda) = \lambda$ . For an occurrence sequence $\sigma \in T^*$, $\ell(\sigma)$ is called a *label sequence*.

In general, a language generated by a Petri net is a set of label sequences corresponding to occurrence sequences of the Petri net. Several varieties of Petri net languages result from the use of labeling policies and the definition of the set of final markings.

**Definition 2.1.** A Petri net language generated by a labeled Petri net $A = (P, T, F, \phi, \iota, M, \Sigma, \ell)$ is called

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

**Definition 2.2.** A Petri net language generated by a labeled Petri net $A = (P, T, F, \phi, \iota, M, \Sigma, \ell)$ is called

- P-*type* if $M$ is the set of all reachable markings from the initial marking $\iota$, i.e., $M = \mathcal{R}(N, \iota)$;
- L-*type* if $M \subseteq \mathcal{R}(N, \iota)$ is a finite set;
- G-*type* if for a given set $M_0 \subseteq \mathcal{R}(N, \iota)$, each marking $\mu \in M$ is greater or equal to any marking $M_0$;
- T-*type* if $M$ is the set of all terminal markings of $N$.

The Petri net in Figure 2.11 with a final marking set $M = \{(0, 0, 1, 0)\}$ results the following different types of Petri net languages:

- P-type: $\{a^m \mid m \geqslant 0\} \cup \{a^m c b^n \mid m \geqslant n \geqslant 0\} \cup \{a^m c b^n d \mid m \geqslant n \geqslant 0\}$;

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



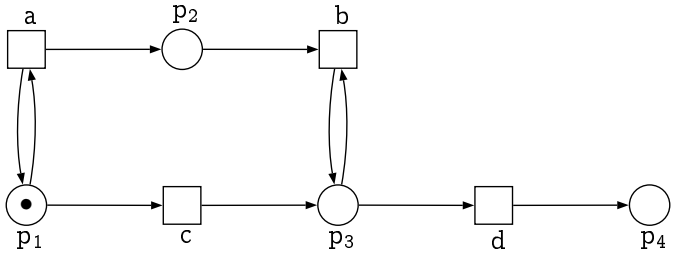


Figure 2.11: A labeled Petri net

- L-type: $\{a^m c b^m \mid m \geqslant 0\}$;
- G-type: $\{a^m c b^n \mid m \geqslant n \geqslant 0\}$;
- T-type: $\{a^m c b^n d \mid m \geqslant n \geqslant 0\}$.

$X^y$ denotes the family of X-type of Petri net languages with $y$-labeling policy where $X \in \{P, L, G, T\}$ and $y \in \{f, -\lambda, \lambda\}$. Then, the relationship of the families of Petri net languages is summarized in Figure 2.12 where an arrow between two classes of languages indicates proper containment.

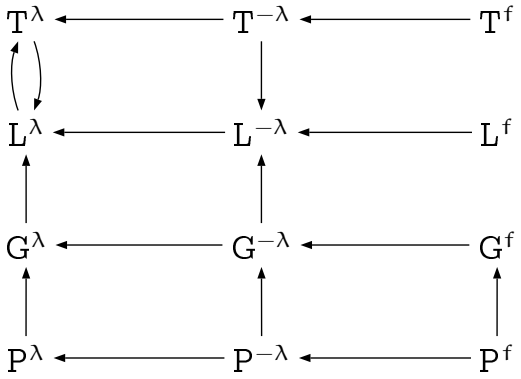


Figure 2.12: The relationship of the families of Petri net languages

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010

# 3

# Semi-Matrix Grammars

## 3.1 Introduction

Although matrix grammars [1] is one of the earliest and well investigated types of grammars with regulated rewriting, they still raise interesting questions, and find theoretical and practical applications. Various variants of matrix grammars such as unordered matrix grammars, vector grammars, simple matrix grammars, and etc., have also been introduced [17, 24, 23, 54, 77, 81, 90, 91]. The monograph [80] is entirely devoted to matrix grammars and their variants and present most of the known results.

Matrix grammars and their varieties are closely related to Petri nets: the restriction used in these grammars can be easily expressed in terms of Petri nets. For instance, in [48], the reduction to the reachability problems of Petri nets helped to solve a number of open problems in regulated rewriting systems, in particular, regularity of matrix languages over one letter alphabet and the emptiness problem for the family of matrix languages. The relationship between vector grammars and Petri nets was investigated in [59].

It will be shown in the next chapter that grammars controlled by specific types of Petri nets generate the families of regulated languages such as matrix and vector languages. In case of matrix and vector languages, controls used in matrix and vector grammars correspond to disjoint chains,

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



and cycles with the only common place in Petri nets, respectively. If we consider Petri nets with disjoint cycles, then grammars controlled by this type of Petri nets generate a new variant of matrix languages, called semi-matrix languages. Thus, it is interesting to investigate the properties of such kind of grammars. In the next section we give the formal definition of semi-matrix grammars, and afterwards we investigate the closure properties and generative capacities of the family of semi-matrix languages.

## 3.2 Definition and Examples

The *shuffle* of $k$ (not necessarily different) strings $w_1, w_2, \ldots, w_k \in X^*$ consists of all words $u$ of the form $u = u_{1,1}u_{1,2}\cdots u_{1,k}\cdots u_{m,1}u_{m,2}\cdots u_{m,k}$ with $u_{i,j} \in X^*$ and $u_{1,j}\cdots u_{m,j} = w_j$, for $1 \leqslant i \leqslant m$, $1 \leqslant j \leqslant k$. The *semi-shuffle* of $k$ (not necessarily different) strings $w_1, w_2, \ldots, w_k \in X^*$ is, informally spoken, the set of those words from the shuffle were only pairwise different strings are interleaved. Formally, let $\{v_1, v_2, \ldots, v_t\}$ be the set of pairwise different strings in $w_1, w_2, \ldots, w_k$ where $v_j$ is contained $n_j$ times. Then the semi-shuffle of $w_1, w_2, \ldots, w_k$ is defined as the shuffle of $v_1^{n_1}, v_2^{n_2}, \ldots, v_t^{n_t}$.

**Definition 3.1.** A *semi-matrix grammar* is a quadruple $G = (V, \Sigma, S, M)$ whose components are defined as for a matrix grammar. Its language $L(G)$ consists of all strings $w \in \Sigma^*$ such that there is a derivation $S \xRightarrow{r_1 r_2 \cdots r_n} w$ where $r_1 r_2 \cdots r_n$ is a semi-shuffle of some matrices $m_{i_1}, m_{i_2}, \ldots, m_{i_k} \in M$, $k \geqslant 1$.

From the definitions of matrix and vector grammars one can see that in a matrix grammar the shuffling of matrices is not allowed while in a vector grammar matrices can be shuffled. Semi-matrix grammars differ from the foregoing grammars by application of matrices in derivations: the shuffling of matrices is allowed only for different matrices. Semi-matrix grammars

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



complete the variations of matrix grammars with respect to the start of a matrix.

The difference of the controls used in matrix, vector and semi-matrix grammars can be seen in the following example.

*Example* 3.1. Let us consider a grammar G with the matrices

$$\begin{aligned} &m_0 : (S \to AB), \\ &m_1 : (A \to aA, B \to aB), \quad m_3 : (A \to a, B \to a), \\ &m_2 : (A \to bA, B \to bB), \quad m_4 : (A \to b, B \to b). \end{aligned}$$

- If G is a matrix grammar then $L(G) = \{ww \mid w \in \{a, b\}^+\}$.
- If G is a vector grammar then

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

- If G is a semi-matrix grammar then

$$\begin{aligned} L(G) \;=\; &\{w_1\{\lambda, ab, ba\}\cdots x \cdot w_1\{\lambda, ab, ba\}\cdots x \mid \\ &w_1, \ldots \in \{a, b\}^*, x \in \{a, b\}\}. \end{aligned}$$

Each derivation of the grammar G begins by using the matrix $m_0$. Then for the further steps of the derivation there are four possible cases:

1. The matrices $m_1$ and $m_2$ are applied without shuffling: the matrices $m_1$ and $m_2$ can be applied any times and in any order. We obtain a string $wAwB$, $w \in \{a, b\}^*$.
2. The matrices $m_1$ and $m_2$ are applied with shuffling: the first rule of $m_1$, then the first rule of $m_2$ or the first rule of $m_2$, then the first rule $m_1$ can be applied and the application of the matrices $m_1$ and $m_2$ are finished in the following orders: the second rule of $m_1$, then the

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



second rule of $m_2$ or the second rule of $m_2$, then the second rule of $m_1$, respectively. We obtain a string

$$w\{\lambda, ab, ba\}Aw\{\lambda, ab, ba\}B, w \in \{a, b\}^*.$$

3. The previous cases can be repeated any times and in any order. We obtain a string

$$w_1\{\lambda, ab, ba\}w_2\{\lambda, ab, ba\}\cdots Aw_1\{\lambda, ab, ba\}w_2\{\lambda, ab, ba\}\cdots B$$

where $w_1, w_2, \ldots \in \{a, b\}^*$.

4. In order to terminate the derivation, $m_3$ or $m_4$ is used. Eventually, we obtain a string

$$w_1\{\lambda, ab, ba\}w_2\{\lambda, ab, ba\}\cdots xw_1\{\lambda, ab, ba\}w_2\{\lambda, ab, ba\}\cdots x$$

where $w_1, w_2, \ldots \in \{a, b\}^*$, $x \in \{a, b\}$.

$sMAT^{\lambda}(sMAT)$ denotes the family of languages generated by the semi-matrix grammars (without erasing rules). The next statement follows immediately from the definition.

**Lemma 3.1.** $sMAT \subseteq sMAT^{\lambda}$.

We define the following binary form for semi-matrix grammars which will be used in the proofs of some statements in the next section.

**Definition 3.2.** A semi-matrix grammar $G = (V, \Sigma, S, M)$ is said to be in a *binary form* if for each rule $A \to \alpha$ in $M$, $|\alpha| \leqslant 2$.

Since each rule $r : A \to x_1x_2\cdots x_n$, $n \geqslant 3$, $x_i \in V \cup \Sigma$, $1 \leqslant i \leqslant n$, in a matrix $m$ of $M$ can be replaced by the sequence

$$A \to x_1B_{m,r}, B_{m,r} \to x_2B_{m,r}, \ldots, B_{m,r} \to x_{n-2}B_{m,r}, B_{m,r} \to x_{n-1}x_n$$

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



of rules where $B_{m,r}$ is a new nonterminal symbol with respect to the rule $r$ and the matrix $m$, one can construct an equivalent semi-matrix grammar $G'$ which is in the binary form, i.e.,

**Lemma 3.2.** For each semi-matrix language L there is a semi-matrix grammar in the binary form which generates the language L.

## 3.3 Closure Properties

By using standard proofs, we show that many closure properties for matrix grammars also hold for semi-matrix grammars.

**Lemma 3.3.** The families sMAT and $sMAT^{\lambda}$ are closed under union and concatenation.

*Proof.* Let $G_1 = (V_1, \Sigma_1, S_1, M_1)$ and $G_2 = (V_2, \Sigma_2, S_2, M_2)$ be semi-matrix grammars (with or without erasing rules). We assume that $V_1 \cap V_2 = \emptyset$. We set

$$V = V_1 \cup V_2 \cup \{S\} \text{ and } \Sigma = \Sigma_1 \cup \Sigma_2$$

where $S$ is a new nonterminal symbol.

We define the grammars $G' = (V, \Sigma, S, M')$ and $G'' = (V, \Sigma, S, M'')$ where

$$M'' = M_1 \cup M_2 \cup \{(S \to S_1), (S \to S_2)\}$$

and

$$M' = M_1 \cup M_2 \cup \{(S \to S_1S_2)\}.$$

It is not difficult to see that

$$L(G') = L(G_1) \cup L(G_2) \text{ and } L(G'') = L(G_1)L(G_2).$$

□

**Lemma 3.4.** The families sMAT and $sMAT^{\lambda}$ are closed mirror image.

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



*Proof.* Let $G = (V, \Sigma, S, M)$ be a semi-matrix grammar (with or without erasing rules). Let $M = \{m_1, m_2, \ldots, m_n\}$ where

$$m_i : (r_{i,1}, r_{i,2}, \ldots, r_{i,k(i)}) \text{ and } r_{i,j} : A_{i,j} \to \alpha_{i,j}, 1 \leqslant i \leqslant n, 1 \leqslant j \leqslant k(i).$$

With each matrix $m_i$, $1 \leqslant i \leqslant n$, we associate the matrix

$$\overline{m}_i : (\overline{r}_{i,1}, \overline{r}_{i,2}, \ldots, \overline{r}_{i,k(i)}), 1 \leqslant i \leqslant n,$$

where $\overline{r}_{i,1} : A_{i,j} \to \overline{\alpha}_{i,j}$ and $\overline{\alpha}_{i,j}$, $1 \leqslant i \leqslant n, 1 \leqslant j \leqslant k(i)$, is the mirror image of $\alpha_{i,j}$, i.e., $\overline{\alpha}_{i,j} = \alpha_{i,j}^R$. Let $\overline{M} = \{\overline{m}_1, \overline{m}_2, \ldots, \overline{m}_n\}$.

Clearly, $L(G') = L(G)^R$ for a semi-matrix grammar $G' = (V, \Sigma, S, \overline{M})$. □

**Lemma 3.5.** The families sMAT and $sMAT^\lambda$ are closed under substitution by λ-free context-free languages. The family $sMAT^\lambda$ is closed under substitution by arbitrary context-free languages.

*Proof.* Let $G = (V, \Sigma, S, M)$ be a semi-matrix grammar (with or without erasing rules) and $s : \Sigma^* \to 2^{\Delta^*}$ be a substitution where $s(a)$ is a context-free language for each $a \in \Sigma$. Let $G_a = (V_a, \Sigma_a, S_a, R_a)$ be a context-free grammar where $L(G_a) = s(a)$ for each $a \in \Sigma$. Without loss of generality we can assume that $V_a \cap V_b = \emptyset$ for all $a, b \in \Sigma$ with $a \neq b$ and $V_a \cap V = \emptyset$ for all $a \in \Sigma$. Let $\phi : V \cup \Sigma \to V \cup \{S_a \mid a \in \Sigma\}$ is the bijection defined by

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

Let $\overline{M}$ be the set of matrices obtained from $M$ by replacing each rule $r : A \to \alpha$ in $M$ by $A \to \phi(\alpha)$. We define a semi matrix grammar

$$G' = (V \cup \bigcup_{a \in \Sigma} V_a, \Sigma \cup \bigcup_{a \in \Sigma} \Sigma_a, S, \overline{M} \cup \bigcup_{a \in \Sigma} \{(A \to \alpha) \mid A \to \alpha \in R_a\}).$$

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



Obviously, $L(G') = s(L(G))$. □

**Lemma 3.6.** The families sMAT and $sMAT^{\lambda}$ are closed under intersection by regular sets.

*Proof.* Let $G = (V, \Sigma, S, M)$ be a semi-matrix grammar in the binary form without erasing rules and $A = (Q, \Sigma, \delta, q_0, F)$ be a finite automaton. We set

$$V' = \{(q_1, x, q_2) \mid q_1, q_2 \in Q, x \in V \cup \Sigma\}$$

and define the following sets of rules:

- for each rule $r : A \to xy$, $x, y \in V \cup \Sigma$ in $M$

$$A(r) = \{(q_1, A, q_2) \to (q_1, x, q)(q, y, q_2) \mid q, q_1, q_2 \in Q\};$$

- for each rule $r : A \to x$, $x \in V \cup \Sigma$ in $M$

$$A(r) = \{(q_1, A, q_2) \to (q_1, x, q_2) \mid q_1, q_2 \in Q\}.$$

Let $A(M)$ be the set of all matrices obtained from $M$ by replacing each rule $r$ by some rule of the set $A(r)$. We also define the set of matrices

$$M_\Sigma = \{((q_1, a, q_2) \to a) \mid q_2 = \delta(q_1, a), q_1, q_2 \in Q, a \in \Sigma\}.$$

For each $q \in F$ we construct the semi-matrix grammar

$$G_q = (V', \Sigma, (q_0, S, q), A(M) \cup M_\Sigma).$$

Then it is not difficult to see that $L(G) \cap L(A) = \bigcup_{q \in F} L(G_q)$ which is again a semi-matrix language. □

Using the same arguments of the proof of Lemma 1.3.2 in [24], one can show that

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



**Lemma 3.7.** The families sMAT and $sMAT^{\lambda}$ are closed under restricted homomorphisms.

Since, from the previous lemmas, the families sMAT and $sMAT^{\lambda}$ are closed under union, intersections with regular sets, substitutions by regular sets and intersections with regular sets, they also closed under inverse homomorphisms, by Theorem 2.1 in Chapter 2. Therefore, these families are semi-AFLs.

**Theorem 3.8.** The families sMAT and $sMAT^{\lambda}$ are semi-AFLs.

By Theorem 2.2 in Chapter 2, the next corollary is immediate

**Corollary 3.9.** The family sMAT and $sMAT^{\lambda}$ are closed under right and left derivatives.

## 3.4 Generative Capacity

First, we investigate the relationship of matrix-languages to matrix languages.

**Lemma 3.10.** $sMAT^{[\lambda]} \subseteq MAT^{[\lambda]}$.

*Proof.* Let $G = (V, \Sigma, S, M)$ be a semi-matrix grammar where

$$M = \{m_1, m_2, \ldots, m_n\}, m_i : (r_{i,1}, r_{i,2}, \ldots, r_{i,k(i)}), 1 \leqslant i \leqslant n.$$

We set

$$\begin{aligned} V' = V &\cup \{S', D_1, D_2, \ldots, D_n\} \\ &\cup \{D_{i,j} \mid 1 \leqslant i \leqslant n, 1 \leqslant j \leqslant k(i)\} \\ &\cup \{D'_{i,j} \mid 1 \leqslant i \leqslant n, 1 \leqslant j \leqslant k(i) - 1\} \end{aligned}$$

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



and $\Sigma' = \Sigma \cup \{c\}$ where

$$S', D_i, D_{i,j}, D'_{i,l}, 1 \leqslant i \leqslant n, 1 \leqslant j \leqslant k(i), 1 \leqslant l \leqslant k(i) - 1,$$

are new nonterminals and $c$ is a new terminal. With each matrix $m_i$, $1 \leqslant i \leqslant n$, we associate the matrices

$$(D_i \to D_{i,1}, r_{i,1}, D_{i,1} \to D'_{i,1})$$
$$(D'_{i,j} \to D_{i,j+1}, r_{i,j+1}, D_{i,j+1} \to D'_{i,j+1}), 1 \leqslant j \leqslant k(i) - 2$$
$$(D'_{i,k(i)-1} \to D_{i,k(i)}, r_{i,k(i)}, D_{i,k(i)} \to D_i).$$

Further we add the matrices

$$(S' \to SD_1D_2 \cdots D_n) \text{ and } (D_i \to c), 1 \leqslant i \leqslant n.$$

We define a matrix grammar $G' = (V', \Sigma', S', M')$ where $M'$ consists of the above-defined matrices.

It is not difficult to see that $L(G') = \{wc^k \mid w \in L(G)\}$, i.e.,

$$L(G) = \partial^r_{c^k}(L(G')).$$

Since the family of matrix languages are closed under right derivative, $L(G)$ is also a matrix language. □

**Lemma 3.11.** $\text{MAT}^{[\lambda]} \subseteq \text{sMAT}^{[\lambda]}$.

*Proof.* Let $G = (V, \Sigma, S, M)$ be a matrix grammar where

$$M = \{m_1, m_2, \ldots, m_n\}, m_i : (r_{i,1}, r_{i,2}, \ldots, r_{i,k(i)}), 1 \leqslant i \leqslant n.$$

We associate with each matrix $m_i$, $1 \leqslant i \leqslant n$, the matrix

$$m_i : (D \to D', r_{i,1}, r_{i,2}, \ldots, r_{i,k(i)}, D' \to D), 1 \leqslant i \leqslant n. \qquad (3.1)$$

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



where $D, D'$ are new nonterminals. Further we add the matrices

$$(S' \to DS) \text{ and } (D \to c) \qquad (3.2)$$

where $c$ is a new terminal symbol.

Then it is easy to see that the grammar

$$G' = (V \cup \{S', D, D'\}, \Sigma \cup \{c\}, S', M')$$

where $M'$ is the set of all matrices (3.1)-(3.2) generates the language

$$L(G') = \{cw \mid w \in L(G)\}, \text{ i.e., } L(G) = \partial_c^l(L(G')).$$

By Corollary 3.9, $L(G)$ is a semi-matrix language. □

We summarize the results of the previous lemmas and Theorem 2.1.2 in [24] in the following theorem.

**Theorem 3.12.**

$$MAT = sMAT \subseteq VEC \subseteq sMAT^{\lambda} = MAT^{\lambda} = VEC^{\lambda}.$$

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010

# 4

# Extended cf Petri Net Controlled Grammars

## 4.1 Introduction

Petri nets, which are graphical and mathematical modeling tools applicable to many concurrent, asynchronous, distributed, parallel, nondeterministic and stochastic systems, have widely been used in the study of formal languages. One of the fundamental approaches in this area is to consider Petri nets as language generators. If the transitions in a Petri net are labeled with a set of (not necessary distinct) symbols, a sequence of transition firing generates a string of symbols. The set of strings generated by all possible firing sequences defines a language called a Petri net language, which can be used to model the flow of information and control of actions in a system. With different kinds of labeling functions and different kinds of final marking sets, various classes of Petri net languages were introduced and investigated by Hack [47] and Peterson [73]. The relationship between Petri net languages and formal languages were thoroughly investigated by Peterson in [74]. It was shown that all regular languages are Petri net languages and the family of Petri net languages are strictly included in the family of context-sensitive languages but some Petri net languages are not context-free and some context-free languages are not Petri net languages. It was

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



also shown that the complement of a free Petri net language is context-free [19].

Another approach to the investigation of formal languages was considered by Crespi-Reghizzi and Mandrioli [18]. They noticed the similarity between the firing of a transition and application of a production rule in a derivation in which places are nonterminals and tokens are separate instances of the nonterminals. The major difference of this approach is the lack of ordering information in the Petri net contained in the sentential form of the derivation. To accommodate it, they defined the commutative grammars, which are isomorphic to Petri nets. In addition, they considered the relationship of Petri nets to matrix, scattered-context, nonterminal-bounded, derivation-bounded, equal-matrix and Szilard languages in [20].

This idea was used in the following works. By extending the type of Petri nets introduced in [18] with the places for the terminal symbols and arcs for the control of nonterminal occurrences in sentential forms, Marek and Češka showed that for every random-context grammar, an isomorphic Petri net can be constructed, where each derivation of the grammar is simulated by some occurrence sequence of transitions of the Petri net, and vice versa. In [59] the relationship between vector grammars and Petri nets was investigated, partially, hybrid Petri nets were introduced and the equality of the family of hybrid Petri net languages and the family of vector languages was shown. By reduction to Petri net reachability problems, Hauschildt and Jantzen [48] could solve a number of open problems in regulated rewriting systems, specifically, every matrix language without appearance checking over one letter alphabet is regular and the finiteness problem for the families of matrix and random context languages is decidable; In several papers [3, 4, 21, 34], Petri nets are used as minimization techniques for context-free (graph) grammars. For instance, in [3] and [4], algorithms to eliminate erasing and unit (chain) rules, algorithms to remove useless rules using the Petri net concept are introduced.

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



Petri nets have also been applied as control devices in grammar systems theory and automata theory. In [9], Beek and Kleijn demonstrated that Petri nets may be used to control the derivations in grammar systems. As a control mechanism they used a special type of a Petri net, called a Generalized Individual Token Net Controller (GITNC), which is a labeled Petri net with individual tokens – one for each component – which monitor the progress of the components. The transitions of the GITNC are labeled by vectors which describe a synchronous execution of components' actions. Such a vector label has one entry for each component: a nonempty entry represents an action to be executed by the corresponding component, while an empty entry indicates that the component is not involved in the synchronization. The occurrence of a transition implies a combined action of those components whose tokens are used by the transition. A Petri net grammar system was introduced as a system of grammars and a GITNC which describes a concurrent protocol for rewriting in the participating grammars. Each grammar of a PN grammar system has own sentential form. The grammars collaborate by synchronizing their actions according to the control exercised by the GITNC. A derivation in a PN grammar system starts from the axioms of the grammars with the controller in an initial marking. At each moment during the derivation rewritings are applied synchronously according to the labels of the transitions that occur. When all components have derived a terminal string and the GITNC is in one of its final markings, the derivation has been successful. The definition of Petri net control mechanisms for grammar systems creates the possibility for the study of concurrent rewriting protocols.

In [37, 38] concurrent Turing machines are introduced, where a Petri net replaces a finite automaton as finite control therein each token in the Petri net is associated with an individual tape head and tape heads can only be distinguished if they are associated with tokens on different places or have different positions on the tape. For the execution of a transition

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



with multiple input tokens, thus involving more than one head, the heads have to occupy the same position on the tape. Final marking and deadlock acceptance conditions are investigated. In [38] it is shown that concurrent Turing machines are equivalent to sequential Turing machines with respect to the acceptable languages. The definition of the control by Petri nets in Turing machines makes possible to use concurrency in automata theory, which differentiates the concurrent model of a Turing machine from the other kinds of its multi-head models. This model is also adopted to finite automata [36, 57]: the tape heads corresponding to the tokens put into the places of the post-condition of a transition point to the tape position immediately to the right of the previous one, or – in the case of a $\lambda$-move – to the same position.

Since a context-free grammar and its derivation process can also be described by a Petri net, where places correspond to nonterminals, transitions are the counterpart of the production rules, and the tokens reflect the occurrences of symbols in the sentential form, and there is a one-to-one correspondence between the application of (sequence of) rules and the firing of (sequence of) transitions, it is a very natural and very easy idea to control the derivations in a context-free grammar by adding some features to the associated Petri net. In the next section we introduce a Petri net associated with a context-free grammar (i.e., a *context-free Petri net*) and show that derivations of the grammar can be simulated by occurrence sequences of the net. In Section 4.3 we construct Petri net control mechanisms from cf Petri nets by adding new places, and define the corresponding grammars, called k-*Petri net controlled grammars*. Furthermore, we investigate fundamental properties of the families of languages generated by k-Petri net controlled grammars, in particular, we show that these families form infinite hierarchy with respect to the numbers of additional places. In section 4.4 we show that by adding some places and arcs which satisfy special requirements, precisely, the new places with transitions of a cf Petri

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



net form chains and cycles, one can generate families of vector, matrix and semi-matrix languages. Thus the control by Petri nets can be considered as a unifying approach to different types of control.

## 4.2 Context-Free 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 nonterminals and tokens are separate occurrences of nonterminals.

**Definition 4.1.** A *context-free Petri net* (in short, a *cf Petri net*) with respect to a context-free grammar $G = (V, \Sigma, S, R)$ is a septuple $N = (P, T, F, \phi, \beta, \gamma, \iota)$ where

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

*Example* 4.1. Let $G_1$ be a context-free grammar with the rules:

$$r_0 : S \to AB,\ r_1 : A \to aAb,\ r_2 : A \to ab,\ r_3 : B \to cB,\ r_4 : B \to c$$

(the other components of the grammar can be seen from these rules). Figure 4.1 illustrates a cf Petri net $N$ with respect to $G_1$. Obviously, $L(G_1) = \{a^n b^n c^m \mid n, m \geqslant 1\}$.

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



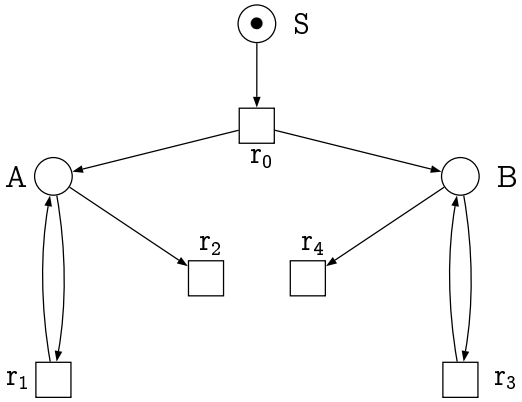


Figure 4.1: A cf Petri net N

The following proposition shows the similarity between terminal derivations in a context-free grammar and successful occurrences of transitions in the corresponding cf Petri net.

**Proposition 4.1.** Let $N = (P, T, F, \phi, \iota, \beta, \gamma)$ be the cf Petri net with respect to a context-free grammar $G = (V, \Sigma, S, R)$. Then $S \xRightarrow{r_1 r_2 \cdots r_n} w$, $w \in \Sigma^*$ is a derivation in G iff $t_1 t_2 \cdots t_n$, $\iota \xrightarrow{t_1 t_2 \cdots t_n} \mu_n$, 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 $\mu_n(p) = 0$ for all $p \in P$.

*Proof.* Let $S \xRightarrow{r_1 r_2 \cdots r_n} w, w \in \Sigma^*$ be a derivation in the grammar G. By induction on the number $1 \leqslant k \leqslant n$ of derivation steps, we show that $t_1 t_2 \cdots t_n$ with $\gamma(t_1 t_2 \cdots t_n) = r_1 r_2 \cdots r_n$ is an occurrence sequence enabled at $\iota$ and finished at the marking $\mu_n$ where $\mu_n(p) = 0$ for all $p \in P$.

Let $k = 1$. $S \Rightarrow_{r_1} w_1$, i.e., the sentential form $w_1$ is obtained from S by the application of a rule $r_1 : S \to w_1 \in R$. Then the transition $t_1 = \gamma^{-1}(r_1)$ also occurs as its input place $\beta^{-1}(S)$ has a token, i.e., by definition, $\iota(\beta^{-1}(S)) = 1$. Let $\iota \xrightarrow{t_1} \mu_1$. Then for each $A \in V$, we have $\mu_1(p) = |w_1|_A$ where $p = \beta^{-1}(A)$.

Suppose $S \xRightarrow{r_1 r_2 \cdots r_m} w_m, w_m \in (V \cup \Sigma)^*$, $1 \leqslant m \leqslant k - 1 < n$, and $t_1 t_2 \cdots t_m$ be an occurrence sequence of transitions of N such that

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



$\gamma(t_1t_2\cdots t_m) = r_1r_2\cdots r_m$. Consider case $m = k$. Then the transition $t_k = \gamma^{-1}(r_k)$, $r_k : A \to \alpha \in R$, can fire since ${}^\bullet t_k = \{\beta^{-1}(A)\}$ and $\mu_k(\beta^{-1}(A)) = |w_k|_A > 0$. If $k = n$, then $\mu_n(p) = 0$ for all $p \in P$ as $w_n \in \Sigma^*$, i.e., $|w_k|_A = 0$ for all $A \in V$.

Let $\nu = t_1t_2\cdots t_n$ be an occurrence sequence of transitions of N enabled at $\iota$ and finished at $\mu_n$ where $\mu_n(p) = 0$ for all $p \in P$. By induction on the number $1 \leqslant k \leqslant n$ of occurrence steps we show that $S \xRightarrow{r_1r_2\cdots r_n} w, w \in \Sigma^*$, is a derivation in G where $r_1r_2\cdots r_n = \gamma(t_1t_2\cdots t_n)$.

For $k = 1$ we have $\iota \xrightarrow{t_1} \mu_1$. Then the rule $r_1 = \gamma^{-1}(t_1) : S \to \alpha \in R$ can also be applied and $S \Rightarrow_{r_1} w_1 = \alpha$. By definition, for each $A \in V$, $|w_1|_A = \mu_1(\beta^{-1}(A))$.

We suppose that for $1 \leqslant m \leqslant k-1 < n$, $S \xRightarrow{r_1r_2\cdots r_m} w_m \in (V \cup \Sigma)^*$ is a derivation in G where $r_1r_2\cdots r_m = \gamma(t_1t_2\cdots t_m)$. Then for each $A \in V$ and $1 \leqslant i \leqslant m$, $|w_i|_A = \mu_i(p)$ where $A = \beta(p)$. If $m = k$, the rule $r_k : A \to \alpha \in R$, $r_k = \gamma(t_k)$, can be applied since $|w_k|_A > 0$. For $k = n$, $\mu_n(p) = 0$ for all $p \in P$ and consequently, $|w_n|_A = \mu_n(\beta^{-1}(A)) = 0$ for all $A \in V$, i.e., $w_n \in \Sigma^*$. □

## 4.3 k-Petri Net Controlled Grammars

### 4.3.1 Definitions and Examples

Now we define a k-*Petri net*, i.e., a cf Petri net with additional k places and additional arcs from/to these places to/from transitions of the net, the pre-sets and post-sets of the additional places are disjoint.

**Definition 4.2.** Let $G = (V, \Sigma, S, R)$ be a context-free grammar with its corresponding cf Petri net $N = (P, T, F, \phi, \beta, \gamma, \iota)$. Let k be a positive integer and let $Q = \{q_1, q_2, \ldots, q_k\}$ be a set of new places called *counters*. A k-*Petri net* is a construct $N_k = (P \cup Q, T, F \cup E, \varphi, \zeta, \gamma, \mu_0, \tau)$ where

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



- $E = \{(t, q_i) \mid t \in T_1^i, 1 \leqslant i \leqslant k\} \cup \{(q_i, t) \mid t \in T_2^i, 1 \leqslant i \leqslant k\}$ such that $T_1^i \subset T$ and $T_2^i \subset T$, $1 \leqslant i \leqslant k$ where $T_l^i \cap T_l^j = \emptyset$ for $1 \leqslant l \leqslant 2$, $T_1^i \cap T_2^j = \emptyset$ for $1 \leqslant i < j \leqslant k$ and $T_1^i = \emptyset$ if and only if $T_2^i = \emptyset$ for any $1 \leqslant i \leqslant k$.

- the weight function $\varphi(x, y)$ is defined by $\varphi(x, y) = \phi(x, y)$ if $(x, y) \in F$ and $\varphi(x, y) = 1$ if $(x, y) \in E$,

- the labeling function $\zeta : P \cup Q \to V \cup \{\lambda\}$ is defined by $\zeta(p) = \beta(p)$ if $p \in P$ and $\zeta(p) = \lambda$ if $p \in Q$,

- the initial marking $\mu_0$ is defined by $\mu_0(\beta^{-1}(S)) = 1$ and $\mu_0(p) = 0$ for all $p \in P \cup Q - \{\beta^{-1}(S)\}$,

- $\tau$ is the final marking where $\tau(p) = 0$ for all $p \in P \cup Q$.

**Definition 4.3.** A k-*Petri net controlled grammar* is a quintuple $G = (V, \Sigma, S, R, N_k)$ where $V, \Sigma, S, R$ are defined as for a context-free grammar and $N_k$ is a k-PN with respect to the context-free grammar $(V, \Sigma, S, R)$.

**Definition 4.4.** The *language* generated by a k-Petri net controlled grammar G consists of all strings $w \in \Sigma^*$ such that there is a derivation

$$S \xRightarrow{r_1 r_2 \cdots r_n} w \text{ where } t_1 t_2 \cdots t_n = \gamma^{-1}(r_1 r_2 \cdots r_n) \in T^*$$

is an occurrence sequence of the transitions of $N_k$ enabled at the initial marking $\iota$ and finished at the final marking $\tau$.

(This definition uses the extended form of the transition labeling function $\gamma : T^* \to R^*$; this extension is done in the usual manner.)

We denote the family of languages generated by k-PN controlled grammars (with erasing rules) by $PN_k$ ($PN_k^\lambda$), $k \geqslant 1$.

We give two examples which will be used in the sequel.

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



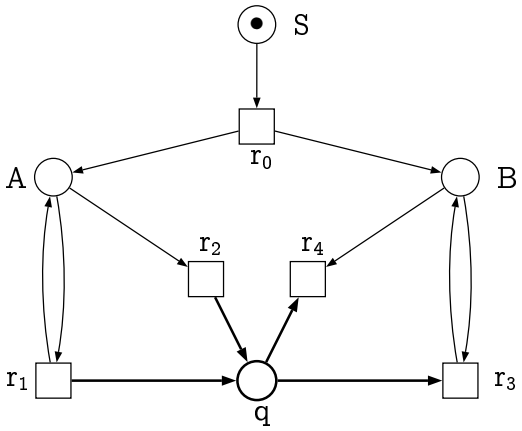


Figure 4.2: A 1-Petri net $N_1$

*Example* 4.2. Figure 4.2 illustrates a 1-Petri net $N_1$ which is constructed from the cf Petri net $N$ in Figure 4.1 adding a single counter place $q$. Let $G_2 = (V, \Sigma, S, R, N_1)$ be the 1-Petri net controlled grammar where $V, \Sigma, S, R$ are defined as for the grammar $G_1$ in Example 4.1. It is not difficult to see that $L(G_2) = \{a^n b^n c^n \mid n \geqslant 1\}$.

*Example* 4.3. Let $G_3$ be a 2-PN controlled grammar with the production rules:

$$\begin{array}{lll} r_0 : S \to A_1B_1A_2B_2, & r_1 : A_1 \to a_1A_1b_1, & r_2 : A_1 \to a_1b_1, \\ r_3 : B_1 \to c_1B_1, & r_4 : B_1 \to c_1, & r_5 : A_2 \to a_2A_2b_2, \\ r_6 : A_2 \to a_2b_2, & r_7 : B_2 \to c_2B_2, & r_8 : B_2 \to c_2 \end{array}$$

and the corresponding 2-Petri net $N_2$ is given in Figure 4.3. Then it is easy to see that $G_3$ generates the language

$$L(G_3) = \{a_1^n b_1^n c_1^n a_2^m b_2^m c_2^m \mid n, m \geqslant 1\}.$$

**Lemma 4.2.** The language $L' = \{a_1^n b_1^n c_1^n a_2^m b_2^m c_2^m \mid n, m \geqslant 1\}$ cannot be generated by a 1-PN controlled grammar.

*Proof.* Suppose the contrary: there is a 1-Petri net controlled grammar

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



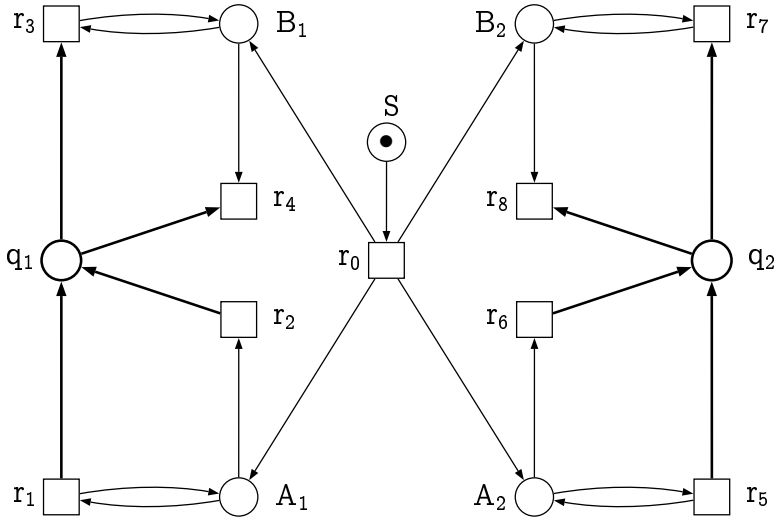


Figure 4.3: A 2-Petri net $N_2$

$G = (V, \Sigma, S, R, N_1)$ where $\Sigma = \{a_1, b_1, c_1, a_2, b_2, c_2\}$ such that $L(G) = L'$. Let $w = a_1^n b_1^n c_1^n a_2^m b_2^m c_2^m$. Since the set $V$ is finite, and if $n$ and $m$ are chosen sufficiently large, every derivation $S \Rightarrow^* w$ in G contains a subderivation of the form D: $A \Rightarrow^* xAy$ where $A \in V$ and $x, y \in \Sigma^*$ with $xy \neq \lambda$. As $L'$ is infinite, there are words with enough large length obtained by iterating such a derivation D arbitrarily many times. Suppose

$$S \Rightarrow^* uAv \Rightarrow^* uxAyv \Rightarrow^* \cdots \Rightarrow^* ux^n Ay^n v \Rightarrow^* w' \in \Sigma^* \qquad (4.1)$$

is also a derivation in G. Then $x^n$ and $y^n$ are substrings of $w'$. By the structure of the words of $L'$, $x$ and $y$ can be only powers of two symbols from $\Sigma \cup \{\lambda\}$. Therefore, in order to generate a word $w = a_1^n b_1^n c_1^n a_2^m b_2^m c_2^m \in L'$ for large $n$ and $m$, we need at least three subderivations of the form

$$D_1 : A_1 \Rightarrow^* x_1 A_1 y_1, \qquad (4.2)$$

$$D_2 : A_2 \Rightarrow^* x_2 A_2 y_2, \qquad (4.3)$$

$$D_3 : A_3 \Rightarrow^* x_3 A_3 y_3 \qquad (4.4)$$

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



where $x_1, x_2, x_3, y_1, y_2, y_3$ are powers of the symbols from $\Sigma$, i.e.,

$$x_i = \alpha_i^{k_i} \text{ and } y_i = \beta_i^{l_i} \text{ where } \alpha_i, \beta_i \in \Sigma \text{ and } k_i + l_i \geqslant 1, i = 1, 2, 3.$$

First, we assume that (4.1) has exactly three subderivations of the form (4.2)–(4.4). According to the production and consumption of tokens by the subderivations (4.2)–(4.4) the following cases can occur:

*Case* 1. One of the derivations (4.2)–(4.4) does not produce and consume any token. Without loss of generality we can assume that this derivation is (4.2). If $S \Rightarrow^* uA_1v \Rightarrow^* uwv \in L'$, then for any $k > 1$ we apply (4.2) $k$ times and get a string which is not in $L'$, i.e.,

$$S \Rightarrow^* uA_1v \Rightarrow^* ux_1A_1y_1v \Rightarrow^* ux_1^2A_1y_1^2v \Rightarrow^* ux_1^kA_1y_1^kv \Rightarrow^* ux_1^kwy_1^kv \notin L'$$

since (4.2) increases only the powers of at most two letters.

*Case* 2. One of the subderivations (4.2)–(4.4) produces tokens and another one consumes tokens. Without loss of generality we assume that (4.2) produces $p \geqslant 1$ tokens and (4.3) consumes $q \geqslant 1$ tokens.

Suppose

$$S \Rightarrow^* u_1A_1u_2A_2u_3 \Rightarrow^* u_1w_1u_2w_2u_3 \in L'.$$

Then the derivation

$$\begin{aligned} S &\Rightarrow^* u_1A_1u_2A_2u_3 \\ &\Rightarrow^* u_1x_1A_1y_1u_2A_2u_3 \Rightarrow^* u_1x_1^kA_1y_1^ku_2A_2u_3 \\ &\Rightarrow^* u_1x_1^kA_1y_1^ku_2x_2A_2y_2u_3 \Rightarrow^* u_1x_1^kA_1y_1^ku_2x_2^lA_2y_2^lu_3 \\ &\Rightarrow^* u_1x_1^kw_1y_1^ku_2x_2^lw_2y_2^lu_3 \end{aligned}$$

where $k, l \geqslant 1$, is also in $G$. It can be done by choosing the numbers $k, l$ in such a way, that $kp - lq = 0$, thus we can choose $k$ and $l$ as $k = q$ and $l = p$ and still get a string $w' \in L'$. Now

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



- if $1 \leqslant |\{\alpha_1, \beta_1, \alpha_2, \beta_2\} \cap \{a_i, b_i, c_i\}| \leqslant 2$, $i = 1$ or $i = 2$ then $w' \notin L'$ as the powers of at most two symbols are increased;

- if $\{\alpha_1, \beta_1, \alpha_2, \beta_2\} \cap \{a_i, b_i, c_i\} \neq \emptyset$ for both $i = 1$ and $i = 2$ then $1 \leqslant |\{\alpha_1, \beta_1, \alpha_2, \beta_2\} \cap \{a_i, b_i, c_i\}| \leqslant 2$ for $i = 1$ or $i = 2$ and again $w' \notin L'$.

From the above it follows that $\{\alpha_1, \beta_1, \alpha_2, \beta_2\} = \{a_i, b_i, c_i, \lambda\}$ for $i = 1$ or $i = 2$. Without loss of generality we assume that $i = 1$. But from the subderivation (4.4) (which produces or consumes tokens) it follows that $\alpha_3, \beta_3 \notin \{a_1, b_1, c_1\}$ and at least one of them belongs to $\{a_2, b_2, c_2\}$. Again we get the contradiction since (4.4) can increase the powers of at most two symbols from $\{a_2, b_2, c_2\}$. If the derivation has the form

$$S \Rightarrow^* u_1 A_1 u_4 \Rightarrow^* u_1 u_2 A_2 u_3 u_4 \Rightarrow^* u_1 u_2 w u_3 u_4,$$

then one gets that $\{x_1, y_1, x_2, y_2\}$ contains only two elements from $\Sigma$ and a contradiction follows as above.

*Case* 3. Two of the subderivations of (4.2)–(4.4) produce (consume) tokens and the other consumes (produces). Without loss of generality we assume that (4.2) and (4.3) produces $p_1$ and $p_2$ tokens, respectively and (4.4) consumes $q$ tokens. If

$$S \Rightarrow^* u_1 A_1 u_2 A_2 u_3 A_3 u_4 \Rightarrow^* u_1 w_1 u_2 w_2 u_3 w_3 u_4 \in L',$$

then the derivation

$$\begin{aligned} S &\Rightarrow^* u_1 A_1 u_2 A_2 u_3 A_3 u_4 \\ &\Rightarrow^* u_1 x_1 A_1 y_1 u_2 x_2 A_2 y_2 u_3 x_3 A_3 y_3 u_4 \\ &\Rightarrow^* u_1 x_1^{k_1} A_1 y_1^{k_1} u_2 x_2^{k_2} A_2 y_2^{k_2} u_3 x_3^{l} A_3 y_3^{l} u_4 \\ &\Rightarrow^* u_1 x_1^{k_1} w_1 y_1^{k_1} u_2 x_2^{k_2} w_2 y_2^{k_2} u_3 x_3^{l} w_3 y_3^{l} u_4 = w' \end{aligned} \qquad (4.5)$$

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



is also in G. By the definition of the final marking, we have $k_1p_1 + k_2p_2 - lq = 0$. For instance, if we choose $k_1, k_2, l$ as $k_1 = p_2q$, $k_2 = p_1q$ and $l = 2p_1p_2$, this equality holds. By structure of a derivation there are two possibilities:

$$\{\alpha_1, \beta_1, \alpha_2, \beta_2, \alpha_3, \beta_3\} = \{a_1, b_1, c_1, a_2, b_2, c_2, \lambda\} \tag{4.6}$$

or

$$\{\alpha_1, \beta_1, \alpha_2, \beta_2, \alpha_3, \beta_3\} = \{a_i, b_i, c_i, \lambda\} \text{ where } i = 1 \text{ or } i = 2. \tag{4.7}$$

Consider (4.6), here we only have the case $\alpha_1 = a_1$, $\beta_1 = b_1$, $\alpha_2 = c_1$, $\beta_2 = a_2$, $\alpha_3 = b_2$ and $\beta_3 = c_2$. It follows that the powers of all symbols of $w'$ are the same. But from (4.5), by continuing the derivation, we get a string which is not in $L'$:

$$\begin{aligned} S &\Rightarrow^* u_1x_1^{k_1}A_1y_1^{k_1}u_2x_2^{k_2}A_2y_2^{k_2}u_3x_3^{l}A_3y_3^{l}u_4 \\ &\Rightarrow^* u_1x_1^{k_1}w_1y_1^{k_1}u_2x_2^{k_2}w_2y_2^{k_2}u_3x_3^{l}A_3y_3^{l}u_4 \\ &\Rightarrow^* u_1x_1^{k_1}w_1y_1^{k_1}u_2x_2^{k_2}w_2y_2^{k_2}u_3x_3^{2l}A_3y_3^{2l}u_4 \\ &\Rightarrow^* u_1x_1^{k_1}w_1y_1^{k_1}u_2x_2^{2k_2}w_2y_2^{2k_2}u_3x_3^{3l}w_3y_3^{3l}u_4 \notin L' \end{aligned}$$

where the powers of four symbols are increased.

Now consider (4.7). Let $i = 1$. From Case 2, we can conclude that one of the following three cases is possible:

$$\begin{aligned} &(a)\quad \{\alpha_1, \beta_1\} = \{a_1, b_1\}, \quad \{\alpha_2, \beta_2\} = \{\lambda\}, \quad \{\alpha_3, \beta_3\} = \{c_1, \lambda\}, \\ &(b)\quad \{\alpha_1, \beta_1\} = \{\lambda\}, \quad \{\alpha_2, \beta_2\} = \{a_1, b_1\}, \quad \{\alpha_3, \beta_3\} = \{c_1, \lambda\}, \\ &(c)\quad \{\alpha_1, \beta_1\} = \{a_1, \lambda\}, \quad \{\alpha_2, \beta_2\} = \{b_1, \lambda\}, \quad \{\alpha_3, \beta_3\} = \{c_1, \lambda\}. \end{aligned}$$

Cases $(a)$ and $(b)$ are similar to *Case 2*. If we choose $k_1 = 3p_2l$, $k_2 = 2p_1l$ and $q = 5p_1p_2$ in case $(c)$, we again get different powers for symbols $a_1, b_1, c_1$, i.e., $w' \notin L'$.

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



Next, we analyze the general case: let the derivation (4.1) have $n \geqslant 4$ subderivations of the form $D_i : A_i \to x_i A_i y_i$ where $A_i \in V$, $x_i = \alpha_i^{l_i}$ and $y_i = \beta_i^{l'_i}$, $\alpha_i, \beta_i \in \Sigma$, $l_i + l'_i \geqslant 1$, $1 \leqslant i \leqslant n$. Without loss of generality we can assume that for some $1 \leqslant s \leqslant n-1$, the derivations $D_i$, $1 \leqslant i \leqslant s$, produce $p_i$ tokens and the derivations $D_j$, $s+1 \leqslant j \leqslant n$, consume $q_j$ tokens. If

$$\begin{aligned} S &\Rightarrow^* u_1 A_1 u_2 A_2 u_3 \cdots u_n A_n u_{n+1} \\ &\Rightarrow^* u_1 w_1 u_2 w_2 u_3 \cdots u_n w_n u_{n+1} = w \in L', \end{aligned} \tag{4.8}$$

then by assumption,

$$\begin{aligned} S &\Rightarrow^* u_1 A_1 u_2 A_2 u_3 \cdots u_n A_n u_{n+1} \\ &\Rightarrow^* u_1 x_1 A_1 y_1 u_2 x_2 A_2 y_2 u_3 \cdots u_n x_n A_n y_n u_{n+1} \\ &\Rightarrow^* u_1 x_1^{k_1} A_1 y_1^{k_1} u_2 x_2^{k_2} A_2 y_2^{k_2} u_3 \cdots u_n x_n^{k_n} A_n y_n^{k_n} u_{n+1} \\ &\Rightarrow^* u_1 x_1^{k_1} w_1 y_1^{k_1} u_2 x_2^{k_2} w_2 y_2^{k_2} u_3 \cdots u_n x_n^{k_n} w_n y_n^{k_n} u_{n+1} = w' \in L'. \end{aligned} \tag{4.9}$$

According to the definition of the final marking, we have

$$\sum_{i=1}^{s} k_i p_i - \sum_{i=s+1}^{n} k_i q_i = 0.$$

and

$$\{\alpha_1, \beta_1, \alpha_2, \beta_2, \ldots, \alpha_n, \beta_n\} = \{a_1, b_1, c_1, a_2, b_2, c_2, \lambda\}.$$

If for some $1 \leqslant i \leqslant n$, $\alpha_i = c_1$ and $\beta_i = a_2$, then all symbols in $w'$ have the same power. Then by continuing two subderivations one of which produces tokens and the other consumes, one increases the powers of at most four symbols, and get a string $w'' \notin L'$.

Let, for some $2 \leqslant i \leqslant n-2$,

$$\{\alpha_1, \beta_1, \alpha_2, \beta_2, \ldots, \alpha_i, \beta_i\} = \{a_1, b_1, c_1, \lambda\} \tag{4.10}$$

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



and

$$\{\alpha_{i+1}, \beta_{i+1}, \alpha_{i+2}, \beta_{i+2}, \ldots, \alpha_n, \beta_n\} = \{a_2, b_2, c_2, \lambda\}. \tag{4.11}$$

It follows that at least one of the subderivations which generate symbols in (4.10) (symbols in (4.11)) produces and another subderivation consumes tokens, since symbols $a_i, b_i, c_i$, $i = 1, 2$, have the same power. Then the tokens produced by a subderivation $D_j$, for some $1 \leqslant j \leqslant i$, can be consumed by a subderivation $D_k$, for some $i + 1 \leqslant k \leqslant n$ as the both group of subderivations use the same counter, which result that the powers of at most two symbols from $a_1, b_1, c_1$ and $a_2, b_2, c_2$ are increased, i.e., a string $w' \notin L'$ is generated. In all cases, we get contradiction to our assumption $L' = L(G)$. □

### 4.3.2 Hierarchy Results

We start with a simple fact.

**Lemma 4.3.** $CF \subset PN_1$.

*Proof.* It is clear that $CF \subseteq PN_1$ if we take $T_1 = T_2 = \emptyset$. From Example 4.2 it follows that $CF \subset PN_1$. □

Now we present some relations to (positive) additive valence languages.

**Lemma 4.4.** $PN_1^{[\lambda]} \subseteq pV^{[\lambda]}$.

*Proof.* Let $G = (V, \Sigma, S, R, N_1)$ be a 1-PN controlled grammar (with or without erasing rules) where $N_1 = (P \cup \{q\}, T, F \cup E, \varphi, \zeta, \gamma, \mu_0, \tau)$ is a corresponding 1-Petri net with the counter $q$ (with the notions of Definition 4.2). We define a positive valence grammar $G' = (V, \Sigma, S, R, \nu)$ where $V, \Sigma, S, R$ are defined as for the grammar $G$ and for each $r \in R$, the mapping $\nu$ is defined by

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



$$\nu(r) = \begin{cases} 1 & \text{if } \gamma^{-1}(r) \in {}^\bullet q, \\ -1 & \text{if } \gamma^{-1}(r) \in q^\bullet, \\ 0 & \text{otherwise.} \end{cases}$$

Let $S \stackrel{\pi}{\Rightarrow} w, w \in \Sigma^*$, $\pi = r_1 r_2 \cdots r_k$, be a derivation in G. Then $\nu = t_1 t_2 \cdots t_k = \gamma^{-1}(r_1 r_2 \cdots r_k)$ is an occurrence sequence of transitions of $N_1$ enabled at the initial marking $\mu_0$ and finished at the final marking $\tau$, i.e.,

$$\mu_0 \xrightarrow{t_1} \mu_1 \xrightarrow{t_2} \cdots \xrightarrow{t_k} \mu_k = \tau$$

By definition, if $|\nu|_t > 0$ for some $t \in {}^\bullet q$ then there is a transition $t' \in q^\bullet$ such that $|\nu|_{t'} > 0$. Let

$$U_1 = \{t_{1,1}, t_{1,2}, \ldots, t_{1,k_1}\} \subseteq {}^\bullet q \text{ where } |\nu|_{t_{1,j}} > 0, 1 \leqslant j \leqslant k_1$$

and

$$U_2 = \{t_{2,1}, t_{2,2}, \ldots, t_{2,k_2}\} \subseteq q^\bullet \text{ where } |\nu|_{t_{2,j}} > 0, 1 \leqslant j \leqslant k_2.$$

Since $\mu_i(q) \geqslant 0$ for each occurrence step $1 \leqslant i \leqslant k$, we have $|\nu|_{U_1} \geqslant |\nu|_{U_2}$, consequently, $\nu(r_1) + \nu(r_2) + \ldots + \nu(r_j) \geqslant 0$ for any $1 \leqslant j < k$ and from $\mu_0(q) = \tau(q) = 0$, $\tau \in M$, it follows that

$$\sum_{t \in U_1} |\nu|_t - \sum_{t \in U_2} |\nu|_t \stackrel{\text{def}}{=} \sum_{i=1}^{k} \nu(r_i) = 0.$$

Hence, $L(G) \subseteq L(G')$.

Let $D : S \xrightarrow{r_1 r_2 \cdots r_k} w \in \Sigma^*$ be a derivation in $G'$ where $\nu(r_1) + \nu(r_2) + \ldots + \nu(r_k) = 0$ and $\nu(r_1) + \nu(r_2) + \ldots + \nu(r_j) \geqslant 0$ for any $1 \leqslant j < k$. By construction of $G'$, D is also a derivation in $(V, \Sigma, S, R)$.

According to the bijection $\gamma : T \to R$, there is an occurrence sequence $\nu = t_1 t_2 \cdots t_k$, $\mu \xrightarrow{t_1} \mu_1 \xrightarrow{t_2} \cdots \xrightarrow{t_k} \mu_k$, in $N_1$ such that $\nu = \gamma^{-1}(r_1 r_2 \cdots r_k)$.

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



$\mu = \mu_0$ since D starts from S, i.e., $\mu_0(\beta^{-1}(S)) = 1$ and $\mu_0(\beta^{-1}(x)) = 0$ for all $x \in (V \cup \Sigma) - \{S\}$ as well as $\mu_0(q) = 0$.

Since $w \in \Sigma^*$, we have $\mu_k(\beta^{-1}(x)) = 0$ for all $x \in V$. From $\sum_{i=1}^{j} \nu(r_i) \geqslant 0$, it follows that $\mu_j(q) \geqslant 0$ for any $1 \leqslant j < k$.

$$\sum_{i=1}^{k} \nu(r_i) \stackrel{\text{def}}{=} \sum_{\gamma^{-1}(r) \in {}^\bullet q} \nu(r) + \sum_{\gamma^{-1}(r) \in q^\bullet} \nu(r) = 0$$

shows that $\mu_k(q) = 0$. Therefore $\mu_k = \tau$. Consequently, $L(G') \subseteq L(G)$. □

**Lemma 4.5.** $aV^{[\lambda]} \subset PN_2^{[\lambda]}$.

*Proof.* Let $G = (V, \Sigma, S, R, \nu)$ be an additive valence grammar (with or without erasing rules). Without loss of generality we can assume that $\nu(r) \in \{1, 0, -1\}$ for each $r \in R$ (Lemma 2.1.10 in [24]).

For each rule $r : A \to \alpha \in R$, $\nu(r) \neq 0$ we add a nonterminal symbol $A_r$ and a pair of rules $r' : A \to A_r$, $r'' : A_r \to \alpha$ and we set

$$\begin{aligned} V' =& V \cup \{A_r \mid r : A \to \alpha \in R, \nu(r) \neq 0\}, \\ R' =& R \cup \{r' : A \to A_r, r'' : A_r \to \alpha \mid r : A \to \alpha \in R, \nu(r) \neq 0\}. \end{aligned}$$

Let $N = (P, T, F, \phi, \beta, \gamma, \iota)$ be a cf Petri net with respect to the context-free grammar $(V', \Sigma, S, R')$. We construct a 2-Petri net $N_2 = (P \cup Q, T, F \cup E, \varphi, \zeta, \gamma, \mu_0, \tau)$ where $Q = \{q, q'\}$ and $E = F_1 \cup F_2$ with

$$\begin{aligned} F_1 =& \{(t, q) \mid t = \gamma^{-1}(r), r \in R \text{ and } \nu(r) = 1\} \\ & \cup \{(t', q') \mid t' = \gamma^{-1}(r'), r \in R \text{ and } \nu(r) = -1\}, \\ F_2 =& \{(q, t) \mid t = \gamma^{-1}(r), r \in R \text{ and } \nu(r) = -1\} \\ & \cup \{(q', t') \mid t' = \gamma^{-1}(r'), r \in R \text{ and } \nu(r) = 1\}. \end{aligned}$$

The rest components of $N_2$ are defined the same as those in the definition. Consider the 2-PN controlled grammar $G' = (V', \Sigma, S, R', N_2)$.

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



Let $D : S \stackrel{\pi}{\Rightarrow} w, w \in \Sigma^*$, $\pi = r_1 r_2 \cdots r_n$, be a derivation in $G'$. Then $\sigma = t_1 t_2 \cdots t_n = \gamma^{-1}(r_1 r_2 \cdots r_n)$ is an occurrence sequence enabled at the initial marking $\mu_0$ and finished at the final marking $\tau$. By construction,

$$\sum_{i=1}^{n} \nu(r_i) = \sum_{t \in {}^\bullet q} |\sigma|_t + \sum_{t \in q'^\bullet} |\sigma|_t - \sum_{t \in q^\bullet} |\sigma|_t - \sum_{t \in {}^\bullet q'} |\sigma|_t = 0$$

since

$$\sum_{t \in {}^\bullet q} |\sigma|_t = \sum_{t \in q^\bullet} |\sigma|_t = \sum_{i=1}^{n} \mu_i(q) \text{ and } \sum_{t \in {}^\bullet q'} |\sigma|_t = \sum_{t \in q'^\bullet} |\sigma|_t = \sum_{i=1}^{n} \mu_i(q').$$

It follows that $D$ is also a derivation in $G$.

Let $D' : S \stackrel{r_1 r_2 \cdots r_n}{\Longrightarrow} w, w \in \Sigma^*$ be a derivation in $G$. For each $1 \leqslant k \leqslant n$,

(1) if $\sum_{i=1}^{k} \nu(r_i) > 0$, then for the rule $r_k$ with $\nu(r_k) \in \{1, 0, -1\}$ in $G$ choose the rule $r_k$ in $G'$;

(2) if $\sum_{i=1}^{k} \nu(r_i) < 0$, then for the rule $r_k$ with $\nu(r_k) \neq 0$ in $G$ choose the rules $r'_k$ and $r''_k$ in $G'$; if $\nu(r_k) = 0$ then choose $r_k$ in $G'$.

(3) if $\sum_{i=1}^{k} \nu(r_i) = 0$, then for the rule $r_k$ with $\nu(r_k) \in \{-1, 0\}$ in $G$ choose the rule $r_k$ in $G'$; if $\nu(r_k) = 1$, then choose $r'_k$, $r''_k$ in $G'$.

Therefore $D'$ is also a derivation in $G'$. The strict inclusion follows from the fact that $\{a_1^n b_1^n c_1^n a_2^m b_2^m c_2^m \mid n, m \geqslant 1\} \in PN_2$ cannot be generated by an additive valence grammar (Example 2.1.7 in [24]). □

The following lemma shows that, for any $n \geqslant 1$, an $n$-PN controlled grammar generates a vector language.

**Lemma 4.6.** For $n \geqslant 1$, $PN_n^{[\lambda]} \subseteq VEC^{[\lambda]}$.

*Proof.* Let $G = (V, \Sigma, S, R, N_n)$ be an $n$-PN controlled grammar (with or without erasing rules) where $N_n = (P \cup Q, T, F \cup E, \varphi, \zeta, \gamma, \mu_0, \tau)$. Let

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



$Q = \{q_1, q_2, \ldots, q_n\}$ and

$$^\bullet q_k = \{t_{k,1,1}, t_{k,1,2}, \ldots, t_{k,1,s(k)}\}$$

where $t_{k,1,i} = \gamma^{-1}(r_{k,1,i})$, $r_{k,1,i} : A_{k,1,i} \to w_{k,1,i}$, $1 \leqslant k \leqslant n$, $1 \leqslant i \leqslant s(k)$, and

$$q_k^\bullet = \{t_{k,2,1}, t_{k,2,2}, \ldots, t_{k,2,l(k)}\}$$

where $t_{k,2,j} = \gamma^{-1}(r_{k,2,j})$, $r_{k,2,j} : A_{k,2,j} \to w_{k,2,j}$, $1 \leqslant k \leqslant n$, $1 \leqslant j \leqslant l(k)$.

Let

$$\beta(p_{k,1,i}) = A_{k,1,i},\ 1 \leqslant k \leqslant n,\ 1 \leqslant i \leqslant s(k)$$

and

$$\beta(p_{k,2,j}) = A_{k,2,j},\ 1 \leqslant k \leqslant n,\ 1 \leqslant j \leqslant l(k).$$

First, we construct a PN controlled grammar $G' = (V', \Sigma, S, R', N')$ in such a way that each counter place of $N'$ has a single input transition and a single output transition, and we show that the grammars G and $G'$ generate the same language.

We set

$$V' = V \cup \{B_{k,i,j}, C_{k,j,i} \mid 1 \leqslant k \leqslant n, 1 \leqslant i \leqslant s(k), 1 \leqslant j \leqslant l(k)\}$$

where $B_{k,i,j}$ and $C_{k,j,i}$, $1 \leqslant k \leqslant n$, $1 \leqslant i \leqslant s(k)$, $1 \leqslant j \leqslant l(k)$, are new nonterminals. $R'$ consists of the following rules

$$\begin{aligned} R' = (R - \{r_{k,1,i}, r_{k,2,j} \mid 1 \leqslant k \leqslant n,\ 1 \leqslant i \leqslant s(k),\ 1 \leqslant j \leqslant l(k)\}) \\ \cup \{r'_{k,1,i,j} : A_{k,1,i} \to B_{k,i,j} \mid 1 \leqslant k \leqslant n, 1 \leqslant i \leqslant s(k), 1 \leqslant j \leqslant l(k)\} \\ \cup \{r''_{k,1,i,j} : B_{k,i,j} \to w_{k,1,i} \mid 1 \leqslant k \leqslant n, 1 \leqslant i \leqslant s(k), 1 \leqslant j \leqslant l(k)\} \\ \cup \{r'_{k,2,j,i} : A_{k,2,j} \to C_{k,j,i} \mid 1 \leqslant k \leqslant n, 1 \leqslant i \leqslant s(k), 1 \leqslant j \leqslant l(k)\} \\ \cup \{r''_{k,2,j,i} : C_{k,j,i} \to w_{k,2,j} \mid 1 \leqslant k \leqslant n, 1 \leqslant i \leqslant s(k), 1 \leqslant j \leqslant l(k)\} \end{aligned}$$

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



and $N' = (P' \cup Q', T', F', \varphi', \zeta', \gamma', \mu_0', \tau')$ where the sets of places, transitions and arcs

$$\begin{aligned}
P' =& P \cup \{p_{k,1,i,j} \mid 1 \leqslant k \leqslant n,\ 1 \leqslant i \leqslant s(k),\ 1 \leqslant j \leqslant l(k)\} \\
& \cup \{p_{k,2,j,i} \mid 1 \leqslant k \leqslant n,\ 1 \leqslant i \leqslant s(k),\ 1 \leqslant j \leqslant l(k)\}, \\
Q' =& \{q_{k,i,j} \mid 1 \leqslant k \leqslant n,\ 1 \leqslant i \leqslant s(k), 1 \leqslant j \leqslant l(k)\}, \\
T' =& (T - \bigcup_{k=1}^{n} ({}^\bullet q_k \cup q_k^\bullet)) \\
& \cup \{t'_{k,1,i,j}, t''_{k,1,i,j} \mid 1 \leqslant k \leqslant n, 1 \leqslant i \leqslant s(k), 1 \leqslant j \leqslant l(k)\} \\
& \cup \{t'_{k,2,j,i}, t''_{k,2,j,i} \mid 1 \leqslant k \leqslant n, 1 \leqslant i \leqslant s(k), 1 \leqslant j \leqslant l(k)\},
\end{aligned}$$

$$\begin{aligned}
F' = (F \cup E - \bigcup_{k=1}^{n} & (\{(p_{k,1,i}, t_{k,1,i}), (t_{k,1,i}, q_k) \mid 1 \leqslant i \leqslant s(k)\} \\
& \cup \{(t_{k,1,i}, p) \mid p = \zeta^{-1}(x),\ |w_{k,1,i}|_x > 0, 1 \leqslant i \leqslant s(k)\} \\
& \cup \{(q_k, t_{k,2,j}), (p_{k,2,j}, t_{k,2,j}) \mid 1 \leqslant j \leqslant l(k)\} \\
& \cup \{(t_{k,2,j}, p) \mid p = \zeta^{-1}(x),\ |w_{k,2,j}|_x > 0, 1 \leqslant j \leqslant l(k)\})) \\
\cup \bigcup_{k=1}^{n} \bigcup_{i=1}^{s(k)} \bigcup_{j=1}^{l(k)} & (\{(p_{k,1,i}, t'_{k,1,i,j}), (t'_{k,1,i,j}, p_{k,1,i,j}), (p_{k,1,i,j}, t''_{k,1,i,j}), \\
& (t''_{k,1,i,j}, q_{k,i,j})\} \cup \{(t''_{k,1,i,j}, p) \mid p = \zeta^{-1}(x),\ |w_{k,1,i}|_x > 0\}) \\
\cup \bigcup_{k=1}^{n} \bigcup_{j=1}^{l(k)} \bigcup_{i=1}^{s(k)} & (\{(p_{k,2,j}, t'_{k,2,j,i}), (t'_{k,1,j,i}, p_{k,2,j,i}), (p_{k,2,j,i}, t''_{k,2,j,i}), \\
& (t''_{k,2,j,i}, q_{k,i,j})\} \cup \{(t''_{k,2,j,i}, p) \mid p = \zeta^{-1}(x),\ |w_{k,2,j}|_x > 0\}).
\end{aligned}$$

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



- The weight function is defined by

$$\varphi'(x,y)=\begin{cases}\varphi(x,y) & \text{if } (x,y)\in F,\\ \varphi(t_{k,1,i},p) & \text{if } x=t_{k,1,i,j}, y=p=\zeta^{-1}(x), |w_{k,1,i}|_x>0,\\ & \quad 1\leqslant k\leqslant n, 1\leqslant i\leqslant s(k), 1\leqslant j\leqslant l(k),\\ \varphi(t_{k,2,j},p) & \text{if } x=t_{k,2,j,i}, y=p=\zeta^{-1}(x), |w_{k,2,j}|_x>0,\\ & \quad 1\leqslant k\leqslant n, 1\leqslant i\leqslant s(k), 1\leqslant j\leqslant l(k),\\ 1 & \text{otherwise.}\end{cases}$$

- The labeling functions are defined by

$$\zeta'(p)=\begin{cases}\zeta(p) & \text{if } p\in P,\\ B_{k,i,j} & \text{if } p=p_{k,1,i,j}, 1\leqslant k\leqslant n, 1\leqslant i\leqslant s(k), 1\leqslant j\leqslant l(k),\\ C_{k,j,i} & \text{if } p=p_{k,2,j,i}, 1\leqslant k\leqslant n, 1\leqslant i\leqslant s(k), 1\leqslant j\leqslant l(k),\\ \lambda, & \text{if } p=q_{k,i,j}, 1\leqslant k\leqslant n, 1\leqslant i\leqslant s(k), 1\leqslant j\leqslant l(k)\end{cases}$$

and

$$\gamma'(t)=\begin{cases}\gamma(t) & \text{if } t\in T,\\ r'_{k,1,i,j} & \text{if } t=t'_{k,1,i,j}, 1\leqslant k\leqslant n, 1\leqslant i\leqslant s(k), 1\leqslant j\leqslant l(k),\\ r''_{k,1,i,j} & \text{if } t=t''_{k,1,i,j}, 1\leqslant k\leqslant n, 1\leqslant i\leqslant s(k), 1\leqslant j\leqslant l(k),\\ r'_{k,2,j,i} & \text{if } t=t'_{k,2,j,i}, 1\leqslant k\leqslant n, 1\leqslant j\leqslant l(k), 1\leqslant i\leqslant s(k),\\ r''_{k,2,j,i} & \text{if } t=t''_{k,2,j,i}, 1\leqslant k\leqslant n, 1\leqslant j\leqslant l(k), 1\leqslant i\leqslant s(k).\end{cases}$$

- The initial marking is defined by $\mu'_0(\zeta^{-1}(S))=1$ and $\mu'_0(p)=0$ for all $p\in P'\cup Q'-\{\zeta^{-1}(S)\}$.

- The final marking is defined by $\tau'(p)=0$ for all $p\in P'\cup Q'$.

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



By the construction of $N'$, an occurrence sequence of the form

$$\mu_1 \xrightarrow{t'_{k,1,i,j}} \mu_2 \xrightarrow{\sigma'} \mu_3 \xrightarrow{t''_{k,1,i,j}} \mu_4 \xrightarrow{\sigma''} \mu_5 \xrightarrow{t''_{k,2,j,i}} \mu_6 \xrightarrow{\sigma'''} \mu_7 \xrightarrow{t'_{k,2,j,i}} \mu_8 \qquad (4.12)$$

where $\sigma', \sigma'', \sigma''' \in T'^*$ can be replaced by

$$\mu_1 \xrightarrow{t'_{k,1,i,j}} \mu_2 \xrightarrow{t''_{k,1,i,j}\cdot\sigma'} \mu_4 \xrightarrow{\sigma''} \mu_5 \xrightarrow{\sigma'''\cdot t''_{k,2,j,i}} \mu_7 \xrightarrow{t'_{k,2,j,i}} \mu_8. \qquad (4.13)$$

Then, it is clear that (4.13) can be replaced in $N_n$ by

$$\mu_1 \xrightarrow{t_{k,1,i}} \mu' \xrightarrow{\sigma'\cdot\sigma''\cdot\sigma'''} \mu'' \xrightarrow{t_{k,2,j}} \mu_8.$$

Conversely, an occurrence sequence of the form

$$\mu_1 \xrightarrow{t_{k,1,i}} \mu_2 \xrightarrow{\sigma} \mu_3 \xrightarrow{t_{k,2,j}} \mu_4$$

in $N_n$ can be replaced in $N'$ by

$$\mu_1 \xrightarrow{t'_{k,1,i,j}} \mu' \xrightarrow{t''_{k,1,i,j}} \mu_2 \xrightarrow{\sigma} \mu_3 \xrightarrow{t'_{k,2,j,i}} \mu'' \xrightarrow{t''_{k,2,j,i}} \mu_4.$$

Correspondingly, without loss of generality we can change the order of the application of rules of derivations in the grammars $G$ and $G'$. Therefore, $L(G) = L(G')$.

Now we show that the grammar $G'$ generates a vector language. By the construction of $N'$, $|{}^\bullet q| = |q^\bullet| = 1$ for all $q \in Q'$.

We associate with each pair of rules $r_1, r_2 \in R'$ where $r_1 = \gamma'(t_1)$, $t_1 \in {}^\bullet q$ and $r_2 = \gamma'(t_2)$, $t_2 \in q^\bullet$, $q \in Q'$, the matrix $m = (r_1, r_2)$ and with each rule $r \in R' - \{r' = \gamma'(t') \mid t' \in {}^\bullet Q' \cup Q'^\bullet\}$, the matrix $m = (r)$. We consider a vector grammar $G'' = (V', \Sigma, S, M)$ where $M$ is the set of all matrices constructed above.

Let $S \overset{\pi}{\Rightarrow} w, w \in \Sigma^*$, $\pi = r_1 r_2 \cdots r_n$, is a derivation in $G'$ where $\iota \xrightarrow{\nu} \tau$ with $\nu = t_1 t_2 \cdots t_n = \gamma'^{-1}(\pi)$.

Let ${}^\bullet q = \{t\}$ and $q^\bullet = \{t'\}$ for some $q \in Q'$. If $t$ in $\nu$, i.e., $|\nu|_t > 0$ then $t'$ is also in $\nu$ and $|t_1 t_2 \cdots t_k|_t \geqslant |t_1 t_2 \cdots t_k|_{t'}$ for each $1 \leqslant k \leqslant n$, moreover, by the definition of the final marking, $|\nu|_t = |\nu|_{t'}$. By the bijection $\gamma'$, $m = (r, r')$, $r = \gamma'(t), r' = \gamma'(t')$ is in $\pi$ and $|r_1 r_2 \cdots r_k|_r \geqslant |r_1 r_2 \cdots r_k|_{r'}$ for each $1 \leqslant k \leqslant n$ as well as $|\pi|_r = |\pi|_{r'}$. Hence, $\pi \in \mathrm{Shuf}^*(M)$.

Let $S \stackrel{\pi}{\Rightarrow} w, w \in \Sigma^*$, $\pi = r_1 r_2 \cdots r_n \in \mathrm{Shuf}^*(M)$, be a derivation in $G''$ then again by the bijection $\gamma'$, $\nu = t_1 t_2 \cdots t_n = \gamma^{-1}(r_1 r_2 \cdots r_n)$ is an occurrence sequence of transitions of $N'$: $\mu_0 \stackrel{\nu}{\rightarrow} \mu_n$. Since the derivation $\pi$ starts from $S$ (i.e., $S$ is the only symbol at the starting sentential form), $\mu_0(\beta^{-1}(S)) = 1$ and $\mu_0(p) = 0$ for all $p \in P - \{\beta^{-1}(S)\}$. It follows that $\mu_0 = \mu'_0$. On the other hand, from $w \in \Sigma^*$, it follows that $\mu_n(\beta^{-1}(x)) = 0$ for all $x \in V$. From $\pi \in \mathrm{Shuf}^*(M)$, if the rules $r, r'$ of a matrix $m = (r, r')$ in $\pi$ then $|r_1 r_2 \cdots r_k|_r \geqslant |r_1 r_2 \cdots r_k|_{r'}$ for each $1 \leqslant k \leqslant n$ and $|\pi|_r = |\pi|_{r'}$. By the bijection $\gamma$, $|t_1 t_2 \cdots t_k|_t \geqslant |t_1 t_2 \cdots t_k|_{t'}$ for each $1 \leqslant k \leqslant n$ where $t = \gamma^{-1}(r), \gamma^{-1}(r')$ and $|\nu|_t = |\nu|_{t'}$. It follows that $\mu_n(q) = 0$ for all $q \in Q'$. Hence, $\mu_n = \tau'$. □

**Theorem 4.7.** For $k \geqslant 1$, $\mathrm{PN}_k^{[\lambda]} \subset \mathrm{PN}_{k+1}^{[\lambda]}$.

*Proof.* We first prove that $\mathrm{PN}_1^{[\lambda]} \subseteq \mathrm{PN}_2^{[\lambda]}$.

Let $G = (V, \Sigma, S, R, N_1)$ be a 1-PN controlled grammar (with or without erasing rules) where $N_1 = (P \cup \{q\}, T, F \cup E, \varphi, \zeta, \gamma, \mu_0, \tau)$ 1-Petri net with the counter place $q$. Let

$${}^\bullet q = \{t_{1,1}, t_{1,2}, \ldots, t_{1,k_1}\}, k_1 \geqslant 1 \text{ and } q^\bullet = \{t_{2,1}, t_{2,2}, \ldots, t_{2,k_2}\}, k_2 \geqslant 1$$

where $t_{i,j} = \gamma^{-1}(r_{i,j})$, $r_{i,j} : A_{i,j} \rightarrow w_{i,j}$, $1 \leqslant i \leqslant 2$, $1 \leqslant j \leqslant k_i$ and by definition ${}^\bullet q \cap q^\bullet = \emptyset$. Let $p_{i,j} = \zeta^{-1}(A_{i,j})$, $1 \leqslant i \leqslant 2$, $1 \leqslant j \leqslant k_i$.

We set

$$V' = V \cup \{B_{i,j} \mid 1 \leqslant i \leqslant 2, 1 \leqslant j \leqslant k_i\}$$

where $B_{i,j}$, $1 \leqslant i \leqslant 2$, $1 \leqslant j \leqslant k_i$, are new nonterminal symbols, introduced

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



for each transition $t_{i,j}$.

For each rule $r_{i,j} : A_{i,j} \to w_{i,j}$, $1 \leqslant i \leqslant 2$, $1 \leqslant j \leqslant k_i$, we add the new rules $r'_{i,j} : A_{i,j} \to B_{i,j}$, $r''_{i,j} : B_{i,j} \to w_{i,j}$. Let $R'$ be the set of all rules of $R$ and all rules constructed above, i.e.,

$$\begin{aligned} R' = R &\cup \{r'_{1,j} : A_{1,j} \to B_{1,j} \mid \gamma^{-1}(A_{1,j} \to w_{1,j}) \in {}^\bullet q, 1 \leqslant j \leqslant k_1\} \\ &\cup \{r''_{1,j} : B_{1,j} \to w_{1,j} \mid \gamma^{-1}(A_{1,j} \to w_{1,j}) \in {}^\bullet q, 1 \leqslant j \leqslant k_1\} \\ &\cup \{r'_{2,j} : A_{2,j} \to B_{2,j} \mid \gamma^{-1}(A_{2,j} \to w_{2,j}) \in q^\bullet, 1 \leqslant j \leqslant k_2\} \\ &\cup \{r''_{2,j} : B_{2,j} \to w_{2,j} \mid \gamma^{-1}(A_{2,j} \to w_{2,j}) \in q^\bullet, 1 \leqslant j \leqslant k_2\}. \end{aligned}$$

We construct a 2-PN controlled grammar $G' = (V', \Sigma, S, R', N_2)$ where $V'$ and $R'$ are defined above and $N_2 = (P', T', F', \varphi', \zeta', \gamma', \mu'_0, \tau')$ is constructed as follows:

$$\begin{aligned} P' &= P \cup \{p'_{i,j} \mid 1 \leqslant i \leqslant 2,\ 1 \leqslant j \leqslant k_i\} \cup \{q, q'\}, \\ T' &= T \cup \{t'_{i,j}, t''_{i,j} \mid 1 \leqslant i \leqslant 2,\ 1 \leqslant j \leqslant k_i\}, \\ F' &= F \cup \bigcup_{i=1}^{2} \bigcup_{j=1}^{k_i} (\{(p_{i,j}, t'_{i,j}), (t'_{i,j}, p'_{i,j}), (p'_{i,j}, t''_{i,j})\} \\ &\quad \cup \{(t''_{i,j}, p) \mid p = \zeta^{-1}(x),\ |w_{i,j}|_x > 0\}) \\ &\quad \cup \{(t''_{1,j}, q') \mid 1 \leqslant j \leqslant k_1\} \cup \{(q', t''_{2,j}) \mid 1 \leqslant j \leqslant k_2\}. \end{aligned}$$

For the weight function we set

$$\varphi'(x,y) = \begin{cases} \varphi(x,y) & \text{if } (x,y) \in F, \\ \varphi(t_{i,j}, p) & \text{if } x = t''_{i,j}, y = p = \zeta^{-1}(x), |w_{i,j}|_x > 0, \\ & \quad 1 \leqslant i \leqslant 2, 1 \leqslant j \leqslant k_i, \\ 1 & \text{otherwise.} \end{cases}$$

The initial and final markings are defined by $\mu'_0(\zeta'^{-1}(S)) = 1$, $\mu'_0(p) = 0$

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



for all $p \in P' - \{\zeta'^{-1}(S)\}$ and $\tau'(p) = 0$ for all $p \in P'$.

The inclusion $L(G) \subseteq L(G')$ is obvious, which directly follows from the construction of $G'$.

Let $S \stackrel{\pi}{\Rightarrow} w, w \in \Sigma^*$, $\pi = r_1 r_2 \cdots r_n$, be a derivation in $G'$ with the occurrence sequence $\nu = t_1 t_2 \cdots t_n = \zeta'^{-1}(\pi)$ of transitions of $N_2$ enabled at the initial marking $\mu'_0$ and finished at the final marking $\tau'$. It is clear that for some $1 \leqslant i \leqslant 2$, $1 \leqslant j \leqslant k_i$, if a rule $r'_{i,j} : A_{i,j} \to B_{i,j}$ in $\pi$, i.e., $|\pi|_{r'_{i,j}} > 0$, then the rule $r''_{i,j} : B_{i,j} \to w_{i,j}$ is also in $\pi$, i.e., $|\pi|_{r''_{i,j}} > 0$, moreover, $|\pi|_{r'_{i,j}} = |\pi|_{r''_{i,j}}$. Without loss of generality we can assume that a rule $r''_{i,j}$ is the next to a rule $r'_{i,j}$ in $\pi$ (as to the nonterminal $B_{i,j}$ only the rule $r''_{i,j}$ is applicable and we can change the order in which the derivation $\pi$ is used). Then we can replace any derivation steps of the form $x_1 A_{i,j} x_2 \Rightarrow_{r'_{i,j}} x_1 B_{i,j} x_2 \Rightarrow_{r''_{i,j}} x_1 w_{i,j} x_2$ by $x_1 A_{i,j} x_2 \Rightarrow_{r_{i,j}} x_1 w_{i,j} x_2$.

Accordingly, the occurrence sequence $t'_{i,j} t''_{i,j}$, $\mu \xrightarrow{t'_{i,j}} \mu' \xrightarrow{t''_{i,j}} \mu''$, is replaced by $t_{i,j}$, $\mu \xrightarrow{t_{i,j}} \mu''$, where $t_{i,j} = \gamma'^{-1}(r_{i,j})$, $t'_{i,j} = \gamma'^{-1}(r'_{i,j})$ and $t''_{i,j} = \gamma'^{-1}(r''_{i,j})$, $1 \leqslant i \leqslant 2$, $1 \leqslant j \leqslant k_i$. Clearly, $L(G') \subseteq L(G)$.

Let us consider the general case $k \geqslant 1$. Let $G = (V, \Sigma, S, R, N_k)$ be a k-Petri net controlled grammar where $N_k = (P \cup Q, T, F \cup E, \varphi, \zeta, \gamma, \mu_0, \tau)$ is a k-Petri net with $Q = \{q_1, q_2, \ldots, q_k\}$. We can repeat the arguments of the proof for $k = 1$ considering $q_k$ instead of $q$ and adding the new counter place $q_{k+1}$.

For $k \geqslant 1$, let the language $L_k$ be defined by

$$L_k = \{\prod_{i=1}^{k} a_i^{n_i} b_i^{n_i} c_i^{n_i} \mid n_i \geqslant 1, 1 \leqslant i \leqslant k\}.$$

Then we can show analogously to Example 4.3 and Lemma 4.2 that, for $k \geqslant 1$,

$$L_{k+1} \in PN_{k+1} \text{ and } L_{k+1} \notin PN_k.$$

Thus the inclusions are strict. □

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



### 4.3.3 Closure Properties

We define the following binary form for k-PN controlled grammars, which will be used in some of the next proofs.

**Definition 4.5.** A k-Petri net controlled grammar $G = (V, \Sigma, S, R, N_k)$ is said to be in a *binary form* if for each rule $A \to \alpha \in R$ the length of $\alpha$ is not greater than 2, i.e., $|\alpha| \leqslant 2$.

**Lemma 4.8** (Binary Form)**.** For each k-Petri net controlled grammar there exists an equivalent k-Petri net controlled grammar in the binary form.

*Proof.* Let $G = (V, \Sigma, S, R, N_k)$ be a k-Petri net controlled grammar with $N_k = (P \cup Q, T, F \cup E, \varphi, \zeta, \gamma, \mu_0, \tau)$.

We denote by $R^{>2}$ the set of all rules of the form $A \to \alpha \in R$ where $|\alpha| > 2$.

For each rule $r = A \to x_1 x_2 \cdots x_n \in R^{>2}$, $x_1, x_2, \ldots, x_n \in V \cup \Sigma$ we set

$$V_r = \{B_1, B_2, \ldots, B_{n-2}\}$$

and

$$R_r = \{A \to x_1 B_1, B_1 \to x_2 B_2, \ldots, B_{n-2} \to x_{n-1} x_n\}$$

where $B_i$, $1 \leqslant i \leqslant n-2$, are new nonterminal symbols, $V_r \cap V_{r'} = \emptyset$ for all $r, r' \in R$, $r \neq r'$, and $V_r \cap V = \emptyset$ for all $r \in R$. Let

$$V' = V \cup \bigcup_{r \in R^{>2}} V_r \text{ and } R' = (R \cup \bigcup_{r \in R^{>2}} R_r) - R^{>2}.$$

We define the context-free grammar $G' = (V', \Sigma, S, R')$ and construct a k-Petri net $N'_k = (P', T', F', \varphi', \zeta', \gamma', \mu'_0, \tau')$ with respect to $G'$ such that

(1) for $A \to \alpha \in R$, $|\alpha| \leqslant 2$,

$$\gamma^{-1}(A \to \alpha) \in {}^\bullet q \cup q^\bullet \text{ iff } \gamma'^{-1}(A \to \alpha) \in {}^\bullet q' \cup q'^\bullet,$$

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



(2) for $A \to \alpha \in R$, $|\alpha| > 2$,

$$\gamma^{-1}(A \to \alpha) \in {}^\bullet q \text{ iff } \gamma'^{-1}(B_{n-2} \to x_{n-1}x_n) \in {}^\bullet q', \tag{4.14}$$

$$\gamma^{-1}(A \to \alpha) \in q^\bullet \text{ iff } \gamma'^{-1}(A \to x_1B_1) \in q'^\bullet \tag{4.15}$$

where $\alpha = x_1x_2\cdots x_n$, $x_i \in V \cup \Sigma$, $1 \leqslant i \leqslant n$.

Let $D : S \xRightarrow{r_1r_2\cdots r_k} w, w \in \Sigma^*$ be a derivation in the grammar G. Then $t_1t_2\cdots t_k = \gamma^{-1}(r_1r_2\cdots r_k)$ is a successful occurrence sequence of transitions in $N_k$. We construct a derivation $D'$ in the grammar $G'$ from D as follows.

If for some $1 \leqslant m \leqslant k$, $r_m : A \to x_1x_2\cdots x_n \in R^{>2}$ then we replace the derivation step

$$y_1Ay_2 \xRightarrow{r_m} y_1x_1x_2\cdots x_ny_2$$

by the derivation steps

$$y_1Ay_2 \xRightarrow{r'_1} y_1x_1B_1y_2 \xRightarrow{r'_2} y_1x_1x_2B_2y_2 \xRightarrow{r'_3} \cdots \xRightarrow{r'_{n-2}} y_1x_1x_2\cdots x_ny_2$$

where $r'_i \in R_{r_m}$, $1 \leqslant i \leqslant n-2$. Correspondingly, $\mu_m \xrightarrow{t_m} \mu_{m+1}$ is replaced by

$$\mu_m \xrightarrow{t'_1t'_2\cdots t'_{n-2}} \mu_{m+1}$$

where $t'_i = \gamma'^{-1}(r'_i)$, $1 \leqslant i \leqslant n-2$. By (4.14)–(4.15), the number of tokens produced and consumed by the transitions $t'_1, t'_2, \ldots, t'_{n-2}$ and the transition $t_m$ are the same. Then $D'$ is a derivation in $G'$, which generates the same word as D does, i.e., $L(G) \subseteq L(G')$.

Inverse inclusion can also be shown using the similar arguments. □

**Lemma 4.9** (Union). The family of languages $PN_k^{[\lambda]}$, $k \geqslant 1$, is closed under union.

*Proof.* Let $G_1 = (V_1, \Sigma_1, S_1, R_1, N_{k,1})$ and $G_2 = (V_2, \Sigma_2, S_2, R_2, N_{k,2})$ be two k-PN controlled grammars with $N_{k,i} = (P_i \cup Q_i, T_i, F_i \cup E_i, \varphi_i, \zeta_i, \gamma_i, \mu_i, \tau_i)$,

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



$i = 1, 2$ (with the notions of Definition 4.2). We assume (without loss of generality) that $V_1 \cap V_2 = \emptyset$. We construct the k-PN controlled grammar

$$G = (V_1 \cup V_2 \cup \{S\}, \Sigma_1 \cup \Sigma_2, S, R_1 \cup R_2 \cup \{S \to S_1, S \to S_2\}, N_k)$$

where $N_k = (P, T, F, \varphi, \zeta, \gamma, \mu_0, \tau)$ is defined by

- the set of places: $P = P_1 \cup P_2 \cup Q_1 \cup \{q\}$ where $q$ is a new place;
- the set of transitions: $T = T_1 \cup T_2 \cup \{t_{01}, t_{02}\}$ where $t_{01}$ and $t_{02}$ are new transitions;
- the set of arcs:

$$\begin{aligned} F = F_1 \cup F_2 \cup E_1 \cup \{&(q, t_{0i}), (t_{0i}, p_{0i}) \mid i = 1, 2\} \\ &\cup \{(t, q_{1i}) \mid (t, q_{2i}) \in E_2, 1 \leqslant i \leqslant k\} \\ &\cup \{(q_{1i}, t) \mid (q_{2i}, t) \in E_2, 1 \leqslant i \leqslant k\} \end{aligned}$$

  where $p_{0i}$ are the places labeled by $S_i$, i.e., $\zeta_i(p_{0i}) = S_i$, $i = 1, 2$;

- the weight function:

$$\varphi(x, y) = \begin{cases} \varphi_i(x, y) & \text{if } (x, y) \in F_i, i = 1, 2, \\ 1 & \text{otherwise;} \end{cases}$$

- the labeling function $\zeta$ is defined by

$$\zeta(p) = \begin{cases} \zeta_1(p) & \text{if } p \in P_1 \cup Q_1, \\ \zeta_2(p) & \text{if } p \in P_2 \\ S & \text{if } p = q; \end{cases}$$

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



- the labeling function $\gamma$ is defined by

$$\gamma(t) = \begin{cases} \gamma_i(t) & \text{if } t \in T_i, i = 1, 2, \\ S \to S_i & \text{if } t = t_{0i}, i = 1, 2; \end{cases}$$

- the initial marking:

$$\mu_0(p) = \begin{cases} 1 & \text{if } p = q, \\ 0 & \text{otherwise;} \end{cases}$$

- the final marking: $\tau(p) = 0$ for all $p \in P$.

By the construction of $N_k$ any occurrence of its transitions can start by firing of $t_{01}$ or $t_{02}$ then transitions of $T_1$ or transitions of $T_2$ can occur, correspondingly we start a derivation with the rule $S \to S_1$ or $S \to S_2$ then we can use rules of $R_1$ or $R_2$.

A string $w$ is in $L(G)$ if and only if there is a derivation $S \Rightarrow S_i \Rightarrow^* w \in L(G_i)$, $i = 1, 2$. On the other hand, we can initialize any derivation $S_i \Rightarrow^* w \in L(G_i)$ with the rule $S \to S_i$, $i = 1, 2$, i.e., $w \in L(G)$. □

**Lemma 4.10** (Concatenation)**.** The family of languages $PN_k$, $k \geqslant 1$, is not closed under concatenation.

*Proof.* Let $L_k$ and $L'_k$ be two languages, with the same structure but disjoint alphabets, given at the end of the proof of Theorem 4.7. Then $L_k, L'_k \in PN_k$ and $L_k \cdot L'_k \notin PN_k$. □

The next lemma shows that the concatenation of two languages generated by k- and m-PN controlled grammars, $k, m \geqslant 1$, can be generated by a $(k + m)$-PN controlled grammar.

**Lemma 4.11.** For $L_1 \in PN_k^{[\lambda]}$, $k \geqslant 1$ and $L_2 \in PN_m^{[\lambda]}$, $m \geqslant 1$,

$$L_1 \cdot L_2 \in PN_{k+m}^{[\lambda]}.$$

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



*Proof.* Let $G_1 = (V_1, \Sigma, S_1, R_1, N_k)$ where $N_k = (P_1, T_1, F_1, \varphi_1, \zeta_1, \gamma_1, \mu_1, \tau_1)$ and $G_2 = (V_2, \Sigma, S_2, R_2, N_m)$ where $N_m = (P_2, T_2, F_2, \varphi_2, \zeta_2, \gamma_2, \mu_2, \tau_2)$ be, respectively, k-Petri net and m-Petri net controlled grammars such that $L(G_1) = L_1$ and $L(G_2) = L_2$. Without loss of generality we assume that $V_1 \cap V_2 = \emptyset$. We set $V = V_1 \cup V_2 \cup \{S\}$ where $S$ is a new nonterminal and

$$R = R_1 \cup R_2 \cup \{S \to S_1 S_2\}.$$

We define a $(k + m)$-PN controlled grammar $G = (V, \Sigma, S, R, N_{k+m})$ with $N_{k+m} = (P, T, F, \varphi, \zeta, \gamma, \mu_0, \tau)$ where

- $P = P_1 \cup P_2 \cup \{p_0\}$ where $p_0$ is a new place;
- $T = T_1 \cup T_2 \cup \{t_0\}$ where $t_0$ is a new transition;
- $F = F_1 \cup F_2 \cup \{(p_0, t_0), (t_0, p_1), (t_0, p_2)\}$ where $\zeta_i(p_i) = S_i$, $i = 1, 2$;
- the weight function $\varphi$ is defined by

$$\varphi(x, y) = \begin{cases} \varphi_i(x, y) & \text{if } (x, y) \in F_i, i = 1, 2, \\ 1 & \text{otherwise;} \end{cases}$$

- the labeling function $\zeta$ is defined by

$$\zeta(p) = \begin{cases} \zeta_i(p) & \text{if } p \in P_i, i = 1, 2, \\ S & \text{if } p = p_0; \end{cases}$$

- the labeling function $\gamma$ is defined by

$$\gamma(t) = \begin{cases} \gamma_i(t) & \text{if } t \in T_i, i = 1, 2, \\ S \to S_1 S_2 & \text{if } t = t_0; \end{cases}$$

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



- the initial marking:

$$\mu_0(p) = \begin{cases} 1 & \text{if } p = p_0, \\ 0 & \text{otherwise;} \end{cases}$$

- the final marking: $\tau(p) = 0$ for all $p \in P$.

It is not difficult to see that $L(G) = L(G_1)L(G_2)$. □

**Lemma 4.12** (Substitution)**.** The family of languages $PN_k$, $k \geqslant 1$, is closed under substitution by context-free languages.

*Proof.* Let $G = (V, \Sigma, S, R, N_k)$ be a k-PN controlled grammar with k-Petri net $N_k = (P \cup Q, T, F \cup E, \varphi, \zeta, \gamma, \mu_0, \tau)$. We consider a substitution $s : \Sigma^* \to 2^{\Delta^*}$ with $s(a) \in CF$ for each $a \in \Sigma$. Let $G_a = (V_a, \Sigma_a, S_a, R_a)$ be a context-free grammar for $s(a)$, $a \in \Sigma$. We can assume that $V \cap V_a = \emptyset$ for any $a \in \Sigma$ and $V_a \cap V_b = \emptyset$ for any $a, b \in \Sigma$, $a \neq b$.

Let $N_a = (P_a, T_a, F_a, \phi_a, \beta_a, \gamma_a, \iota_a)$ be a cf Petri net with respect to the grammar $G_a$, $a \in \Sigma$. We define the k-PN controlled grammar

$$G' = (V \cup \Sigma \cup \bigcup_{a \in \Sigma} V_a, \Delta, S, R' \cup \bigcup_{a \in \Sigma} R_a, N'_k)$$

where $R'$ is the set of rules obtained by replacing each occurrence of $a \in \Sigma$ by $S_a$ in $R$ and $N'_k$ is defined by

$$N'_k = (P \cup Q \cup P_\Sigma \cup \bigcup_{a \in \Sigma} P_a, T \cup \bigcup_{a \in \Sigma} T_a, F \cup F_\Sigma \cup \bigcup_{a \in \Sigma} F_a, \varphi', \zeta', \gamma', \mu'_0, \tau')$$

where

- $P_\Sigma = \{p_a \mid a \in \Sigma\}$ is the set of new places;
- $F_\Sigma = \{(t, p_a) \mid \gamma(t) = A \to \alpha, |\alpha|_a > 0, a \in \Sigma\}$ is the set of new arcs;

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



- the weight function $\varphi'$ is defined by

$$\varphi'(x,y) = \begin{cases} \varphi(x,y) & \text{if } (x,y) \in F, \\ \phi_a(x,y) & \text{if } (x,y) \in F_a, a \in \Sigma, \\ |\alpha|_a, & \text{if } x = t, y = p_a, (t,p_a) \in F_\Sigma, a \in \Sigma; \end{cases}$$

- the labeling function $\zeta'$ is defined by

$$\zeta'(p) = \begin{cases} \zeta(p) & \text{if } p \in (P \cup Q), \\ \beta_a(p) & \text{if } p \in P_a, a \in \Sigma, \\ S_a & \text{if } p = p_a \in P_\Sigma, a \in \Sigma; \end{cases}$$

- the labeling function $\gamma'$ is defined by

$$\gamma'(t) = \begin{cases} \gamma(t) & \text{if } t \in T, \\ \gamma_a(t) & \text{if } t \in T_a, a \in \Sigma; \end{cases}$$

- the initial marking:

$$\mu_0'(p) = \begin{cases} 1 & \text{if } p = \zeta'^{-1}(S), \\ 0 & \text{otherwise;} \end{cases}$$

- the final marking: $\tau'(p) = 0$ for all $p \in P'$;

Obviously, $L(G') \in PN_k$. □

**Lemma 4.13** (Mirror Image)**.** The family of languages $PN_k$, $k \geqslant 1$, is closed under mirror image.

*Proof.* Let $G = (V, \Sigma, S, R, N_k)$ be a k-PN controlled grammar. Let

$$R^- = \{A \to x_n \cdots x_2 x_1 \mid A \to x_1 x_2 \cdots x_n \in R\}.$$

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



The context-free grammar $(V, \Sigma, S, R)$ and its reversal $(V, \Sigma, S, R^{-})$ have the same corresponding cf Petri net $N = (P, T, F, \phi, \beta, \gamma, \iota)$ as $N$ does not preserve the order of the positions of the output places for each transition. Thus we can also use the k-Petri net $N_k$ as a control mechanism for the grammar $(V, \Sigma, S, R^{-})$, i.e., we define $G^{-} = (V, \Sigma, S, R^{-}, N_k)$. Clearly, $L(G^{-}) \in PN_k$. □

**Lemma 4.14** (Intersection with Regular Languages)**.** The family of languages $PN_k$, $k \geqslant 1$, is closed under intersection with regular languages.

*Proof.* We use the arguments and notions of the proof of Lemma 1.3.5 in [24]. Let $G = (V, \Sigma, S, R, N_k)$ be a k-Petri net controlled grammar with a k-Petri net $N_k = (P \cup Q, T, F \cup E, \varphi, \zeta, \gamma, \mu_0, \tau)$ (with the notions of Definition 4.2). Without loss of generality we can assume that $G$ is in a binary form.

Let $\mathcal{A} = (K, \Sigma, s_0, \delta, H)$ be a deterministic finite automaton. We set

$$V' = \{[s, x, s'] \mid s, s' \in K,\ x \in V \cup \Sigma\}.$$

For each rule $r \in R$ we construct the set $R(r)$ in the following way

1. If $r = A \to x_1 x_2,\ x_1, x_2 \in V \cup \Sigma$ then

$$R(r) = \{[s, A, s'] \to [s, x_1, s'][s', x_2, s''] \mid s, s', s'' \in K\}.$$

2. If $r = A \to x,\ x \in V \cup \Sigma$ then

$$R(r) = \{[s, A, s'] \to [s, x, s'] \mid s, s' \in K\}.$$

Further we define the set of rules

$$R_\Sigma = \{[s, a, s'] \to a \mid s' = \delta(s, a), s, s' \in K, a \in \Sigma\}.$$

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



Let

$$R' = \bigcup_{r \in R} R(r) \cup R_\Sigma.$$

We define the context-free grammar $G_s = (V', \Sigma, [s_0, S, s], R')$ for each $s \in H$. Let $N_s = (P_s, T_s, F_s, \phi_s, \beta_s, \gamma_s, \iota_s)$ be a cf Petri net with respect to the grammar $G_s$ where

$$P_s = \{[s, p, s'] \mid s, s' \in K, p \in P\},$$
$$T_s = \{[s, t, s'] \mid s, s' \in K, p \in P\},$$
$$F_s = \{([s_1, x, s_2], [s_1', y, s_2']) \mid s_1, s_2, s_1', s_2' \in K, (x, y) \in F\}.$$

The weight function $\phi_s$ is defined by $\phi([s_1, x, s_2], [s_1', y, s_2']) = \phi(x, y)$ where $s_1, s_2, s_1', s_2' \in K, (x, y) \in F$.

The functions $\beta_s : P_s \to V'$ and $\gamma_s : T_s \to R'$ are bijections, and

$$\iota_s(\beta_s^{-1}([s_0, S, s])) = 1 \text{ and } \iota_s(p) = 0 \text{ for all } P_s - \{\beta_s^{-1}([s_0, S, s])\}.$$

We set

$$F_Q^- = \{((s, t, s'), q) \mid s, s' \in K, q \in Q \wedge t \in {}^\bullet q\}$$

and

$$F_Q^+ = \{(q, (s, t, s')) \mid s, s' \in K, q \in Q \wedge t \in q^\bullet\}.$$

We construct the k-Petri net

$$N_{k,s} = (P_s \cup Q, T_s, F_s \cup F_Q^- \cup F_Q^+, \varphi_s, \zeta_s, \gamma_s, \mu_s, \tau_s)$$

from $N_s$ where

- the weight function $\varphi_s$ is defined by

$$\varphi_s([s_1, x, s_2], [s_1', y, s_2']) = \varphi(x, y), s_1, s_1', s_2, s_2' \in K \text{ and } (x, y) \in F \cup E,$$

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



- the labeling function $\zeta_s$ is defined by

$$\zeta_s([s_1, p, s_2]) = \begin{cases} \beta_s([s_1, p, s_2]) & \text{if } [s_1, p, s_2] \in P_s, \\ \lambda & \text{if } [s_1, p, s_2] \in Q, \end{cases}$$

- the initial marking $\mu_s$ is defined by $\mu_s(\beta_s^{-1}([s_0, S, s])) = 1$ and $\mu_s(p) = 0$ for all $(P_s \cup Q) - \{\beta_s^{-1}([s_0, S, s])\}$,

- the final marking $\tau_s$ is defined by $\tau_s(p) = 0$ for all $p \in P_s \cup Q$,

and define the k-PN controlled grammar $G'_s = (V', \Sigma, (s_0, S, s), R', N_{k,s})$. Then one can see that $L(G) \cap L(A) = \bigcup_{s \in H} L(G'_s)$. □

The results of the previous lemmas are summarized in the following theorem

**Theorem 4.15.** The family of languages $PN_k$, $k \geqslant 1$, is closed under union, substitution, mirror image, intersection with regular languages and it is not closed under concatenation.

## 4.4 Petri Nets with Chains and Cycles

We add new places and arcs, called *control places* and *arcs*, to a cf Petri net such a way that the new places with their input and output transitions compose *chains* or *cycles*.

### 4.4.1 Chain Control

Let $\mathcal{P} = \{\rho_1, \rho_2, \ldots, \rho_n\}$ be a set of disjoint chains where each chain $\rho_i = (P_{\rho_i}, T_{\rho_i}, F_{\rho_i}) \in \mathcal{P}$, $1 \leqslant i \leqslant n$, is defined as

$$\rho = t_{i,1} p_{i,1} t_{i,2} p_{i,2} \cdots p_{i,k_i - 1} t_{i,k_i}$$

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



with the sets of places, transitions and arcs, respectively,

$$P_\rho = \{p_{i,1}, p_{i,2}, \ldots, p_{i,k_i-1}\},$$
$$T_\rho = \{t_{i,1}, t_{i,2}, \ldots, t_{i,k_i}\},$$
$$F_\rho = \{(t_{i,j}, p_{i,j}) \mid 1 \leqslant j \leqslant k_i - 1\} \cup \{(p_{i,j}, t_{i,j+1}) \mid 1 \leqslant i \leqslant k_i - 1\}.$$

*Remark* 4.1. If a chain $\rho \in \mathcal{P}$ consists of a single transitions, i.e., $\rho = t$, then the sets of places and arcs of $\rho$ is considered to be empty, i.e., $P_\rho = F_\rho = \emptyset$.

Let

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

We consider a marked Petri net $N_\mathcal{P} = (P_\mathcal{P}, T_\mathcal{P}, F_\mathcal{P}, \phi, \iota)$ where $\phi(x, y) = 1$ for all $(x, y) \in F_\mathcal{P}$ and $\iota(p) = 0$ for all $p \in P_\mathcal{P}$.

**Proposition 4.16.** $\nu$ is an occurrence sequence of transitions of $N_\mathcal{P}$ enabled at the initial marking $\iota$ and finished at the marking $\mu$ where $\mu(p) = 0$ for all $p \in P_\mathcal{P}$ iff $\nu$ is the shuffle of $tr(\rho_{i_1}), tr(\rho_{i_2}), \ldots, tr(\rho_{i_m})$ for some $\rho_{i_1}, \rho_{i_2}, \ldots, \rho_{i_m} \in \mathcal{P}$, $m \geqslant 1$.

*Proof.* Let $\nu$ be an occurrence sequence of transitions $N_\mathcal{P}$ enabled at $\iota$ and finished at $\mu$ where $\mu(p) = 0$ for all $p \in P_\mathcal{P}$. We denote by $\nu_i \in T^*_{\rho_i}$, $1 \leqslant i \leqslant n$, the scattered substring of transitions of the chain $\rho_i$ in $\nu$. Since transitions of different chains occur concurrently, $\nu$ is a shuffle of the strings $\nu_i$, $1 \leqslant i \leqslant n$.

By definition, any occurrence sequence of transitions of a chain $\rho_i \in \mathcal{P}$, $1 \leqslant i \leqslant n$, starts by firing of $t_{i,1} \in T_{\rho_i}$, and for each $\nu'_i \in \mathrm{Pref}(\nu_i)$,

$$|\nu'_i|_{t_{i,1}} \geqslant |\nu'_i|_{t_{i,2}} \geqslant \cdots \geqslant |\nu'_i|_{t_{i,k_i}}.$$

Moreover,

$$|\nu_i|_{t_{i,1}} = |\nu_i|_{t_{i,2}}| = \cdots = |\nu_i|_{t_{i,k_i}}.$$

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



Let $\nu_{i,j}$, $1 \leqslant j \leqslant |\nu_i|_{t_{i,1}}$, be the sequence of j-th occurrence of the transitions $t_{i,1}, t_{i,2}, \ldots, t_{i,k_i}$. Then, $\nu_i$ is a shuffle of the sequences $\nu_{i,j}$.

Let $\nu = t_1 t_2 \cdots t_m$ be a shuffle of $tr(\rho_{i_1}), tr(\rho_{i_2}), \ldots, tr(\rho_{i_s})$ for some $\rho_{i_1}, \rho_{i_2}, \ldots, \rho_{i_s} \in \mathcal{P}$, $s \geqslant 1$. By definition, any occurrence sequence of transitions of $N_{\mathcal{P}}$ can only start at the initial marking $\iota$. Since for all $1 \leqslant j \leqslant s$,

$$|\nu|_{t_{i_j,1}} = |\nu|_{t_{i_j,2}}| = \cdots = |\nu|_{t_{i_j,k_{i_j}}},$$

$\mu(p) = 0$ for all $p \in P_{\mathcal{P}}$ where $\iota \xrightarrow{\nu} \mu$. □

**Definition 4.6.** Let $G = (V, \Sigma, S, R)$ be a context-free grammar with its corresponding cf Petri net $N = (P, T, F, \phi, \beta, \gamma, \iota)$. Let $T_1, T_2, \ldots, T_n$ be a partition of $T$ and $\mathcal{P} = \{\rho_1, \rho_2, \ldots, \rho_n\}$ be the set of disjoint chains such that $T_{\rho_i} = T_i$, $1 \leqslant i \leqslant n$, and $\bigcup_{\rho \in \mathcal{P}} P_\rho \cap P = \emptyset$.

An *z-Petri net* is a system $N_z = (P \cup Q, T, F \cup E, \varphi, \zeta, \gamma, \mu_0, \tau)$ where

- $Q = \bigcup_{\rho \in \mathcal{P}} P_\rho$ and $E = \bigcup_{\rho \in \mathcal{P}} F_\rho$;
- the weight function $\varphi$ is defined by $\varphi(x, y) = \phi(x, y)$ if $(x, y) \in F$ and $\varphi(x, y) = 1$ if $(x, y) \in E$;
- the labeling function $\zeta : P \cup Q \to V \cup \{\lambda\}$ is defined by $\zeta(p) = \beta(p)$ if $p \in P$ and $\zeta(p) = \lambda$ if $p \in Q$;
- the initial marking $\mu_0$ is defined by $\mu_0(p) = \iota(p)$ if $p \in P$ and $\mu_0(p) = 0$ if $p \in Q$;
- $\tau$ is the final marking where $\tau(p) = 0$ for all $p \in P \cup Q$.

*Example* 4.4. Figure 4.4 illustrates *z*-Petri net $N_z$ with respect to the context-free grammar $G_4 = (\{S, A, B\}, \{a, b\}, S, R)$ where $R$ consists of

$$r_0 : S \to AB,$$
$$r_1 : A \to \lambda, \qquad r_2 : B \to \lambda, \qquad r_3 : A \to aA,$$
$$r_4 : B \to aB, \qquad r_5 : A \to bA, \qquad r_6 : B \to bB.$$

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



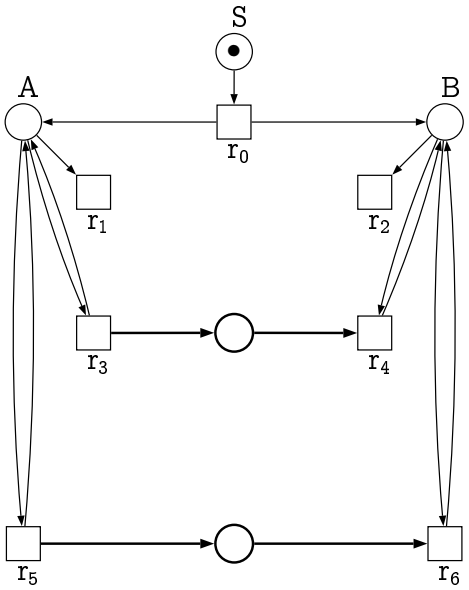


Figure 4.4: A $z$-Petri net $N_z$

### 4.4.2 Cyclic Control

Let $\mathcal{P} = \{\rho_1, \rho_2, \ldots, \rho_n\}$ be a set of disjoint cycles where each cycle $\rho_i = (P_{\rho_i}, T_{\rho_i}, F_{\rho_i}) \in \mathcal{P}$, $1 \leqslant i \leqslant n$, is defined as

$$\rho = p_{i,1}t_{i,1}p_{i,2}t_{i,2}\cdots p_{i,k_i}t_{i,k_i}p_{i,1}$$

with the sets of places, transitions and arcs, respectively,

$$P_{\rho_i} = \{p_{i,1}, p_{i,2}, \ldots, p_{i,k_i}\},$$
$$T_{\rho_i} = \{t_{i,1}, t_{i,2}, \ldots, t_{i,k_i}\},$$
$$F_{\rho_i} = \{(t_{i,j}, p_{i,j}) \mid 1 \leqslant j \leqslant k_i\} \cup \{(t_{i,j}, p_{i,j+1}) \mid 1 \leqslant i \leqslant k_i - 1\} \cup \{(t_{i,k_i}, p_{i,1})\}.$$

Let

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

We define a marked Petri net $N_{\mathcal{P}} = (P_{\mathcal{P}}, T_{\mathcal{P}}, F_{\mathcal{P}}, \phi, \iota)$ where $\phi(x, y) = 1$

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



for all $(x, y) \in F_{\mathcal{P}}$, and for each $\rho_i \in \mathcal{P}$, $1 \leqslant i \leqslant n$, $\iota(p_{i,1}) = 1$ and $\iota(p) = 0$ for all $p \in P_{\rho_i} - \{p_{i,1}\}$.

**Proposition 4.17.** $\nu$ is an occurrence sequence of transitions of $N_{\mathcal{P}}$ enabled and finished at the initial marking $\iota$ iff $\nu$ is the semi-shuffle of strings $tr(\rho_{i_1}), tr(\rho_{i_2}), \ldots, tr(\rho_{i_m})$ for some $\rho_{i_1}, \rho_{i_2}, \ldots, \rho_{i_m} \in \mathcal{P}$, $m \geqslant 1$.

*Proof.* Let $\nu = t_1 t_2 \cdots t_n$ be an occurrence sequence of transitions of $N_{\mathcal{P}}$ enabled and finished at the initial marking $\iota$. Let $\nu_i \in T^*_{\rho_i}$, $1 \leqslant i \leqslant n$, be a scattered substring of transitions of the cycle $\rho_i$ in $\nu$. Similarly to the case of chains, transitions of different cycles can occur concurrently, thus $\nu$ is a shuffle of substrings $\nu_i$, $1 \leqslant i \leqslant n$.

By definition, if transition $t_{i,1} \in T_{\rho_i}$, $1 \leqslant i \leqslant n$, occurs, then in order to return the token to place $p_{i,1}$, transitions $t_{i,2}, t_{i,3}, \ldots, t_{i,k_i}$ have to occur. Moreover, transition $t_{i,1}$ can occur again after transitions $t_{i,2}, t_{i,3}, \ldots, t_{i,k_i}$ have occurred in the given order, i.e., the occurrence of $tr(\rho_i)$, $1 \leqslant i \leqslant n$, can start a second time after its started occurrence has finished. Therefore, $\nu$ is a semi-shuffle of $tr(\rho_{i_1}), tr(\rho_{i_2}), \ldots, tr(\rho_{i_m})$ for some $\rho_{i_1}, \rho_{i_2}, \ldots, \rho_{i_m} \in \mathcal{P}$, $m \geqslant 1$.

Let $\nu = t_1 t_2 \cdots t_n$ be a semi-shuffle of $tr(\rho_{i_1}), tr(\rho_{i_2}), \ldots, tr(\rho_{i_s})$ for some $\rho_{i_1}, \rho_{i_2}, \ldots, \rho_{i_s} \in \mathcal{P}$, $s \geqslant 1$. By definition, any occurrence sequence of transitions of $N_{\mathcal{P}}$ starts at the initial marking $\iota$. On the other hand, any occurrence sequence of transitions of a cycle $\rho_i \in \mathcal{P}$, $1 \leqslant i \leqslant n$, in $\nu$ returns a token to place $p_{i,1}$, i.e., to the initial state. Therefore, $\nu$ is an occurrence sequence enabled at and finished at the marking $\iota$. □

**Definition 4.7.** Let $G = (V, \Sigma, S, R)$ be a context-free grammar with its corresponding cf Petri net $N = (P, T, F, \phi, \beta, \gamma, \iota)$. Let $T_1, T_2, \ldots, T_n$ be a partition of $T$ and $\mathcal{P} = \{\rho_1, \rho_2, \ldots, \rho_n\}$ be the set of disjoint cycles such that $T_{\rho_i} = T_i$, $1 \leqslant i \leqslant n$, and $\bigcup_{\rho \in \mathcal{P}} P_\rho \cap P = \emptyset$.

A *c-Petri net* is a system $N_c = (P \cup Q, T, F \cup E, \varphi, \zeta, \gamma, \mu_0, \tau)$ where

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



- $Q = \bigcup_{\rho \in \mathcal{P}} P_\rho$ and $E = \bigcup_{\rho \in \mathcal{P}} F_\rho$;
- the weight function $\varphi$ is defined by $\varphi(x, y) = \phi(x, y)$ if $(x, y) \in F$ and $\varphi(x, y) = 1$ if $(x, y) \in E$;
- the labeling function $\zeta : P \cup Q \to V \cup \{\lambda\}$ is defined by $\zeta(p) = \beta(p)$ if $p \in P$ and $\zeta(p) = \lambda$ if $p \in Q$;
- the initial marking $\mu_0$ is defined by $\mu_0(p) = \iota(p)$ if $p \in P$, and $\mu_0(p_{i,1}) = 1$, $\mu_0(p_{i,j}) = 0$ where $p_{i,j} \in P_i$, $1 \leqslant i \leqslant n$, $2 \leqslant j \leqslant k_i$;
- $\tau$ is the final marking where $\tau(p) = 0$ if $p \in P$, and $\tau(p_{i,1}) = 1$, $\tau(p_{i,j}) = 0$ where $p_{i,j} \in P_i$, $1 \leqslant i \leqslant n$, $2 \leqslant j \leqslant k_i$.

*Example* 4.5. Figure 4.5 illustrates a $c$-Petri net $N_c$ with respect to the context-free grammar given in Example 4.4.

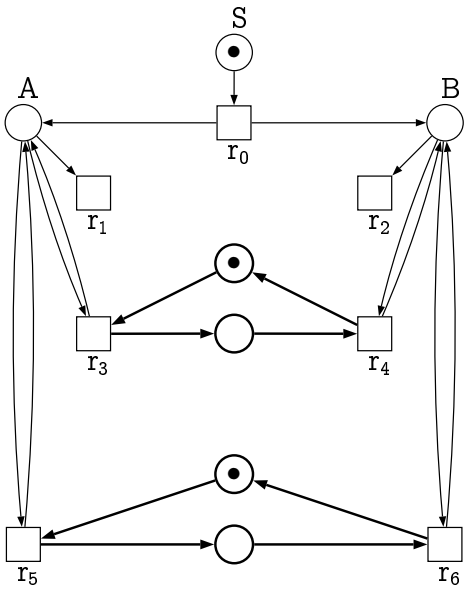


Figure 4.5: A $c$-Petri net $N_c$

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



### 4.4.3 Supervised Cyclic Control

Let $\mathcal{P} = \{\rho_1, \rho_2, \ldots, \rho_n\}$ be a set of cycles such that

$$P_{\rho_1} \cap P_{\rho_2} \cap \cdots P_{\rho_n} = \{p_0\}$$

where each cycle $\rho_i = (P_{\rho_i}, T_{\rho_i}, F_{\rho_i}) \in \mathcal{P}$, $1 \leqslant i \leqslant n$, is defined as

$$\rho = p_0 t_{i,1} p_{i,1} t_{i,2} \cdots p_{i,k_i-1} t_{i,k_i} p_0$$

with the sets of places, transitions and arcs, respectively,

$$\begin{aligned}
P_{\rho_i} &= \{p_0, p_{i,1}, p_{i,2}, \ldots, p_{i,k_i-1}\},\\
T_{\rho_i} &= \{t_{i,1}, t_{i,2}, \ldots, t_{i,k_i}\},\\
F_{\rho_i} &= \{(p_{i,j}, t_{i,j+1}) \mid 1 \leqslant j \leqslant k_i - 1\} \cup \{(t_{i,j}, p_{i,j}) \mid 1 \leqslant i \leqslant k_i - 1\}\\
&\quad \cup \{(p_0, t_{i,1})\} \cup \{(t_{i,k_i}, p_0)\}.
\end{aligned}$$

Let $X \in \{P, T, F\}$ and $X_{\mathcal{P}} = \bigcup_{\rho \in \mathcal{P}} X_\rho$. We define a marked Petri net $N_{\mathcal{P}} = (P_{\mathcal{P}}, T_{\mathcal{P}}, F_{\mathcal{P}}, \phi, \iota)$ where $\phi(x, y) = 1$ for all $(x, y) \in F_{\mathcal{P}}$, and $\iota(p_0) = 1$ and $\iota(p) = 0$ for all $p \in P_{\rho_i} - \{p_0\}$.

By construction, each occurrence sequence of transitions of $N_{\mathcal{P}}$ starts by firing of the first transition $t_{i,1}$ of some cycle $\rho_i$, $1 \leqslant i \leqslant n$, and transitions of a second cycle can fire only after the single token has been returned to the initial marking $\iota$. It happens if we fire the rest transitions $t_{i,2}, t_{i,3}, \ldots, t_{i,k_i}$ of $\rho_i$. It follows that

**Proposition 4.18.** $\nu$ is an occurrence sequence of transitions of $N_{\mathcal{P}}$ enabled and finished at the initial marking $\iota$ iff $\nu$ is the concatenation of strings $tr(\rho_{i_1}), tr(\rho_{i_2}), \ldots, tr(\rho_{i_m})$ for some $\rho_{i_1}, \rho_{i_2}, \ldots, \rho_{i_m} \in \mathcal{P}$, $m \geqslant 1$.

**Definition 4.8.** Let $G = (V, \Sigma, S, R)$ be a context-free grammar with its corresponding cf Petri net $N = (P, T, F, \phi, \beta, \gamma, \iota)$. Let $T_1, T_2, \ldots, T_n$ be

a partition of T. Let $\mathcal{P} = \{\rho_1, \rho_2, \ldots, \rho_n\}$ be the set of cycles such that $T_{\rho_i} = T_i$, $1 \leqslant i \leqslant n$, $P_1 \cap P_2 \cap \cdots \cap P_n = \{p_0\}$ and $\bigcup_{\rho \in \mathcal{P}} P_\rho \cap P = \emptyset$.

An *s-Petri net* is a system $N_s = (P \cup Q, T, F \cup E, \varphi, \zeta, \gamma, \mu_0, \tau)$ where

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

*Example* 4.6. Figure 4.6 illustrates a $s$-Petri net $N_s$ with respect to the context-free grammar given in Example 4.4.

### 4.4.4 Grammars, Languages and Examples

Here we define grammars controlled by $z$ $(c, s)$-Petri nets introduced in the previous subsection.

**Definition 4.9.** (i) An *x-Petri net controlled grammar* is a quintuple $G = (V, \Sigma, S, R, N_x)$ where $V, \Sigma, S$, and $R$ are defined as for a context-free grammar and $N_x = (P \cup Q, T, F \cup E, \varphi, \zeta, \gamma, \mu_0, \tau)$ is a $x$-Petri net with respect to the context-free grammar $(V, \Sigma, S, R)$ where $x \in \{z, c, s\}$.
(ii) The language generated by a $x$-Petri net controlled grammar $G$, denoted by $L(G)$, consists of all strings $w \in \Sigma^*$ such that there is a derivation $S \xRightarrow{r_1 r_2 \cdots r_k} w \in \Sigma^*$ and a successful occurrence sequence of transitions $\nu = t_1 t_2 \cdots t_k$ of $N_x$ such that $r_1 r_2 \cdots r_k = \gamma(t_1 t_2 \cdots t_k)$.

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



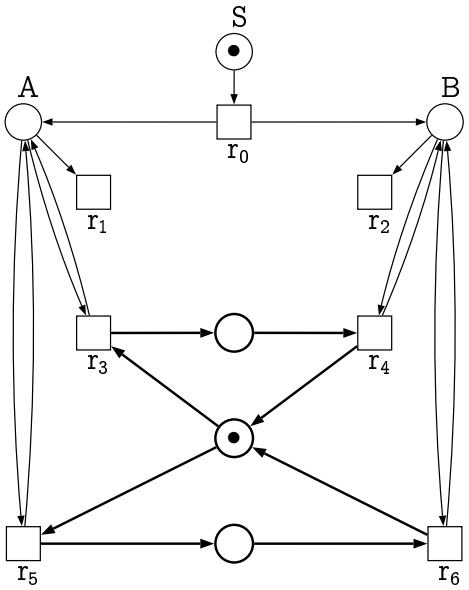


Figure 4.6: An $s$-Petri net $N_s$

*Example* 4.7. Let $G_5 = (V, \Sigma, S, R, N_z)$ be a $z$-Petri net controlled grammar where components $V, \Sigma, S, R$ are defined as for the context-free grammar $G_4$ in Example 4.4, and $N_z$ is the $z$-Petri net depicted in Figure 4.4.

After transition $t_0 = \zeta^{-1}(S \to AB)$ fires, transitions $t_1 = \zeta^{-1}(A \to \lambda)$, $t_2 = \zeta^{-1}(B \to \lambda)$, $t_3 = \zeta^{-1}(A \to aA)$, and $t_4 = \zeta^{-1}(A \to bA)$ are enabled. Transitions $t_3$ and (or) $t_5$ 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 $t_4 = \zeta^{-1}(B \to aB)$ and $t_5 = \zeta^{-1}(B \to bB)$ occur as many times as $t_1$ and (or) $t_3$ occur, respectively. To go to the final marking, transition $t_1$ and $t_2$ occur. We can see that $G_5$ generates a vector language

$$L(G_5) = \{wxw'x \mid x \in \{a, b\}, w \in \{a, b\}^*, w' \in \mathrm{Perm}(w)\}.$$

*Example* 4.8. We consider $c$-PN controlled grammar $G_6 = (V, \Sigma, S, R, N_c)$ where the components $V, \Sigma, S, R$ are defined as for the grammar $G_4$ in Example 4.4 and the $c$-Petri net $N_c$ is illustrated in Figure 4.5 (the transitions

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



of $N_c$ have the same labels as those of $N_z$).

After transition $t_0$ occurs transitions $t_1, t_2, t_3, t_5$ are enabled. Further we have the following cases: (1) the sequence $t_3t_4$ or (and) $t_5t_6$ occurs many times (in any order); (2) the sequence $t_3t_5$ or $t_5t_3$ occurs, then $t_4t_6$ or $t_6t_4$ occurs, respectively; (3) cases (1) and (2) repeat in any order; (4) to go to a final marking the sequence $t_1t_2$ occurs. It is easy to see that $G_6$ generates a semi-matrix language

$$L(G_6) = \{w_1\{\lambda, ab, ba\}w_2\{\lambda, ab, ba\}\cdots w_1\{\lambda, ab, ba\}w_2\{\lambda, ab, ba\}\cdots \mid w_1, w_2, \ldots \in \{a, b\}^*\}.$$

*Example* 4.9. Let $G_7 = (V, \Sigma, S, R, N_s)$ be a $s$-Petri net controlled grammar where components $V, \Sigma, S, R$ are defined as for the context-free grammar $G_4$ in Example 4.4, and $N_s$ is the $s$-Petri net depicted in Figure 4.6 (the transitions of $N_s$ have the same labels as those of $N_z$).

The execution of $N_s$ starts with the occurrence of $t_0$, and transitions $t_1, t_2, t_3$, and $t_5$ are enabled. If $t_1$ occurs, then only $t_2$ occurs. If $t_2$ occurs, then in order to reach the final marking, $t_1$ has to occur. If $t_3$ ($t_5$) occurs then only $t_4$ ($t_6$) occurs. We can see that $G_7$ generates a matrix language

$$L(G_7) = \{ww \mid w \in \{a, b\}^*\}.$$

We denote the families of languages generated by $x$-Petri net controlled grammars (with erasing rules) by $PN_x$, ($PN_x^\lambda$) where $x \in \{z, c, s\}$.

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

**Lemma 4.19.** $VEC^{[\lambda]} = PN_z^{[\lambda]}$, $sMAT^{[\lambda]} = PN_c^{[\lambda]}$, $MAT^{[\lambda]} = PN_s^{[\lambda]}$

*Proof.* We give here a proof for the first equality. First, we show that the inclusion $VEC^{[\lambda]} \subseteq PN_z^{[\lambda]}$ holds. Let $G = (V, \Sigma, S, M)$ be a vector grammar

where

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

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 respect to the context-free grammar $(V, \Sigma, S, R)$.

For each matrix $m_i = r_{i,1} r_{i,2} \cdots r_{i,k_i} \in M$, $1 \leqslant i \leqslant n$, we define a chain $\rho_i = t_{i,1} p_{i,1} t_{i,2} p_{i,2} \cdots t_{i,k_i-1} p_{i,k_i-1} t_{i,k_i}$ where $P_{\rho_i} = \{p_{i,1}, p_{i,2}, \ldots, p_{i,k_i-1}\}$ is a set of new places and $T_{\rho_i} = \{t_{i,1}, t_{i,2}, \ldots, t_{i,k_i}\} \subseteq T$ such that $\gamma(t_{i,j}) = r_{i,j}$, $1 \leqslant i \leqslant n$, $1 \leqslant j \leqslant k_i$. Since G is without repetition, the sets $T_{\rho_i}$, $1 \leqslant i \leqslant n$, are pairwise disjoint. It follows that $\mathcal{P} = \{\rho_1, \rho_2, \ldots, \rho_n\}$ is the set of disjoint chains. Therefore, we can construct the $z$-Petri net $N_z = (P \cup Q, T, F \cup E, \varphi, \zeta, \gamma, \mu_0, \tau)$ with respect to the grammar $(V, \Sigma, S, R)$ (with the notions of Definition 4.6), and define $z$-Petri net controlled grammar $G' = (V, \Sigma, S, R, N_z)$.

Let $D : S \xRightarrow{r_1 r_2 \cdots r_s} w \in \Sigma^*$ be a derivation in the vector grammar G. Then, by definition, $r_1 r_2 \cdots r_s$ is a shuffle of some matrices

$$m_{j_1}, m_{j_2}, \ldots, m_{j_l} \in M, l \geqslant 1.$$

It follows that D is also a derivation in the context-free grammar $(V, \Sigma, S, R)$, and by Proposition 4.1, $\nu = t_1 t_2 \cdots t_s = \gamma^{-1}(r_1 r_2 \cdots r_s)$ is an occurrence sequence of transitions of the cf Petri net N enabled at the initial marking $\iota$ and finished at the marking $\mu(p) = 0$ for all $p \in P$.

On the other hand, $\nu$ is a shuffle of $tr(\rho_{j_1}), tr(\rho_{j_2}), \ldots, tr(\rho_{j_l})$ since $\gamma(tr(\rho_i)) = m_i$, $1 \leqslant i \leqslant n$. By Proposition 4.16, $\mu_0(p) = \tau(p) = 0$ for all $p \in Q$. Therefore, D is also a derivation in $G'$.

Let $D' : S \xRightarrow{r_1 r_2 \cdots r_s} w \in \Sigma^*$ be a derivation in the $z$-Petri net controlled grammar $G'$. Then, $\nu = t_1 t_2 \cdots t_s = \gamma^{-1}(r_1 r_2 \cdots r_s)$ is an occurrence se-

quence of transitions of $N_z$ enabled at the initial marking $\iota$ and finished at the final marking $\tau$. By Proposition 4.16, the occurrence sequence $\nu$ is a shuffle of $tr(\rho_{i_1}), tr(\rho_{i_2}), \ldots, tr(\rho_{i_s})$ for some $\rho_{i_1}, \rho_{i_2}, \ldots, \rho_{i_s} \in \mathcal{P}$, $s \geqslant 1$, as $\iota(p) = \tau(p) = 0$ for all $p \in Q$. On the other hand, by the definition of the bijection $\gamma$, $\gamma(tr(\rho_{i_j})) = m_{i_j}$, $1 \leqslant j \leqslant s$. Therefore, $r_1 r_2 \cdots r_s$ is a shuffle of the matrices $m_{i_1}, m_{i_2}, \ldots, m_{i_s}$. It follows that $D'$ is also a derivation in $G$.

Next we show that $PN_z^{[\lambda]} \subseteq VEC^{[\lambda]}$ holds.

Let $G = (V, \Sigma, S, R, N_z)$ be a $z$-Petri net controlled grammar where $N_z = (P \cup Q, T, F \cup E, \varphi, \zeta, \gamma, \mu_0, \tau)$. By definitions, the set $T$ of transitions is divided into disjoint subsets $T_1, T_2, \ldots, T_n$ such that

$$T = \bigcup_{i=1}^{n} T_i, Q = \bigcup_{i=1}^{n} P_i, E = \bigcup_{i=1}^{n} F_i$$

and $\rho_i = (P_i, T_i, F_i)$, $1 \leqslant i \leqslant n$, are disjoint chains. According to the bijection $\gamma$, we can construct the set of matrices

$$M = \{m_i = \gamma(tr(\rho_i)) \mid 1 \leqslant i \leqslant n\},$$

and define the vector grammar $G' = (V, \Sigma, S, M)$. For the remaining part of the proof we can repeat the arguments of the first part of the proof.

The other equalities can be proven in the same manner using Propositions 4.17 and 4.18. □

The results of Lemma 4.19, Theorem 2.1.2 in [24] and Theorem 3.12 can be summarized in the next theorem

**Theorem 4.20.**

(1) $PN_s = MAT \subseteq PN_z = VEC \subseteq PN_z^{\lambda} = PN_c^{\lambda} = PN_s^{\lambda}$,

(2) $MAT \subseteq PN_c = sMAT \subseteq MAT^{\lambda}$.

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010

# 5

# Arbitrary Petri Net Controlled Grammars

## 5.1 Introduction

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

Therefore as correspondence we choose a bijection (between transitions and rules) or a coding (any transition is mapped to a rule) or a weak coding (any transition is mapped to a rule or the empty word) which agree with the classical three variants of Petri net languages (see e.g. [47, 93, 94]).

We consider two types of acceptance from the theory of Petri net languages: only those occurrence sequences belonging to the languages which transform the initial marking into a marking from a given finite set of mark-

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



ings or all occurrence sequences are taken (independent of the obtained marking). If we use only the occurrence sequence leading to a marking in a given finite set of markings we say that the Petri net controlled grammar is of t-type; if we consider all occurrence sequences, then the grammar is of r-type. We add a further type which can be considered as a complement of the t-type. Obviously, if we choose a finite set $M$ of markings and require that the marking obtained after the application of the occurrence sequence is smaller than at least one marking of $M$ (the order is componentwise), then we can choose another finite set $M'$ of markings and require that the obtained marking belongs to $M'$. The complementary approach requires that the obtained marking is larger than at least one marking of the given set $M$. The corresponding class of Petri net controlled grammars is called of g-type. Therefore, we obtain nine classes of Petri net controlled grammars since we have three different types of correspondence and three types of the set of admitted occurrence sequences. These types of control are generalizations of those types of control considered in the previous chapter, too, where instead of arbitrary Petri nets only such Petri nets have been considered where the places and transitions correspond in a one-to-one manner to nonterminals and rules, respectively.

In Section 5.2 we introduce the concept of control of derivations in context-free grammars by arbitrary Petri nets. Section 5.3 contains the results on the influence of the labeling function on the generative power. In Section 5.4 we discuss the effect of different types of final markings on the generative power.

## 5.2 Grammars and Their Languages

We now introduce the concept of control by an arbitrary Petri net.

**Definition 5.1.** An *arbitrary Petri net controlled grammar* is a tuple $G = (V, \Sigma, S, R, N, \gamma, M)$ where $V, \Sigma, S, R$ are defined as for a context-free

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



grammar and $N = (P, T, F, \varphi, \iota)$ is a (marked) Petri net, $\gamma : T \to R \cup \{\lambda\}$ is a transition labeling function and $M$ is a set of final markings.

**Definition 5.2.** The *language* generated by a Petri net controlled grammar $G$, denoted by $L(G)$, consists of all strings $w \in \Sigma^*$ such that there is a derivation $S \xRightarrow{r_1 r_2 \cdots r_k} w \in \Sigma^*$ and an occurrence sequence $\nu = t_1 t_2 \cdots t_s$ which is successful for $M$ such that $r_1 r_2 \cdots r_k = \gamma(t_1 t_2 \cdots t_s)$.

Definition 5.2 uses the extended form of the transition labeling function $\gamma : T^* \to R^*$; this extension is done in the usual manner.

Obviously, if $\gamma$ maps any transition to a rule, then $k = s$ in Definition 5.2.

*Example* 5.1. Let $G_1 = (\{S, A, B, C\}, \{a, b, c\}, S, R, N_a, \gamma_1, M_1)$ be a Petri net controlled grammar where $R$ consists of

$$\begin{array}{lll} S \to ABC, & & \\ A \to aA, & B \to bB, & C \to cC, \\ A \to a, & B \to b, & C \to c \end{array}$$

and $N_a$ is illustrated in Figure 5.1. If $M_1$ is the set of all reachable markings, then $G_1$ generates the language

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

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

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

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

**Definition 5.3.** A Petri net controlled grammar $G = (V, \Sigma, S, R, N, \gamma, M)$ is called

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



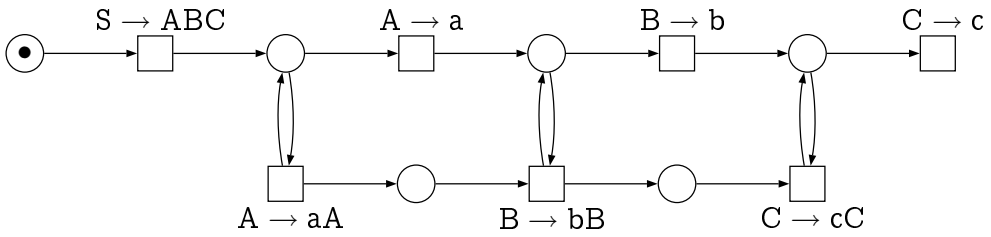


Figure 5.1: A labeled Petri net $N_a$

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

**Definition 5.4.** A Petri net controlled grammar $G = (V, \Sigma, S, R, N, \gamma, M)$ is called

- *r-type* if $M$ is the set of all reachable markings from the initial marking $\iota$, 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 $\mu \geqslant \mu'$.

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

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

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



The following inclusions are obvious.

**Lemma 5.1.** For $x \in \{f, -\lambda, \lambda\}$ and $y \in \{r, t, g\}$, $PN(x, y) \subseteq PN^{\lambda}(x, y)$.

## 5.3 The Effect of Labeling on the Computational Power

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

**Lemma 5.2.** For $y \in \{r, t, g\}$,

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

We now prove that the reverse inclusions also hold.

**Lemma 5.3.** For $y \in \{r, t, g\}$, $PN^{[\lambda]}(-\lambda, y) \subseteq PN^{[\lambda]}(f, y)$.

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

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

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

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

of new rules. Correspondingly, we set

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

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



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

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

and arcs

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

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

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

the weight function $\varphi'$ is defined by

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

the initial marking $\iota'$ is defined by

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

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

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



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

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

Let

$$S \xrightarrow{r_1 \cdots r_j} w_j \overset{r}{\Rightarrow} w'_j \xrightarrow{r_{j+1} \cdots r_k} w_k \in \Sigma^*$$

be a derivation in G where $r : A \to \alpha \in R^{>1}$. Then the rule $r : A \to \alpha$ can be replaced by the pair $A \to A_t$, $A_t \to \alpha$ for some $t \in T^{>1}$ in one-to-one correspondence with the transition $t$ of N where $\gamma(t) = r$, by the transitions $c_t^1$ and $c_t^2$ of $N'$, and vice versa. Hence $L(G) = L(G')$. □

**Lemma 5.4.** For $y \in \{r, t, g\}$, $PN(\lambda, y) \subseteq PN(-\lambda, y)$.

*Proof.* Let $G = (V, \Sigma, S, R, N, \gamma, M)$ be a $(\lambda, y)$-Petri net controlled grammar with $N = (P, T, F, \iota)$. Let

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

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

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

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

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

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

$$Adj^+(t) = Adj^*(t) - T_\lambda.$$

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



Let $T_\lambda = \{t_1, t_2, \ldots, t_n\}$. For each $t_i \in T_\lambda$, $1 \leqslant i \leqslant n$, we define the set of new transitions

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

We introduce the set $R(t_i)$ of new rules with respect to each $t_i \in T_\lambda$, $1 \leqslant i \leqslant n$,

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

Further, we define a $(-\lambda, y)$-Petri net controlled grammar

$$G' = (V, \Sigma, S, R', N', \gamma', M')$$

where $R' = R \cup \bigcup_{t_i \in T_\lambda} R(t_i)$ and $N' = (P, T', F', \iota)$ where

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

The weight function $\varphi'$ is defined by

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

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

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

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



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

$$t'_{11} \cdots t'_{1k(1)} t_1 t'_{21} \cdots t'_{2k(2)} t_2 \cdots t_n t'_{n+11} \cdots t'_{n+1k(n+1)} \tag{5.1}$$

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

Each $\lambda$-transition $t'_{ij}$, $1 \leqslant i \leqslant n$, $1 \leqslant j \leqslant k(i)$ in (5.1) can be replaced by the transition $t''_{ij}$ in $N'$, $1 \leqslant i \leqslant n$, $1 \leqslant j \leqslant k(i)$ with the label $A_i \to A_i$ where $A_i$ is the left side of the rule $r_i$, $\gamma(r_i) = t_i$, $1 \leqslant i \leqslant n$. Then

$$t''_{11} \cdots t''_{1k(1)} t_1 t''_{21} \cdots t''_{2k(2)} t_2 \cdots t''_{n1} \cdots t''_{nk(n)} t_n \tag{5.2}$$

is a successful occurrence sequence in $N'$ and correspondingly

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

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

It is easy to see that the proof of Lemma 5.4 holds for grammars with erasing rules, too. We present another proof in the following lemma since its construction has a smaller increase of the number of places, transitions and edges.

**Lemma 5.5.** For $\gamma \in \{r, t, g\}$, $PN^\lambda(\lambda, \gamma) \subseteq PN^\lambda(-\lambda, \gamma)$.

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

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

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

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



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

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

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

- the weight function is defined by

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

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

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

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

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

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



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

On the other hand, for each derivation

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

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

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

**Theorem 5.6.** For $y \in \{r, t, g\}$,

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

## 5.4 The Effect of Final Markings on the Generative Power

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

**Lemma 5.7.** $PN^{[\lambda]}(\lambda, r) \subseteq PN^{[\lambda]}(\lambda, t)$.

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

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

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

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



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

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

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

**Corollary 5.8.** $PN^{[\lambda]}(\lambda, r) \subseteq PN^{[\lambda]}(\lambda, g)$.

*Proof.* Let $G = (V, \Sigma, S, R, N, \gamma, M)$ be a $(\lambda, r)$-PN controlled grammar (with or without erasing rules) where $N = (P, T, F, \varphi, \iota)$. We construct a $(\lambda, g)$-PN controlled grammar $G'' = (V, \Sigma, S, R, N', \gamma', M')$ where $V$, $\Sigma$, $S$, $R$, $N'$ and $\gamma'$ are defined as for the grammar $G'$ in the proof of Lemma 5.7. If we define $M'$ as the set of any marking $\mu \in \mathcal{R}(N', \iota)$ which is greater than or equal to $\mu' = (0,0,\ldots,0)$, then the inclusion follows immediately. □

**Lemma 5.9.** $PN^{[\lambda]}(\lambda, g) \subseteq PN^{[\lambda]}(\lambda, t)$.

*Proof.* Let $G = (V, \Sigma, S, R, N, \gamma, M)$ be a $(\lambda, g)$-PN controlled grammar (with or without erasing rules) where $N = (P, T, F, \varphi, \iota)$ and $M$ is the set of all markings such that for every marking $\mu \in M$ there is a marking $\mu'$

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



of a given finite set $M_0 \subseteq \mathcal{R}(N, \iota)$ such that $\mu \geqslant \mu'$. Let $p_0$ be a new place. We define the following sets of new transitions and arcs.

$$T_{M_0} = \{t_\mu \mid \mu \in M_0\},$$
$$T_P = \{t_p \mid p \in P\}$$

are the sets of new transitions and

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

are the sets of new arcs. We construct the Petri net

$$N' = (P \cup \{p_0\}, T \cup T_{M_0} \cup T_P, F \cup F^-_{M_0} \cup F^+_{M_0} \cup F_P, \varphi', \iota')$$

where

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

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

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

Let $D : S \overset{\pi}{\Rightarrow} w \in \Sigma^*$, $\pi = r_1 r_2 \cdots r_n$, be a derivation in G, then there is an occurrence sequence $\nu = t_1 t_2 \cdots t_s$ such that $\iota \xrightarrow{\nu} \mu$ where $\gamma(\nu) = \pi$ and

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



$\mu \in M$. By definition, there is a marking $\mu' \in M_0$ such that $\mu \geqslant \mu'$. It follows that the transition $t'_\mu$ can occur and the place $p_0$ receives a token, and the rest tokens in places of P can be removed by firing transitions $t_p$. It is not difficult to see that D is also a derivation in G′.

If $D' : S \overset{\pi}{\Longrightarrow} w \in \Sigma^*$, $\pi = r_1 r_2 \cdots r_n$, is a derivation in G′ with a successful occurrence sequence $\nu = t_1 t_2 \cdots t_s$ where $\gamma'(\pi) = \nu$, then $\iota' \xrightarrow{\nu} \mu'$ where $\mu'(p) = 0$ for all $p \in P$ and $\mu'(p_0) = 1$. Since $\mu'(p_0) = 1$, $|\nu|_{t_\mu} = 1$ for some $\mu \in M_0$. Without loss of generality we can assume that $\nu = \nu' \cdot \nu''$ where $\nu'$ contains only transitions of T and $\nu''$ contains only transitions of $T_P$ and the transition $t_\mu$. Then, $\gamma'(\nu') = \pi$, $\gamma'(\nu'') = \lambda$ and $\iota' \xrightarrow{\nu'} \mu''$ where $\mu'' \geqslant \mu$. It follows that D′ is also a derivation in G. □

**Lemma 5.10.** $PN^{[\lambda]}(\lambda, g) \subseteq PN^{[\lambda]}(\lambda, r)$.

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

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

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

Let

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

We define a $(\lambda, r)$-Petri net controlled grammar

$$G' = (V \cup \overline{\Sigma}, \Sigma, S, \overline{R} \cup R_\Sigma, N', \gamma', M')$$

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



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

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

where $p', p''$ are new places,

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

are sets of new transitions,

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

are sets of new arcs.

- The weight function $\varphi'$ is defined by $\varphi'(x, y) = \varphi(x, y)$ if $(x, y) \in F$, $\varphi'(p, t_\mu) = \mu(p)$ if $\mu \in M_0$ and $\varphi'(x, y) = 1$ if $(x, y) \in F_{p'} \cup F_{p''} \cup F_\Sigma$.
- The initial marking $\iota'$ is defined by $\iota'(p) = \iota(p)$ if $p \in P$ and $\iota'(p') = 1$, $\iota'(p'') = 0$.
- The bijection $\gamma'$ is defined by $\gamma'(t) = A \to \phi(\alpha)$ if $t \in T$ and $\gamma(t) = A \to \alpha$, $\gamma'(t) = \lambda$ if $t \in T_{M_0}$, and $\gamma'(t_a) = \overline{a} \to a$ for all $a \in \Sigma$.
- For each $\tau' \in M'$, $\tau'(p') = 0$ and $\tau'(p'') = 1$.

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

Let $w = a_{i_1} a_{i_2} \cdots a_{i_k}$, $a_{i_1}, a_{i_2}, \ldots, a_{i_k} \in \Sigma$, $k \geqslant 1$. We construct a derivation $D'$ in $G'$ with respect to $D$ as follows:

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

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



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

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

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

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

**Lemma 5.11.** For $x \in \{f, -\lambda, \lambda\}$ and $y \in \{r, t, g\}$,

$$PN^\lambda(x, y) \subseteq MAT^\lambda.$$

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

We set $V' = V \cup \overline{P} \cup \{S', B\}$ where $\overline{P} = \{\overline{p} \mid p \in P\}$ is a set of new nonterminal symbols and $S', B$ are new nonterminal symbols. Let for $t \in T$,

$$^\bullet t = \{p_{i_1}, p_{i_2}, \ldots, p_{i_k}\}$$

and

$$t^\bullet = \{p_{j_1}, p_{j_2}, \ldots, p_{j_m}\}.$$

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



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

$$\sigma_{i_1} : \underbrace{\overline{p}_{i_1} \to \lambda, \overline{p}_{i_1} \to \lambda, \ldots, \overline{p}_{i_1} \to \lambda}_{\varphi(p_{i_1},t)} \tag{5.3}$$

$$\sigma_{i_2} : \underbrace{\overline{p}_{i_2} \to \lambda, \overline{p}_{i_2} \to \lambda, \ldots, \overline{p}_{i_2} \to \lambda}_{\varphi(p_{i_2},t)} \tag{5.4}$$

$$\ldots$$

$$\sigma_{i_k} : \underbrace{\overline{p}_{i_k} \to \lambda, \overline{p}_{i_k} \to \lambda, \ldots, \overline{p}_{i_k} \to \lambda}_{\varphi(p_{i_k},t)} \tag{5.5}$$

$$\sigma_B : B \to B\overline{p}_{j_1}^{\varphi(t,p_{j_1})}\overline{p}_{j_2}^{\varphi(t,p_{j_2})} \cdots \overline{p}_{j_m}^{\varphi(t,p_{j_m})} \tag{5.6}$$

and define the matrix

$$m_r = (\sigma_{i_1}, \sigma_{i_2}, \ldots, \sigma_{i_k}, \sigma_B, r) \tag{5.7}$$

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

$$m_0 = (S' \to SB \cdot \prod_{p \in P} \overline{p}^{|\iota(p)|}) \tag{5.8}$$

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

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

$$m_{\tau,\lambda} = (B \to \lambda, \underbrace{\overline{p}_1 \to \lambda, \ldots, \overline{p}_1 \to \lambda}_{\tau(p_1)}, \ldots, \underbrace{\overline{p}_n \to \lambda, \ldots, \overline{p}_n \to \lambda}_{\tau(p_n)}). \tag{5.9}$$

*Case* $y = r$.

$$m_{p,\lambda} = (\overline{p} \to \lambda) \text{ for each } p \in P \text{ and } m_{B,\lambda} = (B \to \lambda) \tag{5.10}$$

*Case* $y = g$. Here we consider matrices (5.9) together with matrices

(5.10).

We consider the matrix grammar $G' = (V', \Sigma, S', M)$ where $M$ consists of all matrices of (5.7)-(5.8) and matrices (5.9) (Case $y = t$), matrices (5.10) (Case $y = r$) or matrices (5.9)-(5.10) (Case $y = g$).

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

We construct the derivation $D'$ in $G'$ which simulates the derivation D. The derivation $D'$ starts with $S' \Rightarrow SB \cdot \prod_{p \in P} \overline{p}^{|\iota(p)|}$ applying the matrix (5.8), then for each pair of a transition $t$ in $\nu$ and the corresponding rule $r = \gamma(t)$, we choose a matrix of the form (5.7). When the terminal string $w \in \Sigma^*$ is generated, in order to erase the remaining symbols from $\overline{P}$ and the symbol B we use matrices of the form (5.9), (5.10) or (5.9) and (5.10) depending on $y \in \{r, t, g\}$.

Let $D' : S' \xRightarrow{m_0} SB \cdot \prod_{p \in P} \overline{p}^{|\iota(p)|} \xRightarrow{m_{i_1} m_{i_2} \cdots m_{i_n}} w_n = w \in \Sigma^*$ be a derivation in $G'$. Since $V \cap \overline{P} = \emptyset$, we can write a derivation $D'' : S \xRightarrow{r_{j_1} r_{j_2} \cdots r_{j_k}} w_{j_k} = w \in \Sigma^*$ where $r_{j_i}$ is the rule of the non-erasing matrix $m_{r_{j_i}}$, $1 \leqslant i \leqslant k$ in $D'$ and we omit those steps in $D'$ in which erasing matrices are used.

The application of a matrix $m_r$ of the form (5.7) in $D'$ shows that there are at least $\varphi(p_{i_1}, t)$ pieces of $\overline{p}_{i_1}$, etc., and at least $\varphi(p_{i_k}, t)$ pieces of $\overline{p}_{i_k}$ in the sentential form, i.e., the input places $p_{i_1}, p_{i_2}, p_{i_k}$ of $t$ have at least $\varphi(p_{i_1}, t)$, $\varphi(p_{i_2}, t)$, ..., $\varphi(p_{i_k}, t)$ tokens, respectively. Thus, the transition $t$, $\gamma(t) = r$ is enabled in N. We can construct the successful occurrence sequence $\iota \xrightarrow{t_{j_1} t_{j_2} \cdots t_{j_k}} \mu_k$ where $\gamma(t_{j_i}) = r_{j_i}$, $1 \leqslant i \leqslant k$. Hence, $D''$ is a derivation in G. Thus $L(G') \subseteq L(G)$.

Now let $E : S \xRightarrow{r_{j_1} r_{j_2} \cdots r_{j_k}} w_{j_k} = w \in \Sigma^*$ be a derivation in G. Then we also have the derivation $E' : S' \xRightarrow{m_0} SB \xRightarrow{m_{j_1} m_{j_2} \cdots m_{j_k}} w'_{j_k} B$ in $G'$ where $w'_{j_k}$ differs from $w_{j_k}$ only in letters $\overline{p}$ with $p \in P$. These letters and B can be erased with matrices (5.9), (5.10) or (5.9) and (5.10) depending on

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



$y \in \{r, t, g\}$. Thus $L(G) \subseteq L(G')$. □

**Lemma 5.12.** $MAT^{[\lambda]} \subseteq PN^{[\lambda]}(-\lambda, r)$.

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

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

Without loss of generality we can assume that $G$ is without repetitions. Let $R = \{r_{i,j} \mid 1 \leqslant i \leqslant n, 1 \leqslant j \leqslant k(i)\}$.

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

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

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

Let

$$\overline{M} = \{\overline{m}_1, \overline{m}_2, \ldots, \overline{m}_n\} \text{ where } \overline{m}_i = (\overline{r}_{i1}, \overline{r}_{i,2}, \ldots, \overline{r}_{i,k(i)}), 1 \leqslant i \leqslant n,$$
$$M_\Sigma = \{(\overline{a} \to a) \mid a \in \Sigma\}.$$

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

We define a $(-\lambda, r)$-Petri net controlled grammar

$$G'' = (V \cup \overline{\Sigma}, \Sigma, S, R', N, \gamma, M')$$

where $R' = R \cup \{\overline{a} \to a \mid a \in \Sigma\}$ and $N = (P, T, F, \varphi, \iota)$ is a control Petri net

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



where the sets of places, transitions and arcs are respectively defined by

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

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

Let

$$S = w_0 \xRightarrow{m_{i_1}} w_1 \xRightarrow{m_{i_2}} \cdots \xRightarrow{m_{i_l}} w_l = w \in \Sigma^* \tag{5.11}$$

be a derivation in $G'$, where $m_{i_j}$, $1 \leqslant j \leqslant l$ is an element of $\overline{M}$ or $M_\Sigma$, and

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

for some $a \in \Sigma$. Then by definition of the function $\gamma$, $\mu_{j-1} \xrightarrow{\nu_j} \mu_j$ where $\nu_j = t_{i_j,1}t_{i_j,2}\cdots t_{i_j,k(i_j)}$ or $\nu_j = t_a$ and $\mu_j = \iota$ for all $1 \leqslant j \leqslant l$. Hence, according to (5.11), we can construct the successful occurrence sequence $\iota \xrightarrow{\nu_1\nu_2\cdots\nu_l} \iota$ of transitions of N. Therefore, $S \xRightarrow{\pi_1\pi_2\cdots\pi_l} w_l \in \Sigma^*$ is a derivation in $G''$, where, for each $1 \leqslant j \leqslant l$, $\pi_j = \overline{r}_{i_j,1}\overline{r}_{i_j,2}\cdots\overline{r}_{i_j,k(i_j)}$ or $\pi_j = \overline{a} \to a$ for some $a \in \Sigma$.

Let $D : S \xRightarrow{r_1r_2\cdots r_l} w \in \Sigma^*$ be a derivation in $G''$ where $\nu = t_1t_2\cdots t_l$, $\gamma(t_i) = r_i$, $1 \leqslant i \leqslant l$, is a successful occurrence sequence of transitions of N.

If $t_{i,1}$ with $1 \leqslant i \leqslant l$ starts in $\nu$, then in the next steps $t_{i,2}, t_{i,3}, \ldots, t_{i,k(i)}$ can only fire in this order. Another $t_{j,1}$, $1 \leqslant j \leqslant l$ or $t_a$ for some $a \in \Sigma$ can fire after $t_{i,k(i)}$ occurs. By definition of $\gamma$, the corresponding label

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



rules $\overline{r}_{i,1}, \overline{r}_{i,2}, \ldots, \overline{r}_{i,k(i)}$ are the elements of one matrix $\overline{m}_i \in \overline{M}$, i.e., $\overline{m}_i = (\overline{r}_{i,1}, \overline{r}_{i,2}, \ldots, \overline{r}_{i,k(i)})$. Thus the application of matrices of $G'$ can be simulated by occurrence sequence of transitions of $N$. It follows that $D$ is also a derivation in $G'$. □

Now we summarize our results in the following theorem.

**Theorem 5.13.** For $x \in \{f, -\lambda, \lambda\}$ and $y \in \{r, t, g\}$,

$$MAT \subseteq PN(x, r) = PN(x, g) \subseteq PN(x, t) \subseteq PN^{\lambda}(x, y) = MAT^{\lambda}.$$

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010

# 6

# Grammars Controlled by Special Petri Nets

## 6.1 Introduction

In the previous chapter we investigated arbitrary Petri net controlled grammars in dependence on the type of labeling functions and on the definitions of final 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 ([54]) or some subclasses of regular languages ([22, 25]) are considered. Thus, it is also natural to investigate grammars controlled by some special classes of Petri nets. We consider (generalized) state machines, (generalized) marked graphs, causal nets, (extended) free-choice nets, asymmetric choice nets and ordinary nets. Similarly to the general case we also investigate the effects of labeling policies and final markings to the computational power, and 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.

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



## 6.2 Grammars and Their Languages

Let $G = (V, \Sigma, S, R, N, \gamma, M)$ be an arbitrary Petri net controlled grammar where $V$ and $\Sigma$ are sets of nonterminal and terminal symbols, respectively, $S$ is the start symbol, $R$ is set of context-free rules, $N = (P, T, F, \iota)$ is a Petri net, $\gamma : T \to R \cup \{\lambda\}$ is a transition labeling function and $M$ is a set of final markings.

The grammar $G$ 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 a *special Petri net* (in short, sPN) when we refer to each special class.

We also use a notation an $(x, y)$-(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 $\gamma$ and $y \in \{r, t, g\}$ shows the type of a set of final markings (for details, see Chapter 5).

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

$$\begin{array}{lllll} SM^{[\lambda]}(x,y), & GSM^{[\lambda]}(x,y), & MG^{[\lambda]}(x,y), & GMG^{[\lambda]}(x,y), & CN^{[\lambda]}(x,y), \\ FC^{[\lambda]}(x,y), & EFC^{[\lambda]}(x,y), & AC^{[\lambda]}(x,y), & ON^{[\lambda]}(x,y), & PN^{[\lambda]}(x,y) \end{array}$$

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

The inclusion $X(x, y) \subseteq X^{\lambda}(x, y)$ immediately follows from the definition where

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



- $X \in \{SM, GSM, MG, GMG, CN, FC, EFC, AC, ON\}$,
- $x \in \{f, -\lambda, \lambda\}$ and $y \in \{r, t, g\}$.

*Example* 6.1. Let $G_1 = (\{S, A, B\}, \{a, b\}, S, R, N_{sm}, \gamma_1, M_1)$ be a SM controlled grammar where R consists of

$$\begin{array}{lll} S \to AB, & & \\ A \to aA, & A \to bA, & A \to \lambda, \\ B \to aB, & B \to bB, & B \to \lambda, \end{array}$$

the Petri net $N_{sm}$ illustrated in Figure 6.1 is a labeled state machine and $M_1 = \{\mu\}$ where $\mu(p_0) = 1$ and $\mu(p) = 0$ for all $p \in P - \{p_0\}$, then

$$L(G_1) = \{ww \mid w \in \{a, b\}^*\} \in SM^{\lambda}(\lambda, t).$$

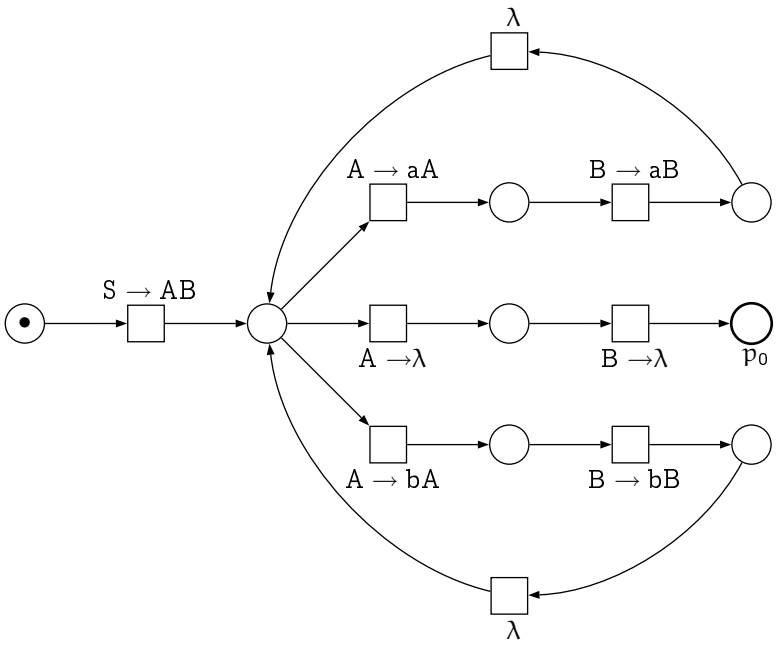


Figure 6.1: A labeled state machine $N_{sm}$

*Example* 6.2. Let $G_2 = (\{S, A, B\}, \{a, b\}, S, R, N_{mg}, \gamma_2, M_2)$ be a MG controlled grammar where R is as for the grammar $G_1$ in Example 6.1, a labeled

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



marked graph $N_{mg}$ is illustrated in Figure 6.2 and $M_1 = \{\mu\}$ where $\mu(p) = 0$ for all $p \in P$. Then

$$L(G_2) = \{ww' \mid w \in \{a, b\}^* \text{ and } w' \in \text{Perm}(w)\} \in MG^{\lambda}(\lambda, t).$$

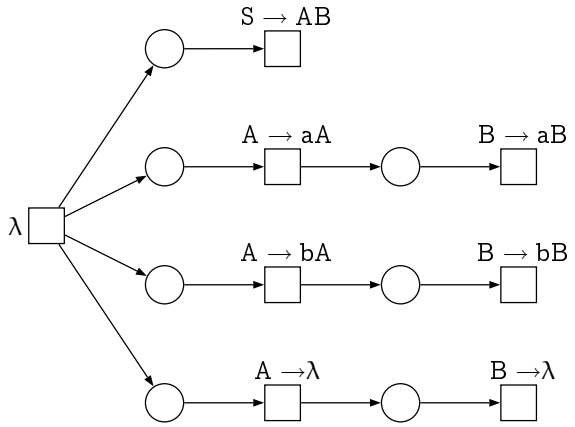


Figure 6.2: A labeled marked graph $N_{mg}$

## 6.3 Results: Labeling Strategies

In this section we investigate the effect of the labeling of transitions on the generative capacities of the introduced families of languages.

From the definition, the next statement follows immediately.

**Lemma 6.1.** For $X \in \{SM, GSM, MG, GMG, CN, FC, EFC, AC, ON\}$ and $y \in \{r, t, g\}$,

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

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

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



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

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

*Proof.* Let $G = (V, \Sigma, S, R, N, \gamma, M)$ 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 $\iota'$ 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 $G' = (V', \Sigma, S, R', N', \gamma', M')$ 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 $\gamma'$ 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 $p \in P_t$.

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



In a derivation $S \xRightarrow{r_1 r_2 \cdots r_n} w \in \Sigma^*$ of G, we replace each rule $r_i : A_i \to \alpha_i$ by the pair $r_{t_i} : A_i \to A_{t_i}$, $r'_{t_i} : A_{t_i} \to \alpha_i$, $1 \leqslant i \leqslant n$, 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 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 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, $w \in L(G')$.

By construction of N′, if a transition $l_t$ for some $t \in T$ in a successful occurrence sequence of transitions $\sigma$ then $l'_t$ is also in $\sigma$, similarly, if $A \to A_t$ in $D : S \overset{\pi}{\Rightarrow} w \in \Sigma^*$ 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, $w \in L(G)$. □

For each $(\lambda, \gamma)$-sPN controlled grammar, if we label each $\lambda$-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 $(-\lambda, \gamma)$-sPN controlled grammar, i.e.,

**Lemma 6.3.** For $X \in \{SM, GSM, MG, GMG, CN, FC, EFC, AC, ON\}$ and $\gamma \in \{r, t, g\}$,

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

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

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



A $(-\lambda, y)$-sPN controlled grammar $G' = (V', \Sigma, S', R', N', \gamma', M')$ 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 G where $\nu = \nu_1 t_1 \cdots \nu_k t_k$, $\gamma(t_i) = r_i$ and $\gamma(\nu_i) = \lambda$ for all $1 \leqslant i \leqslant k$ is an occurrence sequence in N enabled at the initial marking $\iota$ and finishing at a marking $\mu \in M$.

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

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

and $t' \nu_1 t_1 \nu_2 t_2 \cdots \nu_k t_k t''$, $t', t'' \in T'$, is a successful occurrence sequence in

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



$N'$.

On the other hand, for each derivation

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

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

**Lemma 6.4.** For $y \in \{r, t, g\}$ and $X \in \{SM, GSM\}$, $X(\lambda, y) \subseteq X(-\lambda, y)$.

*Proof.* Let $G = (V, \Sigma, S, R, N, \gamma, M)$ be a $(\lambda, y)$-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 $\lambda$-transitions of $N$ and $F_\lambda \subseteq F$ be the set of all incoming and outgoing arcs of the $\lambda$-transitions, i.e., $F_\lambda = \{(p, t), (t, p) \mid p \in P \text{ and } t \in T_\lambda\}$.

We construct a new state machine $N'$ without $\lambda$-transitions by removing each $\lambda$-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 $t_\lambda$ 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 $c_t$ which has the same label as $t$. Let

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

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

$$F' = (F - F_\lambda) \cup \{(p, c_t) \mid p \in {}^\bullet t_\lambda \text{ and } c_t \in T_c(t_\lambda), t_\lambda \in T_\lambda\} \cup \{(c_t, p) \mid c_t \in T_c(t_\lambda) \text{ and } p \in (\mathrm{Adj}^+(t_\lambda))^\bullet, t_\lambda \in T_\lambda\}.$$

We define a $(-\lambda, y)$-SM controlled grammar $G' = (V, \Sigma, S, R, N', \gamma', M)$ where $V, \Sigma, S, R$ are as for the grammar $G$, and $\gamma'(t) = \gamma(t)$ if $t \in T$ and

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



$\gamma'(c_t) = \gamma(t)$ if $c_t \in T_c$ and $c_t$ is the copy of $t \in \mathrm{Adj}^+(t_\lambda)$ for all $t_\lambda \in T_\lambda$.

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

The inverse case can be shown by backtracking the arguments in this paragraph. □

**Lemma 6.5.** For $x \in \{f, -\lambda, \lambda\}$ and $y \in \{r, t, g\}$,

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

*Proof.* Let $G = (V, \Sigma, S, R, N, \gamma, M)$ 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,

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

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



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 $G' = (V', \Sigma, S, R', N', \gamma', M')$ where the labeling function $\gamma'$ 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 $G'$. Thus it is not difficult to see that $L(G) = L(G')$. □

**Lemma 6.6.** For $y \in \{r, t, g\}$ and $X \in \{MG, CN\}$, $X(\lambda, y) \subseteq X(-\lambda, y)$.

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

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

We assume that for each $t \in T_\lambda$, $t$ is not a transition of some cycle $\rho$ 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

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



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 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 $T'_\lambda$ 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 $t_\lambda$ and $t'_\lambda$ are not neighbors. Let $\mathrm{Nbr}(T_\lambda)$ be the set of all maximal neighborhood subsets of $T_\lambda$. Let $T_\lambda(t)$ denote a maximal neighborhood subset of $T_\lambda$ 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 $T_\lambda$,
  - 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

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

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



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

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

– for each $p \in {}^\bullet t_\lambda$, $t_\lambda \in T_\lambda(t)$, $T_\lambda(t) \in \mathbf{Nbr}(T_\lambda)$, $t \in T$, 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

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

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

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

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 $\iota'$ 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 $G' = (V, \Sigma, S, R', N', \gamma', M')$ where

- $V, \Sigma, S$ are defined as for $G$, $R' = R \cup \{A \to A \mid A \to \alpha \in R\}$ and $N'$ is

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



constructed above;

- the labeling function $\gamma'$ 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

$$S \stackrel{\pi}{\Rightarrow} w \in \Sigma^*,\ \pi = r_1 r_2 \cdots r_n \tag{6.1}$$

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

$$\iota \xrightarrow{t_1} \mu_1 \xrightarrow{t_2} \mu_2 \xrightarrow{t_3} \cdots \xrightarrow{t_m} \mu_m \tag{6.2}$$

We construct a successful occurrence sequence $\sigma'$ of transitions of $N'$ on the base of (6.2) as follows: all transitions in (6.2) which are from $T - T_\lambda$ also remain in $\sigma'$. If in order to fire a transition $t \in \mathrm{Adj}^+(T_\lambda)$ which is in (6.2), some transitions $t_{\lambda,i_1}, t_{\lambda,i_2}, \ldots, t_{\lambda,i_l} \in T_\lambda(t)$, $l \geqslant 1$, where $t_{\lambda,i_{j+1}} \in \mathrm{Adj}^1(t_{\lambda,i_j})$, $2 \leqslant j \leqslant l-1$, 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 $G'$ is constructed from (6.1) 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, $L(G) \subseteq L(G')$.

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 $t_\lambda$ and its label $A \to A$ is removed in the derivation and the same string is generated. □

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



**Lemma 6.7.** For $X \in \{EFC, AC, ON\}$ and $y \in \{r, t, g\}$,

$$X(\lambda, y) \subseteq X(-\lambda, y).$$

*Proof.* Let $G = (V, \Sigma, S, R, N, \gamma, M)$ be a $(\lambda, y)$-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

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

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

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

We define a $(-\lambda, y)$-extended free-choice (asymmetric-choice, ordinary) net controlled grammar $G' = (V, \Sigma, S, R', N', \gamma', M')$ 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 $t \in T$

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



and $\gamma'([t]) = A \to A \in R(t_\lambda)$ where $[t] \in T(t_\lambda)$ and $t \in Adj^+(t_\lambda)$ 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 G. Then

$$t'_{1,1} \cdots t'_{1,k(1)} t_1 t'_{2,1} \cdots t'_{2,k(2)} t_2 \cdots t_n t'_{n+1,1} \cdots t'_{n+1,k(n+1)} \tag{6.3}$$

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

Each $\lambda$-transition $t'_{i,j}$, $1 \leqslant i \leqslant n$, $1 \leqslant j \leqslant k(i)$ in (6.3) can be replaced by the transition $t''_{i,j}$ in $N'$, $1 \leqslant i \leqslant n$, $1 \leqslant j \leqslant k(i)$ with the label $A_i \to A_i$ where $A_i$ is the left side of the rule $r_i = \gamma^{-1}(t_i)$, $1 \leqslant i \leqslant n$. Then

$$t''_{1,1} \cdots t''_{1,k(1)} t_1 t''_{2,1} \cdots t''_{2,k(2)} t_2 \cdots t''_{n,1} \cdots t''_{n,k(n)} t_n \tag{6.4}$$

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 $G'$ where

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

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

**Lemma 6.8.** For $y \in \{r, t, g\}$, $PN^{[\lambda]}(\lambda, y) \subseteq FC^{[\lambda]}(\lambda, y)$.

*Proof.* Let $G = (V, \Sigma, S, R, N, \gamma, M)$ 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, $1 \leqslant i \leqslant \varphi(p, t)$, 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 $(t, p_j[t, p])$, $(p_j[t, p], t_j[t, p])$,

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



$(t_j[t,p],p)$ whose weights are 1's, $1 \leqslant j \leqslant \varphi(t,p)$. Let

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

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

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

We define an FC controlled grammar $G' = (V, \Sigma, S, R, N', \gamma', M')$ where the components $V, \Sigma, S, R$ are defined as for the grammar $G$, 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 $0 \leqslant \tau'(p) \leqslant \tau'(p')$ where $p' \in {}^\bullet({}^\bullet p)$.

Let $D : S \xRightarrow{r_1 r_2 \cdots r_m} w \in \Sigma^*$ be a derivation in $G$. 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 $\nu$ by $\nu' = \nu_1 t_1 \nu_2 \cdots \nu_n t_n$ in $N'$ where for all $1 \leqslant i \leqslant n$, $\nu_i \in \mathrm{Perm}({}^\bullet({}^\bullet t_i))$ where

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

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



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

Let $t_1 t_2 \cdots t_n$ be a successful occurrence sequence for $M'$. By construction, each occurrence of $t_i$, $1 \leqslant i \leqslant n$, 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

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

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

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 $t_i$, $1 \leqslant i \leqslant n$, and erase $\nu_i^\lambda$, $1 \leqslant i \leqslant n+1$. 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 $G'$ where $r_1 r_2 \cdots r_n = \gamma'(\nu)$ is also a derivation in $G$ and $r_1 r_2 \cdots r_n = \gamma'(\nu')$. □

The immediate consequence of this lemma is

**Corollary 6.9.** For $X \in \{EFC, AC, ON\}$ and $y \in \{r, g, t\}$,

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

**Lemma 6.10.** For $y \in \{r, t, g\}$, $FC(\lambda, y) \subseteq FC(-\lambda, y)$.

*Proof.* The proof is based on the following idea: for a FC controlled grammar $G$, $L(G) \in FC(\lambda, y)$, we construct an equivalent EFC controlled grammar $G'$, $L(G') \in EFC(-\lambda, y)$ (with an extended free-choice net $N'$ without $\lambda$-transitions), according to Lemma 6.4, next we again transform the grammar $G'$ into an equivalent FC controlled grammar $G''$, $L(G'') \in FC(-\lambda, y)$, which is equivalent to $G$.

Let $G = (V, \Sigma, S, R, N, \gamma, M)$ be a free-choice net controlled grammar

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



with $N = (P, T, F, \iota)$. Let

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

$t_\lambda \in T_\lambda$, be the set of new transitions. By Lemma 6.4, we define an EFC net controlled grammar $G' = (V, \Sigma, S, R', N', \gamma', M')$ (with the notions of the proof of the lemma), which is equivalent to the grammar $G$, 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 $\iota''(p) = 0$

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



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 $G'' = (V, \Sigma, S, R', N'', \gamma'', M'')$ where $V, \Sigma, S, R'$ are defined as for $G'$ and the net $N''$ is constructed above.

The labeling function $\gamma''$ 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 6.4).

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 $0 \leqslant \tau''(p[p, t]) \leqslant \tau''(p)$ where $p \in {}^\bullet({}^\bullet p[p, t])$.

Further we can repeat the arguments of the proof of Lemma 6.4. □

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

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

## 6.4 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.

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



From the structural properties of special Petri nets and Lemmas 6.5, 6.8, the next statement follows immediately

**Theorem 6.12.** For $X \in \{FC, EFC, AC, ON\}$ and $x \in \{f, -\lambda, \lambda\}$, $y \in \{r, g, t\}$,

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

**Lemma 6.13.** $SM^{[\lambda]}(\lambda, r) \subseteq SM^{[\lambda]}(\lambda, t)$.

*Proof.* Let $G = (V, \Sigma, S, R, N, \gamma, M)$ 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.,

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

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 [44]). □

From Lemma 6.13 the next statements follow

**Corollary 6.14.** $SM^{[\lambda]}(\lambda, r) \subseteq SM^{[\lambda]}(\lambda, g)$ and $SM^{[\lambda]}(\lambda, g) \subseteq SM^{[\lambda]}(\lambda, t)$.

*Proof.* 1. If a finite set of final marking is defined as the set of all reachable markings, Lemma 6.13 also holds for "g"-case.

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

Let $M = \{\tau \mid \tau(p) \geqslant \tau'(p)$ for all $p \in P$ 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 $\tau \in M$, $\tau(p) \leqslant \sum_{p \in P} \iota(p)$ for all $p \in P$. We define a set of final markings as

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

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

**Lemma 6.15.** $SM^{[\lambda]}(\lambda, t) \subseteq SM^{[\lambda]}(\lambda, r)$.

*Proof.* Let $G = (V, \Sigma, S, R, N, \gamma, M)$ 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* $G$.

For each $\tau_i \in M$, $1 \leqslant i \leqslant k$, we define a state machine controlled grammar $G_i = (V_i, \Sigma, S_i, R_i, N_i, \gamma_i, M_i)$ 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 $\psi_i : V \cup \Sigma \to V_i \cup \Sigma$, $1 \leqslant i \leqslant k$, 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

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

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



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

- the initial marking $\iota_i$ is defined by $\iota_i(\phi_i(p)) = \iota(p)$ for all $p \in P$;
- the labeling function $\gamma_i$ 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 not to generate strings of the language of a grammar* $G_i$, $1 \leqslant i \leqslant k$, *before reaching the final marking* $\mu_i$, *we change each terminal symbol* $a \in \Sigma$ *to new nonterminal symbols* $\bar{a}$, *and add new places, transitions and arcs to* $N_i$ *such a way that the nonterminal symbols* $\bar{a}$ *are changed back to* $a$ *if and only if the final marking is reached.*

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

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

We construct a state machine $N'_i = (P'_i, T'_i, F'_i, \iota'_i)$ 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 $p \in \overline{P}_i$.

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



A state machine controlled grammar $G'_i = (V'_i, \Sigma, S_i, R'_i, N'_i, \gamma'_i, M'_i)$ is defined as follows:

- $V'_i = V_i \cup \bigcup_{a\in\Sigma} V_{l,j,a}$ where $V_{l,j,a} = \{a_{l,j} \mid 1 \leqslant l \leqslant n, 1 \leqslant j \leqslant k_l\}$ is the set of new nonterminal symbols;
- Let $\bar{R}_i = \{A \to \varphi_i(\alpha) \mid A \to \alpha \in R_i\}$ where the weight 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 $a \in \Sigma$,

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

  and define $R'_i = \bar{R}_i \cup \bigcup_{a\in\Sigma} R_{i,a}$;
- the labeling function $\gamma'$ is defined by
  - $\gamma'_i(t) = A \to \varphi_i(\alpha) \in \bar{R}_i$ if $t \in T_i$ and $\gamma_i(t) = A \to \alpha \in R_i$,
  - $\gamma'_i(t'_{l,j}) = \lambda$ for $1 \leqslant l \leqslant n, 1 \leqslant j \leqslant k_l$,
  - $\gamma'_i(t_{l,j,a}) = a_{l,j} \to a_{l,j+1}$ for $1 \leqslant l \leqslant n, 1 \leqslant j \leqslant k_l - 1$,
  - $\gamma'_i(t_{l,k_l,a}) = a_{l,k_l} \to a_{l+1,1}$ for $1 \leqslant l \leqslant n-2$,
  - $\gamma'_i(t_{n,k_n,a}) = a_{n,k_n} \to a$;
- 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 $\Sigma$, $\sum_{p\in P_i} \mu'_i(p) = \sum_{p\in P_i} \iota'_i(p)$ 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 state machine controlled grammar* $G''$ *that the language generated by this grammar is the union of languages generated by the grammars* $G'_i$, $1 \leqslant i \leqslant k$, *constructed in Step II.*

We define a SM controlled grammar $G'' = (V'', \Sigma, S'', R'', N'', \gamma'', M'')$ 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

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

- the state machine $N'' = (P'', T'', F'', \iota'')$ is defined by

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

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

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

- 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 $1 \leqslant j \leqslant k$ and $\iota''(p'_i) = 0$ for all $1 \leqslant i \neq j \leqslant k$;

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



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

**Lemma 6.16.** $MAT^{[\lambda]} \subseteq SM^{[\lambda]}(f, t)$.

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

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

Without loss of generality we can assume that $G$ is without repetitions. Let $R = \{r_{i,j} \mid 1 \leqslant i \leqslant n, 1 \leqslant j \leqslant k(i)\}$. We define an $(f, t)$-SM controlled grammar $G' = (V, \Sigma, S, R, N, \gamma, \{\mu\})$ where the sets of places, transitions and arcs of the a SM $N = (P, T, F, \iota)$ are defined by

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

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

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

and the final marking $\mu$ is the same as the initial marking $\iota$.

Let

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

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



be a derivation in G, where $m_{i_j} \in M$, $1 \leqslant j \leqslant l$, 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 $\gamma$, $\gamma^{-1}(m_{i_j}) = \sigma_j$ where $\sigma_j = t_{i_j,1} t_{i_j,2} \cdots t_{i_j,k(i_j)}$ for all $1 \leqslant j \leqslant l$. 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 $G'$.

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

**Lemma 6.17.** For $y \in \{r, g, t\}$, $SM^{[\lambda]}(\lambda, y) \subseteq rC^{[\lambda]}$.

*Proof.* Let $G = (V, \Sigma, S, R, N, \gamma, M)$ 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 $\mu$ to state $\mu'$ with label t iff marking $\mu'$ is obtained from marking $\mu$ 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 $\sigma$ is a successful occurrence sequence of transitions of N. Let $K = \{\gamma(\sigma) \mid \sigma \in L(\mathcal{A})\}$. Therefore, $L(G) = L(G')$ where $G' = (V, \Sigma, S, R, K)$ is a regularly controlled grammar. □

From Theorem 6.11 and Lemmas 6.16, 6.17, we have

**Corollary 6.18.** For $x \in \{f, -\lambda, \lambda\}$ and $y \in \{r, g, t\}$, $MAT^{[\lambda]} = SM^{[\lambda]}(x, y)$.

**Lemma 6.19.** $VEC^{[\lambda]} \subseteq MG^{[\lambda]}(f, t) \cap CN^{[\lambda]}(f, t) \cap GSM^{[\lambda]}(f, t)$.

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



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

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

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

We define an $(f, t)$-ON controlled grammar $G' = (V, \Sigma, S, R, N, \gamma, M)$ with $N = (P, T, F, \iota)$ where

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

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

- 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 $\gamma(t_{i,j}) = r_{i,j}$, $1 \leqslant i \leqslant n, 1 \leqslant j \leqslant k_i$, 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 $\gamma(tr(\rho_i)) = m_i$, $1 \leqslant i \leqslant n$, and the firing of the transitions of a path $\rho_i$ simulates the application of the rules of the matrix $m_i$. Moreover, a derivation $S \xRightarrow{r_1 r_2 \cdots r_l} w \in \Sigma^*$ in the context-free grammar $(V, \Sigma, S, R)$ is a derivation in G if $r_1 r_2 \cdots r_l$ is a shuffle of some matrices $m_{i_1}, m_{i_2}, \ldots, m_{i_k} \in M$, and a derivation in $G'$ if $t_1 t_2 \cdots t_l = \gamma^{-1}(r_1 r_2 \cdots r_l)$ is a shuffle of $tr(\rho_{i_1}), tr(\rho_{i_2}), \ldots, tr(\rho_{i_k})$ where

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



$\gamma(tr(\rho_{i_j})) = m_{i_j}$, $1 \leqslant j \leqslant k$. Thus, it is easy to see that each derivation in G can be simulated by a derivation in $G'$ and vise versa. □

**Lemma 6.20.** $sMAT^{[\lambda]} \subseteq SM^{[\lambda]}(f,t) \cap MG^{[\lambda]}(f,t)$.

*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.

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

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

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

We define an $(f,t)$-ON controlled grammar $G' = (V, \Sigma, S, R, N, \gamma, M)$ with $N = (P, T, F, \iota)$ where

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

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

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

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

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

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



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 $\gamma(tr(\rho_i)) = m_i$, $1 \leqslant i \leqslant n$. Using the same arguments of the proof of Lemma 6.19, one can easily show that $L(G) = L(G')$. □

Now we summarize our results in the following theorem.

**Theorem 6.21.** The relations in Figure 6.3 hold where $x \in \{f, -\lambda, \lambda\}$, $y \in \{r, g, t\}$, $X \in \{\mathrm{FC}, \mathrm{EFC}, \mathrm{AC}, \mathrm{ON}\}$, $Y \in \{\mathrm{SM}, \mathrm{GSM}, \mathrm{FC}, \mathrm{EFC}, \mathrm{AC}, \mathrm{ON}\}$ and $Z \in \{\mathrm{MG}, \mathrm{GMG}, \mathrm{CN}\}$; the lines (arrow) denote (proper) inclusions of the lower families into the upper families.

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



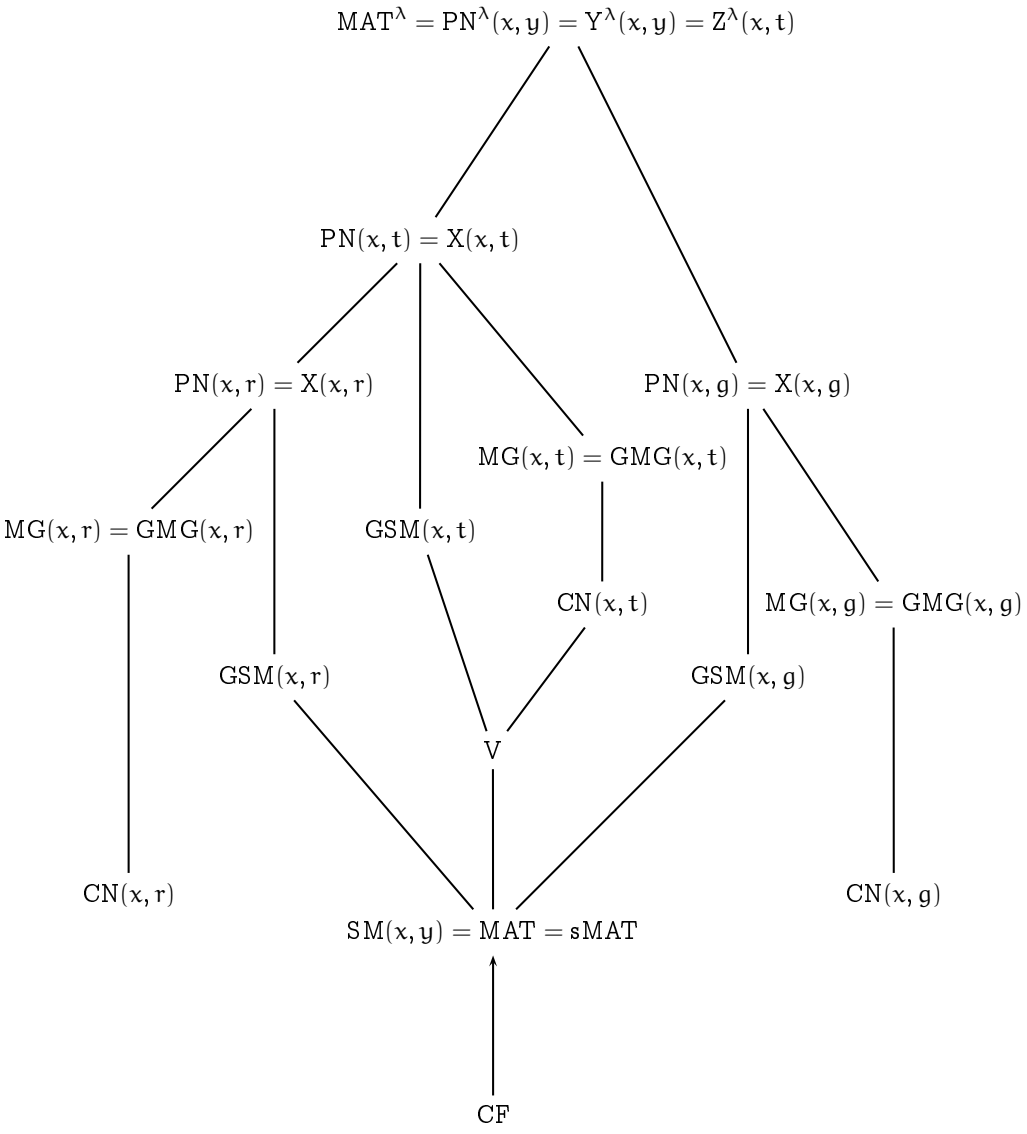


Figure 6.3: The hierarchy of language families generated by Petri net controlled grammars

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010

# 7

# Capacity-Bounded Grammars

## 7.1 Introduction

The close relationship between Petri nets and language theory has been extensively studied for a long time [18, 24]. Results from the theory of Petri nets have been applied successfully to provide elegant solutions to complicated problems from language theory [35, 48].

A context-free grammar can be associated with a context-free (communication-free) Petri net, whose places and transitions, correspond to the nonterminals and the rules of the grammar, respectively, and whose arcs and weights reflect the change in the number of nonterminals when applying a rule. In Chapter 4 we used extended cf Petri nets, i.e., cf Petri nets enriched by additional components (places and arcs) in order to define regulation mechanisms for context-free grammars. Moreover, we considered generalizations of these types of control mechanisms in Chapter 5, where instead of extended cf Petri nets we used arbitrary (place/transition) Petri nets whose transitions correspond to production rules. In this chapter we continue the research in this direction by restricting to (context-free, extended or arbitrary) Petri nets with place capacities.

Quite obviously, a context-free Petri net with place capacity regulates the defining grammar by permitting only those derivations where the number of each nonterminal in each sentential form is bounded by its capac-

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



ity. Similar mechanisms have been introduced and investigated by several authors. Grammar with *finite index* (the index of a grammar is the maximal number of nonterminals simultaneously appearing in its complete derivations (considering the most economical derivations for each string)) were first considered by Brainerd [10]. *Nonterminal-bounded* grammars (a grammar a nonterminal-bounded if the total number of nonterminals in every sentential form does not exceed an upper bound) were introduced by Altman and Banerji in [5, 6, 8]. A "weak" variant of nonterminal-bounded grammars (only the complete derivations are required to be bounded) were defined by Moriya [70]. Ginsburg and Spanier introduced *derivation-bounded* languages in [43] (all strings which have complete derivation in a grammar G consisting of sentential forms each of which does not contain more than k nonterminals collected in the set $L_k(G)$)). There it was shown that grammars regulated in this way generate the family of context-free languages of finite index, even if arbitrary nonterminal strings are allowed as left-hand sides of production rules. Finite index restrictions to regulated grammars have also been investigated [40, 39, 76, 78, 84, 86, 87, 88]. There it was shown that the families of most regulated languages are collapse.

In this chapter we show that capacity-bounded context-free grammars have a larger generative power than context-free grammars of finite index while the family of languages generated by capacity-bounded phrase structure grammars (due to Ginsburg and Spanier) and several families of languages generated by grammars controlled by extended cf Petri nets with place capacities coincide with the family of matrix languages of finite index.

## 7.2 Capacity-Bounded Grammars

We will now introduce capacity-bounded grammars and show some relations to similar concepts known from the literature.

**Definition 7.1.** A *capacity-bounded* grammar is a tuple $G = (V, \Sigma, S, R, \kappa)$

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



where $G' = (V, \Sigma, S, R)$ is a grammar and $\kappa : V \to \mathbb{N}$ is a capacity function. The derivation relation $\Rightarrow_G$ is defined as $\alpha \Rightarrow_G \beta$ iff $\alpha \Rightarrow_{G'} \beta$ and $|\alpha|_A \leqslant \kappa(A)$ and $|\beta|_A \leqslant \kappa(A)$, for all $A \in V$. The language of G is defined as $L(G) = \{w \in \Sigma^* \mid S \Rightarrow_G^* w\}$.

The families of languages generated by capacity-bounded GS grammars and by context-free capacity-bounded grammars are denoted by $GS_{cb}$ and $CF_{cb}$, respectively. The capacity function mapping each nonterminal to 1 is denoted by 1. The notions of finite index and bounded capacities can be extended to matrix, vector and semi-matrix grammars. The corresponding language families are denoted by

$$MAT_{fin}^{[\lambda]},\ VEC_{fin}^{[\lambda]},\ sMAT_{fin}^{[\lambda]},\ MAT_{cb}^{[\lambda]},\ VEC_{cb}^{[\lambda]},\ sMAT_{cb}^{[\lambda]}.$$

Capacity-bounded grammars are very similar to derivation-bounded grammars, which were studied in [43]. A *derivation-bounded* grammar is a quintuple $G = (V, \Sigma, S, R, k)$ where $G' = (V, \Sigma, S, R)$ is a grammar and $k \in \mathbb{N}$ is a bound on the number of allowed nonterminals. The language of G contains all words $w \in L(G')$ that have a derivation $S \Rightarrow^* w$ such that $|\beta|_V \leqslant k$, for each sentential form $\beta$ of the derivation.

Other related concepts are nonterminal-bounded grammars and grammars of finite index. A context-free grammar $G = (V, \Sigma, S, R)$ is *nonterminal-bounded* if $|\beta|_V \leqslant k$ for some fixed $k \in \mathbb{N}$ and all sentential forms $\beta$ derivable in G. The *index* of a derivation in G is the maximal number of nonterminal symbols in its sentential forms. G is of *finite index* if every word in $L(G)$ has a derivation of index at most k for some fixed $k \in \mathbb{N}$. The family of context-free languages of finite index is denoted by $CF_{fin}$.

Note that there is a subtle difference between the first two and the last two concepts. While context-free nonterminal-bounded and finite index grammars are just context-free grammars with a certain structural property (and generate context-free languages by definition), capacity-bounded and

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



derivation-bounded grammars are special cases of *regulated rewriting* (and could therefore generate non-context-free languages). However, it has been shown that the family of derivation bounded languages is equal to $CF_{fin}$, even if arbitrary grammars due to Ginsburg and Spanier are permitted [43]. We will now give two examples of capacity-bounded grammars generating non-context-free languages.

*Example* 7.1. Let $G = (\{S, A, B, C\}, \{a, b, x, y\}, S, R, 1)$ be the capacity-bounded context-free grammar where $R$ consists of

$$r_0 : S \to AC, \quad r_1 : A \to aBb, \quad r_2 : B \to aCb, \quad r_3 : C \to aAb,$$
$$r_4 : A \to xC, \quad r_5 : C \to xB, \quad r_6 : B \to xA, \quad r_7 : A \to y.$$

A derivation in $G$ proceeds as follows. After the first step $S \Rightarrow AC$ the sentential form contains two nonterminals from $\{A, B, C\}$. As long as the terminating rule $r_7$ is not applied, the capacity bound requires that one of the nonterminals must be rewritten by a word containing the third nonterminal symbol. For instance, in the sentential form $AC$, one must either replace $A$ by $aBb$ or $C$ by $xB$. In this way the subderivations of the two nonterminals are coupled.

We consider only the generation of terminal words from

$$M = a^+x^3yb^+a^+x^2yb^+.$$

Words from $M$ can be generated exactly by the derivations of the form

$$\begin{aligned} S &\overset{r_0}{\Rightarrow} AC \overset{r_1}{\Rightarrow} aBbC \overset{r_3}{\Rightarrow} aBbaAb \overset{r_2}{\Rightarrow} a^2Cb^2aAb \overset{r_1}{\Rightarrow} a^2Cb^2a^2Bb^2 \\ &\overset{r_3}{\Rightarrow} a^3Ab^3a^2Bb^2 \overset{r_2}{\Rightarrow} a^3Ab^3a^3Cb^3 \\ &\overset{c}{\Rightarrow} a^{3n}Ab^{3n}a^{3n}Cb^{3n},\ c = (r_1r_3r_2r_1r_3r_2)^{n-1}, n \geqslant 1 \\ &\overset{c}{\Rightarrow} a^{3n}x^3yb^{3n}a^{3n}x^2yb^{3n},\ c \in \{r_5r_4r_6r_5r_7r_4r_7, r_5r_4r_6r_7r_5r_4r_7\}. \end{aligned}$$

The intersection of $L(G)$ with the regular set $M$ is not context-free. Hence,

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



$L(G)$ is a non-context-free language.

*Example* 7.2. Let $G = (\{S, A, B, C, D, E, F\}, \{a, b, c\}, S, R, 1)$ be the capacity-bounded grammar where $R$ consists of the rules:

$$r_1 : S \to ABCD, \quad r_2 : AB \to aEFb, \quad r_3 : CD \to cAD, \quad r_4 : EF \to EC,$$
$$r_5 : EF \to FC, \quad r_6 : AD \to FD, \quad r_7 : AD \to ED, \quad r_8 : EC \to AB,$$
$$r_9 : FD \to CD, \quad r_{10} : FC \to AF, \quad r_{11} : AF \to \lambda, \quad r_{12} : ED \to \lambda.$$

The possible derivations are exactly those of the form

$$\begin{aligned} S &\overset{r_1}{\Rightarrow} ABCD \\ &\overset{(r_2 r_3 r_4 r_6 r_8 r_9)^n}{\Longrightarrow} a^n ABb^n c^n CD \\ &\overset{r_2 r_3}{\Longrightarrow} a^{n+1} EFb^{n+1} c^{n+1} AD \\ &\overset{r_5 r_7}{\Longrightarrow} a^{n+1} FCb^{n+1} c^{n+1} ED \\ &\overset{r_{10} r_{11} r_{12}}{\Longrightarrow} a^n b^n c^n \end{aligned}$$

(in the last phase, the sequences $r_{10} r_{12} r_{11}$ and $r_{12} r_{10} r_{11}$ could also be applied with the same result). Therefore,

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

The above examples show that capacity-bounded grammars – in contrast to derivation bounded grammars – can generate non-context-free languages. Moreover, any context-free language generated by a grammar of $G$ of finite index is also generated by the capacity-bounded grammar $(G, \kappa)$ where $\kappa$ is capacity function constantly $k$.

The generative power of capacity-bounded grammars will be studied in detail in the following two sections. In the end of this section we show an important technical result: one can restrict to grammars with capacities bounded by 1. Let $CF^1_{cb}$ and $GS^1_{cb}$ be the language families generated by context-free and arbitrary grammars with capacity function 1.

**Lemma 7.1.** $CF_{cb} = CF^1_{cb}$ and $GS_{cb} = GS^1_{cb}$.

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



*Proof.* Let $G = (V, \Sigma, S, R, \kappa)$ be a capacity-bounded phrase structure grammar. We construct the capacity-bounded grammar $G' = (V', \Sigma, (S, 1), R', 1)$ with capacity function 1 and

$$\begin{aligned} V' &= \{(A, i) \mid A \in V, 1 \leqslant i \leqslant \kappa(A)\}, \\ R' &= \{\alpha' \to \beta' \mid h(\alpha') \to h(\beta') \in R\} \end{aligned}$$

where $h : (V' \cup \Sigma)^* \to (V \cup \Sigma)^*$ is the letter-to-letter homomorphism defined by $h(a) = a$, for $a \in \Sigma$, and $h((A, i)) = A$, for $A \in V$, $1 \leqslant i \leqslant \kappa(A)$.

To show $L(G') \subseteq L(G)$ we will prove by induction on the number $n$ of derivation steps that $(S, 1) \Rightarrow^*_{G'} \gamma'$ implies $S \Rightarrow^*_G h(\gamma')$. The induction basis $n = 0$ is clear as $h((S, 1)) = S$. Now suppose that the statement has been shown for $n$ and that $\gamma'$ is derivable in $G'$ in $n + 1$ steps. Then there are words $\alpha', \beta', \gamma'_1, \gamma'_2$ such that $\gamma' = \gamma'_1\beta'\gamma'_2$, $\alpha' \to \beta' \in R'$ and $\gamma'_1\alpha'\gamma'_2$ is derivable in $G'$ in $n$ steps. By induction hypothesis, $h(\gamma'_1\alpha'\gamma'_2)$ is derivable in $G$. Moreover, $h(\alpha') \to h(\beta')$ is in $R$ and applying this rule to $h(\gamma'_1)h(\alpha)h(\gamma'_2)$ yields $h(\gamma'_1)h(\beta')h(\gamma'_2) = h(\gamma')$. Finally, $|\gamma'|_{(A,i)} \leqslant 1$ holds for all $A \in V$, $1 \leqslant i \leqslant \kappa(A)$. Consequently $|h(\gamma')|_A \leqslant \kappa(A)$, for all $A \in V$, and $h(\gamma')$ is derivable in $G$.

To show $L(G) \subseteq L(G')$ we will prove by induction on the number $n$ of derivation steps that $S \Rightarrow^*_G \gamma$ implies $(S, 1) \Rightarrow^*_{G'} \gamma'$, for some $\gamma' \in h^{-1}(\gamma)$. Again, the induction basis $n = 0$ is trivial. Now suppose that the statement is true for $n$ and that $\gamma$ is derivable in $n + 1$ steps in $G$. Then there are $\alpha, \beta, \gamma_1, \gamma_2$ such that $\gamma = \gamma_1\beta\gamma_2$, $\alpha \to \beta \in R$ and $\gamma_1\alpha\gamma_2$ is derivable in $n$ steps in $G$. By induction hypothesis, some word from $h^{-1}(\gamma_1\alpha\gamma_2)$ is derivable in $G'$ and it has the form $\gamma'_1\alpha'\gamma'_2$ with $h(\gamma'_1) = \gamma_1$, $h(\alpha') = \alpha$, $h(\gamma'_2) = \gamma_2$, as $h$ is letter-to-letter. Since $\gamma'_1\alpha'\gamma'_2$ is derivable in $G'$ and $\gamma = \gamma_1\beta\gamma_2$ is derivable in $G$, we have $|\gamma'_1\gamma'_2|_{(A,i)} \leqslant 1$ and $|\beta|_A \leqslant \kappa(A) - |\gamma_1\gamma_2|_A$, for all $A \in V, 1 \leqslant i \leqslant \kappa(A)$. For a word $\zeta' \in (V' \cup \Sigma)^*$ and $A \in V$, define

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



the set

$$\mathit{Free}_A(\zeta') \;=\; \{i \mid 1 \leqslant i \leqslant \kappa(A) \text{ and } |\zeta'|_{(A,i)} = 0\}.$$

The cardinality of $\mathit{Free}_A(\gamma_1'\gamma_2')$ is equal to $\kappa(A) - |\gamma_1\gamma_2|_A$ and hence at least $|\beta|_A$. Consequently, there is a word $\beta' \in h^{-1}(\beta)$ such that $|\beta'|_{(A,i)} = 0$, if $i \notin \mathit{Free}_A(\gamma_1'\gamma_2')$, and $|\beta'|_{(A,i)} \leqslant 1$ if $i \in \mathit{Free}_A(\gamma_1'\gamma_2')$. This implies $|\gamma_1'\beta'\gamma_2'|_{(A,i)} \leqslant 1$, for all $A \in V, 1 \leqslant i \leqslant \kappa(A)$. By definition of $G'$, the rule $\alpha' \to \beta'$ is in $R'$, its application on $\gamma_1'\alpha'\gamma_2'$ yields $\gamma' = \gamma_1'\beta'\gamma_2' \in h^{-1}(\gamma)$. As $|\gamma'|_{(A,i)} \leqslant 1$, for all $A \in V, 1 \leqslant i \leqslant A$, $\gamma'$ is derivable in $G'$. □

## 7.3 The Power of Capacity-Bounded GS Grammars

It will be shown in this section that capacity-bounded GS grammars generate exactly the family of matrix languages of finite index. This is in contrast to derivation bounded grammars which generate only context-free languages of finite index [43].

**Lemma 7.2.** $GS_{cb} \subseteq MAT_{fin}$.

*Proof.* Consider some language $L \in GS_{cb}$ and let $G = (V, \Sigma, S, R, 1)$ be a capacity-bounded GS grammar such that $L = L(G)$. A word $\alpha \in (V \cup \Sigma)^*$ can be uniquely decomposed as

$$\alpha = x_1\beta_1x_2\beta_2 \cdots x_n\beta_nx_{n+1}$$

where $x_1, x_{n+1} \in \Sigma^*, x_2, \ldots, x_n \in \Sigma^+, \beta_1, \ldots, \beta_n \in V^+$. The subwords $\beta_i$ are referred to as the *maximal nonterminal blocks* of $\alpha$. Note that the length of a maximal block in any sentential form of a derivation in $G$ is bounded by $|V|$. We will first construct a capacity-bounded grammar $G'$ with $L(G') = L$ such that all words of $L$ can be derived in $G'$ by rewriting

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



a maximal nonterminal block in every step. Let $G' = (V, \Sigma, S, R', 1)$ where

$$R' = \{\alpha_1\alpha\alpha_2 \to \alpha_1\beta\alpha_2 \mid \alpha \to \beta \in R, \alpha_1, \alpha_2 \in V^*, \\ |\alpha_1\alpha\alpha_2|_A \leqslant 1, \text{ for all } A \in V\}.$$

The inclusion $L(G) \subseteq L(G')$ is obvious since $R \subseteq R'$. On the other hand, any derivation step in $G'$ can be written as

$$\gamma_1\underline{\alpha_1\alpha\alpha_2}\gamma_2 \Rightarrow_{G'} \gamma_1\underline{\alpha_1\beta\alpha_2}\gamma_2, \text{ where } \alpha \to \beta \in R,$$

implying that the same step can be performed in G as

$$\gamma_1\alpha_1\underline{\alpha}\alpha_2\gamma_2 \Rightarrow_{G,1} \gamma_1\alpha_1\underline{\beta}\alpha_2\gamma_2.$$

Thus $L(G') \subseteq L(G)$ holds as well. Moreover, any derivation step in G,

$$\gamma_1\alpha_1\underline{\alpha}\alpha_2\gamma_2 \Rightarrow_{G,1} \gamma_1\alpha_1\underline{\beta}\alpha_2\gamma_2,$$

$\alpha_1\alpha\alpha_2$ being a maximal nonterminal block, can be performed in $G'$ replacing the maximal nonterminal block $\alpha_1\alpha\alpha_2$ by $\alpha_1\beta\alpha_2$.

In the second step we construct a context-free matrix grammar H which simulates exactly those derivations in $G'$ that replace a maximal nonterminal block in each step. We introduce two alphabets

$$[V] = \{[\alpha] \mid \alpha \in V^+, |\alpha|_A \leqslant 1, \text{ for all } A \in V\} \text{ and } \overline{V} = \{\overline{A} \mid A \in V\}.$$

The symbols of $[V]$ are used to encode each maximal nonterminal block as single symbols, while $\overline{V}$ is a disjoint copy of $V$. Any word

$$\alpha = x_1\beta_1x_2\beta_2\cdots x_n\beta_nx_{n+1}, x_1, x_{n+1} \in \Sigma^*, x_2, \ldots, x_n \in \Sigma^+, \beta_1, \ldots \beta_n \in V^+$$

such that $|\alpha|_A \leqslant 1$, for all $A \in V$, can be represented by the word $[\alpha] = x_1[\beta_1]x_2[\beta_2]\cdots x_n[\beta_n]x_{n+1}$, where the maximal nonterminal blocks in $\alpha$

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



are replaced by the corresponding symbols from $[V]$. The desired matrix grammar is obtained as $H = (V_H, \Sigma, S', M)$, with $V_H = [V] \cup V \cup \overline{V} \cup \{S'\}$ and the set of matrices defined as follows. For any rule $r = \alpha \to \beta$ in $R'$, $M$ contains the matrix $m_r$ consisting of the rules

- $[\alpha] \to [\beta]$ (note that $\alpha \in [V]$, but $\beta \in ([V] \cup \Sigma)^*$),
- $A \to \overline{A}$, for all $A \in V$ such that $|\alpha|_A = 1$ and $|\beta|_A = 0$,
- $\overline{A} \to A$, for all $A \in V$ such that $|\alpha|_A = 0$ and $|\beta|_A = 1$.

(The order of the rules in $m_r$ is arbitrary). Additionally, $M$ contains the starting and the terminating matrices

$$(S' \to [S]S\overline{A}_1 \cdots \overline{A}_m) \text{ and } (\overline{S} \to \lambda, \overline{A}_1 \to \lambda, \ldots, \overline{A}_m \to \lambda),$$

where $V = \{S, A_1, \ldots, A_m\}$. Intuitively, $H$ generates sentential forms of the shape $[\beta]\gamma$ where $[\beta] \in ([V] \cup \Sigma)^*$ encodes a sentential form $\beta$ derivable in $G'$ and $\gamma \in (V \cup \overline{V})$ counts the nonterminal symbols in $\beta$ as follows: $|\gamma|_A + |\gamma|_{\overline{A}} = 1$ and $|\gamma|_A = |\beta|_A$. Formally, it can be shown by induction that a sentential form over $V_H \cup \Sigma$ can be generated after applying $k \geqslant 1$ matrices (except for the terminating) iff it has the form $[\beta]\gamma$ where

- $\beta \in (V \cup \Sigma)^*$ can be derived in $G'$ in $k - 1$ steps,
- $\gamma \in \{S, \overline{S}\}\{A_1, \overline{A}_1\} \cdots \{A_m, \overline{A}_m\}$ and $|\gamma|_A = 1$ iff $|\beta|_A = 1$.

□

**Lemma 7.3.** $\mathrm{MAT}_{fin} \subseteq \mathrm{GS}_{cb}$.

*Proof.* We will simulate a step of a programmed grammar of finite index (including the information of the next rule to apply) by a series of steps of a capacity-bounded GS grammar. Let $G = (V, \Sigma, S, R, \sigma)$ be a programmed grammar of finite index $k$. The rules of $G$ are given as $R = \{A_1 \to$

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



$\alpha_1, \ldots, A_r \to \alpha_r\}$ and labeled by the symbols $Lab = \{l_1, \ldots, l_r\}$. The success field function $\sigma$ can thus be seen as a function $\sigma : \{1, \ldots, r\} \to 2^{\{1,\ldots,r\}}$. Moreover, define the words $\rho_i = l_1 \cdots l_{i-1} l_{i+1} \cdots l_r$. To construct the equivalent capacity-bounded GS grammar $G = (V', \Sigma, S', R', \kappa)$ we define the nonterminal set

$$V' = V \cup Lab \cup \{S', C, D, X_1, X_2, Y, [, ]\}$$

and the capacity function $\kappa : V' \to \mathbb{N}$ given by

$$\kappa(A') = \begin{cases} 1, & \text{for } A' \in Lab \cup \{X_1, X_2, Y\} \\ k+1, & \text{otherwise.} \end{cases}$$

Let $h : V^* \to (V')^*$ be the homomorphism defined by

$$h(x) = \begin{cases} x, & \text{if } x \in \Sigma \\ [xC], & \text{otherwise.} \end{cases}$$

The set of rules in $G'$ is constructed as follows. For each rule $l_i : A_i \to \alpha_i$, $1 \leqslant i \leqslant r$, and any $1 \leqslant j \leqslant k$, $R'$ contains the rules

$$\begin{aligned}
[A_i C] &\to [l_i X_1 Y], \\
[l_i X_1 Y] &\to [l_i X_2 Y], \\
[l_i X_2 Y] &\to h(\alpha_i)[C], \\
[\rho_i D^j X_2] &\to [\rho_i C^j], \\
[\rho_i C^j] &\to [\rho_i D^j X_1], \\
[\rho_i D^j X_1] &\to [\rho_m D^{j+1-|\alpha_i|_V} X_2], \text{ for } m \in \sigma(i).
\end{aligned}$$

Moreover, $R'$ contains the rules

$$S' \to [SC][\rho_i D^k X_2],\ [\rho_i D^{k+1} X_2] \to \lambda, 1 \leqslant i \leqslant r, \text{ and } [C] \to \lambda.$$

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



Consider a sentential form of the shape $h(\beta)[\rho_i D^j X_2]$ such that $|h(\beta)|_C + j = k + 1$. It can be read as follows: The original programmed grammar G has derived the sentential form $\beta$ with a number of C nonterminals and will apply the rule labeled $l_i$ in the next step. Indeed, if any rule is applicable, then it is $[A_i C] \to [l_i X_1 Y]$ and $h(\beta)$ has the factorization $h(\beta) = h(\beta_1)[A_i C]h(\beta_2)$. (The application of a rule $[A_j C] \to [l_j X_1 Y]$ with $j \neq i$ would introduce a second symbol $l_j$; the application of $[\rho_i D^j X_2] \to [\rho_i C^j]$ would generate a total count of $k + 1$ C's; the left-hand sides of the other rules do not appear in the sentential form.) The next seven steps are also forced, thus giving the following sequence of derivation steps in $G'$:

$$
\begin{aligned}
h(\beta_1)\underline{[A_i C]}h(\beta_2)[\rho_i D^j X_2] &\Rightarrow h(\beta_1)\underline{[l_i X_1 Y]}h(\beta_2)[\rho_i D^j X_2],\\
h(\beta_1)[l_i X_1 Y]h(\beta_2)\underline{[\rho_i D^j X_2]} &\Rightarrow h(\beta_1)[l_i X_1 Y]h(\beta_2)\underline{[\rho_i C^j]},\\
h(\beta_1)\underline{[l_i X_1 Y]}h(\beta_2)[\rho_i C^j] &\Rightarrow h(\beta_1)\underline{[l_i X_2 Y]}h(\beta_2)[\rho_i C^j],\\
h(\beta_1)[l_i X_2 Y]h(\beta_2)\underline{[\rho_i C^j]} &\Rightarrow h(\beta_1)[l_i X_2 Y]h(\beta_2)\underline{[\rho_i D^j X_1]},\\
h(\beta_1)\underline{[l_i X_2 Y]}h(\beta_2)[\rho_i D^j X_1] &\Rightarrow h(\beta_1)\underline{h(\alpha_i)[C]}h(\beta_2)[\rho_i D^j X_1],\\
h(\beta_1)h(\alpha_i)\underline{[C]}h(\beta_2)[\rho_i D^j X_1] &\Rightarrow h(\beta_1)h(\alpha_i)\_h(\beta_2)[\rho_i D^j X_1],\\
h(\beta_1)h(\alpha_i)h(\beta_2)\underline{[\rho_i D^j X_1]} &\Rightarrow h(\beta_1)h(\alpha_i)h(\beta_2)\underline{[\rho_m D^{j+1-|\alpha_i|_V} X_2]},\\
&\qquad \text{where } m \in \sigma(i).
\end{aligned}
$$

Consequently, every sentential form reachable after seven steps from $h(\beta)\,[\rho_i D^j X_2]$ with $|\beta|_V + j = k + 1$ has the form $h(\beta')[\rho_m D^{j'} X_2]$ with $|\beta'|_V + j' = k + 1$, where $\beta'$ is directly derived in G using the rule labeled $l_i$ and $m \in \sigma(i)$. Conversely, any sentential form of the above form can be derived in $G'$ using the seven steps from above.

The sentential forms reachable after the first step in $G'$ are exactly those of the form $[S][\rho_i D^k X_2]$, $1 \leqslant i \leqslant r$. We can conclude by induction that the sentential forms derivable in $G'$ in $(7n + 1)$ steps are exactly those of the forms $h(\beta)[\rho_i D^j X_2]$ with $|\beta|_V + j = k + 1$ such that $\beta$ is derivable in G in

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



$n$ steps and the rule $\rho_i$ is applicable. Moreover, the only way to generate a terminal word $w$ in $G'$ is to generate a sentential form $w[\rho_i D^{k+1} X_2]$ in $7n+1$ steps and then to apply the terminating rule $[\rho_i D^{k+1} X_2] \to \lambda$. This is by the above arguments if and only if $w$ is derivable in $G$, and thus $L(G') = L(G)$. □

## 7.4 Capacity-Bounded Context-Free Grammars

In this section, we investigate capacity-bounded context-free grammars. It turns out that they are strictly between context-free languages of finite index and matrix languages of finite index. Closure properties of capacity-bounded languages with respect to AFL operations are shortly discussed at the end of the section.

As a first result we show that the family of context-free languages with finite index is properly included in $CF_{cb}$.

**Lemma 7.4.** $CF_{fin} \subset CF_{cb}$.

*Proof.* Any context-free language generated by a grammar $G$ of index $k$ is also generated by the capacity-bounded grammar $(G, \kappa)$ where $\kappa$ is the capacity function constantly $k$. The properness of the inclusion follows from Example 7.1. □

An upper bound for $CF_{cb}$ is given by the inclusion $CF_{cb} \subseteq GS_{cb} = MAT_{fin}$. We can prove the properness of the inclusion by presenting a language from $MAT_{fin} \setminus CF_{cb}$. In order to show that a language is not in $CF_{cb}$, we can make use of the following "replacement lemma".

**Theorem 7.5.** For any infinite language $L \in CF_{cb}$, there are a constant $n$ and a finite set $\mathcal{L}$ of infinite languages from $CF_{cb}$ such that, for every word $z \in L$ with $|z| \geqslant n$, there are a decomposition $z = uvw$, $|v| \leqslant n$, and a language $L' \in \mathcal{L}$ such that $uv'w \in L$, for all $v' \in L'$.

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



*Proof.* Consider a capacity-bounded cf grammar $G = (V, \Sigma, S, R, 1)$ such that $L = L(G)$. For $A \in V$, let $G_A = (V, \Sigma, R, A, 1)$ and $L_A = L(G_A)$. The following holds for any derivation in G involving $A$: If $\alpha A \beta \Rightarrow_G^* uvw$, where $\alpha, \beta \in (V \cup \Sigma)^*$, $u, v, w \in \Sigma^*$ and $v$ is the yield of $A$, then $v \in L_A$.

The nonterminal set $V$ can be decomposed as $V = V_{inf} \cup V_{fin}$, where

$$\begin{aligned} V_{inf} &= \{A \in V \mid L_A \text{ is infinite}\}, \\ V_{fin} &= \{A \in V \mid L_A \text{ is finite}\}. \end{aligned}$$

We choose $\mathcal{L} = \{L_A \mid A \in V_{inf}\}$ and $n = r \cdot \max\{|w| \mid w \in \bigcup_{A \in V_{fin}} L_A\}$, where r is the longest length of a right side in a rule of R. For a derivation of $z \in L$ with $|z| > n$, consider the last sentential form with a symbol from $V_{fin}$. Let this symbol be $A$ and the sentential form be $\alpha_1 A \alpha_2$. All nonterminals in $\alpha_1 \alpha_2$ are from $V_{fin}$, and none of them generates a subword containing $A$ in the further derivation process. We get thus another derivation of $z$ in G by postponing the rewriting of $A$ until all other nonterminals have vanished by applying on them the derivation sequence of the original derivation. This new derivation has the form

$$S \Rightarrow_G^* \alpha_1 A \alpha_2 \Rightarrow_G^* uAw \Rightarrow_G^* uvw = z.$$

The length of $v$ can be estimated by $|v| \leqslant n$, as $A$ is in the first step replaced by a word over $(\Sigma \cup V_{fin})$ of length at most r. Finally, note that $uAw \Rightarrow_G^* uv'w$ holds for all $v' \in L_A$. Hence, any word $uv'w$ with $v' \in L_A$ can be derived in G. □

**Corollary 7.6.** $L = \{a^n b^n c^n \mid n \geqslant 1\} \notin CF_{cb}$.

*Proof.* Suppose, for contradiction, that $L \in CF_{cb}$ and that $n$ and $\mathcal{L}$ are the constant and the set of infinite languages from Theorem 7.5 with respect to L.

Consider the word $z = a^{n+1} b^{n+1} c^{n+1} \in L$. By Theorem 7.5, there

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



are a decomposition $z = uvw$, $|v| \leqslant n$, and a language $L' \in \mathcal{L}$ such that $uv'w \in L$, for all $v' \in L'$. For the decomposition, there are two cases: $u \in a^n b^+$ or $w \in b^+ c^n$. We discuss only the first case; the other is similar to the first one. Consider any $v' \in L'$ with $|v'| > |v|$. If $|v'|_a > 0$ then $uv'w$ is in $a^+b^+\{a,b,c\}^*a\{a,b,c\}^*$ and thus not in L. If $|v'|_a = 0$ then $|uv'w| > |uvw| = 3n+3$ and $|uv'w|_a = n+1$, and hence $uv'w \notin L$. □

The results can be summarized as follows:

**Theorem 7.7.** $CF_{fin} \subset CF_{cb} \subset GS_{cb} = MAT_{fin}$.

As regards closure properties, we remark that the constructions showing the closure of CF under homomorphisms, union, concatenation and Kleene closure can be easily extended to the case of capacity-bounded languages.

**Theorem 7.8.** $CF_{cb}$ is closed under homomorphisms, union, concatenation and Kleene closure.

*Proof.* Let $G_1 = (V_1, \Sigma_1, S_1, R_1, 1)$ and $G_2 = (V_2, \Sigma_2, S_2, R_2, 1)$ be capacity bounded grammars with $V_1 \cap V_2 = \emptyset$, and let $h : \Sigma_1^* \to \Delta^*$ be a homomorphism. Let $G_3, G_4, G_5, G_6$ be the capacity-bounded context-free grammars defined as

$$
\begin{aligned}
G_3 &= (V_1, \Delta, S_1, R_3, 1), \text{ with } R_3 = \{A \to g(\beta) \mid A \to \beta \in R_1\} \\
&\quad \text{where } g : (\Sigma_1 \cup V_1)^* \to (\Delta \cup V_1)^* \text{ is the homomorphism} \\
&\quad \text{defined by } g(A) = A, \text{ for } A \in V_1,\ g(a) = h(a), \text{ for } a \in \Sigma_1; \\
G_4 &= (V_1 \cup V_2 \cup \{S_4\}, \Sigma_1 \cup \Sigma_2, S_4, R_4) \text{ with} \\
&\quad R_4 = R_1 \cup R_2 \cup \{S_4 \to S_1, S_4 \to S_2\}; \\
G_5 &= (V_1 \cup V_2 \cup \{S_5\}, \Sigma_1 \cup \Sigma_2, S_5, R_5) \text{ with } R_5 = R_1 \cup R_2 \cup \{S_4 \to S_1 S_2\}; \\
G_6 &= (V_1 \cup \{S_6\}, \Sigma_1, S_6, R_6) \text{ with } R_6 = R_1 \cup \{S_6 \to S_1 S_6, S_6 \to \lambda\}.
\end{aligned}
$$

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



Then, $L(G_3) = h(L(G_1))$, $L(G_4) = L(G_1) \cup L(G_2)$, $L(G_5) = L(G_1)L(G_2)$, $L(G_6) = L(G_1)^*$. We will give a correctness proof only for the last equation.

To show $L(G_6) \subseteq L(G_1)^*$, one proves by induction on the number $n$ of derivation steps that any sentential form derivable in $G_6$ has the shape $\beta_1\beta_2\cdots\beta_k$ or $\beta_1\beta_2\cdots\beta_k S_6$ where $k \geqslant 0$ and $\beta_i$, $1 \leqslant i \leqslant k$ is derivable in $G_1$. Clearly, the statement is true for $n = 0$ as the word $S_6$ is of the claimed form. Now suppose that the statement holds for $n$. The only possible shapes of sentential forms derivable in $n+1$ steps are

| | |
|---|---|
| $\beta_1\beta_2\cdots\beta_k$ | ($S_6 \to \lambda$ applied on $\beta_1\beta_2\cdots\beta_k S_6$); |
| $\beta_1\beta_2\cdots\beta_k S_1 S_6$ | $S_6 \to S_1 S_6$ applied on $\beta_1\beta_2\cdots\beta_k S_6$); |
| $\beta_1\cdots\beta_{i-1}\beta_i'\beta_{i+1}\cdots\beta_k$ | (a rule from $R_1$ applied on substring $\beta_i$ of $\beta_1\beta_2\cdots\beta_k$); |
| $\beta_1\cdots\beta_{i-1}\beta_i'\beta_{i+1}\cdots\beta_k S_6$ | (a rule from $R_1$ applied on substring $\beta_i$ of $\beta_1\beta_2\cdots\beta_k S_6$). |

In the first two cases it is obvious that the sentential forms have the claimed shape. For the other cases, note that $|\beta_1\cdots\beta_{i-1}\beta_i'\beta_{i+1}\cdots\beta_k|_A \leqslant 1$ has to hold for all $A \in V_1$ which implies $|\beta_i'|_A \leqslant 1$, for all $A \in V_1$, and thus $\beta_i \Rightarrow_{G_1} \beta_i'$.

Conversely, any word $w = w_1w_2\cdots w_k$ with $w_i \in L_1$, for $1 \leqslant i \leqslant k$, can be obtained in $G_6$ by the derivation

$$S_6 \Rightarrow S_1S_6 \Rightarrow^* w_1S_6 \Rightarrow w_1S_1S_6 \Rightarrow^* w_1w_2S_6 \Rightarrow^* w_1w_2\cdots w_kS_6 \Rightarrow w_1w_2\cdots w_k$$

where the subwords $w_i$ are derived from $S_1$ as in $G_1$. □

Regarding intersection with regular sets and inverse homomorphisms, we can show non-closure properties.

**Theorem 7.9.** $CF_{cb}$ is neither closed under intersection with regular sets nor under inverse homomorphisms.

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



*Proof.* Let $L \in CF_{cb}$ be the language generated by the capacity-bounded grammar from Example 7.1. As discussed in the example, the intersection of $L$ with the regular set $M$ given there is

$$L_1 = \{a^{3n}x^3yb^{3n}a^{3n}x^2yb^{3n} \mid n \geqslant 1\}.$$

Moreover, let $g : \{a, b, c, d\}^* \to \{a, b, x, y\}$ be the homomorphism defined by $g(a) = a, g(b) = b, g(c) = x^3y, g(d) = x^2y$. It is easy to see that $g^{-1}(L) = L_2 = \{a^{3n}cb^{3n}a^{3n}db^{3n} \mid n \geqslant 1\}$. With the help of Theorem 7.5 it can be shown that $L_1, L_2 \notin CF_{cb}$. □

## 7.5 Control by Petri Nets with Place Capacities

Control by Petri nets can in a natural way be adapted to Petri nets with place capacities. A context-free grammar is controlled by its context-free Petri net with place capacity by only allowing derivations that correspond to valid firing sequences respecting the capacity bounds. The (trivial) proof for the equivalence between context-free grammars and grammars controlled by cf Petri nets can be immediately transferred to context-free grammars and Petri nets with capacities:

**Theorem 7.10.** Grammars controlled by context-free Petri nets with place capacity functions generate the family of capacity-bounded context-free languages.

Let us now turn to grammars controlled by extended cf Petri nets with capacities. Since an extended cf Petri net $N_x$, $x \in \{z, c, s\}$, has two kinds of places, i.e. places labeled by nonterminal symbols and *control* places, it is interesting to consider two types of place capacities in the Petri net: first, we demand that only the places labeled by nonterminal symbols are with capacities (*weak capacity*), and second, all places of the net are with capacities (*strong capacity*).

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



An $x$-Petri net $N_x = (P \cup Q, T, F \cup E, \varphi, \zeta, \gamma, \mu_0, \tau)$ is with *weak capacity* if the corresponding cf Petri net $(P, T, F, \phi, \iota)$ is with place capacity, and *strong capacity* if the Petri net $(P \cup Q, T, F \cup E, \varphi, \mu_0)$ is with place capacity. A grammar controlled by an $x$-Petri net with *weak* (*strong*) *capacity* is an $x$-Petri net controlled grammar $G = (V, \Sigma, S, R, N_x)$ where $N_x$ is with weak (strong) place capacity. We denote the families of languages generated by grammars (with erasing rules) controlled by $x$-Petri nets with weak and strong place capacities by $wPN_{cx}$, $sPN_{cx}$ ($wPN^{\lambda}_{cx}$, $sPN^{\lambda}_{cx}$), respectively, where $x \in \{z, c, s\}$.

With the similar manner, we also define grammars controlled by arbitrary Petri nets with place capacities. Let $G = (V, \Sigma, S, R, N, \gamma, M)$ be an arbitrary Petri net controlled grammar. G is called a grammar controlled by an arbitrary Petri net with place capacity if N is a Petri net with place capacity. The families of languages generated by grammars controlled by arbitrary Petri nets with place capacities (with erasing rules) is denoted by $PN_{cb}(x, y)$ ($PN^{\lambda}_{cb}(x, y)$) where $x \in \{f, -\lambda, \lambda\}$ and $y \in \{r, t, g\}$.

Before deriving results on the generative power of grammars controlled by extended Petri nets with capacities, we will first study the generative power of capacity-bounded matrix and vector grammars, which are closely related to these Petri net grammars.

**Theorem 7.11.** $MAT_{fin} = VEC^{[\lambda]}_{cb} = MAT^{[\lambda]}_{cb} = sMAT^{[\lambda]}_{cb}$.

*Proof.* We give the proof of $MAT_{fin} = VEC^{\lambda}_{cb}$. The other equalities can be shown in an analogous way. Since $MAT_{fin} = VEC_{fin} = VEC^{\lambda}_{fin}$, it suffices to prove $VEC_{fin} \subseteq VEC^{\lambda}_{cb}$ and $VEC^{\lambda}_{cb} \subseteq VEC^{\lambda}_{fin}$. The first inclusion is obvious because any vector grammar of finite index k is equivalent to the same vector grammar with capacity function constantly k.

To show $VEC^{\lambda}_{cb} \subseteq VEC^{\lambda}_{fin}$, consider a capacity-bounded vector grammar $G = (\{A_0, A_1, \ldots, A_m\}, \Sigma, A_0, M, 1)$. (The proof that it suffices to consider the capacity function 1 is like for usual grammars.) To construct an equiv-

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



alent vector grammar of finite index, we introduce the new nonterminal symbols $B_i, B_i'$, $0 \leqslant i \leqslant m$, $C, C'$. For any rule $r : A \to \alpha$, we define the matrix $\mu(r) = (C \to C', s_0, s_1, \ldots, s_m, r, C' \to C)$ such that $s_i = B_i \to B_i'$ if $A = A_i$ and $|\alpha|_A = 0$, $s_i = B_i' \to B_i$ if $A \neq A_i$ and $|\alpha|_{A_i} = 1$, and $s_i$ is empty, otherwise.

Now we can construct $G' = (V', \Sigma, S', M')$ where

$$V' = V \cup \{B_i, B_i' \mid 0 \leqslant i \leqslant m\} \cup \{S', C, C'\}$$

and $M'$ contains

- for any matrix $m = (r_1, \ldots, r_k)$, the matrix $m' = (\mu(r_1), \ldots, \mu(r_k))$,
- the start matrix $(S' \to A_0 B_0 B_1' \cdots B_m' C)$,
- the terminating matrix $(C \to \lambda, B_0' \to \lambda, B_1' \to \lambda, \ldots, B_m' \to \lambda)$.

The construction of $G'$ allows only derivation sequences where complete submatrices $\mu(r)$ are applied: when the sequence $\mu(r)$ has been started, there is no symbol $C$ before $\mu(r)$ is finished, and no other submatrix can be started. It is easy to see that $G'$ can generate after applying complete submatrices exactly those words $\beta\gamma C$ such that $\beta \in (V \cup \Sigma)^*$, $\gamma \in \{B_0, B_0'\}\{B_1, B_1'\} \cdots \{B_m, B_m'\}$ such that $\beta$ can be derived in $G$ and $|\gamma|_{B_i} = 1$ iff $|\beta|_{A_i} = 1$. Moreover, $G'$ is of index $2|V| + 1$. □

By constructions similar to those in Theorem 4.20 and Theorem 7.11, we can show with respect to weak capacities:

**Theorem 7.12.** For $x \in \{z, c, s\}$, $\mathrm{MAT}_{fin} = w\mathrm{PN}_{cx}^{[\lambda]}$.

*Proof.* We give only the proof for $x = z$. The other equations can be shown using analogous arguments. By Theorem 7.11 it is sufficient to show the inclusions $\mathrm{VEC}_{fin} \subseteq w\mathrm{PN}_{cz}$ and $w\mathrm{PN}_{cz}^{\lambda} \subseteq \mathrm{VEC}_{cb}^{\lambda}$.

As regards the first inclusion, let $L$ be a vector language of finite index (with or without erasing rules), and let $\mathrm{ind}(L) = k$, $k \geqslant 1$. Then, there is

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



a vector grammar $G = (V, \Sigma, S, M)$ such that $L = L(G)$ and $ind(G) \leqslant k$. Without loss of generality we assume that G is without repetitions. Let R be the set of the rules of M. By Lemma 4.19, we can construct a $z$-Petri net controlled grammar $G' = (V, \Sigma, S, R, N_z)$, $N_z = (P \cup Q, T, F \cup E, \varphi, \zeta, \gamma, \mu_0, \tau)$, which is equivalent to the grammar G. By definition, for every sentential form $w \in (V \cup \Sigma)^*$ in the grammar G, $|w|_V \leqslant k$. It follows that $|w|_A \leqslant k$ for all $A \in V$. By bijection $\zeta : P \cup Q \to V \cup \{\lambda\}$ we have $\mu(p) = \mu(\zeta^{-1}(A)) \leqslant k$ for all $p \in P$ and $\mu \in \mathcal{R}(N_z, \mu_0)$, i.e. the corresponding cf Petri net $(P, T, F, \phi, \beta, \gamma, \iota)$ is with k-place capacity. Therefore $G'$ is with weak place capacity.

On the other hand, the construction of an equivalent vector grammar for a $z$-Petri net controlled grammar, can be extended to the case of weak capacities just by assigning the capacities of the corresponding places to the nonterminal symbols of the grammar. □

As regards strong capacities, there is no difference between weak and strong capacities for grammars controlled by $c$- and $s$-Petri nets because the number of tokens in every circle is limited by 1. This yields:

**Corollary 7.13.** For $x \in \{c, s\}$, $MAT_{fin} = sPN^{[\lambda]}_{cx}$.

The only families not characterized yet are $sPN^{[\lambda]}_{cz}$. We conjecture that they are also equal to $MAT_{fin}$.

The next statement indicates that the language generated by a grammar controlled by an arbitrary Petri net with place capacity iff it is generated by a matrix grammar.

**Theorem 7.14.** For $x \in \{f, -\lambda, \lambda\}$ and $y \in \{r, t, g\}$,

$$PN_{cb}(x, y) = MAT \subseteq PN^{\lambda}_{cb}(x, y) = MAT^{\lambda}.$$

*Proof.* First, we show that the inclusion $PN^{\lambda}_{cb}(x, y) \subseteq MAT^{\lambda}$ holds for all $x \in \{f, -\lambda, \lambda\}$ and $y \in \{r, t, g\}$.

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



Let $G = (V, \Sigma, S, R, N, \gamma, M)$ be a grammar controlled by Petri net $N = (P, T, F, \varphi, \iota)$ with place capacity. Since the net $N$ is with place capacity, the set of all reachable markings, i.e., $\mathcal{R}(N, \iota)$ is finite. Thus, we can construct a finite automaton $\mathcal{A}$ whose states are the markings of the net $N$ and there is an arc from state $\mu$ to state $\mu'$ with label $t$ iff marking $\mu'$ is obtained from marking $\mu$ 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 $M' = \mathcal{R}(N, \iota)$ and the state-transition function $\delta : M' \times T \to M'$ is defined as $\delta(\mu, t) = \mu'$ iff $\mu \xrightarrow{t} \mu'$. It follows that $\sigma = t_1 t_2 \cdots t_n \in L(\mathcal{A})$ iff $\sigma$ is a successful occurrence sequence of transitions of $N$. Let $K = \{\gamma(\sigma) \mid \sigma \in L(\mathcal{A})\}$. Then, it is not difficult to see that $L(G) = L(G')$ where $G' = (V, \Sigma, S, R, K)$ is a regularly controlled grammar. Therefore, for all $x \in \{f, -\lambda, \lambda\}$ and $y \in \{r, t, g\}$, we have

$$PN_{cb}^{\lambda}(x, y) \subseteq MAT^{\lambda}. \tag{7.1}$$

Next, we show that the inverse inclusion also holds. By direct observation, we can see that Theorem 5.6 in Chapter 5 also holds for the grammars controlled by Petri nets with place capacities, i.e., for $y \in \{r, t, g\}$,

$$PN_{cb}^{[\lambda]}(f, y) = PN_{cb}^{[\lambda]}(-\lambda, y) = PN_{cb}^{[\lambda]}(\lambda, y). \tag{7.2}$$

Lemma 5.12 in Chapter 5 also maintains for $PN_{cb}^{[\lambda]}(-\lambda, r)$:

$$MAT^{[\lambda]} \subseteq PN_{cb}^{[\lambda]}(-\lambda, r). \tag{7.3}$$

Lemma 6.13 and Corollary 6.14 in Chapter 6 also hold for grammars controlled by Petri nets with place capacities as each place of a Petri net with place capacity has at most the upper bound, the set of reachable markings

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



is finite. Thus,

$$PN_{cb}^{[\lambda]}(\lambda, r) \subseteq PN_{cb}^{[\lambda]}(\lambda, g) \subseteq PN_{cb}^{[\lambda]}(\lambda, t) \tag{7.4}$$

From (7.1)-(7.3), we get

$$MAT^{[\lambda]} \subseteq PN_{cb}^{[\lambda]}(\lambda, r) \subseteq PN_{cb}^{[\lambda]}(\lambda, g) \subseteq PN_{cb}^{[\lambda]}(\lambda, t) \tag{7.5}$$

From (7.1) and (7.5) it follows that $MAT^{[\lambda]} = PN_{cb}^{[\lambda]}(x, y)$. □

We summarize our results in the following theorem.

**Theorem 7.15.** The relations in Figure 7.1 hold where $x \in \{f, -\lambda, \lambda\}$, $y \in \{r, g, t\}$, $i \in \{z, c, s\}$ and $j \in \{c, s\}$; the lines (arrows) denote (proper) inclusions of the lower families into the upper families.

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



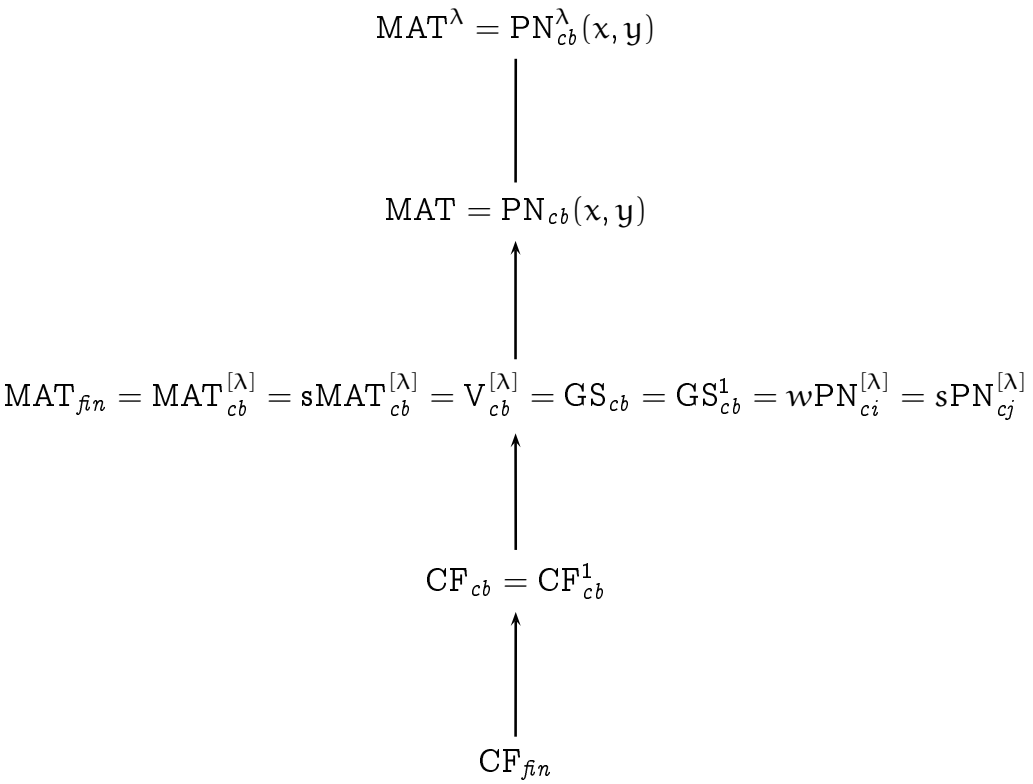


Figure 7.1: The hierarchy of language families generated by grammars with bounded capacities

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010

# 8

# Conclusions and Further Research

The thesis dealt with a new type of grammars with regulated rewriting; it introduced Petri net controlled grammars, and studied their computational and closure properties. It also concerned the close related topics: semi-matrix and capacity-bounded grammars. Though the theme of regulated grammars is one of the classic topics in formal language theory, a Petri net controlled grammar is still interesting subject for the investigation for many reasons. On the one hand, this type of grammars can successfully be used in modeling new problems emerging in manufacturing systems, systems biology and other areas, about which we spoke in Introduction. On the other hand, the graphically illustrability, the ability to represent both a grammar and its control in one structure, and the possibility to unify different regulated rewritings make this formalization attractive for the study. Moreover, control by Petri nets introduces the concept of concurrency in regulated rewriting systems.

In Chapter 3 we introduced semi-matrix grammars and contented our research only those properties of these grammars which were used in proofs of some statements in the next chapters. We investigated the closure properties of families of semi-matrix languages, and established the relationship to the families of matrix and vector languages. There still remain a lot of themes for the separate investigation, for instance, decision problems for

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010

semi-matrix grammars and computational, closure properties of unordered semi-matrix grammars. But the most interesting question, in our opinion, is to investigate the relationship of the families of matrix languages and vector languages. Since by structure of control, semi-matrix grammars are "between" matrix and vector grammars, they may establish the final connection matrix languages to vector languages.

In Chapter 4 we defined several concepts of control by Petri nets, by transforming a context-free grammar into an isomorphic Petri net called a cf Petri net, and enriching it by additional components (places and arcs). First, we defined k-Petri net controlled grammars, i.e., grammars controlled by cf Petri nets with k additional places in Section 4.3. We study the generative power and closure properties of the family of languages generated by k-Petri net controlled grammars. The main contribution of this section is the fact that the families of languages generated by k-Petri net controlled grammars form infinite hierarchy with respect to the number of additional places. In Section 4.4 we showed that some well-known grammars with regulated rewritings can be simulated by extended cf Petri net controlled grammars.

There are various related problems that deserve further investigation.

1. The study of the decision power of this families of languages remains untouched. k-Petri net controlled grammars at most generate vector languages, it suggests that we may prove the decidability for many problems.

2. Our primary investigations reveals that the matrix grammars, vector grammars, semi-matrix grammars, random context grammars [67], and other grammars can be simulated by Petri net controlled grammars. This motivates us to study the unified theory of grammars with regulated rewritings in future.

3. In the section we investigated the hierarchy problem with respect

ISBN:978-84-693-1536-1/DL:T-644-2010

to the number of additional places. It is also interesting to study this problem with regards to the fixed number of additional arcs. Let $PN(n, m_1, m_2)$ denote the family of languages generated by grammars controlled by $(n, m_1, m_2)$-Petri nets where $n$ is the number of additional places, $m_1$ is the number additional incoming arcs to additional places, and $m_2$ is the number of additional outgoing arcs from additional places. For instance, the families $PN(1, 2, 1)$ and $PN(1, 1, 2)$ are incomparable as it is not difficult to see that

$$\{a^{k_1}b^{k_2}c^{k_3} \mid k_1 + k_2 = k_3\} \in PN(1, 2, 1) - PN(1, 1, 2)$$

while

$$\{a^{k'_1}b^{k'_2}c^{k'_3} \mid k'_1 = k'_2 + k'_3\} \in PN(1, 1, 2) - PN(1, 2, 1).$$

4. In the investigated grammars, the control in extended cf Petri nets are restricted to some specific subnets: places, chains or cycles. One may ask what family of languages is generated if arbitrary subnet is considered? The study of this topic may be interesting at least for the reason to complete the variations of extended cf Petri net controlled grammars.

In Chapter 5 we defined arbitrary Petri net controlled grammars which are generalizations both regularly controlled grammars and extended cf Petri net controlled grammars. In the former case, instead of a finite automaton, a Petri net is associated with a context-free grammar and required that the sequence of applied rules corresponds to an occurrence sequence of the Petri net. In the latter case, instead of two bijections, only one transition labeling function is considered. We studied nine families of languages generated by arbitrary Petri net controlled grammar resulted from the different labeling policies and the different definitions of final marking

ISBN:978-84-693-1536-1/DL:T-644-2010

sets, and showed that the labeling does not effect the generative capacity. It is also shown that the introduced families between the families MAT and $\text{MAT}^{\lambda}$. Thus it gives hope that one can solve the problem of strictness of the inclusion $\text{MAT} \subseteq \text{MAT}^{\lambda}$ using Petri net controlled grammars.

In Chapter 6 we continued our study of arbitrary Petri net controlled grammars by restricting Petri nets to structural subclasses, i.e., special Petri nets such as state machines, marked graphs, free-choice nets, and many others. The observation showed that the labeling policy in special Petri nets does not also effect on computational power. On the other hand, we obtained some interesting results: first, it was demonstrated that the families of matrix languages and languages generated by state machine controlled grammars have the same generative power; second, 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. But there still remain several questions to investigate such as the generative capacity and other mathematical properties of the families of languages generated by generalized state machine, (generalized) marked graph, casual net controlled grammars; the relationship of the family of vector languages to the family of languages generated by marked graph controlled grammars.

In Chapter 7 we concentrated our attention to study the behavior of Petri net controlled grammars under dynamical changes of Petri nets, namely we considered cf, extended cf and arbitrary Petri nets with place capacities, and investigated generative power and closure properties corresponding families of languages. We also defined capacity-bounded grammars as counterparts of grammars controlled by cf Petri nets with place capacities. As the main results in this chapter we mark out that capacity-bounded context-free grammars have a larger generative power than context-free grammars of finite index while the family of languages generated by capacity-bounded GS grammars (due to phrase structure Ginsburg and

ISBN:978-84-693-1536-1/DL:T-644-2010

Spanier) and several families of languages generated by grammars controlled by extended cf Petri nets with place capacities coincide with the family of matrix languages of finite index.

In our further investigations we would like to consider grammars controlled by k-Petri nets with places capacities. There is also another very interesting direction for the study. If we notice the definitions of derivation-bounded [43] or nonterminal-bounded grammars [5, 6, 8] only nonterminal strings are allowed as left-hand sides of production rules. Here, an interesting question is emerged, what kind of languages can be generated if we derestrict this condition, i.e., allow any string in the left-hand side of the rules? Let us consider the grammar $G = (\{S, A, B, C, D\}, \{a, b, c\}, S, R)$ where R consists of

$$\begin{array}{lll} S \to abc, & S \to aAbc, & aA \to aaB, \\ Bb \to bbC, & Cb \to bC, & Cc \to Dcc, \\ Cc \to cc, & bD \to Db, & aD \to aA. \end{array}$$

It is not difficult to see that the grammar G generates non-context-free grammar $\{a^n b^n c^n \mid n \geqslant 1\}$ where in each sentential form there is only one nonterminal symbol. Thus, it may be an interesting topic for further study to investigate derivation-bounded, nonterminal-bounded and capacity-bounded grammars with arbitrary phrase structure production rules.

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010

# Bibliography

[1] A. Abraham. Some questions of phrase-structure grammars. *Comput. Linguistics*, 4:61–70, 1965.

[2] A.V. Aho. Indexed grammars. An extension of context-free grammars. *Journal of the ACM*, 15:647–671, 1968.

[3] M. Al-A'ali and A.A. Khan. Removing useless productions of a context-free grammar through Petri net. *Journal of Computer Science*, 3(7):494–498, 2007.

[4] M. Al-A'ali, A.A. Khan, and N. Al-Shamlan. Simplification of context-free grammar through Petri net. *Computers and Structures*, 58:1055–1058, 1996.

[5] E. Altman. The concept of finite representability. Systems Research Center Report SRC 56-A-64-20, Case Institute of Technology, 1964.

[6] E. Altman and R. Banerji. Some problems of finite representability. *Information and Control*, 8:251–263, 1965.

[7] R.B. Altman, M. Peleg, and I. Yeh. Modelling biological processes using workflow and Petri net models. *Bioinformatics*, 18(6):825–837, 2002.

[8] R. Banerji. Phrase structure languages, finite machines, and channel capacity. *Information and Control*, 6:153–162, 1963.

[9] M.H. ter Beek and H.C.M. Kleijn. Petri net control for grammar systems. In *Formal and Natural Computing*, volume 2300 of *LNCS*, pages 220–243. Springer, 2002.

[10] B. Brainerd. An analog of a theorem about context-free languages. *Information and Control*, 11:561–567, 1968.

[11] M. Chen. Modelling and simulation of metabolic networks. In *Proc. ESM – 2002*, pages 441–444, 2002.

[12] N. Chomsky. Three models for the description of languages. *IRE Trans. on Information Theory*, 2(3):113–124, 1956.

[13] N. Chomsky. *Syntactic structure.* Mouton, Gravenhage, 1957.

[14] N. Chomsky. On certain formal properties of grammars. *Information and Control*, 2:137–167, 1959.

[15] D. Cook, L. Holder, and J. Kukluk, editors. *Learning node replacement graph grammars in metabolic pathways. International Conference on Bioinformatics and Computational Biology*, 2007.

[16] A.B. Cremers, H.A. Maurer, and O. Mayer. A note on leftmost restricted random context grammars. *Inform. Proc. Letters*, 2:31–33, 1973.

[17] A.B. Cremers and O. Mayer. On vector languages. *J. Comp. Syst. Sci.*, 8:158–166, 1974.

[18] S. Crespi-Reghizzi and D. Mandrioli. Petri nets and commutative grammars. Internal Report 74-5, Laboraterio di Calcolatori, Instituto di Elettrotecnica ed Elettromca del Politecnico di Milano, Italy, 1974.

[19] S. Crespi-Reghizzi and D. Mandrioli. Properties of firing sequences. In *Proc. MIT Conf. Petri Nets and Related Methods*, pages 233–240, MIT, Cambridge, Mass., 1975.

[20] S. Crespi-Reghizzi and D. Mandrioli. Petri nets and Szilard languages. *Inform. and Control*, 33:177–192, 1977.

[21] P. Darondeau. On the Petri net realization of context-free graphs. *Theor. Computer Sci.*, 258:573–598, 2001.

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



[22] J. Dassow. Subregularly controlled derivations: Context-free case. *Rostock. Math. Kolloq.*, 34:61–70, 1988.

[23] J. Dassow, H. Fernau, and Gh. Păun. On the leftmost derivation in matrix grammars. *International Journal of Foundations in Computer Science*, 10:61–79, 1999.

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

[25] J. Dassow and B. Truthe. Subregularly tree controlled grammars and languages. In E. Csuhaj-Varjú and Z. Esik, editors, *Proc. the 12th International Conference AFL 2008*, pages 158–169, Balatonfured, Hungary, 2008.

[26] J. Dassow and S. Turaev. Arbitrary Petri net controlled grammars. In G. Bel-Enguix and M.D. Jiménez-López, editors, *Linguistics and Formal Languages. Second International Workshop on Non-Classical Formal Languages In Linguistics, Tarragona, Spain*, pages 27–39, 2008. ISBN 978-84-612-6451-3.

[27] J. Dassow and S. Turaev. $k$-Petri net controlled grammars. Preproceedings of Second International Conference on Language and Automata Theory and Applications. Report 36/08, Research Group on Mathematical Linguistics, Universitat Rovira i Virgili, Tarragona, Spain, 2008.

[28] J. Dassow and S. Turaev. $k$-Petri net controlled grammars. In C. Martín-Vide, F. Otto, and H. Fernau, editors, *Language and Automata Theory and Applications. Second International Conference, LATA 2008. Revised Papers*, volume 5196 of *LNCS*, pages 209–220. Springer, 2008.

[29] J. Dassow and S. Turaev. Grammars controlled by special Petri nets. In A.H. Dediu, A.-M. Ionescu, and C. Martín-Vide, editors, *Language and Automata Theory and Applications, Third International Conference, LATA 2009*, volume 5457 of *LNCS*, pages 326–337. Springer, 2009.

[30] J. Dassow and S. Turaev. Petri net controlled grammars: the case of special Petri nets. *Journal of Universal Computer Science*, 15(14):2808–2835, 2009.

[31] J. Dassow and S. Turaev. Petri net controlled grammars: the power of labeling and final markings. *Romanian Jour. of Information Science and Technology*, 12(2):191–207, 2009.

[32] J. Dassow and S. Turaev. Petri net controlled grammars with a bounded number of additional places. *Acta Cybernetica*, 2009. (To appear).

[33] A. Doi, S. Fujita, H. Matsuno, S. Miyano, and M. Nagasaki. Towards biopathways modeling and simulation. In *Applications and Theory of Petri Nets 2003*, volume 2679 of *LNCS*, pages 3–22. Springer, 2003.

[34] X. Erqing. A Pr/T-Net model for context-free language parsing. In *Fifth World Congress on Intelligent Control and Automation*, volume 3, pages 1919–1922, 2004.

[35] J. Esparza. Petri nets, commutative context-free grammars, and basic parallel processes. *Fundam. Inf.*, 31(1):13–25, 1997.

[36] B. Farwer, M. Jantzen, M. Kudlek, H. Rölke, and G. Zetzsche. Petri net controlled finite automata. *Fundamenta Informaticae*, 85(1-4):111–121, 2008.

[37] B. Farwer, M. Kudlek, and H. Rölke. Petri-net-controlled machine models. Technical Report 274, FBI-Bericht, Hamburg, 2006.

[38] B. Farwer, M. Kudlek, and H. Rölke. Concurrent Turing machines. *Fundamenta Informaticae*, 79(3-4):303–317, 2007.

[39] H. Fernau and M. Holzer. Conditional context-free languages of finite index. In *New Trends in Formal Languages – Control, Cooperation, and Combinatorics (to Jürgen Dassow on the occasion of his 50th birthday)*, volume 1218 of *LNCS*, pages 10–26, 1997.

[40] H. Fernau and M. Holzer. Regulated finite index language families collapse. Technical Report WSI-96-16, Universität Türbingen, Wilhelm-Schickard-Institut für Informatik, 2008.

[41] I. Fris. Grammars with partial ordering of the rules. *Inform. Control*, 12:415–425, 1968.

[42] S. Ginsburg and E.H. Spanier. Contol sets on grammars. *Math. Syst. Th.*, 2:159–177, 1968.

[43] S. Ginsburg and E.H. Spanier. Derivation bounded languages. *J. Comput. Syst. Sci.*, 2:228–250, 1968.

[44] R.L. Graham, Groetschel M., and L. Lovász, editors. *Handbook of Combinatorics*, volume 1–3. Elsevier, North-Holland, Amsterdam and MIT Press, Cambridge, Mass., 1996.

[45] S.A. Greibach and J.E. Hopcroft. Scatered context grammars. *J. Comput. Syst. Sci.*, 3:232–247, 1969.

[46] M. Hack. *Decidablity questions for Petri nets.* PhD thesis, Dept. of Electrical Engineering, MIT, 1975.

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



[47] M. Hack. Petri net languages. Computation Structures Group Memo, Project MAC 124, MIT, Cambridge Mass., 1975.

[48] D. Hauschildt and M. Jantzen. Petri nets algorithms in the theory of matrix grammars. *Acta Informatica*, 31:719–728, 1994.

[49] R. Heinrich and Schuster S. The modeling of metabolic systems. Structure, control and optimality. *Biosystems*, 47:61–77, 1998.

[50] Meinecke F. Hofestädt, R. Interactive modelling and simulation of biochemical networks. *Computers in Biology and Medicine*, 25(3):321–334, 1995.

[51] R.A. Hofestädt. Grammatical formalization of metabolic processes. In *Proceedings of the 1st International Conference on Intelligent Systems for Molecular Biology*, pages 181–189. AAAI Press, 1993.

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

[53] J.E. Hopcroft and J.D. Ullman. *Introduction to automata theory, languages, and computation.* Addison-Wesley Longman Publishing Co., Inc., 1990.

[54] O.H. Ibarra. Simple matrix grammars. *Inform. Control*, 17:359–394, 1970.

[55] M. Jantzen. On the hierarchy of Petri net languages. *RAIRO Onformatique Thérique/Theoretical Informatics*, 13(1):19–30, 1979.

[56] M. Jantzen. Language theory of Petri nets. In *Advances in Petri Nets*, volume 254-I of *LNCS*, pages 397–412. Springer, 1987.

[57] M. Jantzen, M. Kudlek, and G. Zetzsche. Language classes defined by concurrent finite automata. *Fundamenta Informaticae*, 85(1-4):267–280, 2008.

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



[58] M. Jantzen and R. Valk. Formal properties of place/transition nets. In W. Brauer, editor, *Net Theory and Applications*, volume 84 of *LNCS*, pages 165–212. Springer, 1980.

[59] C.J. Jiang. Vector grammar and PN machine. *Sci. Chin. (Ser. A)*, 24:1315–1322, 1996.

[60] S.A. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theor. Biol.*, 22:437–467, 1969.

[61] S.A. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theor. Biol.*, 22:437–467, 1969.

[62] J. Kelemen. Conditional grammars: motivations, definitions and some properties. In I. Peak and J. Szep, editors, *Proc. Conf. Autom., Lang., Math. Sci.*, pages 110–123, Salgofarjan, 1984.

[63] M.C. Kohn and W.J. Letzkus. A graph-theoretical analysis of metabolic regulation. *J. Theor. Biol.*, 100:293–304, 1983.

[64] M.N. Leibman, M.L. Mavrovouniotis, and V.N. Reddy. Petri net representation in metabolic pathways. In *Proc. First ISMB*, pages 328–336, 1993.

[65] M.K. Levitina. On some grammars with global productions. *NTI Ser.*, 2(3):32–36, 1972.

[66] R. Liptop. The reachability problem requires esponential space. Technical Report 62, Yale University, 1976.

[67] V. Marek and M. Češka. Petri nets and random-context grammars. In *Proc. of the 35th Spring Conference: Modelling and Simulation of Systems*, pages 145–152, MARQ Ostrava, Hardec nad Moravicí, 2001.

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



[68] C. Martín-Vide, V. Mitrana, and Gh. Păun, editors. *Formal languages and applications*. Springer-Verlag, Berlin, 2004.

[69] E. Mjolsness and G. Yosiphon. Towards the inference of stochastic biochemical network and parameterized grammar models. *Computational Systems Biology*, 2009. to appear.

[70] E. Moriya. Associate languages and derivational complexity of formal grammars and languages. *Information and Control*, 22:139–162, 1973.

[71] T. Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, 1989.

[72] M. Parigot and E. Pelz. A logical approach of Petri net languages. *Theoretical Computer Science*, 39:911–924, 1985.

[73] J.L. Peterson. Computation sequence sets. *J. Computer and System Sciences*, 13:1–24, 1976.

[74] J.L. Peterson. *Petri net theory and modeling of systems*. Prentice-Hall, Englewood Cliffs, NJ, 1981.

[75] C.A. Petri. *Kommunication mit Automaten*. PhD thesis, University of Bonn, 1962.

[76] Gh. Păun. On the index of grammars and languages. *Inf. Contr.*, 35:259–266, 1977.

[77] Gh. Păun. Linear simple matrix languages. *EIK*, 14:377–384, 1978.

[78] Gh. Păun. On the family of finite index matrix languages. *JCSS*, 18(3):267–280, 1979.

[79] Gh. Păun. A new generative device: valence grammars. *Rev. Roum. Math. Pures Appl.*, 25:911–924, 1980.

[80] Gh. Păun. *Matrix grammars*. Scientific and Encyclopadic Publ., Bucharest, 1981.

[81] Gh. Păun. On leftmost derivation restriction in regulated rewriting. *Rev. Roum. Math. Pures Appl.*, 30:751–758, 1985.

[82] W. Reisig and G. Rozenberg, editors. *Lectures on Petri Nets I: Basic Models*, volume 1491 of *LNCS*, Berlin, 1998. Springer.

[83] L.P.A. Robichaud. *Signal flow graphs: Applications*. Prentice-Hall, Englewood Cliffs, NJ, 1962.

[84] G. Rozenberg. More on ET0L systems versus random context grammars. *IPL*, 5(4):102–106, 1976.

[85] G. Rozenberg and A. Salomaa, editors. *Handbook of formal languages*, volume 1–3. Springer, 1997.

[86] G. Rozenberg and D. Vermeir. On ET0L systems of finite index. *Inf. Contr.*, 38:103–133, 1978.

[87] G. Rozenberg and D. Vermeir. On the effect of the finite index restriction on several families of grammars. *Inf. Contr.*, 39:284–302, 1978.

[88] G. Rozenberg and D. Vermeir. On the effect of the finite index restriction on several families of grammars; Part 2: context dependent systems and grammars. *Foundations of Control Engineering*, 3(3):126–142, 1978.

[89] D.J. Rozenkrantz. Programmed grammars and classes of formal languages. *Journal of the ACM*, 16:107–131, 1969.

[90] A. Salomaa. Matrix grammars with leftmost restriction. *Inform. Control*, 20:143–149, 1972.

UNIVERSITAT ROVIRA I VIRGILI
PETRI NET CONTROLLED GRAMMARS
Sherzod Turaev
ISBN:978-84-693-1536-1/DL:T-644-2010



[91] R. Siromoney. On equal matrix grammars. *Inform. Control*, 14:135–151, 1972.

[92] R. Siromoney and Krithivasan. Parallel context-free languages. *Inform. Control*, 24:155–162, 1974.

[93] P.H. Starke. Free Petri net languages. In *Mathematical Foundations of Computer Science 1978*, volume 64 of *LNCS*, pages 506–515, Berlin, 1978. Springer.

[94] P.H. Starke. *Petri-Netze*. Deutscher Verlag der Wissenschaften, 1980.

[95] R. Stiebe and S. Turaev. Capacity bounded grammars. *Journal of Automata, Languages and Combinatorics*, 2009. (Submitted).

[96] R. Stiebe and S. Turaev. Capacity bounded grammars and Petri nets. *EPTCS*, 3:193–203, 2009.

[97] R. Stiebe and S. Turaev. Capacity bounded grammars and Petri nets. In J. Dassow, G. Pighizzini, and B. Truthe, editors, *Eleventh International Workshop on Descriptional Complexity of Formal Systems, Magdeburg, Germany*, pages 247–258, 2009.

[98] J.U. Thoma. *Introduction to bond graphs and their applications*. Pergamon Press, New York, 1975.

[99] S. Turaev. Semi-matrix grammars. In *Second Doctoral Workshop on Mathematical and Engineering Methods in Computer Science, MEMICS 2006*, pages 245–252, Mikulov, Czechia, 2006. ISBN 80-214-3287-X.

[100] S. Turaev. Petri net controlled grammars. In *Third Doctoral Workshop on Mathematical and Engineering Methods in Computer Science, MEMICS 2007*, pages 233–240, Znojmo, Czechia, 2007. ISBN 978-80-7355-077-6.