Technische Universität München
Fakultät für Informatik

Lehrstuhl für Effiziente Algorithmen

Generalized Petri Nets: Algorithms and Complexity

Jeremias Weihmann

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr. H. Seidl

Prüfer der Dissertation:

1. Univ.-Prof. Dr. E. W. Mayr
2. Univ.-Prof. Dr. Dr. h. c. J. Esparza

Die Dissertation wurde am 13.03.2014 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 02.03.2015 angenommen.

Abstract

Classical computational problems for Petri nets, like the reachability problem, exhibit in general high computational complexities. For many classes of *ordinary* Petri nets, not only better upper bounds can be shown but also completeness-results have been found. The arguments used to obtain such tight upper bounds usually exploit the restrictions imposed on the Petri nets of the class in combination with their property of being ordinary. Since it is often not obvious how to adapt these arguments for Petri nets with arbitrary edge multiplicities, some classes consisting of such Petri nets are still not very well understood even if the corresponding classes of ordinary Petri nets are.

The main goal of this thesis is to gain a better understanding of such generalized classes. Recurring motives in our investigation are permutation techniques and canonical firing sequences. We provide a framework for approaching classical computational problems like the reachability, liveness, boundedness, covering, equivalence and containment problems, given the Petri nets of the considered class allow for canonical firing sequences that lead to reachable markings and have the following properties: almost all transitions are contained in short loops (i. e., sequences with nonnegative effect on the marking), and the backbone sequence resulting from removing all loops is still a firing sequence. Depending on the length of the backbone and the loops, space bounds for our problems of interest can be given.

We apply this framework to four generalized classes: conservative Petri nets, generalized communication-free Petri nets (gcf-PNs), inverse generalized communication-free Petri nets, and generalized S-systems. These classes are the natural generalizations of ordinary 1-conservative Petri nets, communication-free Petri nets, inverse communication-free Petri nets, and S-systems/state machines, respectively. Here, the difficult task is to show that the prerequisites of the framework are satisfied. By additionally providing lower bounds, we obtain **PSPACE**-completeness for almost all classical problems of these classes. Exceptions are the equivalence and containment problems, for which we find upper bounds that depend on the concrete form of the respective canonical firing sequences. We also investigate several problems involving the concept of home spaces, for which we give upper bounds, using the semilinear set representations provided by the framework.

In addition to the classes named above, we also take a closer look at communication-free Petri nets (cf-PNs). In contrast to gcf-PNs, for which the problems mentioned earlier are at least **PSPACE**-hard, we discover very efficient (linear or quadratic time) algorithms for a number of distinct problems of cf-PNs. Other classes we investigate in this thesis are generalized conflict-free Petri nets, and ring Petri nets with arbitrary edge multiplicities. For the first class, we show that the reachability problem is in Σ_2^p. For ring Petri nets, we show that this problem is in **coNP**, and even decidable in polynomial time if all edge multiplicities are powers of the same number. Furthermore, we consider two new classes of grammars and commutative grammars. We show that the uniform word problem of exponent-sensitive commutative grammars, which are the natural counterpart of gcf-PNs, is **PSPACE**-complete. In contrast to this, we find that the uniform word problem of exponent-sensitive grammars, which are their non-commutative equivalent, is undecidable.

Acknowledgments

First of all, I'd like to express my deep gratitude to my supervisor Ernst W. Mayr. He gave me the chance at the *Chair for Efficient Algorithms* to investigate a very interesting topic, encouraged my interest and drive, and supported me with his immense knowledge. I thank Volker Heun, one of my professors when I was studying computational biology, for drawing my interest in the direction of efficient algorithms, which should later lead me to this chair. I'd like to thank Javier Esparza for taking his time to discuss some topics on Petri nets with me. I thank Hanjo Täubig for inviting me to take part in his research, for the excellent teamwork in research and teaching, and for his support. I'm very grateful to Johannes Krugel for his support, and for his outstanding work with the student tutors. I also thank him for reading a part of a draft of my thesis and for his feedback. I'd like to thank my colleagues Harald Räcke, Chris Pinkau, Chintan Shah, Stefan Toman, and Werner Meixner for providing a very enjoyable working atmosphere and excellent teamwork. I thank Edith Zeitlmann for managing our publications, and Ernst Bayer for providing technical support. Special thanks go to Christine Lissner who not only manages a large amount of the organization at the chair but was also able to revive one of my plants I already had given up on. I also thank all current and former members of the chair which I didn't mention by name. Last but not least, I thank my family for the constant support over the many years I have been working on this thesis.

Contents

1 Introduction

1.1 Motivation

Petri nets are an important tool for modeling discrete transition systems that are for instance characterized by parallel, concurrent, nondeterministic behavior or shared resources. The reasons for their importance are manifold. Petri nets are intuitive. It is not difficult to understand the transitional semantics and the meaning of places and transitions within this semantics. Petri nets have a transitional semantics that is defined in terms of discrete steps and discrete states of the system. This makes them suited for many applications which cannot easily be modeled or approximated by, e. g., continuous models. Petri nets are powerful but not too powerful. They are powerful enough to be the model of choice for many applications. However, they are not Turing-complete, and therefore, many decision problems are decidable whose counterparts of Turing complete formalisms are not. Moreover, many subclasses of Petri nets allow for low upper bounds for the complexity of these problems. This often makes Petri nets of such subclasses suited as over- or under-approximations of the modeled system since the properties of these Petri nets can easily be analyzed.

The theory concerning how Petri nets can be applied as a modeling tool is often called *applied Petri net theory* [Pet81]. Its counterpart is often called *pure Petri net theory*, and concentrates on providing the concepts, tools, and techniques for the application of Petri nets. On a basic level, this involves gaining fundamental insight into the behavior of Petri nets as a device of computation. The focus of this thesis lies on pure Petri net theory. Of particular interest are some of the most classical and arguably most important computational problems involving Petri nets:

- The reachability problem asks if a given marking is reachable in a given Petri net.
- The liveness problem asks if each reachable marking of a given Petri net enables, for each transition t, a transition sequence containing t.
- The boundedness problem asks if there are finitely many markings reachable in a given Petri net.
- The covering problem asks if there is a marking reachable in a given Petri net that is component wise at least a given marking.
- The equivalence problem asks if the sets of reachable markings of two given Petri nets are the same.
- The containment problem asks if, given two Petri nets, the set of reachable marking of the first net is a subset of the set of reachable markings of the second net.

These problems have been studied for many years. It turned out that, in general, most of them have a very high computational complexity. The reachability, boundedness, and liveness problems require at least $2^{c\sqrt{s}}$ space infinitely often, where s is the encoding size of the corresponding vector addition system and end vector [Hac74b; Lip76]. The equivalence and containment problems are even undecidable [Bak73; Hac73; Hac74a; Hac76]. The high complexities of these problems result from the high computational power of Petri nets combined with their property of being infinite state systems, exhibiting the state explosion problem as a symptom. For many restricted subclasses of Petri nets, the complexities of these problems turned out to be much better. This, and the desire for more insight into the behavior of Petri nets in general led researchers to not only investigate Petri nets in general but also restricted subclasses.

In [May84], Mayr proposed a non-primitive recursive algorithm for the general Petri net reachability problem, thus proving its decidability. However, the nets of many Petri net classes for which the complexity of the reachability problem could be refined are forward-ordinary, i. e., they are subject to the restriction that all edges from places to transitions have multiplicity one. Well-known examples of forward-ordinary Petri nets with **NP**-complete reachability problems are communication-free Petri nets, [Esp97; Yen97], conflict-free Petri nets [HR88; How+89], normal Petri nets, and sinkless Petri nets [How+89] (for the latter two, the promise problem variation of the reachability problem was considered). Other forward-ordinary Petri nets for which completeness-results were obtained include 1-save nets [Che+95] (**PSPACE**-complete), and many subclasses of free-choice Petri nets (see, e. g., [Che+95; Esp98] as well as [DE95] for an introduction to the extensive theory on free-choice Petri nets).

Notable examples for Petri net classes with arbitrary edge multiplicities and matching lower and upper bounds for the reachability problem are single-path Petri nets [How+93] (**PSPACE**-complete), reversible Petri nets [Car+76; MM82] (**EXPSPACE**-complete), cycle-free Petri nets [HI88; Mur89; Ste91][1] (**NP**-complete). For more comprehensive overviews we refer to [EN94; Mur89].

It is well known that the reachability problem of Petri nets with arbitrary edge multiplicities can be reduced in polynomial time to the reachability problem of ordinary Petri nets. However, this does not automatically imply that such a reduction can be performed for a class of general Petri nets that are, for instance, topologically restricted to the problem involving the corresponding class of ordinary Petri nets. In general, it cannot be ruled out (in particular as long as common assumptions like **NP** $\subsetneq$ **PSPACE** are not proven to be false) that such a reduction is either impossible to begin with or needs superpolynomial time. (An example for this observation is the case of communication-free Petri nets and generalized communication-free Petri nets, as we will see later). Another problem in the context of Petri nets with arbitrary edge multiplicities is that nice characterizations for, e. g., markings to be reachable or Parikh vectors to be enabled, are hard to find. In some cases, even if such characterizations exist, they are algorithmically hard to check. While topological structures like traps, siphons, or strongly connected components can often be used for such characterizations in the case of forward-ordinary Petri nets, it is most often not obvious how to apply them in case of general Petri nets.

One reason is the relationship between places and transitions. The places of forward-ordinary Petri nets have a “binary” relationship with the transitions: If a transition increases the number of tokens at some place, then we can say for certain that this place has enough tokens for any transition. That is, each place that has at least one token is not responsible if some transition is not enabled. This intrinsic property of ordinary Petri nets is lost in case of Petri nets with arbitrary edge multiplicities. If a transition t of such a net puts tokens onto a place p, then the answer to the question whether the place prevents some transition t' depends on many different aspects: the number of tokens p contained before t took place, the number of tokens t removed from p and put onto p, and the number of tokens t' needs from p in order to be enabled. These mutual dependencies often make it difficult or even impossible to characterize enabled firing sequences, enabled Parikh vectors, or reachable markings in terms of elegant and easily checkable structural and behavioral properties.

As a consequence, results for many classes of Petri nets that are not restricted w. r. t. edge multi-

[1] Only cycle-free Petri nets with edge multiplicity one for all edges were considered but **NP**-membership also holds in the general case.

plicities have been obtained, e. g., for properties that only depend on structural features of the net and are independent of the initial marking, like structural unboundedness or liveness, or for classical problems (like reachability) subjected to strict conditions (e. g., the Petri net is assumed to be bounded or live). Similar observations apply to the other problems named earlier.

The main goal of this work is to obtain a better understanding of classes of generalized Petri nets and complexity results for problems involving such classes. Our most important contributions for this purpose are the following. We develop a framework, based on canonical permutations and canonical firing sequences, that can be used to determine upper bounds for our problems of interest. The core of this framework is built around the concept of (simple structurally) f-g-canonical classes of Petri nets. Petri nets of classes that are simple structurally f-g-canonical have the property that each reachable marking can be reached by a canonical firing sequence with nice properties. The most important properties are that almost all transitions are contained in short loops (i. e., self-covering sequences), and that the backbone of the sequence which results from removing all loops from the sequence is short and fireable. The lengths of the loops and the backbone are bounded by the function f, while the number of parts of the backbone that are separated by the loops is bounded by g. For such classes, space bounds in terms of f and g for the problems named before and for the construction of semilinear set representations of reachability sets can be given. Another important part of the framework is a sufficient condition for a class to be f-f-canonical. This condition is satisfied if it can be shown that each firing sequence leading to the empty marking in an extension of the Petri net can be permuted in such a way that each marking obtained when firing the permutation is small. Based on the canonical firing sequences, the framework also provides time bounded constructions for semilinear set representations (SLSRs) for the reachability sets of the Petri nets under consideration. These SLSRs are used to provide space bounds for the equivalence and containment problems, which is a standard approach in Petri net theory.

We apply this framework to conservative Petri nets, generalized communication-free Petri nets (characterized by the property that each transition has at most one incoming edge), their inverse Petri nets, and generalized S-system Petri nets. This yields, among other things, **PSPACE**-membership for the reachability, the liveness, the boundedness, and the covering problems. By additionally showing how to simulate **PSPACE**-Turing machines with these nets, we obtain **PSPACE**-completeness. Exceptions are the equivalence and containment problems, for which we find upper bounds that are between polynomial space and doubly exponential space, depending on the concrete functions f and g. We also investigate several problems involving the concept of home spaces, for which we give upper bounds, using the semilinear set representations provided by the framework. The difficult part in applying the framework is to show that its prerequisites are satisfied. Our main tool is a permutation technique. Starting with a given firing sequence, we iteratively permute it until we obtain the canonical permutation, which is then used to argue that the class under consideration is f-g-canonical. Important recurring motives for this and similar approaches are the extraction of loops from long sequences, the decomposition of long loops into short loops, and using sufficient conditions for the property of being enabled for transition sequences and Parikh vectors.

These results emphasize the power of permutation techniques and canonical firing sequences. Even though one could argue that such techniques and sequences have implicitly been applied or obtained in many instances where complexity results for various classical problems of Petri nets have been shown, there are classes (other than those investigated in this work) for which the application of permutation techniques or canonical firing sequences is most apparent. These include (among other classes) single-path Petri nets [How+93], communication-free Petri nets [Yen97], as well as conflict-free and persistent Petri nets [HR88; LR78].

Another generalization of a well-known class consists of generalized conflict-free Petri nets, characterized by the property that the displacement of each transition is nonnegative for each place that has more than one outgoing edge. We show that the reachability problem of this class is contained in Σ_2^p, the second level of the polynomial hierarchy.

Originally, Petri nets were introduced as nets without edge multiplicities (i. e., what we call *ordinary* Petri nets today). Petri nets with edge multiplicities have been called *generalized Petri nets* or sometimes *general Petri nets*. The term *generalized* in the title of this work is meant to emphasize two things: we mainly investigate classes of Petri nets with arbitrary edge multiplicities, as well as classes of Petri nets that generalize well-known classes of (forward-)ordinary Petri nets.

In the next section, we give an outline of the thesis, including a more comprehensive (although not complete) overview of our results.

1.2 Outline

In Chapter 2, we introduce the notation, formalisms, and basic facts, and present first results, which are used in the following chapters. We formally define classical computational problems of Petri nets that are of major importance in general and in particular in the context of this work, namely the reachability, zero-reachability, liveness, boundedness, covering, equivalence, and containment problems, as well as the RecLFS problem. We state basic results and literature revolving around these problems. These problems are investigated for almost all classes of Petri nets we consider in later chapters.

In Chapter 3, we investigate several computational problems of communication-free Petri nets (cf-PNs). We first revisit the equivalence problem of cf-PNs. Yen [Yen97] proposed a construction for a semilinear set representation of the reachability set of cf-PNs which he used to propose a doubly exponential time algorithm for this problem. We address a gap in this construction and give a solution, retaining a large portion of his construction and proof, as well as preserving the size of the representation. Furthermore, we obtain a **coNEXPTIME** upper bound for this problem by applying known results of context-free commutative grammars. The following sections are dedicated to the development of very efficient polynomial time or even linear time algorithms for several variations of the boundedness and liveness problems of cf-PNs. For several more complex notions of boundedness, as well as for the covering problem, we show **NP**-completeness. In the last section, we use our results for cf-PNs to give linear time algorithms for related problems of context-free (commutative) grammars. The proofs in this chapter showcase the power of topological characterizations for properties of restricted classes of forward-ordinary Petri nets.

In Chapter 4, we develop our framework for obtaining upper bounds for our problems of interest. We define a characterization of Petri net classes in terms of canonical firing sequences leading to reachable markings. We show that for classes that can be characterized in this way, our problems of interest are decidable within certain space or time bounds, depending on the concrete parameters characterizing the class. Furthermore, we present a sufficient condition for a class to be characterizable in this way. This condition is based on the existence of certain permutations of firing sequences leading to the empty marking in the wipe-extensions of the nets of the class. For the equivalence and containment problems, we use semilinear set representations (SLSRs) for the reachability sets of Petri nets of such classes as an intermediate structure, and apply known results for SLSRs.

In Chapter 5, we apply our framework to the class of conservative Petri nets. This yields **PSPACE**-

membership for the reachability, liveness, and covering problems. For the equivalence and containment problems, we show **PSPACE**-membership without the framework. Furthermore, we show **PSPACE**-hardness for these problems, even if restricted to ordinary 1-conservative Petri nets. The lower bounds are obtained by simulating polynomial space Turing machines deciding languages by problem specific ordinary 1-conservative Petri nets. Similar approaches are used in the next chapter in much more involved forms.

In Chapter 6, we investigate several classes of Petri nets with arbitrary edge multiplicities, and related grammars: Generalized communication-free Petri nets (gcf-PNs) are characterized by a single topological constraint, namely, that each transition has at most one input place, connected by an edge with arbitrary multiplicity. Inverse gcf-PNs (igcf-PNs) are exactly those Petri nets that can be obtained from gcf-PNs by inverting the direction of all edges. Generalized S-system Petri nets (gss-PNs) are those Petri nets that are gcf-PNs and igcf-PNs at the same time. Exponent-sensitive grammars (ESGs) and exponent-sensitive commutative grammars (ESCGs) are structurally the natural counterparts of gcf-PNs. In the first section, we show how polynomial space Turing machines deciding languages can be simulated by gcf-PNs and igcf-PNs with edge multiplicities $\{1, 2\}$ and by gss-PNs with edge multiplicities $\{1, 2, 3\}$. This yields **PSPACE**-hardness for our problems of interest (with the exception of the liveness problem for which we obtain this result only for gcf-PNs). In the second section, we lay the foundation for applying our framework to gcf-PNs by using a permutation technique to obtain canonical permutations. In the third section, we apply our framework to obtain canonical firing sequences for gcf-PNs which we use to obtain canonical firing sequences also for igcf-PNs. Applying our framework in combination with these canonical firing sequences yields **PSPACE**-membership, and thus **PSPACE**-completeness, for the reachability, covering, and boundedness problems of gcf-PNs, igcf-PNs, and gss-PNs, as well as for the liveness problem of gcf-PNs. Furthermore, we find that the equivalence and containment problems are decidable for gcf-PNs in doubly exponential space, and for igcf-PNs in exponential space. The SLSRs for the reachability sets of gcf-PNs and igcf-PNs provided by our framework are then used to solve several problems for gcf-PNs and igcf-PNs involving home spaces in doubly exponential time. In the last section, we show that the uniform word problem of ESCGs is **PSPACE**-complete, and that the uniform word problem of ESGs is undecidable.

In Chapter 7, we investigate generalized conflict-free Petri nets (gcnf-PNs). We show that the RecLFS and the reachability problems are in **coNP** and $\mathbf{\Sigma}_2^p$, respectively.

In Chapter 8, we investigate ring Petri nets which consist of single directed cycles. We show that the reachability problem for ring-PNs is in **coNP**. Furthermore, we show that the RecLFS and the reachability problems are decidable in polynomial time if all edge multiplicities are powers of the same natural number.

In Chapter 9, we conclude the thesis, and briefly discuss some open problems and potential approaches based on the results of this work.

2 Preliminaries

2.1 Basic notation

Throughout this thesis, we use the following notation to avoid confusion between elements of vectors or sequences and indexed elements of a set. We use $v_{[i]}$ in the few occasions we need to refer to the i-th element of a vector, sequence, or word v. The notation v_i is reserved for indexed elements of a set (e. g., p_1, p_2, $\ldots$). $\mathbb{Z}$, $\mathbb{N}_0$, and $\mathbb{N}$ denote the set of all integers, nonnegative integers, and positive integers, respectively. $\mathbb{Q}$, $\mathbb{Q}_{\geq 0}$, $\mathbb{Q}_{>0}$, $\mathbb{Q}_{<0}$ denote the set of all rational numbers, nonnegative rational numbers, positive rational numbers, and negative rational numbers, respectively. $\mathbb{R}$ denotes the set of real numbers. The binary logarithm is denoted by ld. For $a, b \in \mathbb{Z}$, we define the intervals $[a, b] := \{a, a+1, \ldots, b\} \subsetneq \mathbb{Z}$, and $[a] := [1, a] \subsetneq \mathbb{N}$. For two vectors $u, v \in \mathbb{R}^k$, we write $u \geq v$ if $u_{[i]} \geq v_{[i]}$ for all $i \in [k]$, and we write $u > v$ if $u \geq v$ and $u_{[i]} > v_{[i]}$ for some $i \in [k]$. When k is understood, $\vec{a}$ denotes, for a number $a \in \mathbb{R}$, the k-dimensional vector with $\vec{a}_{[i]} = a$ for all $i \in [k]$. For a vector $v \in \mathbb{R}^k$, $\|v\|_1 := \sum_{i \in [n]} |v_{[i]}|$ and $\|v\|_\infty := \max\{|v_{[i]}| \mid i \in [n]\}$ denote the 1-norm and the ∞-norm of v, resp., while $\max(v) := \max\{v_{[i]} \mid i \in [n]\}$ and $\min(v) := \min\{v_{[i]} \mid i \in [n]\}$ are the largest and smallest component of v, resp. For convenience, we declare that the maximum and minimum of the empty set is 0. For a matrix $A \in \mathbb{R}^{n \times m}$, we define $\|A\|_{1,\infty} := \max\{\sum_{j \in [m]} |a_{i,j}| \mid i \in [n]\}$, where $a_{i,j}$ is the i, j-th entry of A.

For two sequences or words σ and φ, $\sigma \cdot \varphi$ denotes the concatenation of σ and φ, $\sigma_{[i..j]}$ denotes the subsequence $\sigma_{[i]} \cdot \sigma_{[i+1]} \cdots \sigma_{[j]}$, $\sigma_{[..i]} := \sigma_{[1..i]}$ the prefix of length i of σ, and $\sigma_{[i..]} := \sigma_{[i]} \cdot \sigma_{[i+1]} \cdots$ the suffix of σ starting at position i (if σ is infinitely long, then $\sigma_{[i..]}$ is also infinitely long).

If σ is finite, then $|\sigma|$ denotes the length of σ. Both the empty sequence and the empty word (in the context of grammars) are denoted by ϵ. For two sequences σ, φ, we write $\sigma \leftarrow \varphi$ if σ is assigned the value of φ, and σ has the characteristics of a variable whose value can (and will) change during an argumentation, proof, or algorithm. If we define a sequence whose value cannot change, we usually use ":=". In this sense, "$\sigma \leftarrow \ldots$" can be thought of a suggestion for the reader to be attentive w. r. t. the value of σ.

For a fixed n, $e_k \in \{0, 1\}^n$ denotes the k-th standard unit vector, i. e., the vector whose k-th component is 1 and whose other components are 0. For two functions $f, g : X \to \mathbb{R}$, we write $f(.) \overset{\mathrm{P}}{\leq} g(.)$ if there is a polynomial p such that $f(x) \leq p(g(x))$ for all $x \in X$. Note that $\overset{\mathrm{P}}{\leq}$ is transitive.

2.2 Complexity theory and the model of computation

In this section, we introduce the model of computation, and some complexity-theoretic definitions and important results. We will agree on one specific definition for each of the concepts Turing machine, configuration, and time/space complexity of Turing machines which accept languages. There exist indeed many different definitions of these concepts, which are largely equivalent but can, under very special circumstances, exhibit different properties. Our definitions are inspired by those given by Hopcroft and Ullman [HU79] and Arora and Barak [AB09], and are specifically formulated in such a way that they are most convenient in the context of this thesis.

Turing machines and time/space-bounded computation. A *nondeterministic Turing machine* (NDTM) is a 7-tuple $(Q, \Gamma, \Sigma, \delta, q_0, \square, q_{\mathrm{acc}})$, where Q is a set of states, Γ is the tape alphabet,

$\Sigma \subsetneq \Gamma$ (with $Q \cap \Sigma = \emptyset$) is the input alphabet, $\delta \subseteq (Q \setminus \{q_{\text{acc}}\} \times \Gamma) \times (Q \times \Gamma \times \{-1, 0, 1\})$ is the transition relation, $q_0 \in Q$ is the initial state, $\Box \in \Gamma \setminus \Sigma$ is the blank symbol, and q_{acc} is the unique accepting state. All its tapes are one-way infinite, it may have multiple work tapes, and may have a read-only input tape as well as a write-once write-only output tape. An NDTM is a *(deterministic) Turing machine* (TM) if δ is a partial function of $Q \setminus \{q_{\text{acc}}\} \times \Gamma$. The computational semantics is defined as usual. A *configuration* of a Turing machine consists of the contents of the already visited positions of the tapes, the respective positions of the heads, and the state of the machine.

A *(formal) language* is, for some alphabet Σ, a subset $L \subseteq \Sigma^*$ of strings consisting of symbols of Σ. The complement of a language $L \subseteq \Sigma^*$ is defined as $\overline{L} := \{x \in \Sigma^* \mid x \notin L\}$. An *oracle NDTM* with oracle access to a language $L \subseteq \Gamma^*$ is an NDTM with three special states q_{query}, q_{yes}, q_{no}, and a work tape that is designated as its oracle tape. When it enters q_{query}, it moves into the state q_{yes} if the content of the oracle tape is in L, and to q_{no} otherwise.

Through the remainder of this section, let R denote a class of functions $f : \mathbb{N}_0 \to \mathbb{R}$. A TM M computes a function $f : \Sigma^* \to \Sigma^*$ if, on input $x \in \Sigma^*$, M halts with $f(x)$ written on its first work tape (or on its designated output tape if such a tape exists). M computes f in R-time if there is a $T \in R$ such that, for each input $x \in \Sigma^*$, it halts after at most $T(|x|)$ steps. M computes f in R-space if there is an $S \in R$ such that, for each input $x \in \Sigma^*$, it visits at most $S(|x|)$ positions of each *work tape*, while it is allowed to visit any number of positions of some designated read-only input tape, and write a string of any length onto some designated write-once write-only output-tape (if these designated tapes exist).

For an input $x \in \Sigma^*$ and a Turing machine M, we write $M(x) = 1$ if there is a computation path ending in the accepting state q_{acc}, and $M(x) = 0$ otherwise. An (oracle) NDTM M *accepts* a language L if, for each input $x \in \Sigma^*$, $x \in L \Leftrightarrow M(x) = 1$. M *decides* L in R-time if there is a $T \in R$ such that each computation path of M on x has length at most $T(|x|)$. M *decides* L in R-space if there is an $S \in R$ such that each computation path of M on x visits at most $S(|x|)$ positions on each work tape. In the context of time or space bounded computations, the terms *log/logarithmic*, *polynomial*, *exponential*, and *doubly exponential* refer to the families of functions $\{c \cdot \log(n) \mid c \in \mathbb{R}, c > 0\}$, $\{p(n) \mid p \text{ is polynomial}\}$, $\{2^{p(n)} \mid p \text{ is polynomial}\}$, and $\{2^{2^{p(n)}} \mid p \text{ is polynomial}\}$, respectively. For instance, an NDTM deciding a language in time $p(n)$, where p is a polynomial, is called polynomial time NDTM.

We remark that the literature distinguishes between a Turing machine *accepting* a language and a Turing machine *deciding* a language (in R-time/R-space). In case of time- and space-constructible functions (that are at least logarithmic), this distinction is without significance when defining some well-known complexity classes. Since we exclusively use such functions, we will only use NDTMs *deciding* languages for our definitions and constructions. When giving upper bounds for the running time of an algorithm, the random access machine (RAM) with logarithmic word size is assumed to be our model of computation.

Reductions. We can now define the concept of reduction and the complexity classes of interest. A language $L_1 \subseteq \Sigma_1^*$ is R-time (R-space, resp.) many-one reducible to a language $L_2 \subseteq \Sigma_2^*$ if there is a TM M computing a function $f : \Sigma_1^* \to \Sigma_2^*$ in R-time (R-space, resp.) such that, for each $x \in \Sigma_1^*$, $x \in L_1 \Leftrightarrow f(x) \in L_2$. There exist other kinds of reductions like Turing-reductions. However, in the context of this thesis, the term *reduction* will usually be used as an abbreviation for many-one reduction.

Let C denote a class of languages. A language L is C-hard under R-time (R-space) many-one reductions if each $L' \in C$ is R-time (R-space) many-one reducible to L. A language L is complete for C under R-time (R-space) many-one reductions if $L \in C$ and L is C-hard under R-time (R-space) many-one reductions. co$C$ denotes the class of languages consisting of the respective complements of the languages of C, i. e., co$C = \{\overline{L} \mid L \in C\} = \{L \mid \overline{L} \in C\}$, where $\overline{L} = \{x \mid x \in \Sigma^*, x \notin L\}$ for the alphabet Σ of L.

Complexity classes. We now define the complexity classes that are of relevance in context of this thesis. For each of these classes C, C-hardness is defined using a specific class R_C of functions serving as the time or space bound for the reduction. For the definition of co$C$-hardness, the same class R_C of functions is used. Hence, we will only define C-hardness explicitly. **P** (**2EXPTIME**) is the class of all languages decidable by some polynomial time TM (doubly exponential time TM, resp.). **PSPACE** (**NPSPACE**) is the class of all languages decidable by some polynomial space TM (NDTM, resp.). A language is called **PSPACE**-hard if it is **PSPACE**-hard under polynomial-time many-one reductions. The well-known theorem of Savitch [Sav70] implies that **PSPACE** = **NPSPACE**. Furthermore, by Immerman [Imm88] and Szelepcsényi [Sze88], we have **PSPACE** = **coPSPACE**. In the following chapters, we will make use of these identities without explicitly referring to them. For languages $L \in$ **PSPACE**, we define a *Turing machine in standard form* deciding L.

Definition 2.1 (TM in standard form)**.** Let $L \in$ **PSPACE** be a language decided by a TM $M = (Q, \Gamma, \Sigma, \delta, q_0, \Box, q_{\text{acc}})$. Then, M is in standard form if it is a single-tape polynomial space TM, and if, on input $x \in \Sigma^*$, it exhibits the following behavior:

- at the beginning, the tape contains the word x in the first $|x|$ positions, and all other positions contain $\Box$,
- M only uses the first $\ell_S := \lceil (|x| + 2)^{c_1} \rceil$ tape positions at each step of the computation for some constant $c_1 \in \mathbb{N}$,
- M halts after at most $\ell_T := \lceil 2^{(|x|+2)^{c_2}} \rceil$ steps for some constant $c_2 \in \mathbb{N}$, and
- if M enters the state q_{acc}, then it immediately halts, all tape positions contain $\Box$, and the head is over the first tape position.

A transition $d \in \delta$ of M is called M-transition.

It's not hard to see that every language $L \in$ **PSPACE** has a TM in standard form deciding L. We will use Turing machines in standard form in Chapters 5 and 6, where we simulate such machines by Petri nets to obtain lower bounds for a number of computational problems.

NP is the class of all languages L for which a polynomial p and a polynomial time TM M exist such that, for each input $x \in \Sigma^*$, $x \in L \Leftrightarrow \exists w \in \{0,1\}^{p(|x|)} : M(x, w) = 1$. The string w is called certificate or witness for $x \in L$. Alternatively, **NP** is the class of all languages accepted by a polynomial time NDTM.

For the definition of the polynomial hierarchy, we use the recursive oracle-based approach. We first define $\mathbf{\Sigma}_1^p :=$ **NP**. For $i \in \mathbb{N} \setminus \{1\}$, $\mathbf{\Sigma}_i^p$ denotes the class of all languages decided by a polynomial time NDTM with oracle access to some language in $\mathbf{\Pi}_{i-1}^p$, where $\mathbf{\Pi}_i^p$ denotes the complement $\overline{\mathbf{\Sigma}_i^p}$ of $\mathbf{\Sigma}_i^p$. For $i \in \mathbb{N}$, a language is called $\mathbf{\Sigma}_i^p$-hard if it is $\mathbf{\Sigma}_i^p$-hard under polynomial time many-one reductions. In particular, $\mathbf{\Sigma}_2^p$ is the class of all languages decidable by a polynomial time NDTM with oracle access to a language in **coNP**. (We remark that hardness for these classes is often defined in terms of logspace-reductions instead.)

The following complexity classes are the exponential time analogues of **NP** and $\mathbf{\Sigma}_i^p$. **NEXPTIME** is the class of all languages decided by an exponential time NDTM. We define $\mathbf{\Sigma}_1^e := \mathbf{NEXPTIME}$. For $i \in \mathbb{N} \setminus \{1\}$, $\mathbf{\Sigma}_i^e$ denotes the class of all languages decided by an exponential time NDTM with oracle access to some language in $\mathbf{\Pi}_{i-1}^e$, where $\mathbf{\Pi}_i^e$ denotes the complement $\overline{\mathbf{\Sigma}_i^e}$ of $\mathbf{\Sigma}_i^e$. For $i \in \mathbb{N}$, a language is called $\mathbf{\Sigma}_i^e$-hard if it is $\mathbf{\Sigma}_i^e$-hard under polynomial time many-one reductions.

Decision problems. The term *decision problem* or simply *problem* is used as a synonym of *formal language*. We often define a decision problem intensionally as a set of strings satisfying a certain property. Such a definition is usually given in form of a question with the possible answers "yes" or "no", asking if a string $x \in \Sigma^*$ satisfies the property. Then, the decision problem consists of all strings for which the answer is "yes".

Such questions can also be asked for mathematical objects that are not strings, and therefore defining sets of object for which the answer is "yes". For example, we could ask: "Given an object $X \in \mathcal{X}$, does X have property Y?" In such a case, we assume a fixed encoding of the objects of $\mathcal{X}$ as strings of $\{0,1\}^*$. Hence, the question above translates to "Given a string $x \in \{0,1\}^*$, does x encode some object of $X \in \mathcal{X}$ (under our fixed encoding) and does X have property Y?" Let XY denote this decision problem.

Formally, the complement $\overline{\text{XY}}$ of XY consists of all strings $x \in \{0,1\}^*$ that don't encode an object $X \in \mathcal{X}$ (under our fixed encoding) or whose encoded object X does not have property Y. However, usually one would call the following decision problem the complement of XY: "Given an object $X \in \mathcal{X}$, does X not have property Y?" This language consists of all strings $x \in \{0,1\}^*$ such that x encodes an object $X \in \mathcal{X}$ (under our fixed encoding) and X does not have property Y. Obviously, this problem is not the same as $\overline{\text{XY}}$. This difference is without significance in the context of this work since the following assumptions are satisfied:

- all complexity classes used in this work are defined in terms of NDTMs which are allowed to run for at least polynomial time, and
- for all problems considered in this work, it can be checked in polynomial time if the input encodes one of the corresponding objects of interest.

Therefore, we can always add a test with polynomial running time, which checks if the input encodes an object of the class of interest, to a given Turing machine deciding a problem of one of these complexity classes, without leaving the complexity class.

Promise problems. The concept of promise problems was introduced by Even et al. [Eve+84], and can be interpreted as a generalization of the concept of decision problems. We use a slight variation of the definition given by Goldreich [Gol06]. A promise problem is a pair $(\Pi_{\text{yes}}, \Pi_{\text{no}})$ such that $\Pi_{\text{yes}}, \Pi_{\text{no}} \subseteq \Sigma^*$ with $\Pi_{\text{yes}} \cap \Pi_{\text{no}} = \emptyset$ for some alphabet Σ. $\Pi_{\text{yes}} \cup \Pi_{\text{no}}$ is called the promise.

The concepts introduced for decision problems (like language recognition, acceptance, and decision, reductions, complexity classes, etc.) can naturally be generalized for promise problems. For instance, the class P is the set of all promise problems $(\Pi_{\text{yes}}, \Pi_{\text{no}})$ for which there is a polynomial time Turing machine M such that $M(x) = 1$ for each input $x \in \Pi_{\text{yes}}$, and $M(x) = 0$ for each input $x \in \Pi_{\text{no}}$. Note that there are no restrictions on the behavior of M for inputs that are not in the promise.

If the promise equals Σ^*, then the promise is called trivial promise, and the promise problem can be identified with the decision problem Π_{yes} with respect to the concepts named above. In the following, we use the term promise problem only for promise problems with nontrivial promise, and the term decision problem for promise problems with trivial promise.

Promise problems are usually formulated in the same way as decision problems, but the interpretation of this formulation is different. Consider again the following defining question: “Given an object $X \in \mathcal{X}$, does X have property Y?” Again, we assume a fixed encoding of the objects of $\mathcal{X}$ as strings of $\{0,1\}^*$. However, if this question is supposed to define a promise problem, then it translates to the following computational problem: “Given a string $x \in \{0,1\}^*$ such that x encodes some object of $X \in \mathcal{X}$ (under our fixed encoding), does X have property Y?” In contrast to a decision problem, we can assume that the input encodes an object of $\mathcal{X}$ (i. e., satisfies the nontrivial promise).

In many cases, it doesn't make a difference whether an intensional definition of a problem is supposed to define a promise problem or a decision problem. Indeed, if, for instance, it can be checked in polynomial time whether the input $x \in \Sigma^*$ for a promise problem $(\Pi_{\text{yes}}, \Pi_{\text{no}}) \in \mathsf{P}$ is in the promise, then $\Pi_{\text{yes}} \in \mathbf{P}$. In some cases, the correct interpretation of an intensional definition is crucial. For instance, the promise problem defined by the question “Given a bounded Petri net $\mathcal{P}$, is $\mathcal{P}$ connected?” is decidable in logarithmic space, while the decision problem defined by the same question requires at least exponential space in the number of places and transitions [Lip76].

In this work, we will occasionally refer to promise problems, namely, when citing certain known results which are relevant in the context of this work, or when pointing out differences regarding the computational complexity of certain promise problems compared to that of their decision problem counterpart. In such cases, we will explicitly state that we are considering a promise problem. Usually, however, each intensional definition is supposed to define a decision problem. For a comprehensive introduction to promise problems, we refer to [Gol06].

Encoding schemes and hardness in the strong sense. We define a succinct encoding scheme for mathematical objects and problem instances. Every number is encoded in binary representation. A vector of $\mathbb{Q}^k$ is encoded as a k-tuple of numbers. If we regard a tuple as an input, then it is encoded as a tuple of the encodings of the particular components. The encoding size of an object x under this encoding scheme is denoted by $\text{size}(x)$. The input size of a problem instance consists of the total size of the encodings of all entities that are declared as being “given” in the respective problem statement. For (weighted) graphs, we assume a representation as an enumeration of nodes and edges together with their respective weights. Throughout this work, we usually use this succinct encoding scheme.

We remark that complexity bounds for intensionally defined problems depend on the encoding scheme. (More precisely, different encoding schemes formally imply different underlying languages.) Since, for many intensionally defined problems, the encoding scheme differs between different publications, care must be taken when using known complexity results. Our encoding scheme is equivalent to the usual standard binary encoding schemes with respect to the complexity classes defined above. The concept of hardness in the strong sense addresses an aspect between standard binary encoding schemes and unary encoding schemes. An intensionally defined decision problem is **NP**-hard (**PSPACE**-hard) in the strong sense if it is still **NP**-hard (**PSPACE**-hard, resp.) under a unary encoding scheme for all numbers encoded in the inputs.

2.3 Petri nets

Historically, Petri nets were invented 1962 by Petri [Pet62] in his PhD thesis. Since then, the concept of Petri nets underwent a sizable evolution such that the modern concept of Petri nets vastly differs from its original definition. Furthermore, even nowadays, the terminology for Petri nets is not unified. Many concepts involving Petri nets are referred to by respectively many different terms, while, on top of that, different authors use the same term for different concepts. We will use the modern terminology, and will choose those definitions and presentations that are most suitable in the context of this thesis.

The model. A Petri net N is a 3-tuple (P, T, F) where P is a finite set of n *places*, T is a finite set of m *transitions* with $P \cap T = \emptyset$, and $F : (P \times T) \cup (T \times P) \to \mathbb{N}_0$ is a *flow function*. Throughout this work, n and m will always refer to the number of places resp. transitions of the Petri net under consideration, and $W = \max\{F(p,t),\ F(t,p) \mid p \in P, t \in T\}$ to the largest value of its flow function. Usually, we assume an arbitrary but fixed order on P and T, respectively. With respect to this ordering of P, we can consider an n-dimensional vector v as a function of P, and, abusing notation, write $v(p)$ for the entry of v corresponding to place p. Analogously, we write $v(t)$ in the context of an m-dimensional vector and a transition t.

A *marking* μ of a Petri net is an assignment of a nonnegative number of *tokens* to each of its places, and is represented by a vector of $\mathbb{N}_0^n$. A pair (N, μ_0) such that μ_0 is a marking of N is called a *marked Petri net*, where μ_0 is the *initial marking* of (N, μ_0), emphasizing the role of μ_0 as the initial state of the Petri net under a transitional semantics, which is explained later. Usually, we omit the term “marked” if the presence of a certain initial marking is clear from the context since we almost exclusively consider marked Petri nets. A place p (set $S \subseteq P$ of places, resp.) is called *marked* at a marking μ if $\mu(p) > 0$ ($\mu(p) > 0$ for *some* place $p \in S$, resp.), and *unmarked* or *empty* otherwise. A Petri net is encoded as an enumeration of its places and transitions followed by an enumeration of the edges with their respective edge multiplicities.

The Petri net graph of a Petri net $\mathcal{P} = (P, T, F, \mu_0)$ is a directed bipartite graph (P, T, A, w_A, w_P) with sets P and T of nodes consisting of the places and transitions of $\mathcal{P}$, a set $A \subseteq (P \times T) \cup (T \times P)$ of directed edges, and weight functions $w_A : A \to \mathbb{N}$ and $w_P : P \to \mathbb{N}_0$ such that

- $A = \{(x, y) \in (P \times T) \cup (T \times P) \mid F(x, y) > 0\}$,
- $w_A(x, y) = F(x, y)$ for all $(x, y) \in A$, and
- $w_P(p) = \mu_0(p)$ for all $p \in P$.

The weight of an edge is called its *multiplicity*. If a Petri net graph is drawn, places usually have the shape of circles, and transitions have the shape of bars. The $w_P(p)$ tokens of place p are drawn as dots within the circle representing p. The respective edge multiplicity is written as a label near its edge, where we omit the label if the multiplicity equals 1. Usually, we identify a Petri net with its Petri net graph. A Petri net is illustrated in (*a*) of Figure 2.1.

The transitional semantics. A sequence $\sigma \in T^*$ is called *transition sequence*. An infinite long sequence $\sigma \in T^\omega$ is called ω-*transition sequence*. A *Parikh vector* Φ, also known as firing count vector, is simply an element of $\mathbb{N}_0^m$. The *Parikh map* $\Psi : T^* \to \mathbb{N}_0^m$ maps each transition sequence σ

to its *Parikh image* $\Psi(\sigma)$ where $\Psi(\sigma)(t) = k$ for a transition t if t appears exactly k times in σ. Note that each Parikh vector Φ is the Parikh image of some transition sequence.

The *displacement* $\Delta : \mathbb{R}^m \cup T^* \to \mathbb{R}^n$ is, for $x \in \mathbb{R}^m$, defined by $\Delta(x)(p) = \sum_{t \in T} x(t) \cdot (F(t,p) - F(p,t))$ for all places p, and, for $\sigma \in T^*$, defined by $\Delta(\sigma) := \Delta(\Psi(\sigma))$. For a Petri net $N = (P, T, F)$ and a marking μ of N, a transition sequence $\sigma \in T^*$ can be *applied* at μ in N, producing a vector $\mu' \in \mathbb{Z}^n$ with $\mu' = \mu + \Delta(t)$. A transition t is *enabled at* μ or *enabled in* (N, μ) if $\mu(p) \geq F(p,t)$ for all $p \in P$. We say that t is *fired at* μ (*fired in* (N, μ)) if t is enabled and applied at μ (in (N, μ), resp.). If t is fired at μ, then the resulting vector μ' is a marking. Intuitively, if a transition is fired, it first removes $F(p,t)$ tokens from p and then adds $F(t,p)$ tokens to p. The displacement of t at some place p equals the change of tokens observed when applying t.

Additional notation. We write, abusing notation, $t \in \Phi$ if $\Phi(t) > 0$, and $t \in \sigma$ if $t \in \Psi(\sigma)$. We write $\mu \xrightarrow[F]{t} \mu'$ (or just $\mu \xrightarrow{t} \mu'$ if F is understood) if t is enabled at μ and produces μ' when fired at μ. For the empty transition sequence ϵ, we define $\mu \xrightarrow{\epsilon} \mu$. For a nonempty transition sequence σ, we write $\mu \xrightarrow{\sigma} \mu'$ if $\mu \xrightarrow{\sigma[1]} \mu'' \xrightarrow{\sigma[2..]} \mu'$ for some marking μ''. Similarly, we write $\mu \xrightarrow{\Phi} \mu'$ if there is a transition sequence σ with $\Psi(\sigma) = \Phi$ and $\mu \xrightarrow{\sigma} \mu'$. We also say that σ (the Parikh vector Φ, resp.)

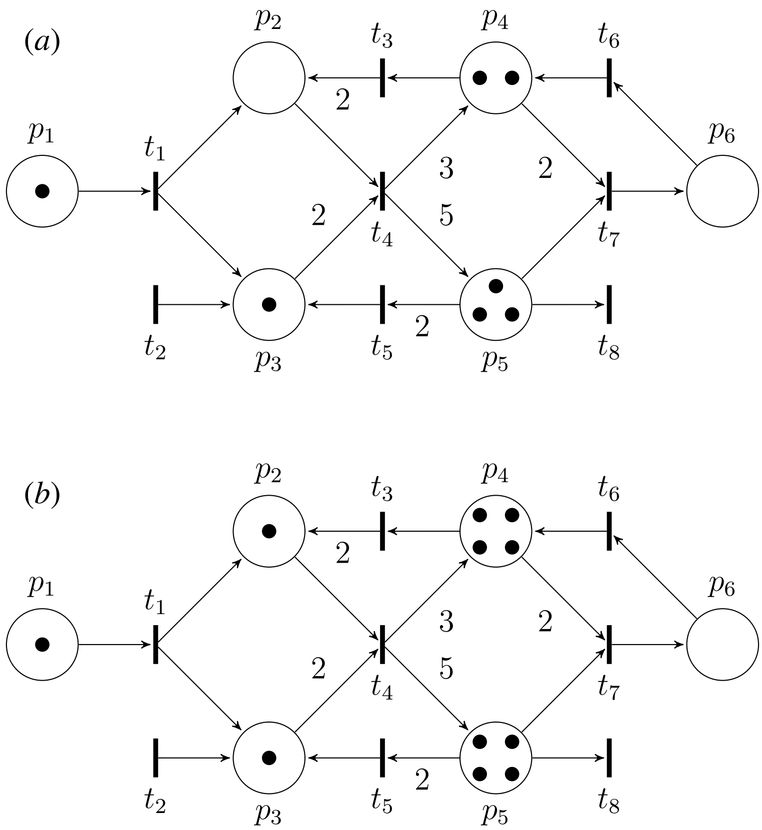


Figure 2.1: (a) illustrates a Petri net $\mathcal{P} = (N, \mu_0)$. The sequence $\sigma := t_3 t_4 t_5^2$ is not a firing sequence of $\mathcal{P}$ since $\mu(p_3) = 1 < 2 = F(p_3, t_4)$ for the marking μ reached by t_3. However, the permutation $\tau := t_3 t_5 t_4 t_5$ of σ is a firing sequence. It has displacement $\Delta(\tau) \triangleq p_2 + 2p_4 + p_5$ and leads to the marking illustrated in (b). Since its displacement is component wise nonnegative, τ is a loop. The Parikh vector $\Phi :\triangleq t_3 + t_4 + 2t_5$ is enabled at μ_0 since there is a transition sequence with Parikh image Φ that is enabled at μ_0 (namely τ). The ω-transition sequence $t_1 \cdot \sigma^\omega$ is an ω-firing sequence.

is enabled at μ and leads from μ to μ'. Analogously, an ω-transition sequence is enabled at μ if each finite prefix is enabled at μ. We write $\mu \xrightarrow{\sigma}$ ($\mu \xrightarrow{\Phi}$, resp.) if $\mu \xrightarrow{\sigma} \mu'$ ($\mu \xrightarrow{\Phi} \mu'$, resp.) and we are not particularly interested in the marking μ', or if σ is an ω-transition sequence enabled at μ. A transition sequence or Parikh vector is enabled in (N, μ_0) if it is enabled at μ_0. Such a transition sequence is also called *firing sequence*. Analogously, an ω-transition sequence enabled at μ_0 is called *ω-firing sequence*.

A marking μ is called *reachable* if $\mu_0 \xrightarrow{\sigma} \mu$ for some σ. The *reachability set* $\mathcal{R}(N, \mu_0)$ of (N, μ_0) consists of all reachable markings. For a Petri net $\mathcal{P} = (P, T, F, \mu_0)$, let σ be a transition sequence, μ a marking, and $S \subseteq P$ a subset of places. The *markings obtained when applying/firing* σ refer to the set $\{\mu \mid \exists i \in [0, |\sigma|] : \mu_0 + \Delta(\sigma_{[..i]}) = \mu\}$. We write $\max(\mu, \sigma, S) := \max\{(\mu + \Delta(\sigma_{[..i]}))(p) \mid i \in [0, |\sigma|], p \in S\}$ for the maximum number of tokens observed at any place $p \in S$ of any marking obtained when applying σ. We define $\max(\mu, \sigma) := \max(\mu, \sigma, P)$, and $\max(\mu, S) := \max(\mu, \epsilon, S)$. By substituting each occurrence of $\max$ by $\min$ in these definitions, we obtain analogous definitions for the minimal number of tokens.

We define the *displacement matrix* $D \in \mathbb{Z}^{n \times m}$ of a Petri net (with a fixed ordering of its places and transitions) as follows: the i-th column of D equals the displacement of the i-th transition. We remark that the transpose D^T is also known as incidence matrix of the Petri net, emphasizing that a Petri net is a bipartite graph.

A Parikh vector or a transition sequence with nonnegative displacement at all places is called *loop* (also known as *self-covering sequence*) since, if it is fired at some marking, the loop can immediately be fired again at the resulting marking. A loop with positive displacement at some place p is a *positive loop* (for p). A Parikh vector or transition sequence with nonpositive displacement at all places is a *nonpositive loop*. A nonpositive loop with negative displacement at some place p is a *negative loop* (for p). A loop that is not a positive loop is called *zero-loop*. A *T-invariant* is an integral vector x (sometimes also defined as nonnegative integral vector) satisfying the equality $Dx = 0$ [Mur89]. T-invariants are a useful tool in Petri net analysis. Note that zero-loops are T-invariants. For a node $x \in P \cup T$, ${}^\bullet x := \{y \mid F(y, x) > 0\}$ ($x^\bullet := \{y \mid F(x, y) > 0\}$) is the *preset* (*postset*, resp.) of x. For a transition sequence σ, we define ${}^\bullet\sigma := \bigcup_{i \in [|\sigma|]} {}^\bullet\sigma_{[i]}$. For transition sequences σ and φ, the sequence that is obtained from σ by deleting the first $\min\{\Psi(\sigma)(t), \Psi(\varphi)(t)\}$ occurrences of every transition t, is denoted by $\sigma \mathbin{\dot{-}} \varphi$. Figure 2.1 illustrates the transitional semantics and some of the terms of this paragraph, using the special notation presented in the following paragraph.

Specifying Petri nets, markings, displacements, and Parikh vectors. Specifying a Petri net $\mathcal{P} = (P, T, F, \mu_0)$ by declaring, for each place $p \in P$ and transition $t \in T$, the value of $F(p, t)$, $F(t, p)$, and $\mu_0(p)$ can be tedious. Instead, we use in several instances the following approach based on [Jon+77]. A vector $v \in \mathbb{Z}^n$ of $\mathcal{P}$ (which can be a marking of $\mathcal{P}$ or the displacement of a Parikh vector or firing sequence) is identified with $\sum_{p \in P} a_p p$ where $v(p) = a_p$ for all $p \in P$. If we want to define v in terms of its associated sum, we write $v :\triangleq \sum_{p \in P} a_p p$. Furthermore, we write $v \triangleq \sum_{p \in P} a_p p$ if v equals a marking associated by the sum on the right-hand side. Similarly, we identify a Parikh vector Φ with $\sum_{t \in T} a_t t$ where $\Phi(t) = a_t$ for all $t \in P$. In order to specify transitions, we write $\sum_{p \in P} a_p p \overset{t}{\underset{F}{\mapsto}} \sum_{p \in P} b_p p$ where $a_p = F(p, t)$ and $b_p = F(t, p)$ for all $p \in P$. (Instead of a sum on the left or right-hand side, we occasionally use its associated marking.) Sometimes, it's more convenient to describe a marking less formally by naming those places that are marked together with their respective number of tokens. In such cases, all places whose token numbers are not in some way specified are empty. We say that a place p contains k tokens at marking μ if $\mu(p) = k$.

Topological and graph-theoretic properties and other concepts. We define some terms that are mostly used in defining certain classes of Petri nets. Let $\mathcal{P} = (P, T, F, \mu_0)$ be a Petri net. $\mathcal{P}$ is *forward-ordinary* (*backward-ordinary*, resp.) if all edges from places to transitions (from transitions to places, resp.) have multiplicity 1. $\mathcal{P}$ is *ordinary* if it is both forward-ordinary and backward-ordinary. We borrow the following four terms from Teruel et al. [Ter+92]. A place p is a *decision* (*attribution*, resp.) if it has at least two outgoing (incoming, resp.) edges. A transition t is a *fork* (*join*, resp.) if it has at least two outgoing (incoming, resp.) edges.

We remark that, even though these terms are less well known, we use them to reduce confusion. For instance, the term *forward-conflict* (as in [Lie76b]) could possibly be used instead of *decision*. However, *conflict-free Petri nets* (see Chapter 7) are not necessarily forward-conflict-free. Furthermore, the class of forward-conflict-free Petri nets is, in addition to decision-freeness, subject to the constraint that each place has exactly one outgoing edge, and would therefore not be completely characterized by the term forward-conflict-free used in this manner. Unfortunately, also the term *decision* can be the source of confusion since Petri nets without decisions are usually called *choice-free Petri nets* [Ter+97].

We now define some graph-theoretic concepts that are of major importance for the analysis of Petri nets in general and of those investigated in this thesis in particular. A *walk* is a sequence $(v_1, \ldots, v_\ell)$ of nodes such that, for all $i \in [\ell - 1]$, (v_i, v_{i+1}) is an edge of the Petri net. A *path* is a walk whose nodes are pairwise different. A *cycle* is a walk whose first and last node are the same. A *circuit* is a cycle $(v_1, \ldots, v_\ell = v_1)$ such that $v_i \neq v_j$ for all $i, j \in [\ell]$ with $\{i, j\} \neq \{1, \ell\}$.

Sometimes it is convenient to only consider those places that are relevant w. r. t. a given set of transitions. For a Petri net $\mathcal{P} = (P, T, F, \mu_0)$, and a subset $T' \subseteq T$ of transitions, the induced Petri net $\mathcal{P}[T']$ consists of all transitions $t \in T'$, all places $p \in \bigcup_{t \in T'} ({}^\bullet t \cup t^\bullet)$, and the flow function F and initial marking μ_0 restricted to these subsets of transitions and places. Furthermore, we define the induced Petri net $\mathcal{P}[\Phi] := \mathcal{P}[\{t \in \Phi\}]$.

On several occasions, we need the so called *wipe-extension* $\widehat{\mathcal{P}} = (P, \widehat{T}, \widehat{F}, \mu_0)$ of a Petri net $\mathcal{P} = (P, T, F, \mu_0)$. $\widehat{T}$ is a superset of T and additionally contains, for each place $p \in P$, a transition $\widehat{t}_p \in \widehat{T} \setminus T$ defined by $p \overset{\widehat{t}_p}{\mapsto} 0$. In other words, the additional transitions can remove tokens from the wipe-extension. We will use the notation $\widehat{\mathcal{P}}$, $\widehat{T}$, $\widehat{F}$, $\widehat{t}_p$ exclusively for the wipe-extension of a Petri net $\mathcal{P}$ with transitions T and flow function F.

Two concepts of Petri nets that are investigated on several occasions are *home states* and *home spaces*. A set $\mathcal{HS}$ is a home space of $\mathcal{P}$ if, for each marking μ reachable in $\mathcal{P}$, there is a marking $\mu' \in \mathcal{HS}$ such that μ' is reachable from μ. A marking μ is a home state if $\{\mu\}$ is a home space. When modeling a system, home states and (non-trivial) home spaces represent states of the system that can be reached from every other state of the system (for instance the main menu).

Extensions of the Petri net model. The Petri net model has been extended in many different directions. Common extensions are states, inhibitor edges, priorities, timing constraints and token colors. In this thesis, we don't consider extended Petri net models.

2.4 Vector addition and replacement systems

Vector addition systems are a tool for the analysis of models of parallel computation introduced by Karp and Miller [KM69]. Formally, a vector addition system (VAS) is a pair $V = (s, B)$ with $B = \{b_1, \ldots, b_m\}$, where $s \in \mathbb{N}_0^n$ and $b_i \in \mathbb{Z}^n$ for all $i \in [m]$. The vector s is the start vector of V, while the vectors $b \in B$ are the displacement or basis vectors of V. A vector $x \in \mathbb{N}_0^n$ is reachable in V if there are displacement vectors $b_{i_1}, \ldots, b_{i_k} \in B$ such that

- $x = s + \sum_{j=1}^{k} b_{i_j}$, and
- $s + \sum_{j=1}^{r} b_{i_j} \geq \vec{0}$ for all $r \in [0, k]$.

The reachability set $\mathcal{R}(V)$ is the set of all vectors reachable in V.

Many results for Petri nets were originally shown for vector addition systems. The importance of this formalism in the context of Petri net analysis is based on the observation that VAS are equivalent to Petri nets in the following sense. A VAS $V = (s, B)$ corresponds to the self-loop-free Petri net $\mathcal{P} = (P, T, F, \mu_0)$ with n places $p_1, \ldots, p_n$ and m transitions $t_1, \ldots, t_m$, such that $\mu_0 = s$ and $\Delta(t_i) = b_i$ for all $i \in [m]$. For V and $\mathcal{P}$, we observe $\mathcal{R}(V) = \mathcal{R}(\mathcal{P})$. Furthermore, it can be shown that general Petri nets are equivalent to ordinary self-loop-free Petri nets in the following sense. For each general Petri net $\mathcal{P} = (P, T, F, \mu_0)$ there is an (ordinary) self-loop-free Petri net $\mathcal{P}' = (P', T', F', \mu_0')$ with $P \subseteq P'$ such that

$$\mathcal{R}(\mathcal{P}) = \{\mu \mid \exists \mu' \in \mathcal{R}(\mathcal{P}') : \ \mu(p) = \mu'(p) \text{ for all } p \in P \text{ and } \mu(p) = 0 \text{ for all } p \notin P'\}.$$

Moreover, $\mathcal{P}'$ can be computed in exponential time from $\mathcal{P}$. Using these relationships, many problems of general Petri nets, (ordinary) self-loop-free Petri nets, and vector addition systems are easily reducible to each other.

Later, Keller [Kel72] (and independently Lien [Lie72; Lie76a] in form of a different but equivalent formalism) introduced vector replacement systems, a formalism similar to vector addition systems and Petri nets. A vector replacement system (VRS) is a triple $V = (s, B, U)$ with $U = \{u_1, \ldots, u_m\}$ and $V = \{b_1, \ldots, b_m\}$, where $s, u_i \in \mathbb{N}_0^n$, $b_i \in \mathbb{Z}^n$, and $u_i + b_i \geq \vec{0}$ for all $i \in [m]$. Again, s is the start vector of V, and the vectors $b \in B$ are the displacement vectors. In contrast to VASs, we additionally have test vectors $u \in U$. A vector $x \in \mathbb{N}_0^n$ is reachable in V if there are displacement vectors $b_{i_1}, \ldots, b_{i_k} \in B$ such that

- $x = s + \sum_{j=1}^{k} b_{i_j}$, and
- $s + \sum_{j=1}^{r-1} b_{i_j} \geq u_{i_r}$ for all $r \in [k]$.

The reachability set $\mathcal{R}(V)$ is the set of all vectors reachable in V. It's not hard to see that VRSs are equivalent to general Petri nets with respect to their reachability sets in a more direct way than VASs. A VRS $V = (s, B, U)$ corresponds to the Petri net $\mathcal{P} = (P, T, F, \mu_0)$ with n places $p_1, \ldots, p_n$ and m transitions $t_1, \ldots, t_m$, such that $\mu_0 = s$ is the initial marking, and $\Delta(t_i) = b_i$ as well as $F(p_j, t_i) = (u_i)_{[j]}$ for all $j \in [n]$ and $i \in [m]$. As in the case of VASs, many results for Petri nets were originally shown using VRSs as the formalism of choice. The equivalence of general Petri nets, ordinary self-loop-free Petri nets, vector addition systems, and vector replacement systems is due to Hack [Hac74a; Hac74b]. For a compact and comprehensive overview of the different formalisms equivalent (in the sense above) to Petri nets, we refer to Peterson [Pet81].

2.5 Commutative grammars

We assume that the reader is familiar with the basic concepts of (formal) grammars, thus we only state some basic definitions. For a comprehensive introduction, we refer to Hopcroft and Ullman [HU79].

A grammar is a quadruple (V_N, V_T, P, s), where V_N is a finite set of variables (i. e., nonterminal symbols), V_T with $V_T \cap V_N = \emptyset$ is a finite set of terminal symbols, $s \in V_N$ is the start symbol (i. e., the axiom), and $P \subset V_N^+ \times (V_N \cup V_T)^*$ is a finite set of productions. For two words $w_1, w_2 \in (V_N \cup V_T)^*$, we write $w_1 \Rightarrow w_2$ if there is a production $(a, b) \in P$ such that $w_1 = xay$ and $w_2 = xby$ for two words $x, y \in (V_N \cup V_T)^*$. The language $L(G)$ of a grammar G is $\{w \in V_T^* \mid s \stackrel{*}{\Rightarrow} w\}$, where $\stackrel{*}{\Rightarrow}$ is the transitive closure of $\Rightarrow$. Of particular interest in the context of this thesis is the class of context-free grammars (CFGs). A grammar is context-free if all productions are of the form $A \rightarrow u$ where $A \in V_N$ and $u \in (V_N \cup V_T)^*$.

Definition 2.2 (Uniform word problem of a class $\mathcal{C}$ of grammars)**.** Given a grammar $G \in \mathcal{C}$, and a word w, is $w \in L(G)$?

In the following, we will occasionally refer to a grammar by the term non-commutative grammar to avoid confusion with commutative grammars.

Commutative grammars were introduced by Crespi-Reghizzi and Mandrioli [CRM76] as a formalism which is, in many aspects, equivalent to Petri nets. Intuitively, a commutative grammar is obtained by enriching a grammar with productions that can switch the order of any two adjacent symbols. In this sense, the order of the symbols within a string is without significance, and only the respective number of occurrences of each symbol is of relevance.

For a formal definition, we use that given by Huynh [Huy83]. A commutative grammar is a quadruple (V_N, V_T, P, s), where V_N is a finite set of variables (i. e., nonterminal symbols), V_T with $V_T \cap V_N = \emptyset$ is a finite set of terminal symbols, $s \in V_N$ is the start symbol (i. e., the axiom), and $P \subset V_N^{\oplus} \times (V_N \cup V_T)^{\circledast}$ is a finite set of productions. Here, $M^{\circledast}$ and $M^{\oplus}$ denote the free commutative monoid and the free commutative semigroup on a set M, respectively. Some classes of commutative grammars, like regular, context-free, context-sensitive, and type 0 commutative grammars (corresponding to well-known classes of non-commutative grammars) were investigated by Huynh [Huy83; Huy84; Huy85]. Another type of commutative grammars, namely commutative semi-groups, were investigated by Mayr and Meyer [MM82]. The class of context-free commutative grammars (CFCGs) is the commutative equivalent of the class of context-free grammars. A commutative grammar is context-free if all productions are of the form $A \rightarrow u$ where $A \in V_N$ and $u \in (V_N \cup V_T)^{\circledast}$. We make use of known results for this class in Section 3.2 when we discuss the equivalence problem of cf-PNs. In Section 6.4, we present a new class of commutative grammars, namely exponent-sensitive commutative grammars (ESCGs), corresponding to generalized communication-free Petri nets, and show results for computational problems involving these grammars. A commutative grammar is exponent-sensitive if $P \subset \{\{x\}^{\oplus} \mid x \in V_N\} \times (V_N \cup V_T)^{\circledast}$. The class of ESCGs properly contains the class of CFCGs.

For $V_N = \{v_1, \ldots, v_p\}$ and $V_T = \{u_1, \ldots, u_q\}$, a commutative word $w \in (V \cup T)^{\circledast}$ is usually written in the form $v_1^{i_1} \cdots v_p^{i_p} u_1^{j_1} \cdots u_q^{j_q}$, where $i_k, j_\ell \in \mathbb{N}_0$ are the number of times v_k and u_ℓ occur in w. As always, a succinct encoding scheme is assumed for w. By fixing an order on the symbols, we can identify commutative words with vectors of $\mathbb{N}_0^{p+q}$, and express productions in terms of such vectors. For two words $w_1, w_2 \in \mathbb{N}_0^{p+q}$, we write $w_1 \Rightarrow w_2$ if there is a production $(a, b) \in \mathbb{N}_0^{2(p+q)}$ with $w_1 - a \geq \vec{0}$ and $w_1 - a + b = w_2$. The language $L(G)$ of a commutative grammar G is $\{w \in V_T^{\circledast} \mid s \stackrel{*}{\Rightarrow} w\}$, where $\stackrel{*}{\Rightarrow}$ is the transitive closure of $\Rightarrow$. In the context of the fixed order of terminal symbols, $L(G)$ can be recognized as a subset of $\mathbb{N}_0^q$. In contrast to non-commutative grammars, this allows for succinct encoding schemes of commutative grammars and words, where we represent a commutative word w by its corresponding vector, and encode this vector, e. g., using a binary alphabet. For known complexity results of problems involving commutative grammars, and new complexity results presented in this thesis, always such succinct encoding schemes are assumed.

We remark that, originally, Crespi-Reghizzi and Mandrioli [CRM76] defined commutative grammars as triples (V_N, V_T, P), i. e., the starting symbol was not part of the grammar. As input instances for a variety of decision problems, pairs $((V_N, V_T, P), s)$ were considered where $s \in (V_N \cup V_T)^{\oplus}$. In other words, under the original definition, s is not a start symbol rather than an initial commutative word, which emphasizes the close relationship between commutative grammars and Petri nets, where the initial word corresponds to the initial marking.

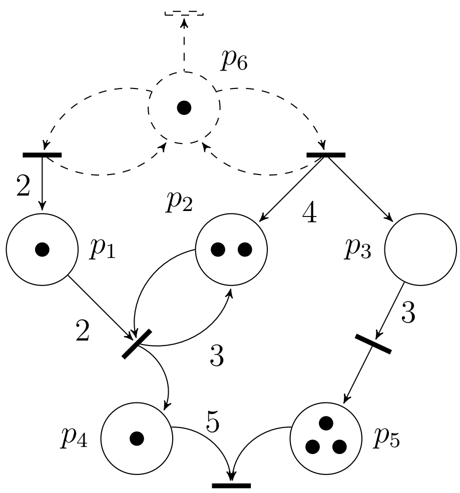
In the context of ESCGs and our applications, the two definitions can be regarded as equivalent since ESCGs can have a production producing an "initial" word from the initial symbol. (This is not the case for every nontrivial class of commutative grammars).

Given a PN $\mathcal{P} = (S, T, F, \mu_0)$, where w. l. o. g. $S = \{p_1, \ldots, p_n\}$ and $T = \{t_1, \ldots, t_m\}$, one can define a commutative grammar $G = (V_N, V_T, P, s)$ such that $\mathcal{R}(\mathcal{P}) = L(G)$. Such a construction can be used as part of a reduction from a problem involving Petri nets to a related problem involving commutative grammars. The main problem lies in the fact that transitions of Petri nets don't necessarily have incoming edges while productions of commutative grammars always must substitute at least one symbol. Unfortunately, this problem was not addressed in [CRM76] or [Huy83]. In the following we fill this gap.

We first define a new Petri net $\mathcal{P}' = (S', T, F', \mu'_0)$ which is obtained from $\mathcal{P}$ in the following way. We add a new place p_{n+1} with $\mu'_0(p_{n+1})$ to the net and set $F'(p_{n+1}, t) = F'(t, p_{n+1}) = 1$ for each $t \in T$ with ${}^\bullet t = \emptyset$. Furthermore, we add a transition that removes one token from p_{n+1}. The place p_{n+1} represents the "void" where tokens are created and vanish. Now, we can define the commutative grammar G of interest. The symbols are $V_N := \{v_i \mid p_i \in S'\} \cup \{s\}$ and $V_T := \{u_i \mid p_i \in S\}$. We associate each place p_i with the variable v_i, $i \in [n+1]$, and each variable v_i with the terminal symbol u_i, $i \in [n]$. For each transition $t \in T'$ there is a production that, for each $i \in [n+1]$, replaces $F'(p_i, t)$ occurrences of variable v_i with $F'(t, p_i)$ occurrences of the same variable. In addition, there is, for each $i \in [n]$, a production that replaces v_i by u_i. Last but not least, we have a production that replaces the initial symbol s with the word that consists of variables only and corresponds to the initial marking μ' of $\mathcal{P}'$. (When using the definition of Crespi-Reghizzi and Mandrioli [CRM76], this production can be replaced by fixing the initial word s appropriately.) This construction is illustrated in Figure 2.2.

Note that $\mathcal{R}(\mathcal{P})$ equals the projection of $\mathcal{R}(\mathcal{P}')$ onto the original places P. This projection, on the other hand, equals $L(G)$. Using this relationship of $\mathcal{R}(\mathcal{P})$ and $L(G)$, we can, in many cases, use results for problems involving certain classes of Petri nets to reason about related problems of corresponding classes of commutative grammars, and vice versa. We call this commutative grammar the canonical commutative grammar for the Petri net.

We also want to define a canonical Petri net for a commutative grammar. For each variable v_i and terminal symbol u_j, there are corresponding places p_i and q_j. We write S_{V_N} (S_{V_T}) for the set of places corresponding to nonterminal (terminal, resp.) symbols. A production replacing a commutative word with k_i occurrences of variable v_i, $i \in [|V_N|]$, by a commutative word with ℓ_i occurrences of variable v_i and ℓ'_j occurrences of the terminal symbol u_j, $j \in [|V_T|]$, is represented by a transition t with $\sum_{i\in[|V_N|]} k_i p_i \overset{t}{\mapsto} \sum_{i\in[|V_N|]} \ell_i p_i + \sum_{i\in[|V_T|]} \ell'_i q_i$. The initial marking μ_0 is the marking corresponding to s. We use an analogue construction for canonical Petri nets of non-commutative grammars. Note that the reachable markings of the canonical Petri net correspond to all commutative words produced



$$V_N = \{v_i \mid i \in [6]\} \cup \{s\}$$
$$V_T = \{u_i \mid i \in [5]\}$$

$$\begin{aligned}
\forall i \in [5] : v_i &\to u_i \\
s &\to v_1 v_2^2 v_4 v_5^3 v_6 \\
v_6 &\to \epsilon \\
v_6 &\to v_1^2 v_6 \\
v_6 &\to v_2^4 v_3 v_6 \\
v_1^2 v_2 &\to v_2^3 v_4 \\
v_3^3 &\to v_5 \\
v_4^5 v_5 &\to \epsilon
\end{aligned}$$

Figure 2.2: A Petri net $\mathcal{P}$ (non-dashed places, transitions and edges), its extension $\mathcal{P}'$ ($\mathcal{P}$ together with the dashed place p_6, transition and edges), and the canonical context-free commutative grammar of $\mathcal{P}$.

by the grammar, and not only those consisting only of terminal symbols.

We remark that, while a suitable construction of canonical commutative grammars from Petri nets is not hard to find, it is not obvious how to define a Petri net $\mathcal{P}$ from a commutative grammar G such that $\mathcal{R}(\mathcal{P}) = L(G)$ under a suitable association of places with a subset of the terminal symbols and variables. Commutative grammars carry out computations using variables. Furthermore, the very definition of $L(G)$ promises that none of its words contains variables. In contrast, Petri nets carry out computations using all their places. However, markings of $\mathcal{R}(\mathcal{P})$ don't distinguish between places that contain the result of a computation and places that were used for the computation itself.

For some computational problems, like the uniform word problem, this is not an issue at all.

Definition 2.3 (Uniform word problem of a class $\mathcal{C}$ of commutative grammars)**.** Given a commutative grammar $G \in \mathcal{C}$, and a commutative word w, is $w \in L(G)$?

This problem can easily be reduced to the reachability problem of Petri nets, which asks, given a PN $\mathcal{P}$ and a marking μ, if $\mu \in \mathcal{R}(\mathcal{P})$. (For CFCGs and cf-PNs, this was done by Esparza [Esp97].) We merely ask if the marking corresponding to the input word is in $\mathcal{R}(\mathcal{P})$. This reduction works for the uniform word problem because it works for a more general problem, namely, if a given commutative word consisting of terminal symbols and variables can be produced by the grammar. In other cases, however, the fundamental intrinsic difference between $L(G)$ and $\mathcal{R}(\mathcal{P})$ can be a problem, as we will see in Section 3.2 when discussing the equivalence problem of cf-PNs.

We note that, acknowledging these differences, there are concepts that, in this sense, are more closely related to Petri nets than to commutative grammars, for instance VASs and VRSs.

2.6 Semilinear sets and their representations

For $\zeta, \pi_1, \ldots, \pi_\ell \in \mathbb{N}_0^k$, the linear set representation (LSR) $\mathcal{L}(\zeta, \{\pi_1, \ldots, \pi_\ell\})$ is the encoding of the tuple $(\zeta, \pi_1, \ldots, \pi_\ell)$, and represents the set $\{\zeta + \sum_{i\in[\ell]} a_i\pi_i \mid a_1, \ldots, a_\ell \in \mathbb{N}_0\}$. A set is linear if it is represented by some LSR. The vector ζ is the constant vector of the LSR, while the vectors π_i, $i \in [\ell]$, are its periods. $\boldsymbol{\mathcal{L}}(\zeta, \{\pi_1, \ldots, \pi_\ell\})$ ($\boldsymbol{\mathcal{L}}$, resp.) denotes the linear set represented by $\mathcal{L}(\zeta, \{\pi_1, \ldots, \pi_\ell\})$ ($\mathcal{L}$, resp.). A semilinear set representation (SLSR) $\mathcal{SL} = \bigodot_{i=1}^{k} \mathcal{L}(\zeta_i, \{\pi_{i,1}, \ldots, \pi_{i,\ell_i}\})$ is the concatenation of LSRs, and represents $\bigcup_{i\in[k]} \boldsymbol{\mathcal{L}}(\zeta_i, \{\pi_{i,1}, \ldots, \pi_{i,\ell_i}\})$. Here, $\odot$ is a concatenation operator used to concatenate two SLSRs. A set is semilinear if it is represented by some SLSR. $\boldsymbol{\mathcal{SL}}$ denotes the semilinear set represented by an SLSR $\mathcal{SL}$. We remark that each (infinite) semilinear set has an infinite number of representations.

Semilinear sets were introduced by Parikh [Par61], and shown by Ginsburg and Spanier [GS66] to be exactly the sets describable by Presburger arithmetic, a decidable theory invented by Presburger [Pre30].

Many problems of Petri nets and related models can be approached by formulating certain structures that are essential for the problem under consideration in terms of SLSs and SLSRs, and then applying known results for semilinear sets. For instance, if the reachability sets of the Petri nets of a class is semilinear and representations can effectively be constructed, then the reachability, boundedness, equivalence, containment, and other problems are decidable for the class. Furthermore, tight complexity results for problems involving SLSRs were discovered by Huynh [Huy80]. Bounds for the size of the vectors encoded in SLSRs representing such a structure can be used to reason about the size of certain vectors of interest (e. g., small witnesses for non-liveness of Petri nets as in Chapter 4). The theory of semilinear sets has been successfully applied to many classes of Petri nets and VASs, including persistent Petri nets [Gra80; May81b; Mül80], VASs up to dimension 5 [HP79], conflict-free Petri nets [HR88], normal and sinkless Petri nets [How+89]. We will use SLSRs for the problems named earlier as well as for other problems. Many results about SLSRs and their applications to different problems of Petri nets and commutative grammars, that are important in the context of this thesis, have been presented by Huynh [Huy80; Huy83; Huy84; Huy85; Huy86] in a series of papers.

Theorem 2.4 ([Huy80], Corollary 4.5; also refer to [Huy86])**.** *The equivalence problem of SLSRs, i. e., the following problem, is Π_2^P-complete: Given SLSRs $\mathcal{SL}_1$ and $\mathcal{SL}_2$, is $\boldsymbol{\mathcal{SL}_1} = \boldsymbol{\mathcal{SL}_2}$?*

The following corollary is a direct consequence of this theorem.

Corollary 2.5. *The containment problem of SLSRs, i. e., the following problem, is Π_2^P-complete: Given SLSRs $\mathcal{SL}_1$ and $\mathcal{SL}_2$, is $\boldsymbol{\mathcal{SL}_1} \subseteq \boldsymbol{\mathcal{SL}_2}$?*

Proof. The containment problem can be reduced in polynomial time to the equivalence problem as follows. We combine the SLSRs $\mathcal{SL}_1$ and $\mathcal{SL}_2$ to a SLSR $\mathcal{SL}'_1$ of the set $\boldsymbol{\mathcal{SL}_1} \cup \boldsymbol{\mathcal{SL}_2}$. Then, $\boldsymbol{\mathcal{SL}_1} \subseteq \boldsymbol{\mathcal{SL}_2}$ holds if and only if $\boldsymbol{\mathcal{SL}'_1} = \boldsymbol{\mathcal{SL}_2}$ holds.

For the other direction also a simple polynomial time reduction exists. Let k be the dimension of the vectors of two SLSRs $\mathcal{SL}'_1$ and $\mathcal{SL}'_2$, and let $\vec{0}_k$ denote the k-dimensional all-0-vector. We obtain the SLSR $\mathcal{SL}_1$ of the set $(\boldsymbol{\mathcal{SL}'_1} \times \vec{0}_k) \cup (\vec{0}_k \times \boldsymbol{\mathcal{SL}'_2})$ by combining the SLSRs resulting from extending the vectors of the SLSRs $\mathcal{SL}'_1$ and $\mathcal{SL}'_2$ at the beginning and end by $\vec{0}_k$, respectively. In the same way, we obtain the SLSR $\mathcal{SL}_2$ of the set $(\boldsymbol{\mathcal{SL}'_2} \times \vec{0}_k) \cup (\vec{0}_k \times \boldsymbol{\mathcal{SL}'_1})$. Then, $\boldsymbol{\mathcal{SL}'_1} = \boldsymbol{\mathcal{SL}'_2}$ holds if and only if $\boldsymbol{\mathcal{SL}'_1} \subseteq \boldsymbol{\mathcal{SL}'_2}$ and $\boldsymbol{\mathcal{SL}'_2} \subseteq \boldsymbol{\mathcal{SL}'_1}$ holds if and only if $\boldsymbol{\mathcal{SL}_1} \subseteq \boldsymbol{\mathcal{SL}_2}$ holds. □

We remark that, even though the theory on SLSRs is very useful, when the reachability set of a Petri net is effectively constructible as an SLSR, it cannot be applied for every Petri net in this manner since, as shown by Hopcroft and Pansiot [HP79], there are Petri nets whose reachability sets are not semilinear.

2.7 Classical computational problems and general related work

In this section, we define some classical computational problems involving Petri nets that are of major interest in the context of this thesis. We investigate these problems for different classes of (forward-)ordinary or generalized Petri nets in the following chapters. Related work involving results for classes of Petri nets that are related to those investigated in these later chapters is discussed at the beginning of the respective chapter. However, classical general results for these problems and basic related work are discussed here. Each definition involves a class $\mathcal{C}$ of Petri nets. For the related work discussed in this section, we assume $\mathcal{C}$ to be the class of all Petri nets. Sometimes, we cite results for the corresponding problem of VASs. In such cases, s denotes the encoding size of the input. We start with one of the most important problems for Petri nets.

Definition 2.6 (Reachability problem of a class $\mathcal{C}$ of Petri nets)**.** Given a Petri net $\mathcal{P} \in \mathcal{C}$ and a marking μ, is μ reachable in $\mathcal{P}$?

The importance of the reachability problems is mainly based on two reasons: A Petri net can be considered as a model of computation or a computational device. This problem asks if the device can enter a certain state. It is in line with various important decision problems involving different kinds of computational devices, like the halting problem of Turing machines. The second reason is that the reachability problem is recursively equivalent to many other Petri net problems (see [Hac74b]), some of which we will discuss later.

The reachability problem has been a problem that withstood each attempt to show its decidability for a considerable amount of time. Before Mayr [May81a; May84] finally presented an algorithm solving this problem, the difficulty of finding a solution led researchers to try gaining insight into the behavior of Petri nets by studying subclasses of Petri nets. Prominent examples include S-systems, marked graphs, conflict-free Petri nets, and persistent Petri nets, which are briefly discussed in Chapters 3 and 7. The algorithm of Mayr [May84] was later simplified by Kosaraju [Kos82] and Lambert [Lam92]. Another two algorithms were presented by Leroux [Ler09] and Leroux [Ler11]. However, even nowadays, researchers investigate subclasses for several reasons: Lipton [Lip76] showed that the reachability problem for VASs requires at least $2^{c\sqrt{s}}$ space (infinitely often) for some constant c. Furthermore, the algorithms named before are not primitive recursive or no satisfying upper bound has been found, yet. On the contrary, subclasses often allow for better complexities, and still potentially provide further insight into the behavior of Petri nets in general. The following restriction of the reachability problem has been shown by Hack [Hac74b] to be recursively equivalent to the reachability problem in general.

Definition 2.7 (Zero-reachability problem of a class $\mathcal{C}$ of Petri nets)**.** Given a Petri net $\mathcal{P} \in \mathcal{C}$, is the empty marking $\vec{0}$ reachable in $\mathcal{P}$?

The reason why we also consider the zero-reachability problem is that, depending on the class of Petri nets under consideration, its complexity can vastly differ from the complexity of the reachability problem (e. g., for communication-free Petri nets, see Chapter 3) or be largely the same (e. g., for generalized communication-free Petri nets, see Chapter 6). A problem closely related to the reachability problem is the liveness problem which involves the notion of liveness. A Petri net $\mathcal{P} = (P, T, F, \mu_0)$ is live if, for each transition $t \in T$ and each marking μ reachable in $\mathcal{P}$, there is a marking μ' that is reachable from μ and enables t.

Definition 2.8 (Liveness problem of a class $\mathcal{C}$ of Petri nets)**.** Given a Petri net $\mathcal{P} \in \mathcal{C}$, is $\mathcal{P}$ live?

Hack [Hac74b] also showed that this problem is recursively equivalent to the reachability problem. Aside from this equivalence, the liveness problem is important for another reason. When modeling a system or a device in terms of Petri nets, the liveness property corresponds to the property of the device to be able to eventually perform any action again, starting from any reachable state of the device. The liveness property is related to many other notions of liveness, some of which we consider in Chapter 3. We remark that both the zero-reachability problem and the liveness problem for VASs require at least $2^{c\sqrt{s}}$ space (infinitely often) for some constant c as well since the (Turing-)reductions given by Hack [Hac74b] can be performed in linear time.

We continue with two problems for which better upper bounds are known. A marking μ of $\mathcal{P}$ can be covered (in $\mathcal{P}$) if there is a marking $\mu' \geq \mu$ that is reachable in $\mathcal{P}$.

Definition 2.9 (Covering problem of a class $\mathcal{C}$ of Petri nets)**.** Given a Petri net $\mathcal{P} \in \mathcal{C}$ and a marking μ, can μ be covered in $\mathcal{P}$?

A Petri net $\mathcal{P}$ is bounded if there is a $k \in \mathbb{N}$ such that $\max(\mu) \leq k$ for every marking μ that is reachable in $\mathcal{P}$. Equivalently, $\mathcal{P}$ is bounded if $\mathcal{R}(\mathcal{P})$ is finite.

Definition 2.10 (Boundedness problem of a class $\mathcal{C}$ of Petri nets)**.** Given a Petri net $\mathcal{P} \in \mathcal{C}$, is $\mathcal{P}$ bounded?

Using a graph, which later became known under the term "coverability tree", Karp and Miller [KM69] showed relatively early that the covering and the boundedness problems are decidable. The algorithm proposed in the paper uses the observation that a Petri net $\mathcal{P} = (N, \mu_0)$ is unbounded if and only if $\mu_0 \xrightarrow{\sigma} \mu_1 \xrightarrow{\tau} \mu_2$ for markings μ_1, μ_2, a sequence σ and a positive loop τ. However, there is an infinite sequence of Petri nets for which the sizes of their coverability trees cannot be bounded by any primitive recursive function, implying a corresponding lower bound for this algorithm (see [Rac78]). A better algorithm, which solves the boundedness problem for VASs and requires at most $2^{c_1 s \operatorname{ld} s}$ space for some constant c_1, was discovered by Rackoff [Rac78]. He showed that if $\mathcal{P}$ is unbounded, then there are σ and τ as above whose lengths are at most doubly exponential. Hence, the encoding sizes of the markings obtained when nondeterministically guessing σ and τ are at most exponential. This is almost optimal since Lipton [Lip76] had shown earlier that, similar to the reachability problem, also the boundedness problem requires at least $2^{c_2\sqrt{s}}$ space (infinitely often) for some constant c_2. Later, Rosier and Yen [RY86] obtained the slightly improved and refined upper bound $2^{c_3 n \operatorname{ld} n}(d + \operatorname{ld} m)$ for some constant c_3 via a multiparameter analysis, where d denotes the maximum over all out- and indegrees of all transitions.

Definition 2.11 (Containment problem of a class $\mathcal{C}$ of Petri nets)**.** Given two Petri nets $\mathcal{P}_1, \mathcal{P}_2 \in \mathcal{C}$, is $\mathcal{R}(\mathcal{P}_1) \subseteq \mathcal{R}(\mathcal{P}_2)$?

Definition 2.12 (Equivalence problem of a class $\mathcal{C}$ of Petri nets)**.** Given two Petri nets $\mathcal{P}_1, \mathcal{P}_2 \in \mathcal{C}$, is $\mathcal{R}(\mathcal{P}_1) = \mathcal{R}(\mathcal{P}_2)$?

By reducing the undecidable Hilbert's tenth problem to the containment problem of VASs, Rabin showed that the latter is undecidable. Rabin's proof was published by Baker [Bak73]. Hack [Hac73; Hac74a] adapted this proof for Petri nets to show the undecidability of the containment problem of general Petri nets. Later, Hack [Hac76] showed that also the equivalence problem of general Petri nets is undecidable via a reduction from the containment problem. Restricted variations of these problems were investigated by Mayr and Meyer [MM81]. They showed that, albeit the containment and equivalence problems of bounded Petri nets are decidable, neither of them can be solved by an algorithm with primitive recursive complexity. Some comprehensive resources surveying classical results for these problems are [EN94; Hac79; Pet81].

Definition 2.13 (RecLFS problem of a class $\mathcal{C}$ of Petri nets)**.** Given a Petri net $\mathcal{P} \in \mathcal{C}$ and a Parikh vector Φ, is Φ enabled in $\mathcal{P}$?

The term RecLFS is an abbreviation for "Recognize legal firing sequence" and is borrowed from Taoka et al. [Tao+03]. The RecLFS problem often occurs as a "subproblem" of the reachability problem. Algorithms solving the reachability problem for classes of Petri nets can often be partitioned into two steps. First, a candidate Parikh vector is determined for which one wants to know if it indeed leads to the end marking of interest. Then, an algorithm solving the RecLFS problem for the Parikh vector is applied.

For many subclasses of Petri nets for which witnessing Parikh vectors have small encoding size, and the RecLFS problem has low complexity, this approach is particularly useful. Examples for such subclasses are communication-free Petri nets and conflict-free Petri nets for which witnessing Parikh vectors have polynomial encoding size and the RecLFS problem is decidable in linear time (for references, see Chapters 3 and 7).

We remark that the RecLFS problem is somewhat excluded from the problems mentioned earlier for two reasons: In literature investigating such subclasses, the RecLFS problem is often not explicitly mentioned but only implicitly investigated and used. Furthermore, the RecLFS problem is in **PSPACE** for general Petri nets (see Lemma 2.19), which means that its complexity is much better than that of the other classical problems of our interest.

This comparatively low complexity makes the approach mentioned earlier also particularly useful for classes for which no improved bound on the RecLFS problem can be shown if it can be shown that the respective reachability problem needs at least polynomial space and the class allows for witnessing Parikh vectors whose encoding sizes are at most polynomial in the size of the input. The search problem variation of the RecLFS problem is the LFS problem, which is, given a Petri net and a Parikh vector Φ, to generate a firing sequence with Parikh image Φ or to state that Φ is not enabled if this is the case. Papers in which the RecLFS or the LFS problem for certain classes of Petri nets are explicitly investigated include [HT01; MW96; TW99; TW06; Tao+03; Wat+89a; Wat+89b].

These are the main problems that are investigated in (almost) every of the following chapters. Besides them, we also investigate other problems for Petri nets or for related formalisms. The definitions and relevant related literature for those can be found in the respective chapter where they are investigated.

2.8 Fundamental facts, observations, and first results

In this section, we collect basic facts and present first complexity results for Petri nets in general, which we frequently use in the following chapters. The following equation is called state equation [Mur77; Mur89]:

$$\mu_0 + \Delta(\Phi) = \mu.$$

Since $\Delta(\Phi) = D \cdot \Phi$ it can also be written as

$$\mu_0 + D \cdot \Phi = \mu.$$

It is well known that a marking μ is only reachable if the state equation is satisfied by some Parikh vector, i. e., by the Parikh image of some transition sequence. However, this is just a necessary condition and, in general, not a sufficient one, which is one of the reasons why the reachability problem is difficult to analyze and has a high complexity.

The following two observations are concerned about the size of minimal solutions of systems of linear Diophantine inequalities.

Theorem 2.14 ([Pot91], Theorem 1)**.** *Let $A \in \mathbb{Z}^{n\times m}$, and $C := \{x \in \mathbb{N}_0^m \mid Ax = 0\}$. Then, for all $x \in \mathcal{H}(C)$, $\|x\|_1 \leq (1 + \|A\|_{1,\infty})^r$, where r is the rank of A.*

Here, the set $\mathcal{H}(C)$ denotes the Hilbert basis of C, which is the set of all minimal elements w. r. t. $\leq$ of C. All elements of C can be expressed as a linear combination of vectors in $\mathcal{H}(C)$ with nonnegative integral coefficients.

Corollary 2.15 ([Pot91], Corollary 1)**.** *Let $A \in \mathbb{Z}^{n\times m}$, $b \in \mathbb{Z}^n$, and $C := \{x \in \mathbb{Z}^m \mid Ax \leq b\}$. Then, there are two finite subsets $C_1, C_2 \subsetneq \mathbb{Z}^m$ such that, for every $x \in \mathbb{Z}^m$*

- *$x \in C \Leftrightarrow x = x_1 + x_2 + \ldots + x_k$, with $x_1 \in C_1$, and $x_2, \ldots, x_k \in C_2$, and*
- *$\forall x \in C_1 \cup C_2 : \|x\|_1 \leq (2 + \|A\|_{1,\infty} + \|b\|_\infty)^n$.*

Using these observations, we find in the following corollary that the set of solutions of the state equation can be represented by a SLSR whose constant vectors are small. This implies that minimal solutions of the state equation are small as well.

Corollary 2.16. *Let $A \in \mathbb{Z}^{n\times m}$, $b \in \mathbb{Z}^n$, and $C := \{x \in \mathbb{N}_0^m \mid Ax = b\}$. Then, C is an SLS represented by an SLSR $\mathcal{SL} = \bigodot_{i=1}^{k} \mathcal{L}(\zeta_i, \{\pi_1, \ldots, \pi_\ell\})$ with $\|\zeta_i\|_1, \|\pi_j\|_1 \leq (2 + \|A\|_{1,\infty} + \|b\|_\infty)^{2n+m}$ for all $i \in [k]$ and $j \in [\ell]$.*

Proof. Let $A' \in \mathbb{Z}^{(2n+m)\times m}$ denote the matrix such that its first n rows are the rows of A, its next n rows are the rows of $-A$, and its next m rows are the rows of $-I_m$, where I_m is the $m \times m$ identity matrix. Furthermore, let $b' \in \mathbb{Z}^{2n+m}$ be obtained from b such that its first n components together are b, its next n components together are $-b$, and its next m components are 0. Let $C' := \{x \in \mathbb{Z}^m \mid A'x \leq b'\}$ denote the set of integral solutions of the system of linear Diophantine inequalities $A'x \leq b'$. Then, it's not hard to see that $C' = C$. If A is the all-0-matrix, then the corollary is obvious. Otherwise, $\|A\|_{1,\infty} = \|A'\|_{1,\infty}$ and $\|b\|_\infty = \|b'\|_\infty$.

Consider the sets C_1 and C_2 resulting from applying Corollary 2.15 to A' and b'. Note that all vectors of C_1 and C_2 must be nonnegative since otherwise there would be a solution with a negative component. Now, we define $\mathcal{SL}$ by letting ζ_i denote the i-th vector of C_1, and π_i the i-th vector of C_2. (In particular, all linear set representations of the semilinear set representation $\mathcal{SL}$ have the same periods.) Then, C is represented by $\mathcal{SL}$. Furthermore, each vector of this representation has a component sum of at most $(2 + \|A'\|_{1,\infty} + \|b'\|_\infty)^{2n+m} = (2 + \|A\|_{1,\infty} + \|b\|_\infty)^{2n+m}$. □

Lemma 2.17. *Let $N = (P, T, F, \mu_0)$ be a Petri net with n places and m transitions, and let W be the largest edge multiplicity of N. Further, let μ be a marking such that there is a Parikh vector Φ' with $\mu_0 + \Delta(\Phi') = \mu$. Then, there is a Parikh vector $\Phi \leq \Phi'$ with $\mu_0 + \Delta(\Phi) = \mu$ and component sum at most $(2 + mW + \max(\mu_0) + \max(\mu))^{2n+m}$.*

Proof. The Parikh vectors Φ with $\mu_0 + \Delta(\Phi) = \mu$ are exactly those that are solutions of the system $\mu_0 + Dx = \mu$. Hence, by applying Corollary 2.16 to D and $\mu - \mu_0$, the lemma follows. □

The next lemma shows that each loop of a Petri net can be decomposed into short loops.

Lemma 2.18. *Let $N = (P, T, F)$ be a Petri net with n places and m transitions, and let W be the largest edge multiplicity of N. Then, there is a finite set $\mathcal{H}(N) = \{\Phi_1, \ldots, \Phi_k\} \subsetneq \mathbb{N}_0^m$ of loops of N such that each loop of $\mathcal{H}(N)$ consists of at most $(1 + (n+m)W)^{n+m}$ transitions, and such that, for each loop Φ of N, there are $a_1, \ldots, a_k \in \mathbb{N}_0$ with $\Phi = a_1\Phi_1 + \ldots + a_k\Phi_k$.*

Proof. Let, w. l. o. g., $P = \{p_1, \ldots, p_n\}$ and $T = \{t_1, \ldots, t_m\}$. Furthermore, let $D \in \mathbb{Z}^{n\times m}$ be the displacement matrix of $\mathcal{P}$, i. e., the i-th column of D equals $\Delta(t_i)$, and $L := \{x \in \mathbb{N}_0^m \mid Dx \geq \vec{0}\}$. Note that L is the set of all loops of N. Let $L' := \{x \in \mathbb{N}_0^{m+n} \mid D'x = \vec{0}\}$, where $D' \in \mathbb{Z}^{n\times(m+n)}$ is the matrix whose first m columns are the columns of D and the next n columns are the columns of $-I_n$. By Theorem 2.14, we observe $\|x\|_1 \leq (1 + (m+n)W)^{m+n}$ for all $x \in \mathcal{H}(L')$. Let $\xi : \mathbb{N}_0^{m+n} \to \mathbb{N}_0^m$ denote the projection of $m+n$-dimensional vectors onto their first m components, and define $\mathcal{H}(N) := \xi(\mathcal{H}(L'))$. Since $L = \xi(L')$, the lemma follows. □

We further make several observations about the RecLFS problem of Petri nets in general. In comparison with many other classical problems, the complexity of the RecLFS problem is small.

Lemma 2.19. *The RecLFS problem of general Petri nets is in* **PSPACE**.

Proof. Let $\mathcal{P} = (P, T, F, \mu_0)$ be a Petri net with largest edge multiplicity W, and Φ a Parikh vector. Assume, Φ is enabled, and let σ be a firing sequence with $\Psi(\sigma) = \Phi$. Then, we have $\max(\mu_0, \sigma) \leq \max(\mu_0) + |\sigma| \cdot W$, i. e., each marking obtained when firing σ has polynomial encoding size. Therefore, the RecLFS problem can be decided in polynomial space by some nondeterministic Turing machine guessing σ step by step. □

Lemma 2.20. *The RecLFS problem of general Petri nets using an (appropriate) unary encoding scheme is* **NP**-*complete in the strong sense.*

Proof. **NP**-hardness in the strong sense has been shown in Watanabe et al. [Wat+89a]. Membership in **NP** is implied by the fact that a nondeterministic Turing machine can guess the order in which the transitions of Φ can be fired. Each step also takes only polynomial time since each marking obtained when firing Φ has polynomial encoding size. □

We note that Watanabe et al. [Wat+89a] also claimed that the RecLFS problem is in **NP**. However, as we will show later in Theorem 5.11, this problem is **PSPACE**-complete even if restricted to ordinary Petri nets, which implies that it cannot be in **NP** if **PSPACE** $\subsetneq$ **NP**. More precisely, assuming **PSPACE** $\subsetneq$ **NP**, it cannot be in **NP** if all numbers of the input are encoded using an alphabet Σ with $|\Sigma| \geq 2$ and if the numbers are not bounded by a polynomial of the input size. (Thus, it is possible that Watanabe et al. [Wat+89a] considered RecLFS with a unary encoding scheme as in Lemma 2.20 or only some restricted variation of the problem, which, however, isn't obvious from the text.) An immediate consequence is the following corollary.

Corollary 2.21. *If* **NP** $\subsetneq$ **PSPACE**, *then the RecLFS problem of general Petri nets is not* **PSPACE**-*hard in the strong sense.*

3 Communication-free Petri nets

In this chapter, we investigate communication-free Petri nets (cf-PNs, also known as BPP-PNs). A Petri net $\mathcal{P} = (P, T, F, \mu_0)$ is communication-free if it is forward-ordinary and $|{}^\bullet t| = 1$ holds for all $t \in T$. For the sake of improved readability, we will occasionally abuse notation by identifying ${}^\bullet t$ with its unique element. The term *communication-free Petri net* was coined by Hirshfeld [Hir94] (who originally considered only cf-PNs that are also backward-ordinary). Communication-free Petri nets are closely related to Basic Parallel Processes [Chr92; Chr+93], a subclass of Milner's process algebra (PA, which is also called Calculus of Communicating Systems, CCS, see [Mil95]). Furthermore, they are closely related to context-free commutative grammars (CFCGs, see [CRM76; Esp97; Huy83] and Section 2.5). For each cf-PN, the corresponding canonical commutative grammar is a CFCG. We will make use of this relationship on several occasions later in this chapter.

There is a large amount of literature involving cf-PNs. However, in most publications (including [Chr92; Chr+93; Hir94]), cf-PNs are used as a tool providing a net semantics for Basic Parallel Processes. The problems considered in these papers usually address model checking, (bi-)simulation, and similar concepts, which are not of major relevance in the context of this thesis. In the following, we focus on literature addressing the classical problems of our interest.

The strong constraints on topology and edge multiplicities limits the computational power of cf-PNs in the sense that they are unable to model synchronizing actions since the fireability of a transition only depends on exactly one place, at which only a single token is required. Esparza [Esp97] showed that the reachability problem of cf-PNs is, nevertheless, **NP**-complete. The **NP**-hardness is based on the fact that, even though no synchronization is possible, cf-PNs can contain decisions and attributions. This allows for a large number of possible firing sequences and enabled Parikh vectors, which can, for instance, be used to carry the different assignments of truth values to the variables of formulas in 3-CNF. (We remark that even in the absence of decisions, the reachability problem is **NP**-hard. In other words, the reachability problem of the subclass of cf-PNs that are also decision-free is **NP**-complete as we will show later.) The result of Esparza [Esp97] yields an alternative proof for the NP-completeness of the uniform word problem for CFCGs, a result that was obtained earlier by Huynh [Huy83]. Another proof for membership of the reachability problem in **NP**, based on canonical firing sequences, was given by Yen [Yen97]. Both proofs (implicitly) rely on the fact that the RecLFS problem for cf-PNs is decidable in polynomial time due to a very easily checkable criterion. A general approach for classes of Petri nets with simple circuits, including conflict-free Petri nets, was given by Yen and Yu [YY03], yielding an alternative proof for **NP**-completeness of the respective reachability (promise) problems. Yen [Yen97] also proposed an exponential time construction for $\mathcal{SL}$s of the reachability sets of cf-PNs. He then used this $\mathcal{SL}$ to argue that the equivalence problem for cf-PNs has a doubly exponential time bound.

Natural subclasses of cf-PNs are S-systems and state-machines. An S-system is an ordinary cf-PNs in which every transition has exactly one incoming and one outgoing edge. Intuitively, tokens in S-systems are only moved between places one by one but never destroyed or created. A state machine is an S-system with exactly one token at the initial marking (sometimes, state machines are defined in the same way as we defined S-systems). State machines are equivalent to finite automata of language theory. The reachability problem of S-systems is decidable in polynomial time [Ha+12; Mur89]. A superclass for ordinary cf-PNs is the class of free-choice Petri nets and well-known extensions of this class (like extended free-choice Petri nets). An ordinary Petri net (P, T, F, μ_0) is a free-choice Petri net if, for each edge $(p, t) \in P \times T$ of the net, either $p^\bullet = \{t\}$ or ${}^\bullet t = \{p\}$. This class was introduced by Hack [Hac72]. In general, the reachability problem of free-choice Petri nets is as hard as that of general Petri nets. However, many restricted subclasses enjoy much better complexities for this and other problems. We refer to Desel and Esparza [DE95] for an introduction on free-choice Petri nets. Other superclasses of cf-PNs are briefly introduced introduced in Chapter 6, also refer to Figure 6.1 in that chapter.

This chapter is organized as follows. In Section 3.1, we provide useful concepts and observations about cf-PNs to establish the foundations for later sections. This section also contains the first complexity results, namely, that the zero-reachability problem of cf-PNs is decidable in linear time, and that the reachability problem of cf-PNs remains **NP**-hard if restricted to cycle-free and decision-free cf-PNs.

In Section 3.2, we address a gap in Yen's [Yen97] construction for a SLSR of the reachability set. We show that, in general, the construction actually computes a proper superset of the reachability set instead. We proceed by fixing the construction in such a way that most parts of his argumentation can be retained while maintaining the bounds for the size and running time of the original construction (in the sense that all specified constants stay the same). Additionally, we apply results of Huynh [Huy85], using the close relationship of cf-PNs and CFCGs, to obtain a **coNEXPTIME** upper bound for the equivalence problem of cf-PNs.

For some notions of boundedness and liveness of BPPs ([Kuč96; May97; May98; May00]), polynomial time algorithms are already known. In addition to these, we investigate a number of other variations of the boundedness problem, and the covering problem for cf-PNs in Section 3.3, and variations of the liveness problem for cf-PNs in section 3.4. For two new variants of the boundedness problem, and for the covering problem, we show NP-completeness. The remaining problems can be solved very efficiently in polynomial time, most of them even in linear time. Some algorithms are also applicable to related well-known problems of BPPs, yielding linear time algorithms for BPPs in standard form. Additionally, our results imply illustrative linear time algorithms for important problems of context-free (commutative) grammars These results can be found in Section 3.5.

Linear time algorithms not only make these problems tractable in practice but also show that cf-PNs are too restricted if we are searching for classes of Petri nets where these problems are hard. We remark that, in this respect, an interesting parallel can be drawn between cf-PNs and conflict-free Petri nets (see Chapter 7). Both are structurally defined in a similar way, both have **NP**-complete reachability problems ([Esp97; HR88; Jon+77]), polynomial time algorithms for the boundedness problem (see Section 3.3 and [Ali+92]), the RecLFS problem (see [Esp97; HR88]), the liveness problem (see Section 3.4 and [Ali+92]), and $\mathbf{\Pi}_2^p$-hard equivalence problems (even $\mathbf{\Pi}_2^p$-complete for conflict-free PNs, see [HR88], [Yen13]). A number of forward-ordinary variations of cf-PNs, which are extended by states, timing or priority constraints, or inhibitor arcs, were investigated by Chen et al. [Che+09]. For some of these extensions, the reachability problem remains **NP**-complete, while for others it turns out to be not decidable at all.

In Chapter 6, we investigate a generalization of cf-PNs, called gcf-PNs, characterized by the sole topological constraint that each transition has at most one incoming edge. In particular, gcf-PNs are not necessarily forward-ordinary. It turns out that almost all problems considered (e. g., RecLFS, (zero-)reachability, liveness, boundedness, covering, RecLFS) are **PSPACE**-complete. The large gap regarding the complexity between many problems for cf-PNs and corresponding problems for gcf-PNs motivates further research of how generalizing classes of Petri nets w. r. t. their edge multiplicities influences the complexity of different problems.

3.1 Fundamental concepts and observations

Some concepts and observations about cf-PNs are needed at several occasions. We collect and prove them in this section. As already mentioned in Chapter 1, useful topological characterizations for properties of Petri nets are often possible if the class under consideration is topologically restricted and forward-ordinary. This also applies to cf-PNs.

Of major interest in the analysis of cf-PNs are the strongly connected components (SCCs). The directed acyclic graph obtained by shrinking all SCCs to super nodes while maintaining the edges between distinct SCC as edges between the corresponding super nodes is called the condensation (of the Petri net). An SCC is a top component if it has no incoming edges in the condensation. For two not necessarily distinct SCCs C_1, C_2, we write $C_1 \geq C_2$ if there is a path from C_1 to C_2 in the condensation (i. e., if for all $v_1 \in C_1$, $v_2 \in C_2$, there is a path from v_1 to v_2 in the Petri net).

Also important is the concept of traps. A subset $Q \subseteq P$ of places is a trap if, for all $t \in T$, ${}^\bullet t \cap Q \neq \emptyset$ implies $t^\bullet \cap Q \neq \emptyset$. In other words, every transition that removes a token from Q also adds a token to Q. Once a trap is marked, it cannot become unmarked by firing a transition. Given a subset $R \subseteq P$ of places, the maximum trap of R is the largest trap $Q \subseteq R$. Note that the maximum trap of R is unique since the union of two traps of R is again a trap of R.

On several occasions, we use the Parikh extension of the Petri net under consideration.

Definition 3.1 (Parikh extension)**.** Let $\mathcal{P} = (P, T, F, \mu_0)$ be a Petri net. The Parikh extension $\mathcal{P}^\Psi = (P^\Psi, T, F^\Psi, \mu_0^\Psi)$ of $\mathcal{P}$ is obtained from $\mathcal{P}$ by adding, for each transition $t \in T$, an unmarked place p_t^Ψ with $F^\Psi(t, p_t^\Psi) = 1$.

Figure 3.1 illustrates a Petri net and its Parikh extension. If we fire a firing sequence σ in the Parikh extension $\mathcal{P}^\Psi$ leading to marking μ, then the new place p_t^Ψ counts how often the transition $t \in T$ is fired, i. e., $\mu(p_t^\Psi) = \Psi(\sigma)(t)$. In other words, the projection of μ onto the new places equals $\Psi(\sigma)$. We remark that the concept of the Parikh extension is closely related to the concept of extended Parikh maps used in [LR78] and [HR88] for persistent and conflict-free Petri nets.

We continue with a number of observations.

Theorem 3.2 ([Esp97], Theorem 3.1)**.** *Let $\mathcal{P} = (N, \mu_0)$ be a cf-PN. A Parikh vector Φ is enabled in $\mathcal{P}$ if and only if*

(a) $\mu_0 + \Delta(\Phi) \geq \vec{0}$, and

(b) each top component of $\mathcal{P}[\Phi]$ has a marked place.

Proof. This theorem is equivalent to Theorem 3.1 of [Esp97]. However, our formulation is better suited for our purposes. The original theorem states that Φ is enabled if and only if (*a*) holds and if, within $\mathcal{P}[\Phi]$, each place is the end node of some path starting at a marked place. □

Lemma 3.3. *Let $\mathcal{P} = (P, T, F, \mu_0)$ be a Petri net, σ a firing sequence in $\mathcal{P}$, and let μ_i, $i \in [|\sigma|]$, be defined by $\mu_0 \xrightarrow{\sigma[1]} \mu_1 \xrightarrow{\sigma[2]} \dots \xrightarrow{\sigma[k]} \mu_k$. Then, for each place p of $\mathcal{P}[\Psi(\sigma)]$, there is an $i \in [0, |\sigma|]$ such that p is marked at μ_i.*

Proof. Each place p of $\mathcal{P}[\Psi(\sigma)]$ is in the pre- or postset of some transition $\sigma_{[i]}$. If $p \in {}^\bullet\sigma_{[i]}$, then p must be marked at μ_{i-1}. If $p \in \sigma_{[i]}{}^\bullet$, then p is marked at μ_i. □

The next lemma demonstrates an alternative way of decomposing loops into subloops with nice properties. In contrast to Lemma 2.18, this lemma is not valid in general.

Lemma 3.4. *Let $\Phi \in \mathbb{N}_0^m$ be a loop of a cf-PN $\mathcal{P} = (P, T, F)$, and let $C_1, \ldots, C_k$, $k \geq 1$, denote the top components of $\mathcal{P}[\Phi]$. Then Φ can be decomposed into loops $\Phi_1, \ldots, \Phi_k$, $k \leq n$, such that*

(a) $\Phi = \sum_{i=1}^{k} \Phi_i$, and

(b) the only top component of $\mathcal{P}[\Phi_i]$ is C_i.

Proof. Let $\vartheta_1, \ldots, \vartheta_k \leq \Phi$ be the minimal (w. r. t. "$\leq$") Parikh vectors such that $C_i \subseteq \mathcal{P}[\vartheta_i]$, and $\vartheta_i(t) = \Phi(t)$ for all $t \in C_i$. Let ϑ be some Parikh vector such that $\vartheta \leq \Phi - \vartheta_1 - \ldots - \vartheta_k$, $\vartheta_1 + \vartheta$ is a loop, C_1 is the only top component of $\mathcal{P}[\vartheta_1 + \vartheta]$, and ϑ is maximal (i. e., there is no Parikh vector $\vartheta' > \vartheta$) with these properties. Note that each ϑ_i is a loop since Φ is a loop. Hence, ϑ always exists and $\Phi_1 := \vartheta_1 + \vartheta$ is a loop, too. We will show that the remaining Parikh vector $\Phi - \Phi_1$ is also a loop, and that the top components of $\mathcal{P}[\Phi - \Phi_1]$ are exactly $C_2, \ldots, C_k$. Then, the lemma follows from the fact that we can iteratively apply this construction to the respective remaining Parikh vector to obtain $\Phi_1, \ldots, \Phi_k$ as given in the lemma.

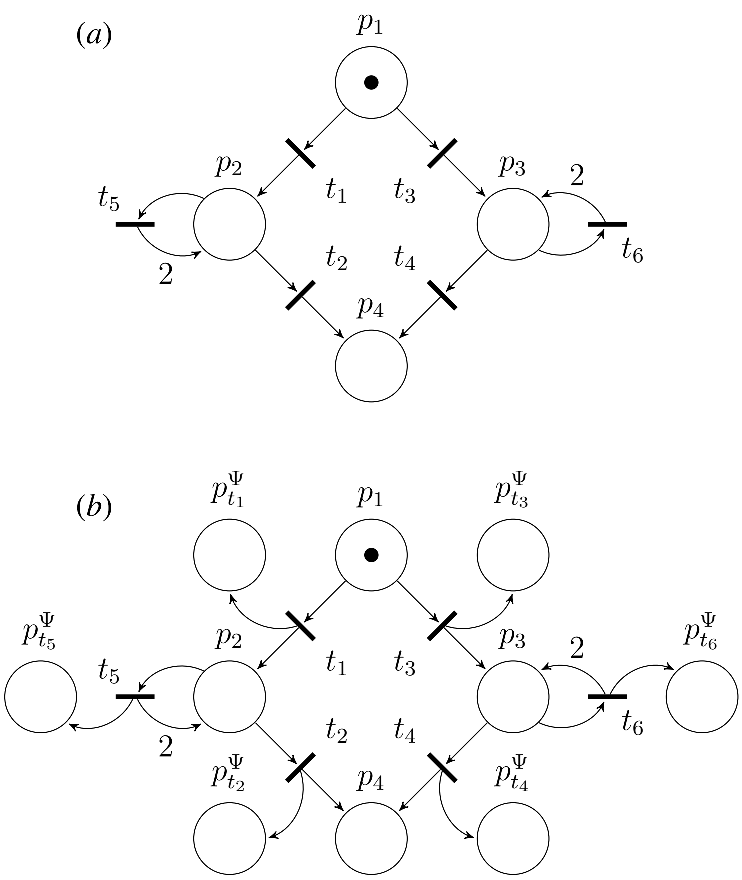


Figure 3.1: (*a*) illustrates a Petri net, and (*b*) its Parikh extension. The net of (*a*) also serves as a counter example for the construction proposed in Proposition 3.12.

Assume for the sake of contradiction that $\Phi - \Phi_1$ is not a loop. Then, there is a place p such that $\Delta(\Phi - \Phi_1)(p) < 0$. This implies that there is a transition $t \in \Phi - \Phi_1$ with $p = {}^\bullet t$. Since Φ is a loop, $\Delta(\Phi) = \Delta(\Phi - \Phi_1) + \Delta(\Phi_1) \geq \vec{0}$, and therefore $\Delta(\Phi_1)(p) > 0$. But then, we observe $\Delta(\Phi_1)(p) + \Delta(t)(p) \geq 0$, i. e., a contradiction to ϑ being maximal.

Next we show that the top components of $\mathcal{P}[\Phi - \Phi_1]$ are exactly $C_2, \ldots, C_k$. Obviously, $C_2, \ldots, C_k$ are top components. Assume for the sake of contradiction that $\mathcal{P}[\Phi - \Phi_1]$ has another top component C. Let $\vartheta_C \leq \Phi - \Phi_1$ be a 0-1-vector such that C is the only SCC of $\mathcal{P}[\vartheta_C]$. ϑ_C must be a loop since otherwise $\Phi - \Phi_1$ wouldn't be a loop. Since C is not a top component of $\mathcal{P}[\Phi]$, there is a place $p \in C$ and a transition t with $t \in {}^\bullet p$, $t \in \Phi$, and $t \notin \Phi - \Phi_1$. (Note that each top component of any (Parikh vector induced) cf-PN contains a place because each transition has an incoming edge.) Since also $t \notin \vartheta_1$ (otherwise C_1 and C would be part of the same SCC), we find $t \in \vartheta$. But then, ϑ isn't chosen maximally since $\vartheta + \vartheta_C$ is larger with respect to the above properties, a contradiction. □

An example for this decomposition is illustrated in Figure 3.2.

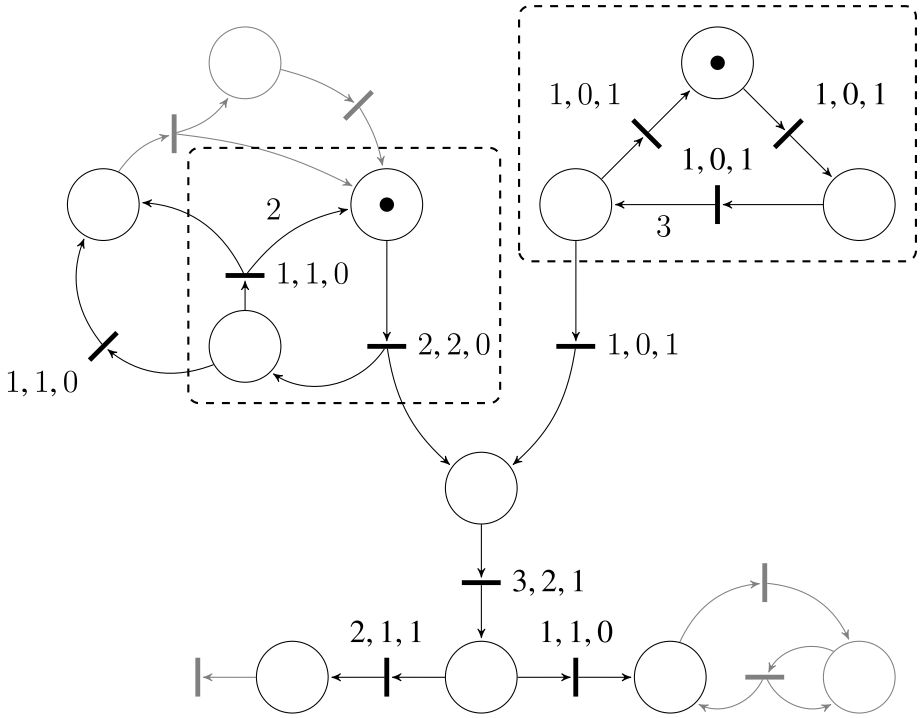


Figure 3.2: This figure illustrates the loop decomposition of Lemma 3.4 applied to a Petri net $\mathcal{P}$ and a loop Φ. Places and transitions that are not part of $\mathcal{P}[\Phi]$ are gray. The top components of $\mathcal{P}[\Phi]$ are surrounded by dashed rectangles. The labels of the transitions t are triples $\Phi(t), \Phi_1(t), \Phi_2(t)$, where Φ_1, Φ_2 are the subloops of the decomposition of Φ. Note that the loop decomposition is in general not unique. In this example, Φ is enabled since all top components of $\mathcal{P}[\Phi]$ are marked.

Using these observations, we can show the following lemma.

Lemma 3.5. *Let $\mathcal{P} = (P, T, F, \mu_0)$ be a cf-PN, and Φ, ϑ Parikh vectors such that ϑ is a loop, and both Φ and $\Phi + \vartheta$ are enabled at μ_0. Then, for each firing sequence σ such that $\mathcal{P}[\Phi]$ is a subnet of $\mathcal{P}[\Psi(\sigma)]$, there are transition sequences $\sigma_1, \ldots, \sigma_{k+1}$ and loops $\vartheta_1, \ldots, \vartheta_k$, $k \leq n$, such that* (*a*) $\sigma = \sigma_1 \cdots \sigma_{k+1}$,

(*b*) $\vartheta = \vartheta_1 + \ldots + \vartheta_k$,

(*c*) $\mathcal{P}[\vartheta_i]$, $i \in [k]$, *has exactly one top component, and this top component is the* i*-th top component of* $\mathcal{P}[\vartheta]$ *using a properly chosen numbering of the top components, and*

(*d*) ϑ_i, $i \in [k]$, *is enabled at marking* μ_i *where* $\mu_0 \xrightarrow{\sigma_1 \cdots \sigma_i} \mu_i$.

Proof. Consider the decomposition of ϑ by Lemma 3.4 into loops $\vartheta_1, \ldots, \vartheta_k$, $k \leq n$, such that $\vartheta = \sum_{i=1}^{k} \vartheta_i$, and the i-th top component C_i of $\mathcal{P}[\vartheta]$ is the unique top component of $\mathcal{P}[\vartheta_i]$.

Let $i \in [k]$. Assume that C_i and $\mathcal{P}[\Phi]$ are disjoint. Then, C_i is a top component of $\mathcal{P}[\Phi + \vartheta]$, and C_i is marked at μ_0 by Theorem 3.2 since $\Phi + \vartheta$ is enabled at μ_0. Therefore, by the same lemma, ϑ_i is enabled at μ_0.

Now, assume that C_i and $\mathcal{P}[\Phi]$ are not disjoint, i. e., they share a place p. Since $\mathcal{P}[\Phi]$ is a subnet of $\mathcal{P}[\Psi(\sigma)]$, Lemma 3.3 implies that, for each place p of $\mathcal{P}[\Phi]$, there is a marking μ reached by some prefix of σ such that p is marked at μ. Therefore, by Theorem 3.2, ϑ_i is enabled at μ .

We conclude that, by splitting the sequence σ at appropriate positions, we obtain transition sequences σ_i and markings μ_i, $i \in [k+1]$, such that $\sigma = \sigma_1 \cdot \sigma_2 \cdots \sigma_{k+1}$, $\mu_0 \xrightarrow{\sigma_1} \mu_1 \cdots \xrightarrow{\sigma_k} \mu_k \xrightarrow{\sigma_{k+1}} \mu_{k+1}$, and ϑ_i is enabled at μ_i where we assume w. l. o. g. that the top components of $\mathcal{P}[\vartheta]$ are appropriately numbered. □

We will later use this lemma to enrich certain "backbone" firing sequences with loops, which will serve as witnesses for unboundedness.

The next few lemmata, which all involve traps of cf-PNs in some way, lay the foundation for many linear time algorithms for various problems presented in later sections.

Lemma 3.6. *Let* $\mathcal{P} = (P, T, F, \mu_0)$ *be a cf-PN and* $R \subseteq P$ *be a set of places. The maximum trap* Q *of* R *can be determined in linear time.*

Proof. We apply the following procedure. We initialize Q by R. As long as there is a transition $t \in T$ such that ${}^\bullet t \in Q$ and $t^\bullet \cap Q = \emptyset$, we remove ${}^\bullet t$ from Q. At the end of the procedure, Q must be a trap since otherwise the procedure wouldn't have stopped. Furthermore, Q is a maximum trap of R since the procedure cannot remove a place p from Q if p is part of the maximum trap of R.

We can implement the procedure in linear time as follows. We use two arrays A and N, and a list L, as well as a collection Q. The collection Q is initialized with the set R. Array A has length $|T|$ and $A[i]$ is initialized with $|t_i^\bullet \cap R|$. Array N has length $|P|$ and $N[i]$ is initialized with an empty list if $p_i \notin R$, and otherwise with a list of all transitions t_j such that $t_j \in {}^\bullet p_i$. The list L is initialized with all transitions t_i for which ${}^\bullet t_i \in Q$ and $A[i] = 0$ hold. It's not hard to see that these data structures can be initialized in linear time.

Now, as long as L is not empty, we do the following. First, we pop some transition t_i from the list. Assume $p_j = {}^\bullet t_i$. Then, if $p_j \in Q$, we remove p_j from Q, and, for each t_k contained in the list $N[j]$, we decrease $A[k]$ by 1, and add t_k to L if $A[k] = 0$ after the decreasing step. When L is empty, Q is the maximum trap of R. The running time of this procedure is linear. □

Lemma 3.7. *Let $\mathcal{P} = (P, T, F, \mu_0)$ be a cf-PN, and $R \subseteq P$ be a subset of places such that no subset $Q \subseteq R$ is a trap. Then, there is a firing sequence σ with $\Delta(\sigma)(p) \geq 0$ for all $p \notin R$, leading to a marking at which R is empty.*

Proof. By definition, if a set $Q \subseteq P$ is not a trap, then there is a transition t with ${}^\bullet t \in Q$ and $t^\bullet \cap Q = \emptyset$. Define the transitions $t_1, \ldots, t_{|R|}$ and the sets $R_1, \ldots, R_{|R|}$ recursively as follows. We start with $R_1 := R$. Given R_i for $i \in [|R|]$, we choose t_i as a transition with ${}^\bullet t_i \in R_i$ and $t_i^\bullet \cap R_i = \emptyset$, and set $R_{i+1} := R_i \setminus {}^\bullet t_i$. This means that $R_{|R|} \subsetneqq \ldots \subsetneqq R_1$, and we can successively empty $R_{|R|}, \ldots, R_1$ by firing each of the transitions $t_{|R|}, \ldots, t_1$ an appropriate number of times. Since these transitions don't remove tokens from places outside of R, the displacement of the firing sequence at these places is nonnegative. □

Lemma 3.8. *Let $\mathcal{P} = (P, T, F, \mu_0)$ be a cf-PN, and $Q \subseteq R \subseteq P$ be the maximum trap of R. Then, the following are equivalent:*

(*a*) *R is empty at some reachable marking μ with $\mu(p) \geq \mu_0(p)$ for all $p \notin R$,*

(*b*) *R is empty at some reachable marking,*

(*c*) *Q is empty at μ_0.*

Proof. "(*a*) ⇒ (*b*)": There is nothing to show.

"(*b*) ⇒ (*c*)": If Q is marked, then $R \supseteq Q$ will always be marked, regardless of the transitions fired.

"(*c*) ⇒ (*a*)": Notice that $R \setminus Q$ doesn't contain a trap by the maximality of Q. Consider the cf-PN $\mathcal{P}'$ which emerges from $\mathcal{P}$ by removing Q and all transitions incident to Q. If $R \setminus Q$ would contain a trap w. r. t. $\mathcal{P}'$, then at least one transition t with ${}^\bullet t \in R \setminus Q$ and $t^\bullet \cap Q \neq \emptyset$ was removed, a contradiction to Q being the maximum trap of R. Therefore, $R \setminus Q$ contains no trap w. r. t. $\mathcal{P}'$, and, by Lemma 3.7, there is a firing sequence σ of $\mathcal{P}'$ removing all tokens from $R \setminus Q$ such that $\Delta(\sigma)(p) \geq 0$ for all $p \notin R \setminus Q$. Firing σ in $\mathcal{P}$ removes all tokens from $R \setminus Q$ without putting any tokens to a place of Q since such transitions don't exist in $\mathcal{P}'$. Hence, the marking μ reached by σ in $\mathcal{P}$ satisfies the properties of (*a*). □

Lemma 3.9. *Given a cf-PN $\mathcal{P} = (P, T, F, \mu_0)$ and a set $R \subseteq P$, we can decide in linear time if there is a reachable marking at which R is empty.*

Proof. Using Lemma 3.6, we find in linear time the maximum trap Q of R and check if it is empty. By Lemma 3.8, this is the case if and only if R can be emptied. □

These observations are already sufficient to show that the zero-reachability problem which, in general, is as hard as the reachability problem (see [Hac74b]), is decidable in linear time for cf-PNs.

Theorem 3.10. *The zero-reachability problem for cf-PNs is decidable in linear time.*

Proof. We simply apply Lemma 3.9 to $\mathcal{P}$ and P. □

Another immediate minor result can be obtained by observing that attributions provide sufficient computational power to make the reachability problem of cf-PNs **NP**-hard.

Theorem 3.11. *The reachability problem of cf-PNs is* **NP***-complete, even if restricted to end marking* $\vec{1}$ *and ordinary cf-PNs without cycles and decisions.*

Proof. Since the reachability problem of cf-PNs is **NP**-complete [Esp97], we only have to show **NP**-hardness for cf-PNs without cycles and decisions. We adapt the proof of Esparza [Esp97] who showed that the reachability problem of cf-PNs is **NP**-complete. Given a formula in 3-CNF over the variables $x_1, \ldots, x_k$ and clauses $C_1, \ldots, C_\ell$, we construct a cf-PN without cycles and decisions in which the all-1-marking $\vec{1}$ is reachable if and only if the formula can be satisfied. An example is illustrated in Figure 3.3. □

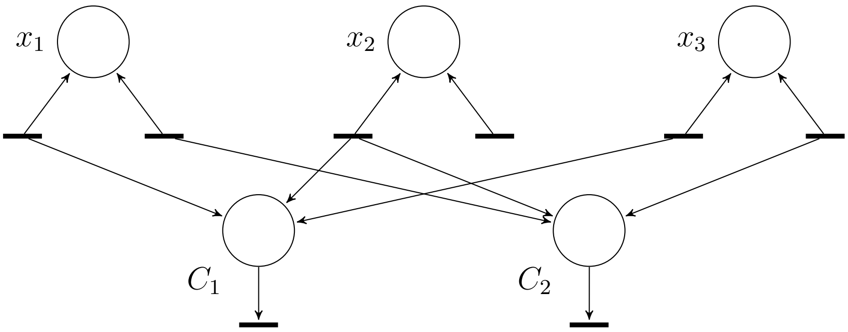


Figure 3.3: The formula $C_1 \wedge C_2 :=(x_1 \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3})$ can be satisfied if and only if the marking $\vec{1}$ is reachable in this ordinary cycle-free and decision-free cf-PN.

3.2 The equivalence problem

In this section we consider the equivalence problem of cf-PNs. In [Yen97], Yen proposed a construction for an SLSR of the reachability set of cf-PNs. The encoding size of the representation obtained by this construction is at most exponential in the size of the cf-PN. The author used the already mentioned result that the equivalence problem of SLSR is in $\mathbf{\Pi}_2^p$ (see Theorem 2.4) to claim a doubly exponential time bound for this problem. The construction of the SLSR is contained in the proof proposed by Yen [Yen97] for the following proposition.

Proposition 3.12 ([Yen97], Theorem 5)**.** *Let $\mathcal{P} = (P, T, F, \mu_0)$ be a cf-PN. For some fixed constants c_1, c_2, d_1, d_2, d_3, we can construct in* **DTIME**$(2^{c_2 \mathsf{size}(\mathcal{P})^3})$ *an SLSR $\mathcal{SL} = \bigodot_{\mu \in B} \mathcal{L}(\mu, \rho_\mu)$ of $\mathcal{R}(\mathcal{P})$ whose size is bounded by $\mathcal{O}(2^{c_1 \mathsf{size}(\mathcal{P})^3})$, where*

1. *B is the set of reachable markings with components at most $2^{d_1 \mathsf{size}(\mathcal{P})^2}$, and*
2. *ρ_μ is the set of all $\Delta(\tau)$ such that*
   (a) *τ is a loop,*
   (b) *$\Delta(\tau)$ has no component larger than $2^{d_2 \mathsf{size}(\mathcal{P})^2}$, and*
   (c) *$\exists\, \alpha, \beta \in T^*$, $\exists$ marking ν,*

(i) $\mu_0 \xrightarrow{\alpha} \nu \xrightarrow{\beta} \mu$,

(ii) τ is enabled at ν,

(iii) $|\tau|, |\alpha\beta| \leq 2^{d_3 \mathrm{size}(\mathcal{P})^2}$.

We show that there are cf-PNs such that $\mathcal{SL}$ contains markings that are not reachable. Consider the cf-PN $\mathcal{P}$ with initial marking $\mu_0 \triangleq p_1$, illustrated in (*a*) of Figure 3.1. The marking $\mu :\triangleq p_4$ is reachable.

In particular we have $\mu_0 \xrightarrow{t_1} \nu :\triangleq p_2 \xrightarrow{t_2} \mu$ as well as $\mu_0 \xrightarrow{t_3} \nu' :\triangleq p_3 \xrightarrow{t_4} \mu$. Notice that we can safely and w. l. o. g. assume $\mu \in B$ since we can blow up the size of the net by adding unrelated places. We observe that the loop t_5 is enabled at ν while t_6 is enabled at ν'. As before, we can safely assume $|t_1t_2|, |t_3t_4|, |t_5|, |t_6| \leq 2^{d_3 \mathrm{size}(\mathcal{P})^2}$. Therefore, we find $\Delta(t_5), \Delta(t_6) \in \rho_\mu$. But then, the unreachable marking $p_2 + p_3 + p_4 \triangleq \mu + \Delta(t_5) + \Delta(t_6)$ is in $\mathcal{L}(\mu, \rho_\mu)$. Hence, the constructed SLSR of $\bigcup_{\mu \in B} \mathcal{L}(\mu, \rho_\mu)$ cannot represent $\mathcal{R}(\mathcal{P})$.

The inclusion $\mathcal{R}(\mathcal{P}) \subseteq \mathcal{SL}$ is proven correctly in [Yen97]. Our goal is to repair the construction in such a way that we can almost completely reuse the proof given for this direction. Our first step is to show that there is a certain subclass of cf-PNs for which the other direction $\mathcal{SL} \subseteq \mathcal{R}(\mathcal{P})$ is also true. To this end, we observe that the crucial property that makes $\mathcal{P}$ a counter example is that μ is reachable by the two firing sequences t_1t_2 and t_3t_4 which have different Parikh images. Indeed, for cf-PNs having the nice property that any two firing sequences leading to the same marking have the same Parikh image, the theorem holds.

Theorem 3.13. *Let $\mathcal{P} = (P, T, F, \mu_0)$ be a cf-PN such that, for all firing sequences σ, σ' leading to the same marking, $\Psi(\sigma) = \Psi(\sigma')$ holds. For some fixed constants c_1, c_2, d_1, d_2, d_3, we can construct in* **DTIME**$(2^{c_2 \mathrm{size}(\mathcal{P})^3})$ *an SLSR $\mathcal{SL} = \bigodot_{\mu \in B} \mathcal{L}(\mu, \rho_\mu)$ of $\mathcal{R}(\mathcal{P})$ whose size is bounded by $\mathcal{O}(2^{c_1 \mathrm{size}(\mathcal{P})^3})$, where*

1. *B is the set of reachable markings with components at most $2^{d_1 \mathrm{size}(\mathcal{P})^2}$, and*

2. *ρ_μ is the set of all $\Delta(\tau)$ such that*

   (*a*) *τ is a loop,*

   (*b*) *$\Delta(\tau)$ has no component larger than $2^{d_2 \mathrm{size}(\mathcal{P})^2}$, and*

   (*c*) *$\exists\, \alpha, \beta \in T^*$, $\exists$ marking ν,*

   (i) $\mu_0 \xrightarrow{\alpha} \nu \xrightarrow{\beta} \mu$,

   (ii) *τ is enabled at ν,*

   (iii) $|\tau|, |\alpha\beta| \leq 2^{d_3 \mathrm{size}(\mathcal{P})^2}$.

Proof. Assume $B \neq \emptyset$, and let $\mu \in B$ and $\mu' \in \mathcal{L}(\mu, \rho_\mu)$ be arbitrarily chosen. Our goal is to show that μ' is reachable. Let $\rho_\mu := \{\Delta(\tau_1), \ldots, \Delta(\tau_\ell)\}$, where τ_i, $i \in [\ell]$, are loops satisfying the properties (*a*)–(*c*) of the lemma. By definition, there are $a_1, \ldots, a_\ell \in \mathbb{N}_0$ such that $\mu' = \mu + \sum_{i=1}^{\ell} a_i \Delta(\tau_i)$. For $\Delta(\tau_i)$, $i \in [\ell]$, let α_i denote the sequence α as defined in the theorem. Since all firing sequences leading to μ have the same Parikh image Φ_μ, $\mathcal{P}[\Psi(\alpha_i)]$ is a subnet of $\mathcal{P}[\Phi_\mu]$.

Let σ be some firing sequence with Parikh image Φ_μ, and let μ_j, $j \in [|\sigma|]$, be defined by $\mu_0 \xrightarrow{\sigma[..j]} \mu_j$. By applying Lemma 3.5 to $\Psi(\alpha_i)$, $\Psi(\tau_i)$, and σ for all $i \in [\ell]$, we find that for any subloop of any

$\Psi(\tau_i)$ there is a $j \in [0, |\alpha|]$ such that the subloop under consideration is enabled at μ_j. Therefore, the Parikh vector $\Psi(\sigma) + \sum_{i=1}^{\ell} a_i \Psi(\tau_i)$ leading to μ' is enabled at μ_0. □

We can use this theorem and the corresponding construction in a mediate way to construct an SLSR of $\mathcal{R}(\mathcal{P})$ in exponential time for every cf-PN $\mathcal{P}$.

Theorem 3.14. *Let $\mathcal{P} = (P, T, F, \mu_0)$ be a cf-PN. For some fixed constants c_1, c_2, d_1, d_2, we can construct in DTIME$(2^{c_2 \text{size}(\mathcal{P})^3})$ an SLSR of the reachability set $\mathcal{R}(\mathcal{P})$ whose size is bounded by $\mathcal{O}(2^{c_1 \text{size}(\mathcal{P})^3})$ where no component of any constant vector is larger than $2^{d_1 \text{size}(\mathcal{P})^2}$ and no component of any period is larger than $2^{d_2 \text{size}(\mathcal{P})^2}$.*

Proof. We compute the Parikh extension $\mathcal{P}^\Psi$ of $\mathcal{P}$ (see Definition 3.1). First notice that $\mathcal{P}^\Psi$ is a cf-PN. Since all firing sequences of $\mathcal{P}^\Psi$ leading to the same marking have the same Parikh image, we can apply the construction given in [Yen97], which is correct for $\mathcal{P}^\Psi$ by Theorem 3.13, in order to obtain a SLSR $\mathcal{SL}$ of $\mathcal{R}(\mathcal{P}^\Psi)$. Now notice that a marking μ is reachable in $\mathcal{P}$ if and only if there is a marking μ' that is reachable in $\mathcal{P}^\Psi$ such that the projection of μ' onto the places of $\mathcal{P}$ equals μ (to see this, simply apply the same firing sequence). Therefore, the projection of $\mathcal{SL}$ onto the places of $\mathcal{P}$ yields an SLSR of $\mathcal{R}(\mathcal{P})$.

The running time of this projection is linear in the size of $\mathcal{SL}$. In turn, the size of $\mathcal{P}^\Psi$ is linear in the size of $\mathcal{P}$. Hence, the constants c_1, c_2, d_1, d_2 may be larger for this theorem than for Theorem 3.13 but all specified constants (like the cube of "size$(\mathcal{P})^3$") are not increased. □

As already mentioned, using the SLSRs of $\mathcal{R}(\mathcal{P}_1)$ and $\mathcal{R}(\mathcal{P}_2)$ constructed in this way, one can use the fact that equivalence of SLSRs is in $\mathbf{\Pi}_2^p$ to decide the equivalence problem of cf-PNs in deterministic doubly exponential time. Obviously, this is not the strongest result which can be obtained by this argument since it follows that this problem is also in $\mathbf{\Pi}_2^e$, the complement of the second level of the exponential hierarchy.

However, following Huynh [Huy85], an even stronger result can be given. There, the *inequivalence* problem of CFCGs is shown to be contained in **NEXPTIME**. The canonical commutative grammar (see Section 2.5) of a cf-PN is a CFCG. Therefore, since the equivalence problem of cf-PNs can be reduced in polynomial time to the equivalence problem of CFCGs, using the canonical commutative grammars of the input cf-PNs, the equivalence problem of cf-PNs is in **coNEXPTIME**, too. One could also translate the arguments given by Huynh [Huy85] for CFCGs into the world of cf-PNs to give the **coNEXPTIME**-bound: The general idea is that one can construct an SLSR of the reachability set consisting of an exponential number of only polynomially sized LSRs. By that, the membership of an exponentially sized marking can be checked in deterministic exponential time instead of using an $\mathbf{\Pi}_2^e$-algorithm. Since, in the case of *inequivalence*, an exponentially sized marking witnessing the inequivalence can be guessed in **NEXPTIME**, it follows that the equivalence problem of cf-PNs is in **coNEXPTIME**.

For an analogue lower bound of the equivalence problem of cf-PNs, it is not obvious how to adapt the proof of $\mathbf{\Pi}_2^p$-hardness for the equivalence problem of CFGs or CFCGs with one terminal symbol, given by Huynh [Huy84]. He presented a reduction from the $\mathbf{\Pi}_2^p$-hard inequivalence problem of integer expressions, using variables (whose number depend on the input) for carrying out computations. However, Yen [Yen13] gave an alternative proof for $\mathbf{\Pi}_2^p$-hardness of the equivalence problem of cf-PNs, by reducing the $\mathbf{\Pi}_2^p$-complete problem $\overline{\Sigma_2^P\text{-SAT}}$ to it. ($\overline{\Sigma_2^P\text{-SAT}}$ denotes the complement of Σ_2^P-SAT, and consists of all Boolean expressions of the form $\forall x \, \exists y : \phi(x, y)$ that are *not* true, where

x and y are vectors of Boolean variables, and ϕ is a Boolean formula in 3-DNF with variables of x and y.) These are the strongest bounds known so far in terms of levels of the polynomial hierarchy.

Theorem 3.15. *The equivalence problem of cf-PNs is* Π_2^p*-hard and contained in* **coNEXPTIME**.

We observe that the gap between the lower and the upper bound is still very large. Improving or even closing this gap is an open problem. The same holds for the equivalence problem of CFCGs.

3.3 Boundedness problems and the covering problem

In this section we investigate several variations of the boundedness problem, and show that the covering problem of cf-PNs is **NP**-complete. We first define the concepts of boundedness we are interested in.

Definition 3.16. Let $\mathcal{P} = (P, T, F, \mu_0)$ be a Petri net, and $R \subseteq P$. A place $p \in P$ is

- unbounded (with unmarked R) if, for all $k \in \mathbb{N}$, there is a reachable marking $\mu \in \mathcal{R}(\mathcal{P})$ such that $\mu(p) \geq k$ (and $\mu(r) = 0$ for all $r \in R$, resp.),
- unbounded on an ω-firing sequence σ if, for all $k \in \mathbb{N}$, there is a finite prefix of σ leading to a marking μ such that $\mu(p) \geq k$ (such a place is also called ω-unbounded),
- persistently unbounded if, for all reachable markings $\mu \in \mathcal{R}(\mathcal{P})$, p is unbounded in the Petri net (P, T, F, μ),
- universally unbounded if p is persistently unbounded and unbounded on each ω-firing sequence.

A set $S \subseteq P$ of places is

- simultaneously unbounded if, for all $k \in \mathbb{N}$, there is a reachable marking $\mu \in \mathcal{R}(\mathcal{P})$ such that $\mu(p) \geq k$ for all $p \in S$.
- simultaneously ω-unbounded if there is an ω-firing sequence σ such that, for all $k \in \mathbb{N}$, there is a finite prefix of σ leading to a marking μ satisfying $\mu(p) \geq k$ for all $p \in S$.

We remark that, for a place, "universally unbounded" implies "persistently unbounded" which in turn implies "unbounded on an ω-firing sequence" which implies "unbounded". Furthermore, by Lemma 3.2 of [LR78] a set $S \subseteq P$ of places is simultaneously ω-unbounded if and only if there is an ω-firing sequence σ such that all places $p \in S$ are unbounded on (the same sequence) σ. Hence, this on first sight weaker characterization yields another definition for the same concept. The notion of universally unboundedness is, to our knowledge, new. The motivation behind this concept is that an universally unbounded place in a certain sense measures the progress of the computation of a Petri net. The concept of a place being unbounded with unmarked R is also new. It is mainly motivated by the fact that theorems using this concept can be used to decide a variety of problems for CFCGs in a very illustrative way.

3.3.1 Concepts of non-simultaneously unboundedness

In this subsection we investigate concepts of unboundedness where the places under consideration are not required to be simultaneously (ω-)unbounded, and provide efficient algorithms for the corresponding problems. We first derive a characterization of unbounded and ω-unbounded places in terms of strongly connected components.

Lemma 3.17. *Let $\mathcal{P} = (P, T, F, \mu_0)$ be a cf-PN, and $p \in P$ a place. Then, the following are equivalent:*

1. *p is unbounded,*
2. *there is a loop τ with $\Delta(\tau)(p) > 0$ enabled at some reachable marking,*
3. *p is unbounded on some ω-firing sequence,*
4. *there are strongly connected components C_1, C_2, C_3, C_4 of $\mathcal{P}$ such that*
   - (*a*) *$p \in C_4$,*
   - (*b*) *$C_1 \geq C_2 \geq C_3 \geq C_4$,*
   - (*c*) *C_1 has a marked place, and*
   - (*d*) *C_2 has a transition t with ${}^\bullet t \in C_2$ and $\sum_{q \in t^\bullet \cap (C_2 \cup C_3)} F(t, q) \geq 2$.*

Proof. "*1. ⇒ 2.*": By definition, we can find an infinite sequence of enabled Parikh vectors Φ_1, Φ_2, ... such that $\Delta(\Phi_i)(p) < \Delta(\Phi_{i+1})(p)$, $i \in \mathbb{N}$. Consider the induced infinite sequence of vectors such that the i-th vector is $(\Phi_i, \mu_0 + \Delta(\Phi_i)) \in \mathbb{N}_0^{m+n}$. It is easy to see that this sequence contains an infinite non-decreasing subsequence $(\Phi_{i_j}, \mu_0 + \Delta(\Phi_{i_j}))$, $i_j < i_{j+1}$, $j \in \mathbb{N}$, i. e., $(\Phi_{i_j}, \mu_0 + \Delta(\Phi_{i_j})) \leq (\Phi_{i_{j+1}}, \mu_0 + \Delta(\Phi_{i_{j+1}}))$ (see, e. g., Lemma 4.1. of [KM69]). Let $\Phi := \Phi_{i_1}$ and $\vartheta := \Phi_{i_2} - \Phi_{i_1}$. Then, Φ and $\Phi + \vartheta$ are enabled at μ_0, and ϑ is a positive loop for p. Therefore, we can apply Lemma 3.5 to Φ, ϑ and some firing sequence σ having Parikh image Φ. Let $\sigma_1, \ldots, \sigma_{k+1}$ and $\vartheta_1, \ldots, \vartheta_k$ be defined as in the lemma. Then we have $\Delta(\vartheta_i)(p) > 0$ for some $i \in [k]$, and ϑ_i is enabled at the marking μ reached by $\sigma_1 \cdots \sigma_i$, concluding the proof.

"*1. ⇒ 4.*": We continue where the proof for "*1. ⇒ 2.*" ended. Let τ be a transition sequence with $\Psi(\tau) = \vartheta_i$ enabled at μ. Further, let C_2^τ be the unique top component of $\mathcal{P}[\Psi(\tau)]$, and C_4^τ the SCC of $\mathcal{P}[\Psi(\tau)]$ containing p. Since τ is enabled at μ, by Theorem 3.2 there are places p_1 and p_2 such that p_1 is marked at μ_0, $\mathcal{P}$ contains a path from p_1 to p_2, p_2 is contained in C_2^τ, and $\mu(p_2) > 0$. Define C_1 as the SCC of $\mathcal{P}$ containing p_1.

Since τ is a positive loop, C_2^τ contains a transition. If there is a transition t of C_2^τ such that $\sum_{p' \in t^\bullet \cap C_2^\tau} F(t, p') \geq 2$, then simply define $C_3^\tau := C_2^\tau$. Now, assume that such a transition doesn't exist. Then, we have $C_4^\tau \neq C_2^\tau$ since the total number of tokens in C_2^τ cannot increase by firing τ. In particular, there is a path $(p_2', t, p_3, \ldots, p)$ from some place $p_2' \in C_2^\tau$ to $p \in C_4^\tau$ where $p_3 \notin C_2^\tau$. Let C_3^τ be the SCC of $\mathcal{P}[\Psi(\tau)]$ containing p_3.

In any case, if $t^\bullet \cap C_2^\tau = \emptyset$, then τ decreases the number of tokens at C_2^τ, a contradiction to τ being a loop. Therefore, $t^\bullet \cap C_2^\tau \neq \emptyset$, and we obtain $\sum_{p' \in t^\bullet \cap (C_2^\tau \cup C_3^\tau)} F(t, p') \geq 2$. Now, let C_i for $i \in [2, 4]$ be the SCC of $\mathcal{P}$ containing C_i^τ, and observe that $C_1, \ldots, C_4$ satisfy the properties (*a*)–(*d*).

"*2. ⇒ 3.*": Let σ be a firing sequence leading to a marking at which τ is enabled. Then, p is unbounded on the ω-firing sequence $\sigma \cdot \tau^\omega$.

"*3.* ⇒ *1.*": This follows immediately from the definitions.

"*4.* ⇒ *1.*": To mark $^\bullet t$, we first fire along a path starting at a marked place of C_1 and ending at $^\bullet t \in C_2$. Then we fire $k \in \mathbb{N}$ times along a cycle containing t. This increases the total number of tokens within C_3 by at least k. These tokens can then be transferred to p. □

The most simple cf-PN, where, for some unbounded place, C_1, C_2, C_3, and C_4 are different components, is illustrated in (*a*) of Figure 3.4. As already mentioned, a Petri net is unbounded if and only if there are a reachable marking μ and a positive loop τ enabled at μ (see [KM69]). By Lemma 3.17, an analogue observation can be made for single places of a cf-PN. In general, however, the latter is not true. In Petri net (*b*) of Figure 3.4, place p is unbounded but there is no positive loop τ for p which is enabled at some reachable marking. We further note that (in contrast to, e. g., persistent Petri nets, see [LR78]) this concept doesn't hold for sets of places of cf-PNs, i. e., a set $S \subseteq P$ of places of a cf-PN is not necessarily simultaneously ω-unbounded if it is simultaneously unbounded. An example is given in Petri net (*c*) of Figure 3.4.

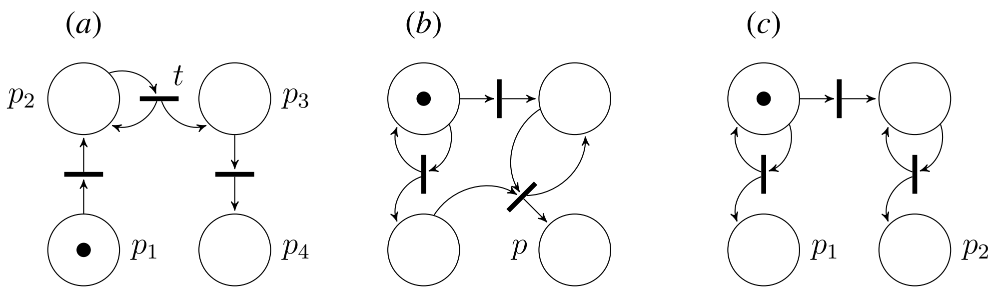


Figure 3.4: The cf-PN (*a*) illustrates Lemma 3.17, where p_4 is unbounded since $C_1 := \{p_1\}$, $C_2 := \{p_2, t\}$, $C_3 := \{p_3\}$, and $C_4 := \{p_4\}$ satisfy the properties of the lemma. In Petri net (*b*), place p is unbounded but not ω-unbounded. In cf-PN (*c*), $\{p_1, p_2\}$ is simultaneously unbounded but not simultaneously ω-unbounded.

We can use the characterization provided by Lemma 3.17 to give efficient algorithms for certain boundedness problems.

Theorem 3.18. *Given a cf-PN $\mathcal{P} = (P, T, F, \mu_0)$ and a set $R \subseteq P$, we can find in linear time all places that are (ω-)unbounded.*

Proof. Using Tarjan's modified depth-first search [Tar72], we find the strongly connected components of $\mathcal{P}$. Then, we use three modified DFSs in the condensation of $\mathcal{P}$ in the following way to find all (ω-) unbounded places. The first DFS finds all C_1-candidates, i. e., SCCs containing a marked place. The second DFS determines all C_2-candidates, i. e., SCCs reachable from a C_1-candidate and containing a transition t with $\sum_{q \in t^\bullet} F(t, q) \geq 2$. For each such transition t contained in a C_2-candidate C, we consider the places $p \in t^\bullet$. If $\sum_{q \in t^\bullet \cap C} F(t, q) \geq 2$, then C is not only a C_2-candidate but also a C_3-candidate. Furthermore, each SCC $C' \neq C$ containing a place of $t^\bullet$ is a C_3-candidate. The last DFS finds all C_4-components, i. e., all SCCs reachable from C_3-candidates. The C_4-components found by this scheme are all components for which appropriates components C_1, C_2, C_3 exists such that they together satisfy the properties of Lemma 3.17. (However, not for every C_1-candidate exists a suitable C_4-component.) By the same lemma, exactly the places of C_4-components are (ω-)unbounded. Note that all these steps can be performed in linear time. □

As a corollary, we can decide the boundedness problem for cf-PNs in linear time.

Corollary 3.19. *The boundedness problem for cf-PNs is decidable in linear time.*

Proof. We use Theorem 3.18 to check if there is no unbounded place. □

Note that it can be checked in linear time if the input encodes a cf-PN. Hence, the above corollary indeed holds for the decision problem variation of the boundedness problem and not only for the promise problem variation. A similar theorem can be shown, if we demand that a certain set R should be unmarked.

Theorem 3.20. *Given a cf-PN $\mathcal{P} = (P, T, F, \mu_0)$ and a set $R \subseteq P$, we can find in linear time all places that are unbounded with unmarked R.*

Proof. If the maximum trap Q of R is marked, then no place is unbounded with unmarked R. Hence, assume that Q is empty. Let $\mathcal{P}'$ result from $\mathcal{P}$ by removing from $\mathcal{P}$ all transitions incident to the maximum trap Q of R. Let U (U', resp.) denote the set of all places that are unbounded with empty R in $\mathcal{P}$ (that are unbounded in $\mathcal{P}'$, resp.). We will show that $U = U'$.

Let $p \in U$. Then, there is, for each $k \in \mathbb{N}$, a firing sequence σ of $\mathcal{P}$ leading to a marking μ such that $\mu(p) \geq k$ and $\mu(r) = 0$ for all $r \in R$. The sequence σ cannot contain a transition incident to Q since otherwise Q would be marked at μ. Therefore, σ is a firing sequence in $\mathcal{P}'$, which implies $p \in U'$.

Now, let $p \in U'$. Then, there is, for each $k \in \mathbb{N}$, a firing sequence σ of $\mathcal{P}'$ leading to a marking μ such that $\mu(p) \geq k$. Furthermore, we observe Q is empty at μ since Q is empty at μ_0 and σ doesn't contain transitions incident to Q. μ is also reachable in $\mathcal{P}$ by σ. By Lemma 3.8, there is a marking $\mu' \geq \mu$ reachable from μ in $\mathcal{P}$ such that R is empty at μ', which implies $p \in U$.

Using Lemma 3.6 and Theorem 3.18, finding Q, checking if Q is empty, and computing $\mathcal{P}'$ and U' can be done in linear time. □

We remark that it was shown in [Kuč96] that boundedness of BPPs can be decided in polynomial time. Our results imply a linear time algorithm for all BPPs in *standard form* (see [Chr+93]).

Theorem 3.21. *Given a cf-PN $\mathcal{P} = (P, T, F, \mu_0)$ and a place $p \in P$, we can decide in linear time if p is persistently unbounded.*

Proof. We use the terminology of Lemma 3.17 and Theorem 3.18. Let C_4 denote the SCC containing p. For the cf-PN $(P, T, F, \vec{1})$ whose initial marking has one token at each place, we determine the set $R \subseteq P$ of all places contained in SCCs C_1 for which SCCs C_2 and C_3 exist such that C_1, C_2, C_3, and C_4 satisfy the properties mentioned in Lemma 3.17. To find these SCCs in linear time, we use a similar scheme as in the proof of Theorem 3.18. The only difference is that we filter out all C_1-candidates for which no suitable C_4-component exists. This can easily be done by first filtering out all C_3-candidates from which no C_4-component can be reached, then filtering out all C_2-candidates for which no suitable remaining C_3-candidate exists, and then filtering out all C_1-candidates from which no remaining C_2-candidate can be reached.

By Lemma 3.17, p is unbounded at each marking μ such that there is a place $r \in R$ with $\mu(r) > 0$. Therefore, p is not persistently unbounded if and only if there is a marking reachable from μ_0 where no place of R is marked. By Lemma 3.8, we only have to determine if the maximum trap of R is marked. By Lemma 3.6, this can be done in linear time. □

A simple characterization of universally unbounded places is also possible.

Lemma 3.22. *Let $\mathcal{P} = (P, T, F, \mu_0)$ be a cf-PN. A place $p \in P$ is universally unbounded if and only if*

(a) *p is persistently unbounded,*

(b) *$F(t, p) \geq 2$ holds for each transition $t \in p^\bullet$, and*

(c) *each cycle of $\mathcal{P}$ that can be marked by some firing sequence contains a transition $t \in {}^\bullet p$.*

Proof. "$\Rightarrow$": We prove the contraposition, i. e., we show that p is not universally unbounded if (a), (b) or (c) doesn't hold. If p is not persistently unbounded, then, by definition, p is not universally unbounded. Hence, assume that (a) holds. Now assume that (b) doesn't hold for $\mathcal{P}$, i. e., there is a transition t such that $p = {}^\bullet t$ and $F(t, p) \leq 1$. First consider the case $F(t, p) = 1$. Since p is persistently unbounded, there is a firing sequence σ leading to μ such that $\mu(p) \geq 1$. We observe that p is not unbounded on σt^ω. Next consider the case $F(t, p) = 0$. Then we can immediately remove all tokens from p as soon as some other transition deposits tokens at p. Hence, there is an ω-firing sequence on which p is not unbounded. (Note that, by (a), at least one ω-firing sequence must exist.) If (c) does not hold, then we can mark that cycle and fire along this cycle infinitely often without increasing the number of tokens at p.

"$\Leftarrow$": Since, by (a), p is persistently unbounded, we only have to show that p is unbounded on each ω-firing sequence σ. Let μ be a marking reached by some finite prefix $\sigma_{[..i]}$ of σ. Then, there is a subsequence $\sigma_{[i+1..j]}$ such that the induced net $\mathcal{P}[\Psi(\sigma_{[i+1..j]})]$ contains a cycle. (The length of this subsequence can depend on the number of tokens at μ. Also note that such a subsequence containing transitions of a cycle must exist since each transition has an incoming edge.) By (b), no transition can decrease the number of tokens at p. Furthermore, by (b) and (c), the cycle, and therefore $\sigma_{[i+1..j]}$, contains a transition that increases the number of tokens at p. Since this argument holds for any such μ, we can partition any ω-firing sequence into infinitely many segments such that each segment increases the number of tokens at p. Therefore, p is unbounded on each ω-firing sequence. □

Using this characterization, we can show the following theorem.

Theorem 3.23. *Given a cf-PN $\mathcal{P} = (P, T, F, \mu_0)$ and a place $p \in P$, we can decide in linear time if p is universally unbounded.*

Proof. Consider the characterization of the universally unboundedness property given at Lemma 3.22. (a) can be checked in linear time using the algorithm described at Theorem 3.21. (b) can trivially be checked in linear time. Now we show, how we can test for (c). Let $\mathcal{P}'$ be the cf-PN resulting from $\mathcal{P}$ by removing all transitions $t \in {}^\bullet p$. A cycle of $\mathcal{P}$ doesn't contain a transition $t \in {}^\bullet p$ if and only if it is also a cycle of $\mathcal{P}'$. These are the potentially problematic cycles. Property (c) is satisfied if and only if no such cycle can be marked. The set S of places contained in these cycles can be determined in linear time by computing the SCCs of $\mathcal{P}'$. Next, we must check if one of these places can be marked in $\mathcal{P}$. To this end, we find all SCCs of $\mathcal{P}$ and color all SCCs C_2 red for which a marked SCC C_1 with $C_1 \geq C_2$ exists. This can be done in linear time, e. g., by computing all SCCs and then using a DFS in the condensation. Now, (c) holds if and only if no place of S is contained in a red SCC. □

We remark that universally unboundedness can easily be checked in cf-PNs because only one place determines if a transition is enabled. Furthermore, this leads to an even stronger property of universally unbounded places of cf-PNs: Their token numbers can never decrease. This doesn't hold in general. Figure 3.5 illustrates such an example. A natural and open question in this context is: Given a Petri net with a universally unbounded place p, which lower bounds for the displacements of firing sequences at p can be given?

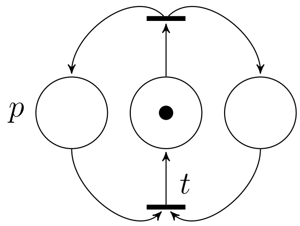


Figure 3.5: p is universally unbounded but $\Delta(t)(p) < 0$.

3.3.2 Simultaneously unboundedness and the covering problem

In this subsection, we consider the covering problem as well as boundedness problems where we ask if many places are simultaneously (ω-)unbounded. First, we formally define the simultaneously-unboundedness problem (SU), and the simultaneously-ω-unboundedness problem (SIU).

Definition 3.24.

- Problem SU of a class C of Petri nets: Given a Petri net $\mathcal{P} = (P, T, F, \mu_0) \in C$ and a subset $S \subseteq P$ of places, is S simultaneously unbounded?
- Problem SIU of a class C of Petri nets: Given a Petri net $\mathcal{P} = (P, T, F, \mu_0) \in C$ and a subset $S \subseteq P$ of places, is S simultaneously ω-unbounded?

Next, we describe some form of canonical firing sequences for markings reachable in cf-PNs. This sequence is reminiscent of that described by Yen in Lemma 2 of [Yen97]. However, we need a small bound on the length of the "backbone" $\bar{\xi}$. Later in that paper, a suitable sequence is constructed but its properties are not explicitly stated in form of a Lemma such that we provide an alternative canonical sequence together with a short proof for the sake of completeness. We also note that that, whenever this lemma is used, one could also use the similarly strong but somewhat more general statement of Corollary 6.17.

Lemma 3.25. *There is a constant $c \in \mathbb{N}$ such that, for each cf-PN $\mathcal{P} = (P,\ T,\ F,\ \mu_0)$ and each reachable marking μ of $\mathcal{P}$, there are $\ell \in \mathbb{N}$ and transition sequences ξ, $\bar{\xi}$, α_1, …, $\alpha_{\ell+1}$, τ_1, …, τ_ℓ with the following properties:*

(a) $\xi = \alpha_1 \cdot \tau_1 \cdot \alpha_2 \cdot \tau_2 \cdots \tau_\ell \cdot \alpha_{\ell+1}$ *is a firing sequence leading from μ_0 to μ,*

(b) $\bar{\xi} = \alpha_1 \cdot \alpha_2 \cdots \alpha_{\ell+1}$ *is a firing sequence of length at most $\|\mu_0\|_1 \cdot n(nW)^n + m \cdot (1 + (n + m)W)^{n+m} \leq c^{\text{size}(\mathcal{P})}$,* (*c*) *each* τ_i, $i \in [\ell]$, *is a positive loop of length at most* $(1 + (n + m)W)^{n+m}$, *and*

(*d*) *at most* $\min\{n, m\} + 1$ *of the sequences* α_i, $i \in [\ell + 1]$, *are nonempty.*

Proof. Let $\vartheta \leq \Psi(\sigma)$ be a loop with the maximum number of transitions such that the Parikh vector $\Phi := \Psi(\sigma) - \vartheta$ satisfies $\mu_0 + \Delta(\Phi) \geq \vec{0}$. We show that $\mathcal{P}[\Phi]$ is cycle-free. Assume for the sake of contradiction that $\mathcal{P}[\Phi]$ contains a cycle. Let ϑ' be the loop that contains each transition of this cycle exactly once, and let $\Phi' := \Phi - \vartheta'$. If $\mu_0 + \Delta(\Phi') \geq \vec{0}$, then we have found a contradiction since $\vartheta + \vartheta'$ is a larger loop than ϑ. Otherwise, there must be a place p with $(\mu_0 + \Delta(\Phi'))(p) < 0$. Hence, there is a transition $t \in \Phi'$ with $p = {}^\bullet t$. Since $(\mu_0 + \Delta(\Phi') + \Delta(\vartheta'))(p) = (\mu_0 + \Delta(\Phi))(p) \geq 0$, we find $\Delta(\vartheta')(p) > 0$. We decrease the entry of t in Φ' by 1 and increase it in ϑ' by 1. Now, we can do the same case analysis as before. Eventually, however, this process must stop with a contradiction since, with each iteration, Φ' gets smaller.

How many tokens can be produced by a firing sequence of $\mathcal{P}[\Phi]$ when the initial marking contains only one token? Every time a transition is fired, a token is consumed from some SCC C and in total at most nW tokens are produced within SCCs $C' \leq C$, $C' \neq C$. Since there are at most n SCCs containing places, such a firing sequence can produce at most $(nW)^n$ tokens at each place. Hence, the total number of occurrences of transitions that consume tokens from a specific place is at most $(nW)^n$. Therefore, the length of such a firing sequence is bounded by $n(nW)^n$. Since the initial marking contains $\|\mu_0\|_1$ tokens, this implies $\|\Phi\|_1 \leq \|\mu_0\|_1 \cdot n(nW)^n$.

We now apply Lemma 2.18 to (P, T, F) and ϑ, and obtain, for some k (and an appropriate indexing of the elements of $\mathcal{H}(P, T, F)$), coefficients $a_i \in \mathbb{N}$ and loops $\vartheta_i \in \mathcal{H}(P, T, F)$, $i \in [k]$, such that $\vartheta = a_1\vartheta_1 + \ldots + a_k\vartheta_k$, and, for all $i \in [k]$, ϑ_i satisfies $\|\vartheta_i\|_1 \leq (1 + (n + m)W)^{n+m}$ and cannot be decomposed into nontrivial loops any further. Note that, by this and Lemma 3.4, $\mathcal{P}[\vartheta_i]$, $i \in [k]$, has exactly one top component.

We choose $r \leq m$ loops $\vartheta_{i_1}, \ldots, \vartheta_{i_r}$ such that $\Phi^* := \Phi + \vartheta_{i_1} + \ldots + \vartheta_{i_r}$ satisfies $\mathcal{P}[\Phi^*] = \mathcal{P}[\Psi(\sigma)]$. By this and since $\mu_0 + \Delta(\Phi^*) \geq \vec{0}$ is implied by $\Delta(\Phi^*) \geq \Delta(\Phi)$ and $\mu_0 + \Delta(\Phi) \geq \vec{0}$, Theorem 3.2 implies that Φ^* is enabled at μ_0. Let $\bar{\xi}$ be some firing sequence with $\Psi(\bar{\xi}) = \Phi^*$. From the discussion above, we find $|\bar{\xi}| \leq \|\Phi\|_1 + m \cdot \max_{j\in[r]} \left\|\vartheta_{i_j}\right\|_1 \leq \|\mu_0\|_1 \cdot n(nW)^n + m \cdot (1 + (n + m)W)^{n+m}$.

By Theorem 3.2 and by the facts that ϑ_i has exactly one top component and $\mathcal{P}[\vartheta_i]$, $i \in [k]$, is a subnet of $\mathcal{P}[\Psi(\bar{\xi})]$, we can find, for each $i \in [k]$, transition sequences α, β, and τ such that $\bar{\xi} = \alpha \cdot \beta$, $\Psi(\tau) = \vartheta_i$, and $\alpha \cdot \tau \cdot \beta$ is a firing sequence in $\mathcal{P}$. Hence, by splitting $\bar{\xi}$ at appropriate positions (and by appropriately numbering the loops), we obtain the lemma. Note that we must indeed split at no more than $\min\{n, m\}$ positions since we can find a set of at most $\min\{n, m\}$ places of $\mathcal{P}[\Psi(\bar{\xi})]$ such that each top component of the loops contains a place of this set. Furthermore, we can discard zero-loops. □

It turns out that the introduction of some kind of implicit "communication" in form of the concept of simultaneousness is enough to make the problems SU and SIU **NP**-complete. Furthermore, we find that, like the reachability problem, the covering problem is also **NP**-complete.

Theorem 3.26. *The problems SU and SIU of cf-PNs, and the reachability and covering problems of cf-PNs are* **NP***-complete, even if restricted to cf-PNs* $\mathcal{P} = (P, T, F, \mu_0)$ *with* $|t^\bullet| = 1$ *and* $F(t, t^\bullet) \leq 2$ *for all* $t \in T$.

Proof. We first show the **NP**-hardness of SU and SIU by giving a polynomial time reduction from 3-SAT to both SU and SIU. Given a formula in 3-CNF over the variables $x_1, \ldots, x_k$ and clauses C_1,

..., C_ℓ, we construct a cf-PN such that a certain subset $S = \{c_i \mid i \in [\ell]\}$ of places is simultaneously (ω-)unbounded if and only if the formula can be satisfied. An example is illustrated in Figure 3.6. Note that the cf-PN produced by this reduction satisfies the additional constraints ("even if...") of the lemma.

Since the reachability problem of cf-PNs is **NP**-complete [Esp97], we only need to show **NP**-hardness of this problem under the additional constraints. To this end, note that the marking $\sum_{i\in[\ell]} C_i$ is reachable in the wipe-extension of the above Petri net if and only if the formula can be satisfied.

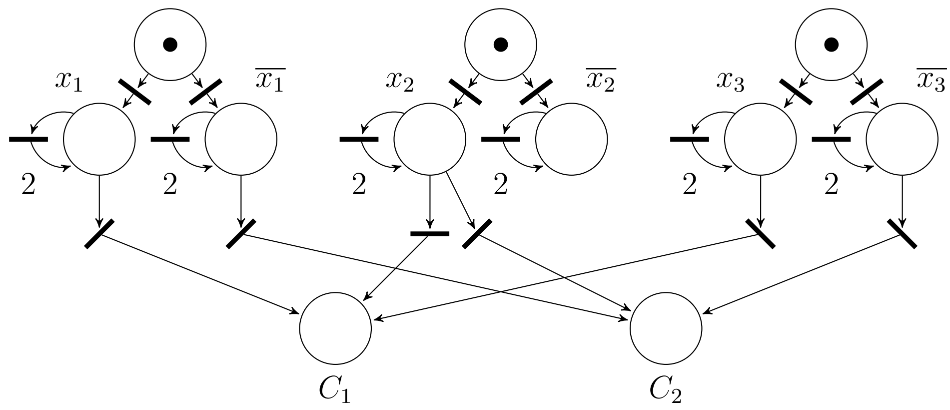


Figure 3.6: The formula $C_1 \wedge C_2 :=(x_1 \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3})$ can be satisfied if and only if $\{C_1, C_2\}$ is simultaneously (ω-)unbounded.

Next, we show that the covering problem is in **NP** by reducing it to the reachability problem. To this end, we observe that the following are equivalent:

- μ can be covered in $\mathcal{P}$,
- μ can be covered in the wipe-extension $\widehat{\mathcal{P}}$, and
- μ is reachable in the wipe-extension $\widehat{\mathcal{P}}$.

Now, we show that SU is in **NP** and that the covering problem is **NP**-hard, even under the additional restrictions of the lemma, by providing an appropriate polynomial time reduction from SU to the covering problem. Let $\mathcal{P}$ be part of the input for SU, and W be the largest edge multiplicity of $\mathcal{P}$'s wipe-extension $\widehat{\mathcal{P}} = (P, \widehat{T}, \widehat{F}, \mu_0)$. Let $\mu :\triangleq \sum_{p\in S}(\mu_0(p) + c^{\text{size}(\widehat{\mathcal{P}})}W + 1) \cdot p$, where c is the constant of Lemma 3.25. Note that μ has polynomial encoding size (in particular, size($\widehat{\mathcal{P}}$) is polynomial in size($\mathcal{P}$)).

Assume that S is simultaneously unbounded in $\mathcal{P}$. Then μ can be covered in $\widehat{\mathcal{P}}$. Now, assume that μ can be covered in $\widehat{\mathcal{P}}$, and is therefore reachable in $\widehat{\mathcal{P}}$. In accordance with Lemma 3.25, let $\xi = \alpha_1 \cdot \tau_1 \cdot \alpha_2 \cdot \tau_2 \cdots \tau_\ell \cdot \alpha_{\ell+1}$ be a firing sequence of $\widehat{\mathcal{P}}$ leading from μ_0 to μ. Then, $|\bar{\xi}| \leq c^{\text{size}(\widehat{\mathcal{P}})}$ which implies that each prefix of $\bar{\xi}$ can increase the number of tokens at a place by at most $c^{\text{size}(\widehat{\mathcal{P}})}W$. Hence, each place $p \in S$ has some i such that $\Delta(\tau_i)(p) > 0$. Therefore, for each $k \in \mathbb{N}$, the marking $\widehat{\nu}$ of $\widehat{\mathcal{P}}$ reached by the firing sequence $\alpha_1 \cdot \tau_1^k \cdot \alpha_2 \cdot \tau_2^k \cdots \tau_\ell^k \cdot \alpha_{\ell+1}$ satisfies $\widehat{\nu}(p) \geq (\mu_0 + \Delta(\bar{\xi}))(p) + k \geq k$ for all $p \in S$ (since $\mu_0 + \Delta(\bar{\xi}) \geq \vec{0}$). By removing all occurrences of the new transitions (which can only remove tokens from the net, and are part of $\widehat{\mathcal{P}}$ but not part of $\mathcal{P}$) from this firing sequence, we obtain a firing sequence of $\mathcal{P}$ leading to a marking ν of $\mathcal{P}$ with $\nu(p) \geq k$ for all $p \in S$. Therefore, S is simultaneously unbounded in $\mathcal{P}$. Note that the covering problem is **NP**-hard even if we only allow inputs satisfying the additional constraints of this lemma. The reason is that the reduction from 3-SAT to SU given above produces cf-PNs $\mathcal{P}$ that satisfy these constraints, implying that also the corresponding wipe-extensions $\widehat{\mathcal{P}}$ satisfy them.

It remains to be shown that SIU is in **NP**. Unfortunately, a reduction in the same fashion as shown above fails. Therefore, we use another approach. Our goal is to give a nondeterministic procedure accepting if and only if the given set $S \subseteq P$ is simultaneously ω-unbounded. Suppose the latter is the case. We use a similar reasoning as in the proof of Lemma 3.17. From the definition, we conclude that there are infinitely many firing sequences σ_k, $k \in \mathbb{N}$, such that σ_k is a prefix of σ_{k+1} and $\Delta(\sigma_k)(p) \geq k$ for all $p \in S$ and $k \in \mathbb{N}$. Consider the induced infinite sequence of vectors $\mu_0 + \Delta(\sigma_k)$, $k \in \mathbb{N}$. As before, we can pick an infinite subsequence with indices $i_1 < i_2 < \ldots$ such that $\mu_0 + \Delta(\sigma_{i_k}) \leq \mu_0 + \Delta(\sigma_{i_{k+1}})$ for all $k \in \mathbb{N}$. However, we can pick these indices in such a way that additionally $(\mu_0 + \Delta(\sigma_{i_k}))(p) < (\mu_0 + \Delta(\sigma_{i_{k+1}}))(p)$ for all $p \in S$ and $k \in \mathbb{N}$ holds. Therefore, $\Phi := \Psi(\sigma_{i_2}) - \Psi(\sigma_{i_1})$ is a positive loop for all $p \in S$, enabled at the marking μ reached by σ_{i_1}, i. e., $\mu_0 \xrightarrow{\sigma_{i_1}} \mu \xrightarrow{\Phi}$.

Consider the decomposition $\Phi = a_1\Phi_1 + \ldots + a_\ell\Phi_\ell$ obtained by applying Lemma 2.18 to Φ. For each $p \in S$, there must be some Φ_i with $\Delta(\Phi_i)(p) > 0$. Hence, the Parikh vector $\Phi^* := \Phi_1 + \ldots + \Phi_\ell$ is a positive loop for all $p \in S$. Note that Φ^* has polynomial encoding size.

Since $\mathcal{P}[\Phi] = \mathcal{P}[\Phi^*]$, Theorem 3.2 implies that Φ^* is enabled at exactly those markings at which Φ is enabled. In particular, a loop is enabled at μ if and only if it is enabled at $\mu^* \in \{0,1\}^n$ where $\mu^*(p) = 1$ if and only if $\mu(p) \geq 1$. Note that μ^* has polynomial encoding size and satisfies $\mu^* \leq \mu$.

Now, we can describe the nondeterministic procedure which accepts if and only if S is simultaneously unbounded on some ω-firing sequence: We guess μ^* and Φ^* in polynomial time and check nondeterministically and in polynomial time if μ^* can be covered and if Φ^* is enabled at μ^*. This completes the proof. □

Note that a further restriction to $F(t, t^\bullet) = 1$ leads to S-systems, a subclass of cf-PNs, which are always bounded. Furthermore, we can decide in linear time if the set P of all places of a cf-PN is simultaneously (ω-)unbounded. This is the case if and only if all top components C contain a marked place and a transition t with $\sum_{p \in t^\bullet \cap C} F(t,p) \geq 2$. Hence, the problems SU and SIU for cf-PNs are hard only if the input set S satisfies $1 < |S| < |P|$. We remark that using the existence of 0-1-markings which enable loops is reminiscent of a similar argument used by Howell and Rosier [HR88] for conflict-free Petri nets.

3.4 Liveness problems

Many different notions of liveness can be found in literature. We are mainly interested in the following.

Definition 3.27. Let $\mathcal{P} = (P, T, F, \mu_0)$ be a Petri net. A transition $t \in T$ is

- L_0-*live* (dead) if there is no firing sequence containing t,
- L_1-*live* (potentially fireable) if it isn't dead,
- L_2-*live* (arbitrarily often fireable) if, for each $k \in \mathbb{N}$, there is a firing sequence containing t at least k times,

- L_3-*live* (infinitely often fireable) if there is an ω-firing sequence containing t infinitely often,
- L_4-*live* (live) if t is L_1-live at each reachable marking,
- L_5-*live* (infinitely often fired) if each firing sequence is prefix of an ω-firing sequence and each ω-firing sequence contains t infinitely often.

For sets of places and the whole Petri net, we have the following definitions.

- A subset $T' \subseteq T$ of transitions is L_i-*live*, $i \in [0, 5]$, if all transitions of S are L_i-live, and
- $\mathcal{P}$ is L_i-*live*, $i \in [0, 5]$, if T is L_i-live.

The notions of L_0, ..., L_4-liveness are referred to in [Mur89]. L_5-liveness is to our knowledge a new concept, corresponding to our new notion of universally unboundedness. Notice, that L_i-liveness implies L_j-liveness, where $5 \geq i \geq j \geq 1$. Using the results of Section 3.3, we can efficiently solve many decision problems involving these notions of liveness.

Theorem 3.28. *Given a cf-PN $\mathcal{P} = (P, T, F, \mu_0)$, and $i \in [0, 3]$, we can find in linear time all transitions that are L_i-live and decide if $\mathcal{P}$ is L_i-live.*

Proof. Consider the Parikh extension $\mathcal{P}^\Psi = (P^\Psi, T, F^\Psi, \mu_0^\Psi)$ of $\mathcal{P}$ (see Definition 3.1). A transition t is not L_0-live iff t is L_1-live iff for the SCC C containing p_t^Ψ there is a marked SCC C' such that $C' \geq C$ (see Theorem 3.2). Hence, to find all L_i-live transitions for $i \in \{0, 1\}$ in linear time, we compute $\mathcal{P}^\Psi$, determine the SCCs of $\mathcal{P}^\Psi$, and investigate them in a similar fashion as in Theorem 3.18. For $i \in [2, 3]$, notice that a transition t is L_2-live iff p_t^Ψ is unbounded iff p_t^Ψ is ω-unbounded (see Lemma 3.17) iff t_t is L_3-live. Hence, we simply apply the algorithm of Theorem 3.18 to $\mathcal{P}^\Psi$ to determine all unbounded places p_t^Ψ corresponding to L_2/L_3-live transitions t. This all can be done in linear time. □

Theorem 3.29. *Given a cf-PN $\mathcal{P} = (P, T, F, \mu_0)$, a transition $t \in T$, and $i \in [4, 5]$, we can decide in linear time if t is L_i-live, and in quadratic time if $\mathcal{P}$ is L_i-live.*

Proof. As before, consider the Parikh extension $\mathcal{P}^\Psi = (P^\Psi, T, F^\Psi, \mu_0^\Psi)$ of $\mathcal{P}$. It is easy to see that a transition t is L_4-live iff p_t^Ψ is persistently unbounded. Furthermore, t is L_5-live iff p_t^Ψ is universally unbounded. Hence, we simply apply the algorithms of Theorem 3.21 and Theorem 3.23 to $\mathcal{P}^\Psi$ and p_t^Ψ. □

Note that, for $i = 4$, this theorem makes a statement about the liveness problem of cf-PNs, which asks if a given cf-PN is live. Mayr [May97] showed that (the BPP-analogon of) L_4-liveness is decidable in polynomial time for BPPs. Our results imply a quadratic time algorithm for all BPPs in standard form (see [Chr+93]). In the same paper, other interesting notions of liveness were investigated, namely the partial deadlock reachability problem and the partial livelock reachability problem. For both problems polynomial time algorithms were proposed for cf-PNs and PA-processes in general. Using our results, linear time algorithms can be given for cf-PNs. These imply linear time algorithms also for BPPs that are in standard form.

Theorem 3.30 (partial deadlock reachability)**.** *Given a cf-PN $\mathcal{P} = (P, T, F, \mu_0)$ and a set $T' \subseteq T$ of transitions, we can decide in linear time if there is a reachable marking μ at which no transition of T' is enabled.*

Proof. By Lemma 3.9, we can check in linear time if there is a marking at which $R := \bigcup_{t \in T'} {}^{\bullet}t$ is empty. □

Theorem 3.31 (partial livelock reachability)**.** *Given a cf-PN $\mathcal{P} = (P, T, F, \mu_0)$ and a set $T' \subseteq T$ of transitions, we can decide in linear time if there is a reachable marking μ such that no marking reachable from μ enables a transition $t \in T'$.*

Proof. We introduce a counting place p and an edge from each transition $t \in T'$ to p. A marking μ as defined in the lemma exists if and only if p is not persistently unbounded. By Theorem 3.21, this can be decided in linear time. □

3.5 Related problems for CFGs and CFCGs

In this section, we apply our results to cf-PNs corresponding to context-free (commutative) grammars. This allows us to efficiently solve many problems involving these kind of grammars. Esparza et al. [Esp+00] provided a generic algorithm deciding in quadratic time if the language of a given context-free grammar is finite. In the same paper, they mentioned that a careful implementation of the algorithm in [HU79] could possibly achieve linear time. Using our results, we can decide a generalization of the finiteness problem of commutative and non-commutative context-free grammars in linear time.

Theorem 3.32. *Given a CFG (CFCG, resp.) $G = (V_N, V_T, P, s)$ and a set $U \subseteq V_T$, we can decide in linear time if $L(G)[U]$ is finite, where $L(G)[U]$ denotes the set of all words $x \in U^*$ ($x \in U^{\circledast}$, resp.) for which a word $y \in L(G)$ exists such that x is obtained from y by deleting all symbols which are not in U.*

Proof. Let $\mathcal{P} = (S_{V_N} \cup S_{V_T}, T, F, \mu_0)$ be the canonical cf-PN of G, obtained by the construction given in Section 2.5, and let $S_U \subseteq S_{V_T}$ denote the set of places corresponding to U. Then, $L(G)[U]$ contains infinitely many (commutative) words if and only if S_U contains a place that is unbounded with unmarked S_{V_N}. By Theorem 3.20, this can be checked in linear time. □

Clearly, for $U = \emptyset$, this algorithm solves the well-known finiteness problem of CFGs and CFCGs. An advantage of our algorithm compared to the one given in [HU79] for this problem is that it does not require the grammar being in Chomsky normal form. In [Esp+00], the authors also provided linear time algorithms for the emptiness problem and the problem of finding nullable variables of context-free grammars. Our results provide alternative linear time algorithms for these problems as well as for corresponding problems of context-free *commutative* grammars.

Theorem 3.33. *Given a context-free (commutative) grammar $G = (V_N, V_T, P, s)$, we can decide in linear time if $L(G)$ is empty.*

Proof. Let $\mathcal{P} = (S_{V_N} \cup S_{V_T}, T, F, \mu_0)$ be the canonical cf-PN of G, obtained by the construction given in Section 2.5. Then, the following are equivalent:

- $L(G) = \emptyset$,
- each (commutative) word produced by the grammar contains a variable, and
- S_{V_N} cannot be emptied.

By Lemma 3.9, the last condition can be checked in linear time. □

Theorem 3.34. *Given a context-free (commutative) grammar $G = (V_N, V_T, P, s)$, we can find in linear time all nullable variables, i. e., all variables $v \in V_N$ for which the empty word ϵ is in $L(V_N, V_T, P, v)$.*

Proof. Let $\mathcal{P} = (S_{V_N} \cup S_{V_T}, T, F, \mu_0)$ be the canonical cf-PN of G, obtained by the construction given in Section 2.5. Further, let Q be the maximum trap of $S_{V_N} \cup S_{V_T}$, and let $\mu_v :\triangleq p_v$ for $v \in V_N$ denote the marking that contains one token at place p_v and is empty at all other places. Then, $\epsilon \in L(V_N, V_T, P, v)$ if and only if the empty marking is reachable in $(S_{V_N} \cup S_{V_T}, T, F, \mu_v)$. By Lemma 3.8, this is the case if and only if $p_v \notin Q$. By Lemma 3.6, we can compute Q in linear time. After that, we simply collect all variables that correspond to places of $S_{V_N} \setminus Q$. □

4 A framework for classes of general Petri nets

In this chapter, we develop our framework for classes of Petri nets with arbitrary edge multiplicities. We remark that, in principle, this framework is also applicable for some classes of ordinary Petri nets. However, the complexity bounds provided by the framework are usually at least polynomial space. Since many problems of classes of ordinary Petri nets have better complexities, the framework is not helpful in obtaining completeness-results for such classes.

We first define the property of being (simple structurally) f-g-canonical. A class of Petri nets that satisfies this property, enjoys upper bounds for various problems and constructions, where these upper bounds depend on the concrete functions f and g.

Definition 4.1 (f-g-canonical class of Petri nets)**.** A class $\mathcal{C}$ of Petri nets is f-g-*canonical* for two monotonically increasing functions $f, g : \mathbb{N}_0^4 \to \mathbb{N}$ if, for each $\mathcal{P} = (P, T, F, \mu_0) \in \mathcal{C}$ and each marking μ reachable in $\mathcal{P}$, there are some $k \in [0, n(\max(\mu) + uW)]$ and transition sequences ξ, $\bar{\xi}$, $\alpha_1, \ldots, \alpha_{k+1}$, $\tau_1, \ldots, \tau_k$ with the following properties, where $u = f(n, m, W, \max(\mu_0))$:

(*a*) $\xi = \alpha_1 \cdot \tau_1 \cdot \alpha_2 \cdot \tau_2 \cdots \alpha_k \cdot \tau_k \cdot \alpha_{k+1}$ is a firing sequence of length at most $(k+1)u$ leading from μ_0 to μ,

(*b*) $\bar{\xi} = \alpha_1 \cdot \alpha_2 \cdots \alpha_{k+1}$ is a firing sequence of length at most u,

(*c*) at most $g(n, m, W, \max(\mu_0))$ elements of $\{\alpha_1, \ldots, \alpha_{k+1}\}$ are nonempty sequences, and

(*d*) each τ_i, $i \in [k]$, is a positive loop with $|\tau_i| \leq u$ enabled at some marking μ^* with $\max(\mu^*) \leq u$ and $\mu^* \leq \mu_0 + \Delta(\alpha_1 \cdot \alpha_2 \cdots \alpha_i)$.

An f-g-canonical class is

- *structurally* f-g-*canonical* if, for each $(N, \mu_0) \in \mathcal{C}$ and each marking μ of N, the Petri net (N, μ) is also in C, and
- *simple* if it can be determined in polynomial space if a given Petri net $\mathcal{P}$ belongs to $\mathcal{C}$, and if f and g are computable functions.

For a reachable marking μ, we call the corresponding firing sequence ξ the *canonical firing sequence* of μ, and $\bar{\xi}$ the *backbone* of ξ. Note that the upper bound $n(\max(\mu) + uW)$ for the number k of positive loops is just for convenience since this bound is implied by the bounds on the length of the backbone, the lengths of the loops, and by the number of tokens at the end marking μ.

The following theorem provides a sufficient condition for a class to be f-f-canonical.

Theorem 4.2. *Let $\mathcal{C}$ be a class of Petri nets for which there is a monotonically increasing function $\tilde{f} : \mathbb{N}_0^4 \to \mathbb{N}_0$ such that, for each $\mathcal{P} \in \mathcal{C}$ with $n > 0$ and each firing sequence σ of $\mathcal{P}$'s wipe-extension $\widehat{\mathcal{P}}$ leading to the empty marking $\vec{0}$, there is a permutation φ of σ enabled in $\widehat{\mathcal{P}}$ with $\max(\mu_0, \varphi) \leq \tilde{f}(n, m, W, \max(\mu_0))$, where n, m, W refer to $\mathcal{P}$. Then, $\mathcal{C}$ is f-f-canonical, where f is defined by $f(n, m, W, K) = (\tilde{f}(n, m, W, K) + 1)^{2n}$.*

Proof. The proof is sketched in Figure 4.1.

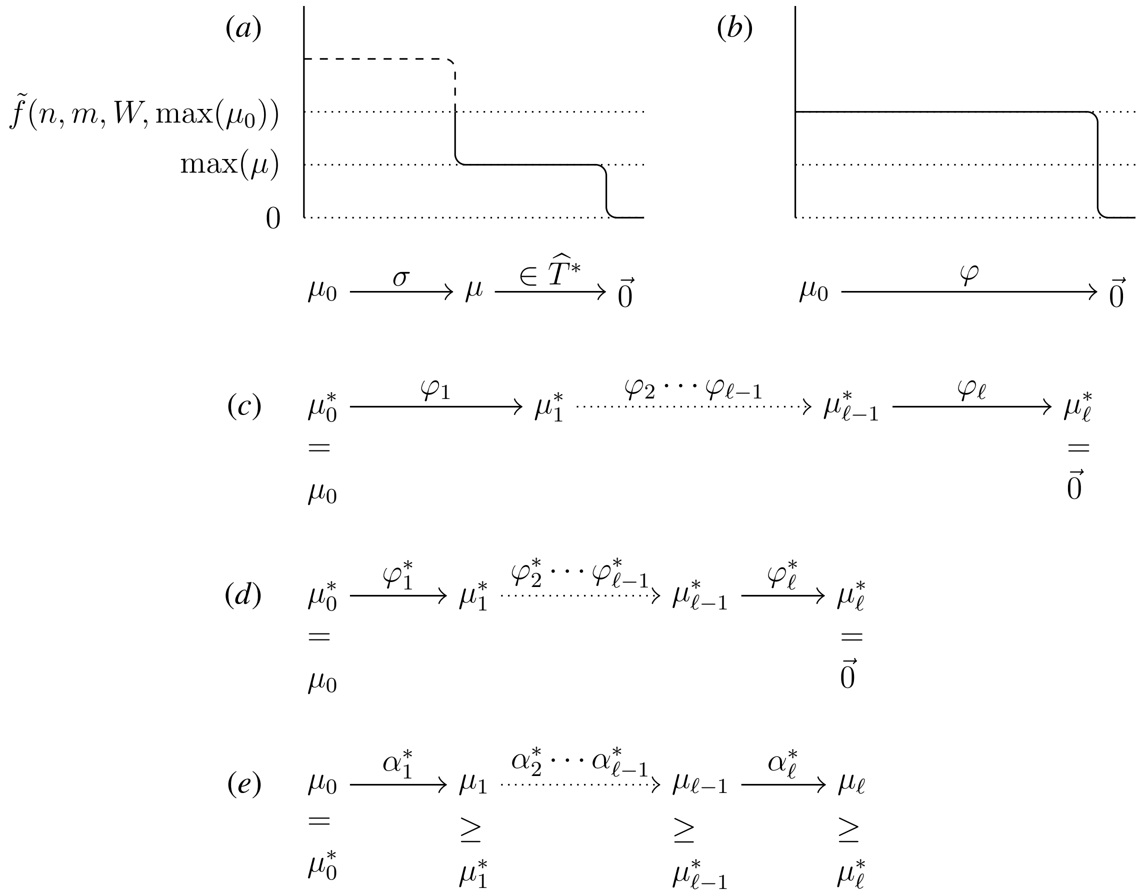


Figure 4.1: A solid curve is an upper bound for the number of tokens, where a dashed curve implies that the number of tokens can become arbitrarily large. (*a*): σ is extended by transitions of $\widehat{T}$, yielding a firing sequence leading to $\vec{0}$ in the wipe-extension. The number of tokens when firing σ can become arbitrarily large. (*b*): The firing sequence is permuted to φ. The number of tokens at the markings obtained when firing φ is bounded. (*c*): φ is split into appropriately chosen subsequences $\varphi_1, \ldots, \varphi_\ell$, each of them witnessing one of these markings. Since these markings are small, the number ℓ of subsequences is small, too. (*d*): For the same reason, long subsequences φ_i contain short zero-loops. These are iteratively cut from the subsequences, resulting in short sequences $\varphi_1^*, \ldots, \varphi_\ell^*$. Those at most k zero-loops containing transitions of $\widehat{T}$ are stored in a multiset L^*. Each zero-loop of L^* is enabled at some marking μ_i^*. (*e*): All occurrences of transitions of $\widehat{T}$ are removed from the sequences and the loops, resulting in subsequences $\alpha_1^*, \ldots, \alpha_\ell^*$ and k positive loops $\tau_1, \ldots, \tau_k$, where $\bar{\xi} := \alpha_1^* \cdots \alpha_\ell^*$ is the backbone, and $\tau_1, \ldots, \tau_k$ are the loops of interest, constituting the canonical firing sequence.

Let $\mathcal{P} = (P, T, F, \mu_0) \in \mathcal{C}$. If one of the values n, m, W is 0, then μ_0 is the only reachable marking, which means that $\mathcal{P}$ can be added to any f-g-canonical class such that the resulting class is still f-g-canonical. Hence, we assume in the following that $n, m, W > 0$. Let $b := \tilde{f}(n, m, W, \max(\mu_0))$, and let μ be a reachable marking of $\mathcal{P}$. Then, there is a firing sequence σ leading to μ. Consider the wipe-extension $\widehat{\mathcal{P}} = (P, \widehat{T}, \widehat{F}, \mu_0)$ of $\mathcal{P}$ with n places, $m + n$ transitions, and largest edge multiplicity W. The sequence σ is also a firing sequence in $\widehat{\mathcal{P}}$ leading to μ, and can be extended by transitions of $\widehat{T} \setminus T$ to a firing sequence σ' leading to the empty marking $\vec{0}$. By the assumption of the lemma, there is a permutation φ of σ' enabled in $\widehat{\mathcal{P}}$ with $\max(\mu_0, \varphi) \leq b$.

Let $M := \{\nu \mid \mu_0 \xrightarrow{\varphi_{[..i]}} \nu, i \in [0, |\varphi|]\}$ be the set of all markings obtained when firing φ. We partition φ into contiguous non-overlapping subsequences $\varphi_1, \ldots, \varphi_\ell$ with $\varphi = \varphi_1 \cdots \varphi_\ell$ such that the markings $\mu_i^* := \mu_0 + \Delta(\varphi_1 \cdots \varphi_i)$, $i \in [0, \ell]$, satisfy the following property: for each $\nu \in M$, there is exactly one $i \in [0, \ell]$ with $\mu_i^* = \nu$. In particular, this means $\mu_i^* \neq \mu_j^*$ for all distinct $i, j \in [0, \ell]$ and $M = \bigcup_{i \in [0, \ell]} \{\mu_i^*\}$. Note that $\ell \leq (b+1)^n$ since there are at most $(b+1)^n$ markings ν with $\max(\nu) \leq b$. In the following, we will iteratively cut out short zero-loops from these subsequences and collect them in a multiset L^*.

Consider the following condition: for all zero-loops $\tau^* \in L^*$, we have $|\tau^*| \leq (b+1)^n$, and there is a marking $\mu^* \in M$ such that τ^* is enabled at μ^* in $\widehat{\mathcal{P}}$. At the beginning, L^* is empty, implying that the above condition is satisfied. Now, assume that, for some sequence $\rho := \varphi_i$, $i \in [\ell]$, we observe $|\rho| \geq (b+1)^n$. Then, with the same argument as before, there are indices i_1, i_2 with $i_1 < i_2$ and $i_2 - i_1 \leq (b+1)^n$ such that both firing sequences $\varphi_1 \cdots \varphi_{i-1} \cdot \rho_{[..i_1]}$ and $\varphi_1 \cdots \varphi_{i-1} \cdot \rho_{[..i_2]}$ lead to the same marking $\mu^* \in M$. Let $\varphi'_j := \varphi_j$ for all $j \in [\ell] \setminus \{i\}$, and $\varphi'_i := \rho_{[..i_1]} \cdot \rho_{[i_2+1..]}$. If the zero-loop $\rho_{[i_1+1..i_2]}$ contains some transition of $\widehat{T} \setminus T$, then we add it to L^*, and discard it otherwise. Then, $\mu_0^* \xrightarrow{\varphi'_1 \cdots \varphi'_i} \mu_i^*$, $i \in [\ell]$, since we merely removed a zero-loop (which has no effect on the markings). In particular, the zero-loop $\rho_{[i_1+1..i_2]}$, which is enabled at $\mu^* \in M$, as well as all zero-loops which were contained in L^* before, (still) satisfy the condition given above.

Iterating this argument yields sequences $\varphi_1^*, \ldots, \varphi_\ell^*$ and multisets L_i^*, $i \in [0, \ell]$, consisting of zero-loops, where $L^* := \bigcup_{i=0}^{\ell} L_i^*$, such that

- $\mu_0 = \mu_0^* \xrightarrow{\varphi_1^*} \mu_1^* \xrightarrow{\varphi_2^*} \cdots \mu_{\ell-1}^* \xrightarrow{\varphi_\ell^*} \mu_\ell^* = \vec{0}$ in $\widehat{\mathcal{P}}$,
- $\tau^* \in L_i^*$ is enabled at μ_i^* in $\widehat{\mathcal{P}}$ and $\max(\mu_i^*) \leq b$ for all $i \in [0, \ell]$,
- $|\varphi_i^*| \leq (b+1)^n$ for all $i \in [\ell]$, and
- $|\tau^*| \leq (b+1)^n$ for all $\tau^* \in L^*$.

All these sequences potentially contain transitions of $\widehat{T} \setminus T$ since they are sequences of $\widehat{P}$. Furthermore, each $\tau^* \in L^*$ certainly contains such transitions which implies $k := |L^*| \leq n \cdot \max(\mu)$. We obtain the sequences α_i^*, $i \in [\ell]$, from φ_i^*, and the multisets L_i, $i \in [\ell]$, from the sets L_i^* by removing all transitions $\widehat{T} \setminus T$ from the respective sequences. Let $L := \bigcup_{i=0}^{\ell} L_i$, and $\mu_i := \mu_0 + \Delta(\alpha_1^* \cdots \alpha_i^*)$, $i \in [\ell]$.

The following properties are satisfied:

- $\bar{\xi} = \alpha_1^* \cdots \alpha_\ell^*$ is a firing sequence of $\mathcal{P}$ with $|\bar{\xi}| \leq (b+1)^{2n} =: f(n, m, W, \max(\mu_0))$,
- $\tau \in L_i$, $i \in [0, \ell]$, is enabled in $\mathcal{P}$ at $\mu_i^* \leq \mu_i$, where $\max(\mu_i^*) \leq b$, and
- $|\tau| \leq (b+1)^n$ for all $\tau \in L$.

Now, we obtain $\tau_1, \ldots, \tau_k$ by numbering the sequences of L, where we number the sequences of L_i before the sequences of L_{i+1}. Furthermore, we obtain the sequences α_i, $i \in [k+1]$, by splitting $\bar{\xi}$ at appropriate positions (where at most $f(n, m, W, \max(\mu_0))$ of the sequences α_i are nonempty). Note that $\xi := \alpha_1 \cdot \tau_1 \cdot \alpha_2 \cdot \tau_2 \cdots \alpha_k \cdot \tau_k \cdot \alpha_{k+1}$ is indeed a firing sequence. In total, all properties of the lemma are satisfied. □

The Petri net $\mathcal{P}$ and its wipe-extension $\widehat{\mathcal{P}}$ of Figure 4.2 illustrate the limits of Theorem 4.2:

The next theorem provides upper bounds for the reachability, covering, and boundedness problems in terms of f and g.

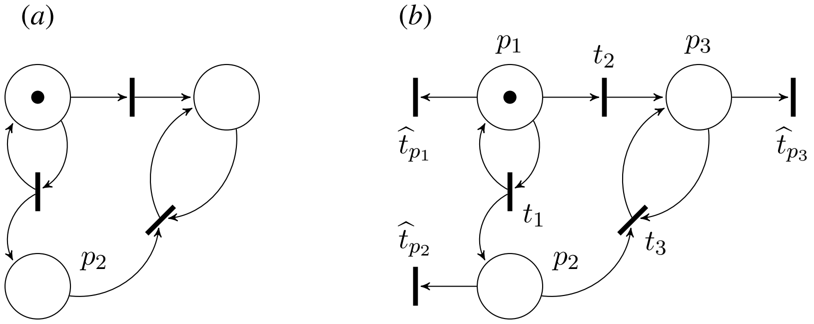


Figure 4.2: The Petri net of (b) is the wipe-extension of Petri net $\mathcal{P}$ of (a). No class $\mathcal{C}$ containing $\mathcal{P}$ can satisfy the prerequisites of Theorem 4.2 since, for each $k \in \mathbb{N}$, the only firing sequence with Parikh image $k \cdot t_1 + t_2 + k \cdot t_3 + \widehat{t}_{p_3}$ (which leads to the empty marking) is $t_1^k \cdot t_2 \cdot t_3^k \cdot \widehat{t}_{p_3}$, and the marking reached by t_1^k contains k tokens at place p_2. This means there is no permutation of σ such that it's possible to bound the number of tokens in terms of n, m, W, and $\max(\mu_0)$ for all markings obtained when firing the permutation.

Theorem 4.3. *Let $\mathcal{C}$ be a simple f-g-canonical class of Petri nets. Then, the reachability, and the covering problems are decidable in space polynomial in*

$$\text{size}(\mathcal{P}) + \text{size}(\mu) + n \operatorname{ld} f(n, m, W, \max(\mu_0)) + r,$$

and the boundedness problem is decidable in space polynomial in

$$\text{size}(\mathcal{P}) + n \operatorname{ld} f(n, m, W, \max(\mu_0)) + r,$$

where r is the space needed to compute $f(n, m, W, \max(\mu_0))$.

Proof. Let $u := f(n, m, W, \max(\mu_0))$. In the following, we assume n, m, W, $u > 0$ since otherwise μ_0 is the only reachable marking and the problems of the lemma can trivially be decided in polynomial space. We first consider the reachability problem. Let μ be a reachable marking, and let ξ, $\bar{\xi}$, α_i, and τ_i be as in Definition 4.1. Observe that $\max(\mu_0 + \Delta(\alpha_1 \cdot \tau_1 \cdots \alpha_i)) \leq \max(\mu) + uW$ for each $i \in [k+1]$, as well as $\max(\mu_0 + \Delta(\alpha_1 \cdot \tau_1 \cdot \alpha_2 \cdots \tau_i)) \leq \max(\mu) + uW$ for each $i \in [k]$ since all τ_j are loops, and $\min(\Delta(\bar{\xi}_{[q..r]})) \geq -uW$ for all $q, r \in [|\bar{\xi}|]$. Together with $\max(\vec{0}, \tau_i) \leq uW$ for all $i \in [k]$, and $\max(\vec{0}, \alpha_i) \leq uW$ for all $i \in [k+1]$, we find $\max(\mu_0, \xi) \leq \max(\mu) + 2uW$. Hence, we can decide the reachability problem (nondeterministically) in space polynomial in

$$\text{size}(\mathcal{P}) + n \operatorname{ld}(\max(\mu) + 2uW) + r \overset{\text{P}}{\leq} \text{size}(\mathcal{P}) + \text{size}(\mu) + n \operatorname{ld}(u) + r$$

by guessing these markings one after another, while only storing the last two markings, and checking if, for any two consecutive markings ν and ν', $\nu \xrightarrow{t} \nu'$ for some transition $t \in T$.

Next, we consider the boundedness problem. Let $\delta := \max(\mu_0) + uW + 1$. Assume that $\mathcal{P}$ is unbounded. Then, there is a reachable marking μ with $\max(\mu) \geq \delta$. Let ξ, $\bar{\xi}$, α_i, and τ_i be the sequences corresponding to μ as of Definition 4.1. Let p be a place with $\mu(p) = \max(\mu)$. Since $|\bar{\xi}| \leq u$ and $|\tau_i| \leq u$, $i \in [k]$, we can find coefficients $a_i \in \{0, 1\}$, $i \in [k]$, such that the reachable marking $\mu' := \mu_0 + \Delta(\bar{\xi}) + \sum_{i \in [k]} a_i \Delta(\tau_i)$ satisfies $\max(\mu') \in [\delta, \delta + uW]$. On the other hand, if a marking μ with $\max(\mu) \in [\delta, \delta + uW]$ exists, then $k \geq 1$, which implies that $\mathcal{P}$ is unbounded since $\alpha_1 \cdot \tau_1^i$ is a firing sequence for any $i \in \mathbb{N}$. Hence, we can decide the boundedness problem (nondeterministically) in space polynomial in

$$\begin{aligned}
&\text{size}(\mathcal{P}) + n\,\text{ld}(\delta + uW) + n\,\text{ld}(u) + r \\
&= \text{size}(\mathcal{P}) + n\,\text{ld}(\max(\mu_0) + 2uW + 1) + n\,\text{ld}(u) + r \\
&\overset{\text{P}}{\le} \text{size}(\mathcal{P}) + n\,\text{ld}(u) + r
\end{aligned}$$

by guessing a marking μ with $\max(\mu) \in [\delta, \delta + uW]$ and checking if μ is reachable.

Last, we consider the covering problem. Let μ^* be the marking for which we want to know if it can be covered. Assume that μ^* can be covered, and let $\mu \ge \mu^*$ be a reachable marking with its corresponding sequences ξ, $\bar{\xi}$, τ_i as of Definition 4.1. Let $M := \{\Delta(\tau_i) \mid i \in [k]\}$, and $\mu' := \mu_0 + \Delta(\bar{\xi}) + \sum_{d \in M} \max(\mu^*) \cdot d$. Marking μ' is reachable because we can construct a firing sequence leading to μ' by appropriately inserting, for each $i \in [k]$, a number of copies of τ_i into $\bar{\xi}$. Furthermore, if $(\mu_0 + \Delta(\bar{\xi}))(p) < \mu^*(p)$ for some place p, then $d(p) > 0$ for some $d \in M$. Therefore, μ' covers μ^*. Note that $\max(d) \in [0, uW]$ for each $d \in M$ and thus $|M| \le (uW + 1)^n$. By the simple observation $\sum_{i \in [k]} \le \prod_{i \in [k]} a_i'$ if $a_1, \ldots, a_k \in \mathbb{N}_0$ and $a_i' \ge \max\{a_i, 2\}$ for all $i \in [k]$, this implies

$$\begin{aligned}
\max(\mu') &\le \max(\mu_0) + \max(\Delta(\bar{\xi})) + \sum_{d \in M} \max(\mu^*) \cdot \max(d) \\
&\le \max(\mu_0) + uW + |M| \cdot \max(\mu^*) \cdot \max\{\max(d) \mid d \in M\} \\
&\le \max(\mu_0) + uW + (uW + 1)^n \cdot \max(\mu^*) \cdot uW \\
&\le (\max(\mu_0) + 2) \cdot 2uW \cdot (2uW)^n \cdot \max(\mu^*) \cdot uW =: d'.
\end{aligned}$$

Hence, we can decide the covering problem (nondeterministically) in space polynomial in

$$\begin{aligned}
&\text{size}(\mathcal{P}) + n\,\text{ld}(d') + n\,\text{ld}(u) + r \\
&= \text{size}(\mathcal{P}) + n\,\text{ld}((\max(\mu_0) + 2) \cdot 2uW \cdot (2uW)^n \cdot \\
&\quad \cdot \max(\mu^*) \cdot uW) + n\,\text{ld}(u) + r \\
&\overset{\text{P}}{\le} \text{size}(\mathcal{P}) + n\,\text{ld}(\max(\mu^*)) + n\,\text{ld}(u) + r \\
&\overset{\text{P}}{\le} \text{size}(\mathcal{P}) + \text{size}(\mu^*) + n\,\text{ld}(u) + r
\end{aligned}$$

by guessing a marking $\mu \ge \mu^*$ with $\max(\mu) \le d'$ and checking if μ is reachable. □

For each Petri net $\mathcal{P}$ of a simple structurally f-g-canonical class of Petri nets, we can use the canonical firing sequences to compute semilinear set representations of the reachability set $\mathcal{R}(\mathcal{P})$. To this end, we use the following strategy. We consider all possible relevant tuples of markings iteratively reached by parts of the backbones of the canonical sequences. Each tuple constitutes its own linear set, where the constant vector is the marking reached by the backbone, and the set of periods is the set of the displacements of all short positive loops enabled at some marking of the tuple.

Theorem 4.4. *Let $\mathcal{C}$ be a simple structurally f-g-canonical class of Petri nets, and $\mathcal{P} \in C$. Then, the SLSR*

$$\mathcal{SL} = \bigodot_{(\mu_1, \ldots, \mu_{u_2}) \in M} \mathcal{L}(\mu_{u_2}, \bigcup_{i \in [0, u_2]} D_{\mu_i}),$$

represents $\mathcal{R}(\mathcal{P})$ *and can be constructed in time*

$$p((\max(\mu_0) + 2u_1W)^{u_2n} \cdot 2^{p(\text{size}(\mathcal{P}))+n\,\mathrm{ld}(u_3)} + r)$$

for some polynomial p*, where*

(*a*) $u_1 = f(n, m, W, \max(\mu_0))$, $u_2 = g(n, m, W, \max(\mu_0))$,
$u_3 = f(n, m, W, \max(\mu_0) + 2u_1W)$, *and* r *is the time needed to calculate* u_1, u_2, *and* u_3,

(*b*) $M = \{(\mu_1, \ldots, \mu_{u_2}) \mid \mu_{i-1} \to \mu_i$ *and* $\max(\mu_i) \le \max(\mu_0) + u_1W$ *for all* $i \in [u_2]\}$, *and*

(*c*) $D_\mu = \{\Delta(\tau) \mid \tau$ *is a loop enabled at* μ *with* $\max(\Delta(\tau)) \le u_1W\}$.

Proof. As before, we assume n, m, W, u_1, u_2, $u_3 > 0$ since otherwise μ_0 is the only reachable marking. Consider a reachable marking μ, and let ξ, $\bar{\xi}$, α_i, τ_i be its corresponding sequences as of Definition 4.1. Let $\alpha'_1, \ldots, \alpha'_{u'_2}$ with $u'_2 \le u_2$ be the nonempty sequences of $\{\alpha_1, \ldots, \alpha_{k+1}\}$ such that $\alpha'_1 \cdots \alpha'_{u'_2} = \bar{\xi}$. Let μ_i, $i \in [0, u'_2]$, be defined by $\mu_0 \xrightarrow{\alpha'_1 \cdots \alpha'_i} \mu_i$, and $\mu_i := \mu_{i-1}$ for all $i \in [u'_2 + 1, u_2]$. Now, it's not hard to see that the properties of these sequences imply $(\mu_1, \ldots, \mu_{u_2}) \in M$ and, for all $i \in [k]$, there is a $j \in [0, u_2]$ with $\Delta(\tau_i) \in D_{\mu_j}$, i. e., $\mu \in \boldsymbol{\mathcal{L}}(\mu_{u_2}, \bigcup_{i\in[0,u_2]} D_{\mu_{u_2}}) \subseteq \boldsymbol{\mathcal{SL}}$. Therefore, $\mathcal{R}(\mathcal{P}) \subseteq \boldsymbol{\mathcal{SL}}$. The other direction, namely $\boldsymbol{\mathcal{SL}} \subseteq \mathcal{R}(\mathcal{P})$, is obvious.

In the following, we show the upper bound for the time needed to construct $\mathcal{SL}$. We first compute a reachability matrix $A \in \{0,1\}^{d\times d}$ where $d := (\max(\mu_0) + 2u_1W + 1)^n$ such that the entry corresponding to two markings μ, μ' with $\max(\mu)$, $\max(\mu') \le \max(\mu_0) + 2u_1W$ is 1 if and only if $\mu \to \mu'$. By Theorem 4.3 and the monotonicity of f, each entry can be determined in space polynomial in

$$\begin{aligned}
&\text{size}(\mathcal{P}) + \text{size}(\mu) + \text{size}(\mu') + n\,\mathrm{ld}(u_3)\\
&\overset{\mathrm{P}}{\le} \text{size}(\mathcal{P}) + n\,\mathrm{ld}(\max(\mu_0) + 2u_1W) + n\,\mathrm{ld}(\max(\mu_0) + 2u_1W) + n\,\mathrm{ld}(u_3)\\
&\overset{\mathrm{P}}{\le} \text{size}(\mathcal{P}) + n\,\mathrm{ld}(u_3).
\end{aligned}$$

The whole matrix can be determined in time $p(d^2 \cdot 2^{p(\text{size}(\mathcal{P})+n\,\mathrm{ld}(u_3))})$ for some polynomial p.

Next, we construct all relevant sets D_{μ^*}. Since $|\bar{\xi}| \le u_1$, we only have to consider the $(\max(\mu_0) + u_1W + 1)^n$ different markings μ^* with $\max(\mu^*) \le \max(\mu_0) + u_1W$. Because $|\tau_i| \le u_1$, we now check, for each such μ^* and all at most $(u_1W + 1)^n$ different markings μ satisfying $\mu \ge \mu^*$ and $\max(\mu - \mu^*) \le u_1W$, if μ can be reached from μ^*. If this is the case for μ^* and μ, then we add $\mu - \mu^*$ to D_{μ^*}. Using A, the total time to construct all sets is polynomial in the size of the matrix A, and thus is at most $p(d^2 \cdot 2^{p(\text{size}(\mathcal{P})+n\,\mathrm{ld}(u_3))})$ for some polynomial p.

Last, we construct, for each $(\mu_1, \ldots, \mu_{u_2}) \in M$, the LSR $\mathcal{L}(\mu_{u_2}, \bigcup_{i\in[0,u_2]} D_{\mu_i})$. We consider all $((\max(\mu_0) + u_1W + 1)^n)^{u_2}$ possible tuples $(\mu_1, \ldots, \mu_{u_2})$ with $\max(\mu_i) \le \max(\mu_0) + u_1W$, $i \in [u_2]$, and check, using A, if $\mu_{i-1} \to \mu_i$ for all $i \in [u_2]$. If this is the case, then we construct the LSR mentioned above. The total time to compute u_1, u_2, u_3, to generate A, all sets D_{μ^*}, and all these LSRs is polynomial in

$$\begin{aligned}
&(\max(\mu_0) + u_1W + 1)^{u_2n} \cdot u_2 \cdot 2^{p_1(\text{size}(\mathcal{P})+n\,\mathrm{ld}(u_3))} \cdot d^2 2^{p_1(\text{size}(\mathcal{P})+n\,\mathrm{ld}(u_3))} + r\\
&\overset{\mathrm{P}}{\le} (\max(\mu_0) + 2u_1W)^{u_2n} \cdot 2^{p_2(\text{size}(\mathcal{P}))+n\,\mathrm{ld}(u_3)} + r
\end{aligned}$$

for two polynomials p_1, p_2. □

We will use this theorem in Chapter 6. We note that the Petri net of Figure 4.2 additionally demonstrates that (not very surprisingly) there are also Petri nets with semilinear reachability sets for which our permutation technique cannot be applied.

Using this SLSR as a tool (which is not explicitly computed), we can also give an upper bound for the liveness problem in terms of f and g. The core of the proof is the observation that if the Petri net is not live, then there are a transition and a small marking serving as witnesses.

Theorem 4.5. *Let $\mathcal{C}$ be a simple structurally f-g-canonical class of Petri nets. Then, the liveness problem of C is decidable in space polynomial in*

$$\text{size}(\mathcal{P}) + n\,\text{ld}(f(n, m, W,\ \max(\mu_0) + f(n, m, W,\ \max(\mu_0))W)) + r,$$

where r is the space needed to compute

$$f(n, m, W,\ \max(\mu_0) + f(n, m, W,\ \max(\mu_0))W).$$

Proof. As before, we assume n, m, W, $u > 0$ since otherwise μ_0 is the only reachable marking, and the liveness property can be checked in polynomial time. Assume that $\mathcal{P} = (N, \mu_0) \in \mathcal{C}$ is not live. Then, there is a transition t and a reachable marking μ such that ξ and $\bar{\xi}$ are its corresponding sequences as of Definition 4.1, and no marking μ' reachable from μ enables t. If $\max(\mu) > \max(\mu_0) + uW$, then $\xi \neq \bar{\xi}$, and the marking ν reached by $\bar{\xi}$ satisfies $\nu < \mu$. Furthermore, for each marking ν' reachable from ν, there is a marking μ' reachable from μ such that $\nu' < \mu'$. Hence, ν is a smaller marking than μ that also witnesses that $\mathcal{P}$ is not live. Thus, if $\mathcal{P}$ is not live, then there is a small marking that witnesses that some transition t is not live.

We now show how we can check for a marking μ with $\max(\mu) \leq \max(\mu_0) + uW$ and a transition t if no marking μ' reachable from μ enables t. Let $u' := f(n, m, W,\ \max(\mu)) \leq f(n, m, W,\ \max(\mu_0) + uW)$, and let $\mathcal{SL} = \bigodot_{i\in[\ell]} \mathcal{L}(\zeta_i, \Pi_i)$ be the SLSR for $\mathcal{R}(N, \mu)$ given in Theorem 4.4. If there is a marking μ' reachable from μ and enabling t, then there is a LSR $\mathcal{L}(\zeta, \Pi) := \mathcal{L}(\zeta_i, \Pi_i)$ for some $i \in [\ell]$ such that $\mu' \in \boldsymbol{\mathcal{L}}(\zeta, \Pi)$, $\max(\zeta) \leq \max(\mu) + u'W$, and $\max(\pi) \leq u'W$ for all $\pi \in \Pi$. Since $\mu' \in \boldsymbol{\mathcal{L}}(\zeta, \Pi)$, if $\zeta(p) < F(p, t)$ for some place $p \in {}^\bullet t$, there is a period $\pi \in \Pi$ with $\pi(p) \in [1, u'W]$. Hence, by appropriately combining the periods (for each place of ${}^\bullet t$ at most W periods), we find a marking $\mu'' \in \boldsymbol{\mathcal{L}}(\zeta, \Pi)$ such that μ'' enables t, and

$$\begin{aligned}
\max(\mu'') &\leq \max(\zeta) + nW \cdot u'W \leq \max(\mu) + u'W + nu'W^2 \\
&\leq (\max(\mu_0) + uW) + 2nf(n, m, W,\ \max(\mu_0) + uW)W^2 \\
&\leq \max(\mu_0) + 3nf(n, m, W,\ \max(\mu_0) + uW)W^2 =: u''.
\end{aligned}$$

That means, we only need to check if no marking μ'' reachable from μ with $\max(\mu'') \leq u''$ enables t in order to ensure that no marking reachable from μ enables t.

Using these bounds, we can now show how to decide the liveness problem. We iterate over all $t \in T$ and all markings μ with $\max(\mu) \leq \max(\mu_0) + uW$ and check if μ is reachable. For each such reachable marking μ, we test, by iterating over all μ'' with $\max(\mu'') \leq u''$, if at least one of these markings is reachable from μ and enables t. $\mathcal{P}$ is live if and only if all tests succeed. By Theorem 4.3, the amount of space needed by this algorithm is at most polynomial in

$$\begin{aligned}
&\text{size}(\mathcal{P}) + n\,\mathrm{ld}(\max(\mu_0) + uW) + n\,\mathrm{ld}(\max(\mu_0) \\
&\quad + 3nf(n, m, W, \max(\mu_0) + uW)W^2) + n\,\mathrm{ld}(f(n, m, W, \max(\mu_0) + uW)) + r \\
&\overset{\mathrm{P}}{\le} \text{size}(\mathcal{P}) + n\,\mathrm{ld}(f(n, m, W, \max(\mu_0))) + n\,\mathrm{ld}(f(n, m, W, \max(\mu_0) + uW)) \\
&\quad + n\,\mathrm{ld}(f(n, m, W, \max(\mu_0) + uW)) + r \\
&\overset{\mathrm{P}}{\le} \text{size}(\mathcal{P}) + n\,\mathrm{ld}(f(n, m, W, \max(\mu_0) + uW)) + r.
\end{aligned}$$

□

The SLSRs of Theorem 4.4 can also be used to decide the equivalence and containment problems. As already mentioned in Section 2.6, using algorithms for problems of SLSRs to solve problems of Petri nets is a standard approach that has been used for many classes of Petri nets before.

Theorem 4.6. *Let $\mathcal{C}$ be a simple structurally f-g-canonical class of Petri nets. Then, for some polynomial p, the equivalence and containment problems of $\mathcal{C}$ are decidable in space*

$$p((K + 2u_1K)^{u_2n} \cdot 2^{p(s+n\,\mathrm{ld}(u_3))} + r),$$

where

- *s is the encoding size of the input,*
- *n is the total number of places of both nets,*
- *m is the total number of transitions,*
- *W is the maximum of all edge multiplicities of both nets,*
- *K is the largest number of tokens appearing at some place at the initial markings,*
- *$u_1 = f(n, m, W, K)$, $u_2 = g(n, m, W, K)$, $u_3 = f(n, m, W, K + 2u_1W)$, and*
- *r is the time needed to calculate u_1, u_2, and u_3.*

Proof. For the input Petri nets $\mathcal{P}_1$ and $\mathcal{P}_2$, we use Theorem 4.4 to compute the SLSRs $\mathcal{SL}_1$ and $\mathcal{SL}_2$ with $\boldsymbol{\mathcal{SL}}_1 = \mathcal{R}(\mathcal{P}_1)$ and $\boldsymbol{\mathcal{SL}}_2 = \mathcal{R}(\mathcal{P}_2)$. By Theorem 2.4 and Corollary 2.5, we can decide the containment and equivalence problems of SLSRs in space polynomial in the size of the SLSRs. Thus, the theorem follows. □

5 Conservative Petri nets

An S-*invariant* is a vector $x \in \mathbb{Z}^n$ (sometimes also defined as a vector $x \in \mathbb{N}_0^n$) satisfying the equation $D^T x = 0$. A Petri net is called x-*conservative* for some $x \geq \vec{1}$ if x is an S-invariant, and is called *conservative* if it is x-conservative for some $x \geq \vec{1}$. Furthermore, a Petri net is 1-*conservative* if it is $\vec{1}$-conservative. The defining property of an x-conservative Petri nets is that, for all reachable markings, the weighted (by x) sum of tokens is the same. This immediately implies that conservative Petri nets are bounded. Conservative Petri nets were introduced by Lien [Lie76a]. Later, Jones et al. [Jon+77] showed (among other things) that the reachability and the containment problems of ordinary 1-conservative Petri nets are **PSPACE**-complete. Furthermore, they showed that the liveness problem of 1-conservative Petri nets is contained in **PSPACE**, but no lower bound was given.

In this chapter, we strengthen these results. Our results hold for all conservative Petri nets, including those with arbitrary edge multiplicities. The arguments used for the upper bounds are in essence the arguments used in [Jon+77], which are appropriately extended for dealing with nets that are not ordinary or 1-conservative. In addition to these generalized upper bounds, we also prove **PSPACE**-hardness (and thus **PSPACE**-completeness) of the liveness problem, as well as **PSPACE**-completeness of the RecLFS problem. As we will see, the property of being conservative leads to straightforward arguments. This gives us an opportunity to demonstrate a number of constructions (e. g., simulating **PSPACE**-machines with Petri nets) that reappear, in much more involved forms, in Chapter 6, where we discuss generalized communication-free Petri nets.

Obviously, a Petri net $\mathcal{P}$ is conservative if there is an $x \in \mathbb{Q}^n$ with $x \geq \vec{1}$ and $D^T x = 0$ since we can scale such a rational valued vector to an integer valued one. Using, for instance, Karmarkar's algorithm [Kar84], we can find in polynomial time such a rational valued vector x (if it exists) which additionally satisfies $\max(x) \leq 2^{p(\mathrm{size}(\mathcal{P}))}$ for some polynomial p. Hence, the recognition problem, which asks if a given Petri net $\mathcal{P}$ is conservative, can be decided in polynomial time. We remark that, for this reason, the boundedness problem of conservative Petri nets is decidable in polynomial time (and not only its promise problem variation, for which trivially always the answer "yes" can be given). The same applies to the zero-reachability problem, which can trivially be decided in polynomial time, since the empty marking is reachable in a conservative Petri net if and only if the initial marking is empty.

We remark that, instead of using Theorem 4.2, we could easily show directly that the class of conservative Petri nets is f-f-canonical for some appropriate function f, or show the complexity results without using our framework at all. For the same reason, however, it lends itself as a good example.

Since we want to demonstrate our framework, we first show that the prerequisites of Theorem 4.2 are satisfied.

Lemma 5.1. *There is a constant d such that, for each reachable marking μ of a conservative Petri net $\mathcal{P} = (P, T, F, \mu_0)$ with $n > 0$, $\max(\mu) \leq 2^{\mathrm{size}(\mathcal{P})^d}$ holds.*

Proof. Since $\mathcal{P}$ is conservative and $n > 0$, there is a constant c such that there is an $x \in \mathbb{Q}^n$ satisfying $x \geq \vec{1}$, $D^T x = 0$, and $\max(x) \leq 2^{\mathrm{size}(\mathcal{P})^c}$. Let Φ be a Parikh vector leading to μ. We observe

$$\begin{aligned}\max(\mu) &\leq \mu^T x = (\mu_0 + \Delta(\Phi))^T x = (\mu_0^T + \Phi^T D^T)x = \mu_0^T x + \Phi^T (D^T x) \\ &= \mu_0^T x \leq n \cdot \max(\mu_0) \cdot \max(x) \leq 2^{\mathrm{size}(\mathcal{P})^d}\end{aligned}$$

for some constant d, where e_k denotes the k-th standard unit vector (as defined in Section 2.1). □

Lemma 5.2. *There is a polynomial p such that, for each conservative Petri net $\mathcal{P}$ with $n > 0$ and each firing sequence σ of its wipe-extension $\widehat{\mathcal{P}}$ leading to the empty marking $\vec{0}$, there is a permutation φ of σ enabled in $\widehat{\mathcal{P}}$ with $\max(\mu_0, \varphi) \leq 2^{p(n+m+\operatorname{ld} W+\operatorname{ld}\max(\mu_0))}$.*

Proof. Let $\widehat{\mathcal{P}} = (P, \widehat{T}, \widehat{F}, \mu_0)$ be the wipe-extension of a conservative Petri net $\mathcal{P} = (P, T, F, \mu_0)$. Let μ' be some marking reached in $\widehat{\mathcal{P}}$ by some Parikh vector Φ', Φ be the projection of Φ' onto the transitions of the original Petri net $\mathcal{P}$, and μ be the marking reached by Φ in $\mathcal{P}$. (Note that Φ is indeed enabled since all transitions we discarded only remove tokens from the net.) Then, by Lemma 5.1, we observe $\max(\mu') \leq \max(\mu) \leq 2^{\text{size}(\mathcal{P})^d}$ for some constant d. □

Lemma 5.3. *There is a polynomial p such that the class $\mathcal{C}$ of conservative Petri nets is simple structurally f-f-canonical, where $f(n, m, W, K) = 2^{p(n+m+\operatorname{ld} W+\operatorname{ld} K)}$.*

Proof. By Theorem 4.2 and Lemma 5.2, there is a polynomial p such that $\mathcal{C}$ is f-f-canonical for the function f defined by $f(n, m, W, K) = 2^{p(n, m, \operatorname{ld} W, \operatorname{ld} K)}$. As argued before, we can check in polynomial time if a Petri net is conservative. Hence, $\mathcal{C}$ is simple. Since the conservation property doesn't depend on the initial marking, $\mathcal{C}$ is structurally f-f-canonical. □

We can apply our framework to obtain upper bounds for a number of problems.

Lemma 5.4. *The reachability, the covering, and the liveness problems of conservative Petri nets are in* **PSPACE**.

Proof. By Theorem 4.3, Lemma 5.3, and by the fact that r is polynomial in size($\mathcal{P}$), the reachability and the covering problems are decidable in space polynomial in

$$\begin{aligned}
&\text{size}(\mathcal{P}) + \text{size}(\mu) + n \operatorname{ld} f(n, m, W, \max(\mu_0)) + r \\
&= \text{size}(\mathcal{P}) + \text{size}(\mu) + n \cdot p(n + m + \operatorname{ld} W + \operatorname{ld}\max(\mu_0)) + r \\
&\overset{\text{P}}{\leq} \text{size}(\mathcal{P}) + \text{size}(\mu),
\end{aligned}$$

where p is the polynomial of Lemma 5.3. Similarly, Theorem 4.5 and Lemma 5.3 imply that the liveness problem is decidable in space polynomial in

$$\begin{aligned}
&\text{size}(\mathcal{P}) + n \operatorname{ld}(f(n, m, W, \max(\mu_0) + f(n, m, W, \max(\mu_0))W)) + r \\
&= \text{size}(\mathcal{P}) + n \operatorname{ld}(2^{p(n+m+\operatorname{ld} W+\operatorname{ld}(\max(\mu_0)+2^{p(n+m+\operatorname{ld} W+\operatorname{ld}\max(\mu_0))}W))}) \\
&\overset{\text{P}}{\leq} \text{size}(\mathcal{P}) + np'(n + m + \operatorname{ld} W + \operatorname{ld}(\max(\mu_0))) \\
&\overset{\text{P}}{\leq} \text{size}(\mathcal{P}),
\end{aligned}$$

where p is the polynomial of Lemma 5.3, and p' is some polynomial. □

We now show that the liveness problem is **PSPACE**-hard in the strong sense. A straightforward adaption of Hack's reduction [Hac74b] from the reachability problem to the liveness problem is not possible since it uses the zero-reachability problem as an intermediate step and, more importantly, it yields Petri nets that are not necessarily conservative. Our approach is, however, similar to that used in [Jon+77] for the reachability and covering problems of ordinary 1-conservative Petri nets.

Let $L \in$ **PSPACE**, and M be a Turing machine in standard form deciding L. Using M, we will define a polynomial time reduction from L to the liveness problem. We will use similar, yet more involved, reductions for problems of generalized communication-free Petri nets and related classes in Chapter 6. Hence, we define the Petri nets used for such reductions in a relatively generic fashion.

Definition 5.5 (Transition gadget)**.** A Petri net $G = (P_G, T_G, F_G)$ is a transition gadget if there are four distinct places $p^{\Gamma}_{\text{in}}, p^{Q}_{\text{in}}, p^{\Gamma}_{\text{out}}, p^{Q}_{\text{out}} \in P_G$. For a specific instance H of G, and a place $p \in P_G$ (transition $t \in T_G$), $p(H)$ ($t(H)$, resp.) denotes place p (transition t, resp.) of H. $p^{\Gamma}_{\text{in}}(H)$ and $p^{\Gamma}_{\text{out}}(H)$ as well as $p^{Q}_{\text{in}}(H)$ and $p^{Q}_{\text{out}}(H)$ may not necessarily denote distinct places.

Definition 5.6 (Petri net $\mathcal{P}^G_{M,x}$)**.** Let $M = (Q, \Gamma, \Sigma, \delta, q_0, \Box, q_{\text{acc}})$ be a Turing machine in standard form, $x \in \Sigma^*$, and $G = (P_G, T_G, F_G)$ be a transition gadget. Then, the Petri net $\mathcal{P}^G_{M,x}$ is defined in the following way:

- for each position $i \in [\ell_S]$, symbol $s \in \Gamma$, and state $q \in Q$ there are places $p^{\Gamma}_{i,s}$ and $p^{Q}_{i,q}$,
- for each position $i \in [\ell_S]$ and M-transition $d = ((q,s),(q',s',y)) \in \delta$, there is an instance $G_{i,d} = (P_{i,d}, T_{i,d}, F_{i,d})$ of gadget G, where $p^{\Gamma}_{\text{in}}(G_{i,d}) = p^{\Gamma}_{i,s}$, $p^{Q}_{\text{in}}(G_{i,d}) = p^{Q}_{i,q}$, $p^{\Gamma}_{\text{out}}(G_{i,d}) = p^{\Gamma}_{i,s'}$, and $p^{Q}_{\text{out}}(G_{i,d}) = p^{Q}_{i+y,q'}$, and
- $\mu_0 \triangleq p^{Q}_{1,q_0} + \sum_{i \in [\ell_S]} p^{\Gamma}_{i,x[i]}$.

The place $p^{Q}_{1,q_{\text{acc}}}$ is also denoted by p_{acc}.

Semantically, the places $p^{\Gamma}_{i,s}$ encode the contents of the tape, while the places $p^{Q}_{i,q}$ encode the position of the head and the state of M. The transitions of the gadgets are used to simulate the transitions of M. Obviously, the gadget G must be appropriately defined for $\mathcal{P}^G_{M,x}$ to be able to simulate M on input x. We use the gadget G illustrated in Figure 5.1. The corresponding Petri net $\mathcal{P}^G_{M,x}$ illustrated in Figure 5.2 is an ordinary 1-conservative Petri net. We remark that, by these definitions, different gadgets are not necessarily place disjoint, and a place can be an input place of some gadget and an output place of the same gadget.

Definition 5.7 (Configuration marking)**.** Let $M = (Q, \Gamma, \Sigma, \delta, q_0, \Box, q_{\text{acc}})$ be a Turing machine in standard form, $x \in \Sigma^*$, and $G = (P_G, T_G, F_G)$ be a transition gadget. The *configuration marking* $\mu \triangleq p^{Q}_{i,q} + \sum_{j \in [\ell_S]} p^{\Gamma}_{j,y_{[j]}}$ *corresponds* to the configuration of M at which only the first ℓ_S positions may be different from $\Box$, these positions contain the string $y \in \Gamma^{\ell_S}$, M is in state q, and the head is over position i. The configuration corresponding to a configuration marking μ is denoted by $\text{conf}(\mu)$, and $\mu_{\text{acc}} :\triangleq p_{\text{acc}} + \sum_{i \in [\ell_S]} p^{\Gamma}_{i,\Box}$ is the configuration marking corresponding to the unique accepting configuration of M.

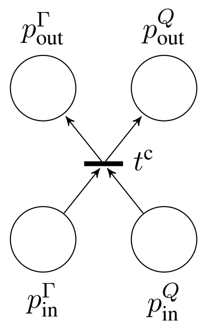


Figure 5.1: Transition gadget G for ordinary 1-conservative Petri nets

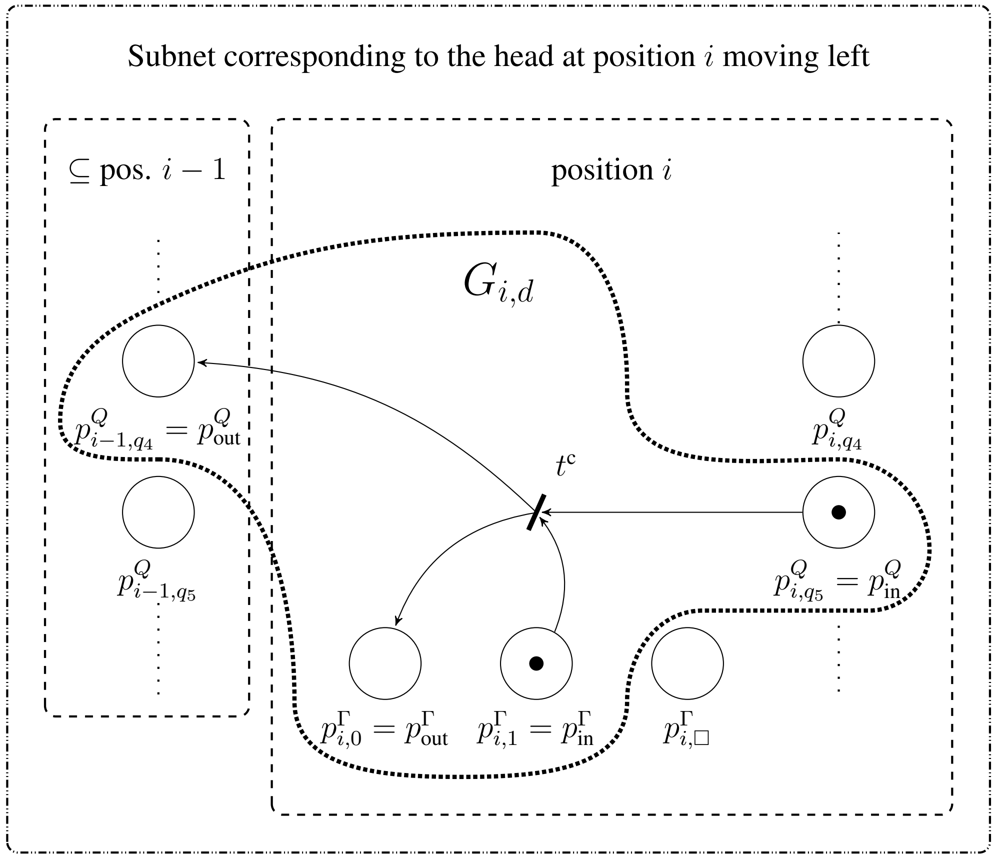


Figure 5.2: This figure illustrates a subnet of $\mathcal{P}^{G}_{M,x}$ where the marking is a configuration marking. In this example, M has tape alphabet $\Gamma = \{0, 1, \Box\}$. Since p^{Q}_{i,q_5} is marked, M is in state q_5 and M's head is at position i with respect to the configuration of M corresponding to the marking. The dotted curve encompasses the gadget $G_{i,d}$ where $d = (q_5, 1, q_4, 0, -1) \in \delta$ is an M-transition. In order for d to be executed, it requires that M is in state q_5, and that the tape contains the symbol 1 at the current position of the head. When d is executed, M switches into state q_4, writes 0 onto the tape, and moves left. For the corresponding configuration of M, this behavior is simulated by the gadget (see Lemma 5.8).

Note that, for each configuration C reachable by M on input x, there is a unique configuration marking corresponding to C. Moreover, the initial marking μ_0 of $\mathcal{P}^{G}_{M,x}$ is a configuration marking corresponding to the initial configuration of M.

Next, we show how firing sequences of $\mathcal{P}^G_{M,x}$ leading to configuration markings and computation paths of M on input x correspond to each other.

Lemma 5.8. *The following properties hold:*

(a) *if a marking μ' is reachable from a configuration marking μ in $\mathcal{P}^G_{M,x}$, then μ' is a configuration marking,*

(b) *if a transition sequence $\sigma = t^c(G_{i_1,d_1}) \cdots t^c(G_{i_k,d_k})$ of $\mathcal{P}^G_{M,x}$ leads from a configuration marking μ to a configuration marking μ', then the sequence of M-transitions $d_1 \cdots d_k$ is a computation path of M and leads from conf(μ) to conf(μ'), and*

(c) *if a computation path $d_1 \cdots d_k$ of M leads from conf(μ) to conf(μ'), then there are $i_1, \ldots,$ $i_k \in [\ell_S]$ such that $t^c(G_{i_1,d_1}) \cdots t^c(G_{i_k,d_k})$ leads from μ to μ' in $\mathcal{P}^G_{M,x}$.*

Proof. Property (a) follows from the observation that if a transition of $\mathcal{P}^G_{M,x}$ is fired at some configuration marking, then the marking we obtain is also a configuration marking. Properties (b) and (c) are also shown easily: Assume that $t^c(G_{i,d})$ of gadget $G_{i,d}$ leads from a configuration marking μ to a configuration marking μ'. Then, we observe that executing d at conf(μ) leads to conf(μ'). Consider an M-transition $d \in \delta$ that is executed at configuration conf(μ), where the head of M is at position i, yielding a configuration C. Then, firing t^c of gadget $G_{i,d}$ at μ yields a marking μ' with $C = \text{conf}(\mu')$. We apply these arguments to all transitions of σ respectively all M-transitions of the computation path. □

The correspondence of computation paths and firing sequences has a number of useful implications, which are collected in the next lemma.

Lemma 5.9. *There is a unique longest firing sequence σ of $\mathcal{P}^G_{M,x}$. This firing sequence σ satisfies $|\sigma| \le \ell_T$. Each firing sequence is a prefix of σ. Furthermore, the following are equivalent:*

- *M accepts x,*
- *μ_{acc} is reachable in $\mathcal{P}^G_{M,x}$,*
- *some marking μ with $\mu(p_{acc}) \ge 1$ is reachable, and*
- *σ leads to μ_{acc}.*

Proof. Since M is deterministic and always terminates (in particular, it terminates in the unique accepting configuration if M accepts x), Lemma 5.8 implies that there is a unique longest firing sequence σ, and each firing sequence is a prefix of σ (i. e., $\mathcal{P}^G_{M,x}$ is a single-path Petri net). By this, by Lemma 5.8, and by the fact that M is in standard form, the rest of the lemma follows. □

We can use the equivalences shown in the last lemma to finally obtain the desired lower bound for the liveness problem of conservative Petri nets.

Lemma 5.10. *The liveness problem of conservative Petri nets is* **PSPACE***-hard in the strong sense, even if restricted to ordinary* 1*-conservative Petri nets.*

Proof. By Lemma 5.9, we can use p_{acc} as a control place, which can be marked if and only if it is eventually marked if and only if M accepts x. We obtain the ordinary 1-conservative Petri net $\mathcal{P} = (P', T', F', \mu_0)$ by modifying $\mathcal{P}^G_{M,x} = (P, T, F, \mu_0)$ in the following way. We add a new place p'_{acc} and two new transitions t, t' with $p'_{\text{acc}} \xmapsto[F']{t} p_{\text{acc}}$ and $p_{\text{acc}} \xmapsto[F']{t'} p'_{\text{acc}}$. These transitions can transfer tokens freely between p_{acc} and p'_{acc}. Note that, once p_{acc} is marked for the first time, at least one of these new places remains marked. For each two places $p, p' \in P \setminus \{p_{\text{acc}}, p'_{\text{acc}}\}$, we add a transition t with $p_{\text{acc}} + p \xmapsto[F']{t} p_{\text{acc}} + p'$. These transitions can be used to transfer tokens freely between the (original) places of P once p_{acc} is marked for the first time. (If necessary, we can transfer a token from p'_{acc} back to p_{acc} to enable these transitions.) Last, we add a transition t^* with $p_{\text{acc}} + p'_{\text{acc}} \xmapsto[F']{t^*} p_{\text{acc}} + p^{\Gamma}_{1,\Box}$. The transition t^* can be used to decrease the total number of tokens at p_{acc} and p'_{acc} to 1 if they together contain at least two tokens. (If the latter is the case, we can use the new transitions to evenly spread the tokens between these two places in order to enable t^*.)

If M doesn't accept x, then none of the new transitions can ever fire. If M accepts x, then σ marks p_{acc}. As long as p_{acc} is marked, we can use the new transitions to freely transfer at least 2 tokens within the net. (Note that the net always contains at least 3 tokens since M is in standard form.) Since $\{p_{\text{acc}}, p'_{\text{acc}}\}$ is a trap, and therefore cannot be unmarked once it is marked, we observe that, for each transition $t \in T'$ and each marking μ reachable from μ_{acc} in $\mathcal{P}$, there is a marking μ' reachable from μ in $\mathcal{P}$ which enables t. Therefore, M accepts x if and only if $\mathcal{P}$ is live. Given M and x, $\mathcal{P}$ can be constructed in time polynomial in $|x|$. Furthermore, the numerical values of all numbers occurring in the encoding of $\mathcal{P}$ are polynomial in size($\mathcal{P}$). Thus, the lemma follows. □

We remark that the proofs for **PSPACE**-hardness of the reachability and the covering problems of ordinary 1-conservative Petri nets given in [Jon+77] actually yield **PSPACE**-hardness in the strong sense. In total, we obtain the following result.

Theorem 5.11. *The reachability, the covering, and the liveness problems of conservative Petri nets are* **PSPACE***-complete in the strong sense, even if restricted to ordinary* 1*-conservative Petri nets.*

Proof. The theorem follows from (the proofs of) Theorem 3.1 and Corollary 3.2 of [Jon+77], and Lemmata 5.4 and 5.10. □

We now consider the RecLFS problem. Even though Corollary 2.21 implies that the RecLFS problem is probably not **PSPACE**-hard in the strong sense, we can, however, show that it is **PSPACE**-hard (in the weak sense) and subsequently **PSPACE**-complete.

Theorem 5.12. *The RecLFS problem is* **PSPACE***-complete, even if restricted to ordinary* 1*-conservative Petri nets.*

Proof. By Theorem 2.19, the RecLFS problem is in **PSPACE**. We now show the lower bound. We obtain the ordinary 1-conservative Petri net $\mathcal{P} = (P', T', F', \mu_0)$ by modifying $\mathcal{P}^G_{M,x} = (P, T, F, \mu_0)$ in the following way. First, we add a new place p^* and a new transition r_1 with $p_{\text{acc}} \xmapsto[F']{r_1} p^*$. Next, we add $n = |P|$ new places $p'_1, \ldots, p'_n$, each marked with $\ell_T \cdot |\delta|$ tokens, where ℓ_T is the bound on the running time of M on input x (see Definition 2.1). Last, we add a transition r_2 with $p^* + \sum_{i\in[n]} p'_i \xmapsto[F']{r_2} p^* + \sum_{p\in P} p$. Consider the Parikh vector $\Phi :\triangleq r_1 + (\ell_T \cdot |\delta|) \cdot r_2 + \sum_{t\in T} \ell_T \cdot t$. Assume that M accepts x.

By Lemma 5.9, $\mathcal{P}$ has a firing sequence σ of length at most ℓ_T leading to a marking at which p_{acc} is marked. Then, $\rho := \sigma \cdot r_1 \cdot r_2^{\ell_T \cdot |\delta|} \cdot \bigodot_{t \in T} t^{\ell_T - \Psi(\sigma)(t)}$ is a firing sequence of $\mathcal{P}$ with $\Psi(\rho) = \Phi$. If, on the other hand, M doesn't accept x, then, by Lemma 5.9, p_{acc} cannot be marked. Consequently, r_1 can never be fired which implies that Φ is not enabled. Note that Φ has polynomial encoding size. As before, this reduction can be performed in polynomial time. □

Last, we investigate the containment and equivalence problems. For conservative Petri nets, using Theorem 4.6 wouldn't yield a good upper bound. Instead, we use a direct approach.

Theorem 5.13. *The containment problem and equivalence problem of conservative Petri nets are* **PSPACE**-*complete in the strong sense, even if restricted to ordinary* 1-*conservative Petri nets.*

Proof. Consider the Petri net $\mathcal{P}^G_{M,x} = (P, T, F, \mu_0)$. We construct the Petri net $\mathcal{P}_1 = (P_1, T_1, F_1, \mu_1)$ from $\mathcal{P}^G_{M,x}$ in the following way: We add $\ell_S + 1$ places $p^*_1, \ldots, p^*_{\ell_S+1}$. The initial marking of $\mathcal{P}_1$ is $\mu_1 :\triangleq \sum_{i \in [\ell_S+1]} p^*_i$. Then we add a transition t^*_1 with $\mu_1 \xmapsto[F_1]{t^*_1} \mu_0$. Now, we obtain $\mathcal{P}_2 = (P_1, T_1 \cup \{t^*_2\}, F_2, \mu_1)$ from $\mathcal{P}_1$ by adding a transition t^*_2 with $\mu_1 \xmapsto[F_2]{t^*_2} \mu_{\text{acc}}$. Either t^*_1 or t^*_2 can be used to mark the original places of P with μ_0 or μ_{acc}, respectively. Both $\mathcal{P}_1$ and $\mathcal{P}_2$ are ordinary 1-conservative Petri nets that can be constructed in polynomial time in $|x|$. The following are equivalent:

- M accepts x,
- μ_{acc} is reachable in $\mathcal{P}^G_{M,x}$,
- $\mathcal{R}(\mathcal{P}_2) \subseteq \mathcal{R}(\mathcal{P}_1)$, and
- $\mathcal{R}(\mathcal{P}_2) = \mathcal{R}(\mathcal{P}_1)$.

Hence, the lower bound of the theorem follows. For **PSPACE**-membership, observe that if $\mathcal{R}(\mathcal{P}_1) \not\subseteq \mathcal{R}(\mathcal{P}_2)$ respectively $\mathcal{R}(\mathcal{P}_1) \neq \mathcal{R}(\mathcal{P}_2)$ holds, then, by Lemma 5.1, there is a polynomially sized reachable marking serving as a witness. Using Lemma 5.4, we can check in polynomial space if such a marking exists. □

We conclude the chapter with an observation about a number of home space problems.

Theorem 5.14. *The following problems for conservative Petri nets are* **PSPACE**-*complete in the strong sense, even if restricted to ordinary* 1-*conservative Petri nets:*

- *home state recognition: Given a conservative Petri net $\mathcal{P}$ and a marking μ, is μ a home state?*
- *home space recognition: Given a conservative Petri net $\mathcal{P}$ and a set $\mathcal{HS}$ of markings (given as an enumeration of markings), is $\mathcal{HS}$ a home space?*
- *home state existence: Given a conservative Petri net, does it have a home state?*

Proof. Consider again the ordinary 1-conservative Petri net $\mathcal{P}^G_{M,x}$. By Lemma 5.9, M accepts x if and only if μ_{acc} is reachable from each reachable marking. Hence, the home state recognition and the home space recognition problems are **PSPACE**-hard in the strong sense for ordinary 1-conservative

Petri nets. Both problems for conservative Petri nets are in **PSPACE** as the following procedure shows: Given a set $\mathcal{HS}$ of places, we enumerate all markings μ with $\max(\mu) \leq 2^{\text{size}(\mathcal{P})^d}$, where d is the constant of Lemma 5.1, and check, for each μ that is reachable, if there is a $\mu' \in \mathcal{HS}$ with $\mu \to \mu'$. This is the case if and only if $\mathcal{HS}$ is a home space. By Theorem 5.11, this procedure only needs space polynomial in $\text{size}(\mathcal{P}) + \text{size}(\mathcal{HS})$.

Next, we show that the complement of the home state existence problem of conservative Petri nets is **PSPACE**-complete in the strong sense, even if restricted to ordinary 1-conservative Petri nets. To this end, we construct a Petri net $\mathcal{P}' = (P', T', F', \mu_0')$ from $\mathcal{P}^G_{M,x} = (P, T, F, \mu_0)$ by adding to places p_1', p_2' and two transitions t_1', t_2' with $p_{\text{acc}} \xmapsto{t_1'} p_1'$ and $p_{\text{acc}} \xmapsto{t_2'} p_2'$. We define the marking $\mu'_{\text{acc}} :\triangleq \sum_{p \in P} \mu_{\text{acc}}(p) \cdot p$ of $\mathcal{P}'$. Let μ_1', μ_2' defined by $\mu'_{\text{acc}} \xrightarrow{t_1'} \mu_1'$ and $\mu'_{\text{acc}} \xrightarrow{t_2'} \mu_2'$. By Lemma 5.9, the following are equivalent:

- M accepts x, and
- μ_1' and μ_2' are reachable in $\mathcal{P}'$.

Furthermore, since M is in standard form, no transition is enabled at μ_1' or μ_2'. Hence, if M accepts x, then $\mathcal{P}'$ has no home state. If M doesn't accept x, then, by Lemma 5.9, neither of the markings μ_1', μ_2' is reachable, and the marking reached by the unique longest firing sequence is a home state. This proves the lower bound **PSPACE**-hardness in the strong sense, even for the restricted problem. To decide the problem in polynomial space, we use a similar algorithm as before, with the extension that we iterate over all candidates $\mathcal{HS} = \{\mu\}$ for home states, where $\max(\mu) \leq 2^{\text{size}(\mathcal{P})^d}$. Since **coPSPACE** $=$ **PSPACE**, the home state existence problem of conservative Petri nets is **PSPACE**-complete in the strong sense, even if restricted to ordinary 1-conservative Petri nets. $\square$

6 Generalized communication-free Petri nets

In this chapter, we investigate generalized communication-free Petri nets (gcf-PNs, also known as join-free Petri nets), inverse generalized communication-free Petri nets (igcf-PNs, also known as fork-free Petri nets), and generalized S-system Petri nets (gss-PNs, also known as weighted state machines). The relationship of the classes defined in the following is illustrated in Figure 6.1. A Petri

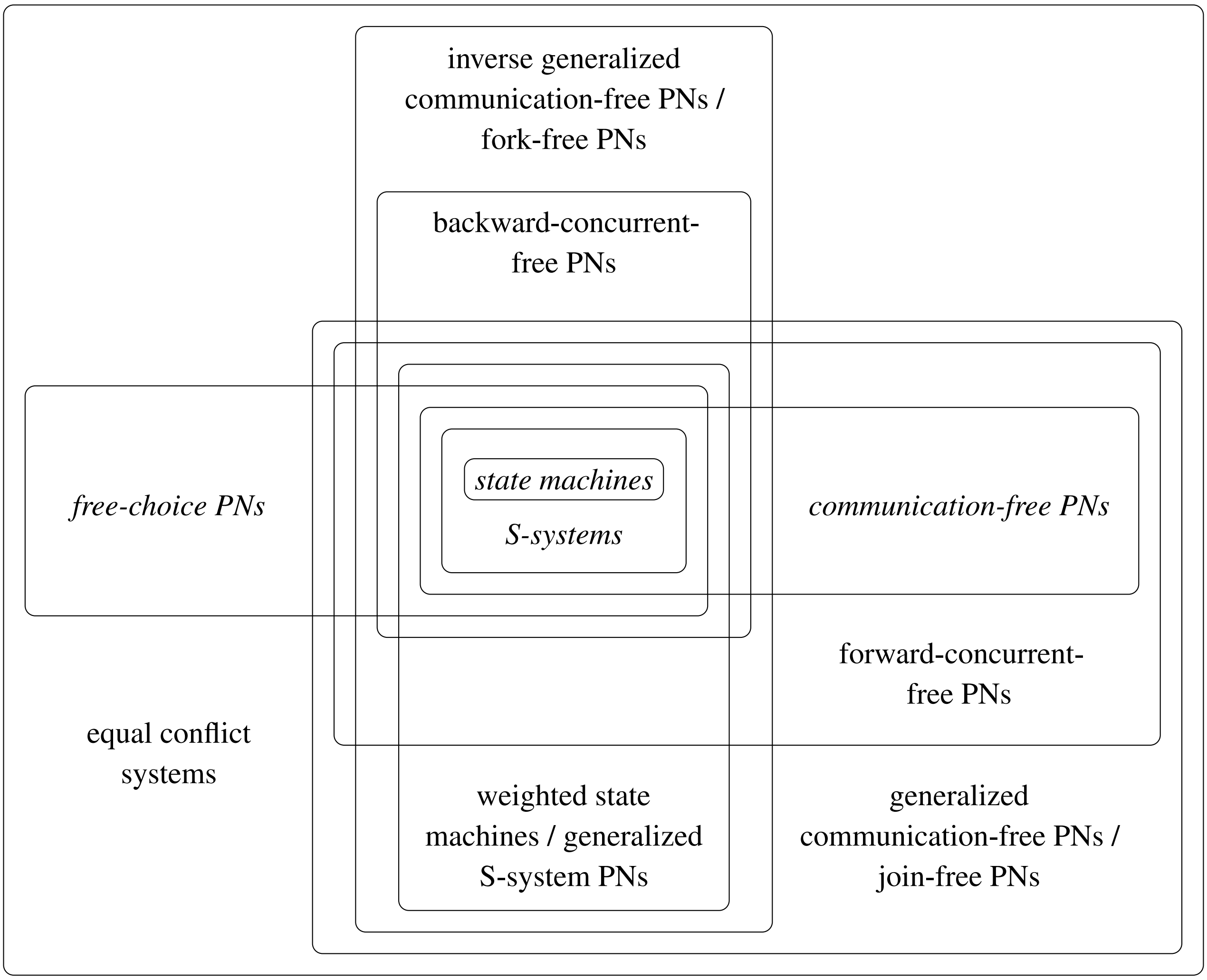


Figure 6.1: This class diagram illustrates illustrates the relationships between classes of Petri nets that are closely related to generalized communication-free Petri nets, their inverse nets, and generalized S-system Petri nets. Classes written in italics consist of (forward-)ordinary Petri nets.

net is a gcf-PN (igcf-PN, resp.) if each transition has at most one incoming (outgoing, resp.) edge. A Petri net is a gss-PN if it is a gcf-PNs and igcf-PNs at the same time. To improve the readability, we will, in the context of a transition t of a gcf-PN (of a igcf-PN, resp.), occasionally abuse notation by identifying ${}^\bullet t$ ($t^\bullet$, resp.) with its unique element if this set is nonempty. The classes of gcf-PNs and gss-PNs are generalizations of the classes of cf-PNs and state machines/S-Systems with respect to edge multiplicities. Very close relatives of the class of gcf-PNs (igcf-PNs) is the subclass of forward-concurrent-free Petri nets (backward-concurrent-free, resp.) which are gcf-PNs (igcf-PNs, resp.) with the additional property that each transition has *exactly* one incoming (outgoing, resp.) edge.

The classes of gcf-PN and igcf-PNs were first considered by Holt and Commoner [HC70]. Lien

[Lie76b] investigated termination properties of forward- and backward-concurrent-free Petri nets, relating properties like the property of being conservative or the existence of component-wise positive zero-loops with each other. Teruel and Silva [TS93; TS94; TS96] published a series of papers on equal conflict systems, a natural generalization of free-choice Petri nets, containing the classes of gcf-PNs and igcf-PNs. Amer-Yahia et al. [AY+99] presented approaches based on techniques of linear algebra to reason about properties of gcf-PNs and igcf-PNs. Recently, Delosme et al. [Del+13] provided a sufficient condition for gcf-PNs to be well-behaved (i. e., bounded and live) which can be checked in polynomial time. Morita and Watanabe [MW96] showed that the RecLFS problem of gss-PNs is **NP**-hard, even if restricted to gss-PNs with exactly two different edge multiplicities. Taoka and Watanabe [TW99] investigated RecLFS for a subclass of gss-PNs with cactus structure. It is noticeable that no completeness-results for the classical problems of our interest for the classes investigated in this chapter have been found so far. Here, we fill this gap.

This chapter is organized as follows. In Section 6.1, we show that the (zero-)reachability, boundedness, covering, equivalence, and containment problems are **PSPACE**-hard for gss-PNs with edge multiplicities $\{1, 2, 3\}$, or for gcf-PNs or igcf-PNs with edge multiplicities $\{1, 2\}$. Additionally, we show **PSPACE**-hardness for the liveness problem of gcf-PNs. In particular the strong lower bounds of these problems for gss-PNs can be surprising since almost all of these problems have very low complexity for (unweighted) S-systems (e. g., they are always bounded since they are conservative, and the reachability problem is decidable in polynomial time [Ha+12; Mur89]). To obtain these lower bounds, we use a similar approach to the one we used in Chapter 5 for conservative Petri nets, based on the simulation of polynomial space Turing machines. However, compared to that of Chapter 5, this approach is technically more involved in order to deal with the topological constraints on the nets.

In Section 6.2, we lay the foundation for applying our framework of Chapter 4 to gcf-PNs by using a permutation technique to obtain canonical permutations. Applying our framework to these canonical permutations yields canonical firing sequences for gcf-PNs. We use them to obtain canonical firing sequences for igcf-PNs which have even stronger properties compared to those for gcf-PNs. Applying our framework to the canonical firing sequences of gcf-PNs and igcf-PNs yields SLSRs of the reachability sets of gcf-PNs and igcf-PNs. The SLSRs for gcf-PNs (for igcf-PNs, resp.) have at most doubly (single, resp.) exponential encoding size in the size of the net.

In Section 6.3, we present the complexity results obtained by combining the lower bounds of Section 6.1 with the results from applying our framework to the canonical firing sequences of gcf-PNs and igcf-PNs. We show that the (zero-)reachability, boundedness, covering, and RecLFS problems of gcf-PNs, igcf-PNs and gss-PNs are **PSPACE**-complete, even under the restrictions on the edge multiplicities mentioned above. Additionally, we show that the liveness problem of gcf-PNs is **PSPACE**-complete. For the equivalence and containment problems, we obtain as upper bounds doubly exponential space for gcf-PNs and exponential space for igcf-PNs. In the last part of this section, we consider several computational problems involving home spaces. We use canonical firing sequences in gcf-PNs and igcf-PNs, as well as the SLSRs of their reachability sets, to check if a gcf-PN has a finite home space, to construct a minimal finite home space of such a net, provided it exists, and to decide if a given SLSR represents a home space of a given igcf-PN, all in doubly exponential space.

In Section 6.4, we consider a new class of commutative grammars called exponent-sensitive commutative grammars (ESCGs) which is closely related to gcf-PNs. Productions of such grammars are allowed to substitute a commutative word whose symbols are all the same by an arbitrary commutative word. In this sense, they are sensitive to context to a certain extent. However, since the substituted word can be longer than its replacement, the class of ESCGs is not contained in the class of context-sensitive commutative grammars (CSCGs) which has been studied by Huynh [Huy83]. On the other hand, not every CSCG is an ESCG since a single production of a CSCGcan substitute different symbols. We use our results for gcf-PNs to show that the uniform word problem of ESCGs is **PSPACE**-complete. In all cases we know of, the complexity of the uniform word problem of a class of commutative grammars is at least the complexity of the uniform word problem of the corresponding class of non-commutative grammars. In order to check if this pattern also holds for CSCGs, we additionally consider the corresponding class of (non-commutative) grammars, called exponent-sensitive grammars (ESG). The productions of these grammars can substitute a word whose symbols are all the same by an arbitrary word. Analogously to before, ESGs and context-sensitive grammars are incomparable. However, contrary to previous observations, we observe that any grammar has a normal form that is an ESG, which implies that the uniform word problem of ESGs is undecidable, and therefore has a higher complexity than the uniform word problem of ESCGs.

6.1 Lower bounds

We say that a gss-PN is *almost ordinary*, abbreviated by *a. o.*, if each edge multiplicity is in $\{1, 2, 3\}$. Furthermore, we call a gcf-PN or an igcf-PN *almost ordinary* if each edge multiplicity is in $\{1, 2\}$. In this section, we show how almost ordinary gss-PNs, gcf-PNs, and igcf-PNs can simulate polynomial space Turing machines. As a result we obtain **PSPACE**-hardness for many problems of interest, in most cases even **PSPACE**-hardness in the strong sense.

The reductions are similar to that used in [Jon+77] and in Chapter 5. However, since we cannot make use of out-communication and/or in-communication, our approach must be more sophisticated.

We first describe four transition gadgets G^{gss}, $\tilde{G}^{\text{gss}}$, G^{gcf}, G^{igcf} which are used to simulate TM-transitions (see Definition 5.5). These gadgets are building blocks for the Petri nets used to simulate polynomial space TMs. For each problem under consideration, the resulting Petri net is further enhanced with problem specific places and transitions. Each gadget has exactly two input places p^{Γ}_{in}, p^{Q}_{in}, two output places p^{Γ}_{out}, p^{Q}_{out}, and a number of control places p^{c}_1, p^{c}_2, p^{c}_3, p^{c}. In our application, an output place of one instance of a gadget can potentially be an input place of the same or some other instance of the gadget. Figure 6.2 illustrates these transition gadgets.

For the following constructions, we fix an arbitrary language $L \in \textbf{PSPACE}$ over some alphabet Σ, and a Turing machine $M = (Q, \Gamma, \Box, \Sigma, \delta, q_0, q_{\text{acc}})$ in standard form deciding L (see Definition 2.1). Let $x \in \Sigma^*$ be an input string for M. Let $\mathcal{P}_{\text{gss}} := \mathcal{P}^{G^{\text{gss}}}_{M,x}$, $\tilde{\mathcal{P}}_{\text{gss}} := \mathcal{P}^{\tilde{G}^{\text{gss}}}_{M,x}$, $\mathcal{P}_{\text{gcf}} := \mathcal{P}^{G^{\text{gcf}}}_{M,x}$, and $\mathcal{P}_{\text{igcf}} := \mathcal{P}^{G^{\text{igcf}}}_{M,x}$ (see Definition 5.6). By $\mathcal{P} = (P, T, F, \mu_0)$ we unspecifically refer to one of the nets $\mathcal{P}_{\text{gss}}$, $\tilde{\mathcal{P}}_{\text{gss}}$, $\mathcal{P}_{\text{gcf}}$, and $\mathcal{P}_{\text{igcf}}$.

Let the sets $P_{i,d}$ for $i \in [\ell_S]$ and $d \in \delta$ be defined as in Definition 5.6. Furthermore, we define the following sets of places for all $i \in [\ell_S]$:

$$P^{Q}_i := \bigcup_{q \in Q} \{p^{Q}_{i,q}\}, \quad P^{Q} := \bigcup_{j \in [\ell_S]} P^{Q}_j, \quad P^{\not Q}_i := \bigcup_{d \in \delta} P_{i,d} \setminus P^{Q}.$$

The membership of the places of the gadgets constituting $\mathcal{P}$ is illustrated in Figure 6.3. Note that $P^{\not Q}_1$, ..., $P^{\not Q}_{\ell_S}$, P^{Q} is a partition of P. By this construction, $\mathcal{P}_{\text{gss}}$ and $\tilde{\mathcal{P}}_{\text{gss}}$ are gss-PNs, $\mathcal{P}_{\text{gcf}}$ is a gcf-PN, and $\mathcal{P}_{\text{igcf}}$ is an igcf-PN.

Similar to the ordinary 1-conservative Petri nets used in Chapter 5, we use $\mathcal{P}$ to simulate M on input x. A subnet of $\tilde{\mathcal{P}}_{\text{gss}}$ is illustrated in Figure 6.4 (also compare Figure 5.2). Intuitively, it's probably not difficult to accept that $\mathcal{P}$ can indeed simulate M. However, to formally show that $\mathcal{P}$ is,

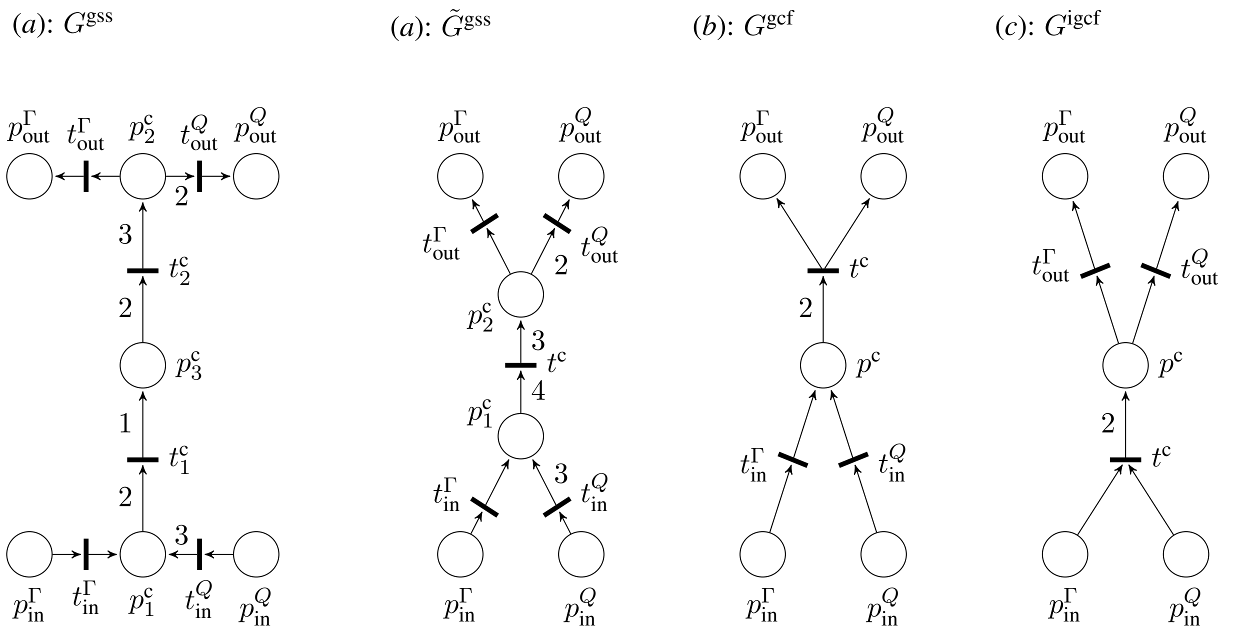


Figure 6.2: Transition gadgets for gss-PNs, gcf-PNs, and igcf-PNs

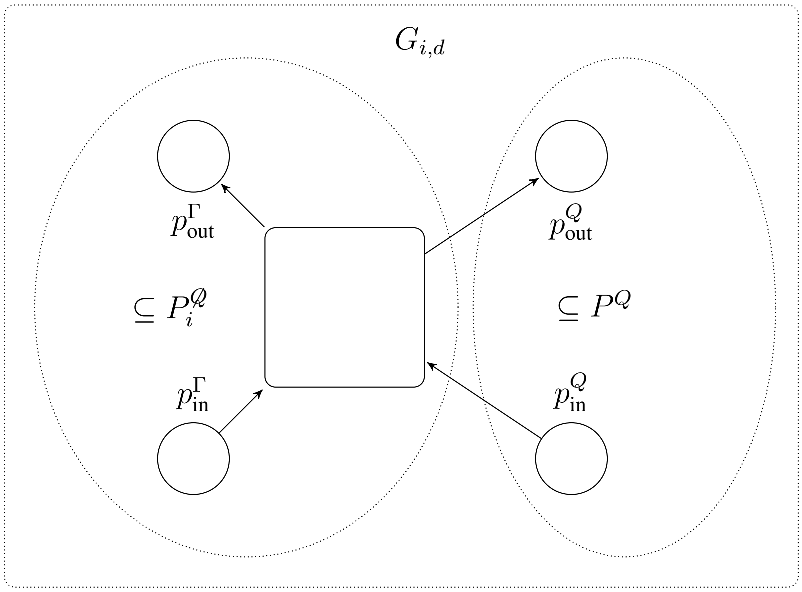


Figure 6.3: For each gadget $G_{i,d}$ of $\mathcal{P}$, the places $p^{Q}_{\text{in}}(G_{i,d})$ and $p^{Q}_{\text{out}}(G_{i,d})$ belong to P^{Q}, while all other places of $G_{i,d}$ belong to $P^{\mathcal{Q}}_i$.

with respect to reachability, not an over-approximation of M on x, we must discuss some technical issues. When reading the proofs of the lemmata in this section, the reader may find it helpful to repeatedly take Figure 6.4 into consideration.

In the following, we make extensive use the terminology and the marking μ_{acc} as defined in Definition 5.7. For a simpler description of the markings reachable in $\tilde{\mathcal{P}}_{\text{gss}}$, we characterize them by certain types. Let $T^{\mathcal{Q}}_i := \{t^{\Gamma}_{\text{in}}(\tilde{G}^{\text{gss}}_{i,d}), t^{\Gamma}_{\text{out}}(\tilde{G}^{\text{gss}}_{i,d}), t^{\text{c}}(\tilde{G}^{\text{gss}}_{i,d}) \mid d \in \delta\}$ denote the set of transitions adjacent only to places of $P^{\mathcal{Q}}_i$. Note that $P^{\mathcal{Q}}_i$ is exactly the set of places of the induced Petri net $\tilde{\mathcal{P}}_{\text{gss}}[T^{\mathcal{Q}}_i]$.

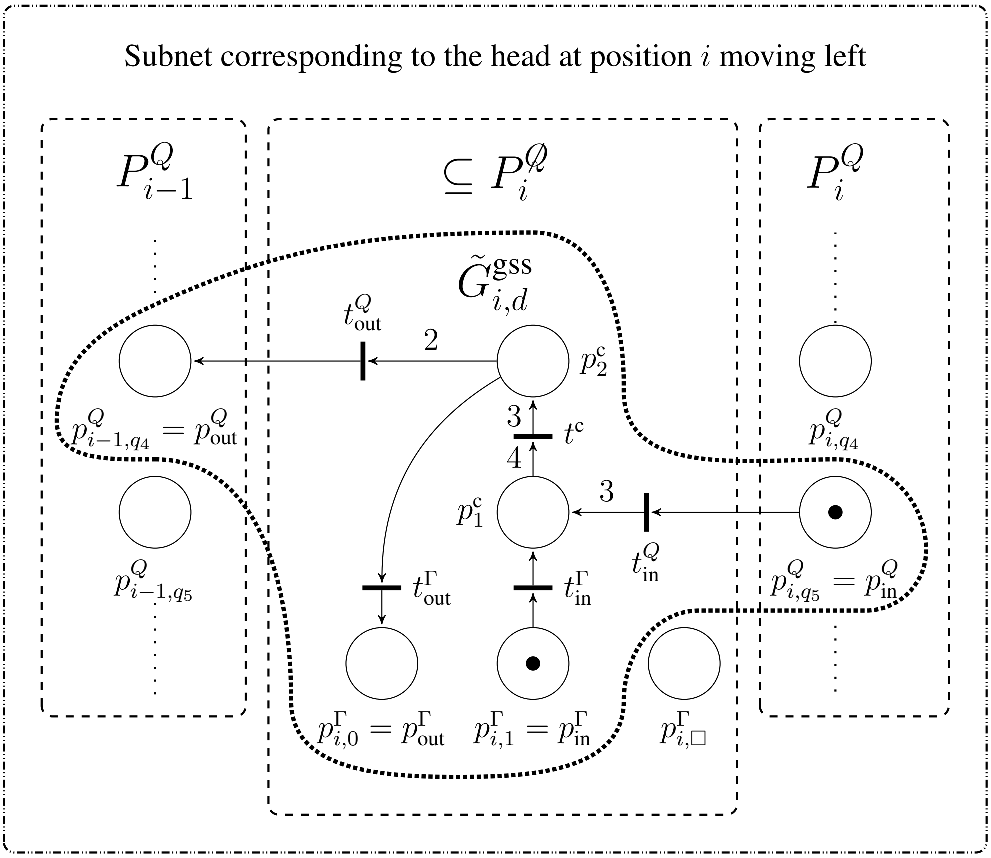


Figure 6.4: This figure illustrates a subnet of $\tilde{\mathcal{P}}_{\text{gss}}$ where the marking is a configuration marking. In this example, M has tape alphabet $\Gamma = \{0, 1, \Box\}$. Since p^Q_{i,q_5} is marked, M is in state q_5 and M's head is at position i with respect to the configuration of M corresponding to the marking. The dotted curve encompasses the gadget $\tilde{G}^{\text{gss}}_{i,d}$ where $d = (q_5, 1, q_4, 0, -1) \in \delta$ is an M-transition. In order for d to be executed, it requires that M is in state q_5, and that the tape contains the symbol 1 at the current position of the head. When d is executed, M switches into state q_4, writes 0 onto the tape, and moves left. For the corresponding configuration of M, this behavior is simulated by the gadget (see Lemma 6.3).

Definition 6.1.
Let μ be a marking reachable in $\tilde{\mathcal{P}}_{\text{gss}}$.

(*a*) μ is of type A if it satisfies the following properties: P^Q contains one token, and, for each $i \in [\ell_S]$, $P^{\not Q}_i$ contains one token.

(*b*) μ is of type B if it satisfies the following properties: P^Q is unmarked, and there are $i^* \in [\ell_S]$ and $d^* \in \delta$ such that,

- for each $i \in [\ell_S] \setminus \{i^*\}$, $P^{\not Q}_i$ contains one token, and
- there are an initial marking ν_0 of $\tilde{\mathcal{P}}_{\text{gss}}[T^{\not Q}_{i^*}]$ and a transition sequence ψ such that
  - $\nu_0 \triangleq 3 \cdot p^{\text{c}}_1(\tilde{G}^{\text{gss}}_{i^*,d^*}) + p$ for some $p \in P^{\not Q}_{i^*} \setminus \{p^{\text{c}}_1(\tilde{G}^{\text{gss}}_{i^*,d^*})\}$,
  - $\psi \in \left(T^{\not Q}_{i^*}\right)^*$, and

– $\nu_0 \xrightarrow{\psi} \mu[P^Q_{i^*}]$, where $\mu[P^Q_{i^*}]$ is the marking μ restricted to the places of $P^Q_{i^*}$.

Intuitively, markings of type A correspond to a situation where the decision which M-transition should be simulated next is not made, yet. Markings of type B correspond to a situation where the Petri net is has not finished simulating an M-transition (and possibly isn't even able to do so, e. g., if a "wrong" transition is fired which leads to a marking from which no configuration marking can be reached anymore). The marking of the subnet corresponding to the position i^* at which currently the M-transition d^* is simulated can be obtained by starting with 3 tokens at $p^c_1(\tilde{G}^{gss}_{i^*,d^*})$ and with 1 token at some other place of $P^Q_{i^*}$, and then firing transitions that transfer tokens *within* $P^Q_{i^*}$.

Lemma 6.2. *Let σ be a firing sequence of $\tilde{\mathcal{P}}_{gss}$ with an initial marking μ_0 of type A, let μ_i, $i \in [|\sigma|]$, be defined by $\mu_0 \xrightarrow{\sigma[..i]} \mu_i$, let $T_2 := \{t^Q_{in}(\tilde{G}^{gss}_{i,d}), t^Q_{out}(\tilde{G}^{gss}_{i,d}) \mid i \in [\ell_S], d \in \delta\}$, let k be the number of occurrences of transitions of T_2 in σ, and let i_j denote position of the j-th such occurrence in σ. Furthermore, let $i_0 := 0$ and $i_{k+1} := |\sigma| + 1$. Then, the following properties hold:*

(*a*) *for each odd $j \in [k]$,*

- *there are $i^* \in [\ell_S]$, $d^* \in \delta$ such that $\sigma_{[i_j]} = t^Q_{in}(\tilde{G}^{gss}_{i^*,d^*})$, and if $j+1 \in [k]$, then $\sigma_{[i_{j+1}]} = t^Q_{out}(\tilde{G}^{gss}_{i^*,d^*})$,*
- *for all $i \in [i_j, i_{j+1} - 1]$, μ_i is of type B,*

(*b*) *for each even $j \in [0, k]$,*

- *for all $i \in [i_j, i_{j+1} - 1]$, μ_i is of type A,*
- *for all $i \in [i_j + 1, i_{j+1} - 1]$, $\sigma_{[i]} \notin T_2 \cup \{t^c(\tilde{G}^{gss}_{i',d}) \mid i' \in [\ell_S], d \in \delta\}$, and*

(*c*) *for each $i^* \in [\ell_S]$, $d^* \in \delta$, $i \in [|\sigma|]$ with $\sigma_{[i]} = t^c(\tilde{G}^{gss}_{i^*,d^*})$, we observe $i > i_1$, and $\sigma_{[i_j]} = t^Q_{in}(\tilde{G}^{gss}_{i^*,d^*})$ for the largest $j \in [k]$ with $i_j < i$.*

Proof. This lemma can be proven by induction on the length ℓ of σ: For $\ell = 0$, all properties are satisfied. Assume the lemma holds for all firing sequences of length at most $\ell - 1$.

First, assume that $\mu_{\ell-1}$ is of type A. If $\sigma_{[\ell]} = t^Q_{in}(G_{i,d})$ for some $i \in [\ell_S]$ and $d \in \delta$, then μ_ℓ is of type B. Moreover, all properties hold. Otherwise, we observe $\sigma_{[\ell]} \in \{t^\Gamma_{in}(\tilde{G}^{gss}_{i,d}), t^\Gamma_{out}(\tilde{G}^{gss}_{i,d}) \mid i \in [\ell_S], d \in \delta\}$ which implies that μ_ℓ is of type A again. As before, all properties hold.

In the following, we assume that $\mu_{\ell-1}$ is of type B, and let, for $\mu_{\ell-1}$, ψ, i^* and d^* be as in Definition 6.1.

Suppose $\sigma_{[\ell]} \notin \{t^Q_{out}(G_{i,d}) \mid i \in [\ell_S], d \in \delta\}$. Then, we observe $\sigma_{[\ell]} \in \{t^c(G_{i^*,d}) \mid d \in \delta\} \cup \{t^\Gamma_{in}(G_{i,d}), t^\Gamma_{out}(G_{i,d}) \mid i \in [\ell_S], d \in \delta\}$. (Note that this observation concludes the proof of (*c*).) This means that $\sigma_{[\ell]}$ does neither mark P^Q nor transfers tokens from P^Q_i to $P^Q_{i'}$ for any two different i, $i' \in [\ell_S]$. Therefore, $\sigma_{[\ell]}$ transfers tokens within P^Q_i for some $i \in [\ell_S]$. If $i \neq i^*$ for this i, then $\sigma_{[\ell]} \in \{t^\Gamma_{in}(\tilde{G}^{gss}_{i,d}), t^\Gamma_{out}(\tilde{G}^{gss}_{i,d}) \mid d \in \delta\}$ which implies that μ_ℓ is of type B again. Moreover, all properties hold. If $i = i^*$, then, by Definition 6.1, we obtain a marking of type B again, and all properties hold.

Now, suppose $\sigma_{[\ell]} \in \{t^Q_{out}(G_{i,d}) \mid i \in [\ell_S], d \in \delta\}$. Then, $\sigma_{[\ell]} = t^Q_{out}(G_{i^*,d^*})$. Furthermore, $t^\Gamma_{out}(G_{i^*,d^*})$ can appear in ψ at most once: if ψ would contain $t^\Gamma_{out}(G_{i^*,d^*})$ at least twice, then directly after the second occurrence there is at most one token at $p^Q_{out}(G_{i^*,d^*})$, and there are in total three tokens within $P^Q_{i^*}$. Since a transition $t^c(G_{i^*,d})$ for some $d \in \delta$ must be fired in order to increase the number of tokens within $P_{i^*}^Q$ but all these transitions are prevented by insufficient number of tokens, $t_{\text{out}}^Q(G_{i^*,d^*})$ wouldn't occur in ψ, a contradiction. Therefore, $\mu_{\ell-1}(p_{\text{out}}^Q(G_{i^*,d^*})) \geq 2$ and there are in total three tokens at $\mu_{\ell-1}$ within $P_{i^*}^Q$. Consequently, μ_ℓ is of type A. Moreover, all properties hold. □

We now show how firing sequences of $\mathcal{P}_{\text{gss}}$ leading to configuration markings and computation paths of M on input x correspond to each other.

Lemma 6.3. *Let μ, μ' be configuration markings of $\mathcal{P}_{gss}$.*

(*a*) *Let $d_1 \cdots d_\ell$ be a computation path of M leading from conf(μ) to conf(μ'). Then, there is a transition sequence σ leading from μ to μ' such that $|\sigma| = 6\ell$, and, for all $j \in [\ell]$, there is a gadget $G := G_{i,d_j}^{gss}$ for some $i \in [\ell_S]$ such that $\varphi_{[7i-6..7i]} = t_{in}^\Gamma(G) \cdot t_{in}^Q(G) \cdot t_1^c(G) \cdot t_1^c(G) \cdot t_2^c(G) \cdot t_{out}^\Gamma(G) \cdot t_{out}^Q(G)$.*

(*b*) *Let σ be a transition sequence leading from μ to μ'. Furthermore, let ℓ denote the number of occurrences of a transition of $\{t_2^c(G_{i,d}^{gss}) \mid i \in [\ell_S], d \in \delta\}$ within σ, and let d_j, $j \in [\ell]$, be defined in such a way that the j-th occurrence is $t_2^c(G_{i,d_j}^{gss})$ for some $i \in [\ell_S]$. Then, the following properties hold:*

- *there is a permutation φ of σ enabled at μ such that $|\varphi| = 6\ell$ and, for all $j \in [\ell]$, there is a gadget $G := G_{i,d_j}^{gss}$ for some $i \in [\ell_S]$ such that $\varphi_{[7i-6..7i]} = t_{in}^\Gamma(G) \cdot t_{in}^Q(G) \cdot t_1^c(G) \cdot t_1^c(G) \cdot t_2^c(G) \cdot t_{out}^\Gamma(G) \cdot t_{out}^Q(G)$, and*
- *$d_1 \cdots d_\ell$ is a computation path of M leading from conf(μ) to conf(μ').*

Proof. We prove the lemma for $\tilde{\mathcal{P}}_{\text{gss}}$ and the set $\{t^c(\tilde{G}_{j,d}^{\text{gss}}) \mid j \in [\ell_S], d \in \delta\}$ instead since this proof is shorter than the proof for the original lemma. Then, however, the original lemma immediately follows because of the following two observations: for each $i \in [\ell_S]$ and $d \in \delta$, two occurrences of $t_1^c(G_{i,d}^{\text{gss}})$ and one occurrence of $t_2^c(G_{i,d}^{\text{gss}})$ can always be bundled (by permuting the sequence appropriately), and can simulate one occurrence of $t^c(\tilde{G}_{i,d}^{\text{gss}})$, and vice versa. Furthermore, the number of occurrences of $t_2^c(G_{i,d}^{\text{gss}})$ must be exactly twice the number of occurrences of $t_1^c(G_{i,d}^{\text{gss}})$ in order to reach a configuration marking again (otherwise, the transition sequence would have a non-zero displacement at some control place).

Proof for (a): Consider an M-transition $d \in \delta$ that is executed at configuration conf(μ), where the head of M is at position i, yielding a configuration C. Let $G := \tilde{G}_{i,d}$. Firing $t_{\text{in}}^\Gamma(G) \cdot t_{\text{in}}^Q(G) \cdot t^c(G) \cdot t_{\text{out}}^\Gamma(G) \cdot t_{\text{out}}^Q(G)$ at μ yields a marking μ'' with $C = \text{conf}(\mu'')$. We use this argument for the all M-transitions of the computation path.

Proof for (b): In the following, we will often make use of Lemma 6.12 without explicitly referring to it. For $|\sigma| = 0$, the claim holds. Now, let $|\sigma| > 0$, and assume that the claim holds for all transition sequences of length less than $|\sigma|$ that lead from a configuration marking to a configuration marking.

Let k and i_j, $j \in [0, k+1]$ be defined as of Lemma 6.2. For an odd $j \in [k-1]$, we call the sequence $\sigma_{[i_j..i_{j+1}]}$ a δ-block. Sequence σ must contain a transition $t^c(G_{i,d})$ for some $i \in [\ell_S], d \in \delta$, since $|\sigma| > 0$, and otherwise μ' wouldn't be a configuration marking. Let $\sigma_{[r]}$ be the first occurrence $t^c(G)$ of such a transition where $G := G_{i^*,d_1}$ for some $i^* \in [\ell_S]$. By Lemma 6.2, $\sigma_{[r]}$ is contained in some δ-block. We observe that it's actually the first δ-block $\sigma_{[i_1..i_2]}$ since the each δ-block must contain a transition $t^c(G_{i,d})$ for some $i \in [\ell_S]$ and $d \in \delta$ in order to enable the last transition of the δ-block.

All transitions of $\sigma_{[..r]}$ belonging to gadgets other than G cannot increase the number of tokens at $p^{\Gamma}_{\text{in}}(G)$ or $p^{Q}_{\text{in}}(G)$. Therefore, they are not useful in order to enable $t^{\text{c}}(G)$, and can be pushed behind $\sigma_{[r]}$ (preserving their relative order).

Let α be the resulting firing sequence. Then, either $\alpha_{[..3]} = t^{Q}_{\text{in}}(G) \cdot t^{\Gamma}_{\text{in}}(G) \cdot t^{\text{c}}(G)$ or $\alpha_{[..3]} = t^{\Gamma}_{\text{in}}(G) \cdot t^{Q}_{\text{in}}(G) \cdot t^{\text{c}}(G)$ holds. W. l. o. g., assume that the latter is the case (we simply permute the first two transitions if necessary). Note that $\alpha_{[..i_2]}$ is a δ-block. With the same argument as in the proof of Lemma 6.2, we find that $t^{\Gamma}_{\text{out}}(G)$ occurs at most once in this δ-block.

Assume that $t^{\Gamma}_{\text{out}}(G)$ occurs once in this δ-block. Then we can push this occurrence and $\alpha_{[i_2]} = t^{Q}_{\text{out}}(G)$ to the 4th and 5th position, resulting in a transition sequence β enabled at μ, where $\beta_{[..5]} = t^{\Gamma}_{\text{in}}(G) \cdot t^{Q}_{\text{in}}(G) \cdot t^{\text{c}}(G) \cdot t^{\Gamma}_{\text{out}}(G) \cdot t^{Q}_{\text{out}}(G)$. The marking μ_β with $\mu \xrightarrow{\beta_{[..5]}} \mu_\beta \xrightarrow{\beta_{[6..]}} \mu'$ is a configuration marking. We observe that the M-transition d_1 leads from conf(μ) to conf(μ_β). Applying the induction hypothesis to μ_β, μ, and $\beta_{[6..]}$ proves the lemma.

Next, assume that $t^{\Gamma}_{\text{out}}(G)$ does not occur in the δ-block $\alpha_{[..i_2]}$. Let μ_α denote the marking with $\mu \xrightarrow{\alpha_{[..i_2]}} \mu_\alpha$. If $t^{\Gamma}_{\text{out}}(G)$ occurs within $\alpha_{[4..i_2]}$, then $\mu_\alpha(p^{\text{c}}(G)) \geq 4$, and μ_α is neither of type A nor of type B, a contradiction to Lemma 6.2. Therefore, $t^{\Gamma}_{\text{out}}(G)$ does not occur within $\alpha_{[4..i_2]}$, and we observe $\mu_\alpha(p^{\text{c}}(G)) = 1$. This token must be removed by either $t^{\Gamma}_{\text{out}}(G)$ or $t^{Q}_{\text{out}}(G)$ to reach the configuration marking μ. Let $\alpha_{[q]}$, $q \in [i_2+1, \ell_S]$, be the first occurrence of such a transition. If $\alpha_{[q]} = t^{\Gamma}_{\text{out}}(G)$, then the same argument as in the last paragraph proves the lemma.

Hence, assume $\alpha_{[q]} = t^{Q}_{\text{out}}(G)$. In order to provide the tokens for this transition, $\alpha_{[i_2+1..q-1]}$ must contain the transition $t^{\text{c}}(G)$ at least once. However, the marking ν which we obtain after the first occurrence of $t^{\text{c}}(G)$ within $\alpha_{[i_2+1..q-1]}$ satisfies $\nu(p^{\text{c}}(G)) = 4$. Hence, ν is neither of type A nor of type B, a contradiction to Lemma 6.2. In total, we find that $t^{\Gamma}_{\text{out}}(G)$ must occur once in the δ-block $\alpha_{[..i_2]}$, which, as shown before, proves the lemma. □

We can use Lemma 6.3 to show analogue lemmata for $\mathcal{P}_{\text{gcf}}$ and $\mathcal{P}_{\text{igcf}}$ very easily.

Lemma 6.4. *Let μ, μ' be configuration markings of $\mathcal{P}_{gcf}$.*

(*a*) *Let $d_1 \cdots d_\ell$ be a computation path of M leading from conf(μ) to conf(μ'). Then, there is a transition sequence σ leading from μ to μ' such that $|\sigma| = 3\ell$, and, for all $j \in [\ell]$, there is a gadget $G := G^{gcf}_{i,\, d_j}$ for some $i \in [\ell_S]$ such that $\varphi_{[3i-2..3i]} = t^{\Gamma}_{in}(G) \cdot t^{Q}_{in}(G) \cdot t^{c}(G)$.*

(*b*) *Let σ be a transition sequence leading from μ to μ'. Furthermore, let ℓ denote the number of occurrences of a transition of $\{t^{c}(G^{gcf}_{i,\, d}) \mid i \in [\ell_S], d \in \delta\}$ within σ, and let d_j, $j \in [\ell]$, be defined in such a way that the j-th occurrence is $t^{c}(G^{gcf}_{i,\, d_j})$ for some $i \in [\ell_S]$. Then, the following properties hold:*

- *there is a permutation φ of σ enabled at μ such that $|\varphi| = 3\ell$ and, for all $j \in [\ell]$, there is a gadget $G := G^{gcf}_{i,\, d_j}$ for some $i \in [\ell_S]$ such that $\varphi_{[3i-2..3i]} = t^{\Gamma}_{in}(G) \cdot t^{Q}_{in}(G) \cdot t^{c}(G)$, and*
- *$d_1 \cdots d_\ell$ is a computation path of M leading from conf(μ) to conf(μ').*

Proof. Proof for (*a*): The argument is the same as in the proof of Lemma 6.3 except that we consider the firing sequence $t^{\Gamma}_{\text{in}}(G) \cdot t^{Q}_{\text{in}}(G) \cdot t^{\text{c}}(G)$.

Proof for (*b*): The important observation is that $t^{\text{c}}_1(G) \cdot t^{\text{c}}_2(G) \cdot t^{\Gamma}_{\text{out}}(G) \cdot t^{Q}_{\text{out}}(G)$ of $\mathcal{P}_{\text{gss}}$ can simulate $t^{\text{c}}(G')$ of $\mathcal{P}_{\text{gcf}}$ and vice versa, where G and G' are the corresponding gadgets. We construct the transition sequence σ' of $\mathcal{P}_{\text{gss}}$ where σ' corresponds to σ and leads from ν to ν' which are the markings of $\mathcal{P}_{\text{gss}}$ corresponding to μ and μ'. We apply Lemma 6.3 to σ' to obtain both the permutation φ' of σ' as well as the computation path $d_1 \cdots d_\ell$. Last, we construct the transition sequence φ of $\mathcal{P}_{\text{gcf}}$ which corresponds to φ' and is a permutation of σ. The sequence φ and the computation path $d_1 \cdots d_\ell$ satisfy (b). □

Lemma 6.5. *Let μ, μ' be configuration markings of $\mathcal{P}_{igcf}$.*

(a) *Let $d_1 \cdots d_\ell$ be a computation path of M leading from conf(μ) to conf(μ'). Then, there is a transition sequence σ leading from μ to μ' such that $|\sigma| = 3\ell$, and, for all $j \in [\ell]$, there is a gadget $G := G^{igcf}_{i,\,d_j}$ for some $i \in [\ell_S]$ such that $\varphi_{[3i-2..3i]} = t^c(G) \cdot t^{\Gamma}_{out}(G) \cdot t^{Q}_{out}(G)$.*

(b) *Let σ be a transition sequence leading from μ to μ'. Furthermore, let ℓ denote the number of occurrences of a transition of $\{t^c(G^{igcf}_{i,\,d}) \mid i \in [\ell_S], d \in \delta\}$ within σ, and let d_j, $j \in [\ell]$, be defined in such a way that the j-th occurrence is $t^c(G^{igcf}_{i,\,d_j})$ for some $i \in [\ell_S]$. Then, the following properties hold:*

- *there is a permutation φ of σ enabled at μ such that $|\varphi| = 3\ell$ and, for all $j \in [\ell]$, there is a gadget $G := G^{igcf}_{i,\,d_j}$ for some $i \in [\ell_S]$ such that $\varphi_{[3i-2..3i]} = t^c(G) \cdot t^{\Gamma}_{out}(G) \cdot t^{Q}_{out}(G)$, and*
- *$d_1 \cdots d_\ell$ is a computation path of M leading from conf(μ) to conf(μ').*

Proof. Proof for (a): The argument is the same as in the proof of Lemma 6.3 except that we consider the firing sequence $t^c(G) \cdot t^{\Gamma}_{\text{out}}(G) \cdot t^{Q}_{\text{out}}(G)$.

Proof for (b): We can use the same strategy as in Lemma 6.4. The only significant difference is that, in this case, $t^{\Gamma}_{\text{in}}(G) \cdot t^{Q}_{\text{in}}(G) \cdot t^{c}_{1}(G) \cdot t^{c}_{2}(G)$ of $\mathcal{P}_{\text{gss}}$ can simulate $t^c(G')$ of $\mathcal{P}_{\text{igcf}}$ and vice versa, where G and G' are the corresponding gadgets. The rest of the argumentation is analogous. □

In the following, we stepwise extend $\mathcal{P}$ to Petri nets $\mathcal{P}_1$, $\mathcal{P}_2$, and $\mathcal{P}_3$. The final Petri net $\mathcal{P}_3$ is then used to perform the reductions to our problems of interest. Each net is obtained from the previous one by adding a new gadget, or a number of transitions. The relationship between these nets is illustrated in Figure 6.5. This illustration can be helpful to understand the constructions and observations, which follow now.

Let $\mathcal{P}_1$ denote the Petri net obtained in the following way from $\mathcal{P}_{\text{gss}}$, $\mathcal{P}_{\text{gcf}}$, or $\mathcal{P}_{\text{gcf}}$. We add an instance G_1 of gadget $G^{(1)}_{\ell_S+1}$ illustrated in Figure 6.6 to the net such that, for each $i \in [\ell_S]$, $p^{\Gamma}_{i,\Box} = p^{(1)}_i(G_1)$, and $p_{\text{acc}} = p^{(1)}_{\ell_S+1}(G_1)$. The initial marking of $\mathcal{P}_1$ at all old places equals the initial marking of $\mathcal{P}_{\text{gss}}$, $\mathcal{P}_{\text{gcf}}$, or $\mathcal{P}_{\text{gcf}}$, respectively. Furthermore, all new places of the gadget G_1 with the exception of $p^{(1)}(G_1)$ are marked by one token. For the following lemma, we use the observation that firing sequences and computation paths correspond to each other in the sense shown above.

Lemma 6.6. *Let $\mu :\triangleq (\ell_S + 1) \cdot p^{(1)}(G_1)$ be a marking of $\mathcal{P}_1$. Then, the following are equivalent:*

(a) *μ is reachable in $\mathcal{P}_1$,*

(b) *μ is reachable in $\mathcal{P}_1$ by some firing sequence that contains every transition t of every gadget $G_{i,\,d}$, $i \in [\ell_S]$, $d \in \delta$, at most $2\ell_T$ times,*

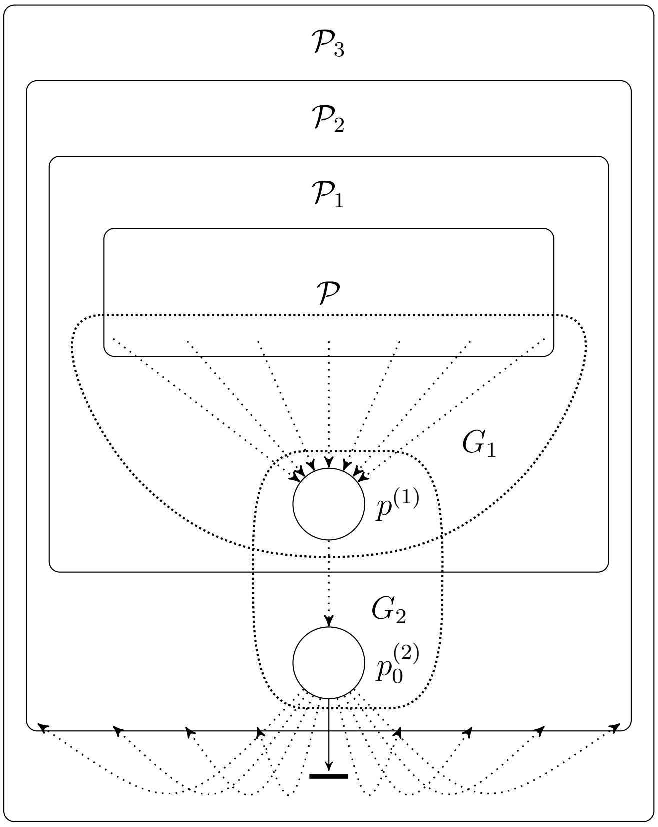


Figure 6.5: Solid borders encompass $\mathcal{P}, \mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3$, dotted borders encompass the gadgets, and dotted lines indicate paths. $\mathcal{P}_1$ emerges from $\mathcal{P}$ by adding gadget G_1. This gadget consists of paths starting at certain places of $\mathcal{P}$ and ending at a place $p^{(1)}$. These paths are used to transfer all tokens from μ_{acc} to $p^{(1)}$. $\mathcal{P}_2$ emerges from $\mathcal{P}_1$ by adding gadget G_2. This gadget consists of a single path from $p^{(1)}$ to a place $p_0^{(2)}$, and is used to convert all tokens at $p^{(1)}$ into a single token at $p_0^{(2)}$. $\mathcal{P}_3$ emerges from $\mathcal{P}_2$ by adding a number of transitions which are used to increase the number of tokens at $p_0^{(2)}$, distribute them over the whole net $\mathcal{P}_3$, or empty $p_0^{(2)}$ and subsequently the whole net.

(*c*) *some marking μ' with $\mu'(p^{(1)}(G_1)) \geq \ell_S + 1$ is reachable in $\mathcal{P}_1$,*

(*d*) *M accepts x.*

Proof. Let σ be a firing sequence of $\mathcal{P}_1$ leading to some marking μ'. W.l.o.g., we assume that $\sigma = \alpha \cdot \beta$, where all occurrences of transitions of G_1 constitute the suffix β of σ since otherwise, we can permute σ by pushing these transitions to the end of the sequence (while maintaining their relative order), and obtain again a firing sequence. Let μ_α be the marking reached by α. We observe the upper bound $\mu'(p^{(1)}(G_1)) \leq \sum_{i \in [\ell_S+1]} \left\lfloor \frac{\Psi(\sigma)(t_i^{(1)}(G_1))+3}{4} \right\rfloor$. The reason is that after firing $t_i^{(1)}(G_1)$ once, we can increase the number of tokens at $p^{(1)}(G_1)$ by 1. After that, we need four additional occurrences of $t_i^{(1)}(G_1)$ to do the same.

If $\mathcal{P}_1$ were constructed from $\tilde{\mathcal{P}}_{\text{gss}}$, then, by Lemma 6.2, μ_α could not have more than 3 tokens at any place $p_i^{(1)}(G_1)$, $i \in [\ell_S + 1]$. We want to argue that this also holds if $\mathcal{P}_1$ is constructed from $\mathcal{P}_{\text{gss}}$. To this end, note that for every marking ν reachable in $\mathcal{P}_{\text{gss}}$ there is a marking ν' reachable in $\tilde{\mathcal{P}}_{\text{gss}}$ such that the projection of ν' onto all input and output places covers the corresponding projection of ν. This can be shown in a similar fashion as the Lemma 6.3 but we won't go into detail here.

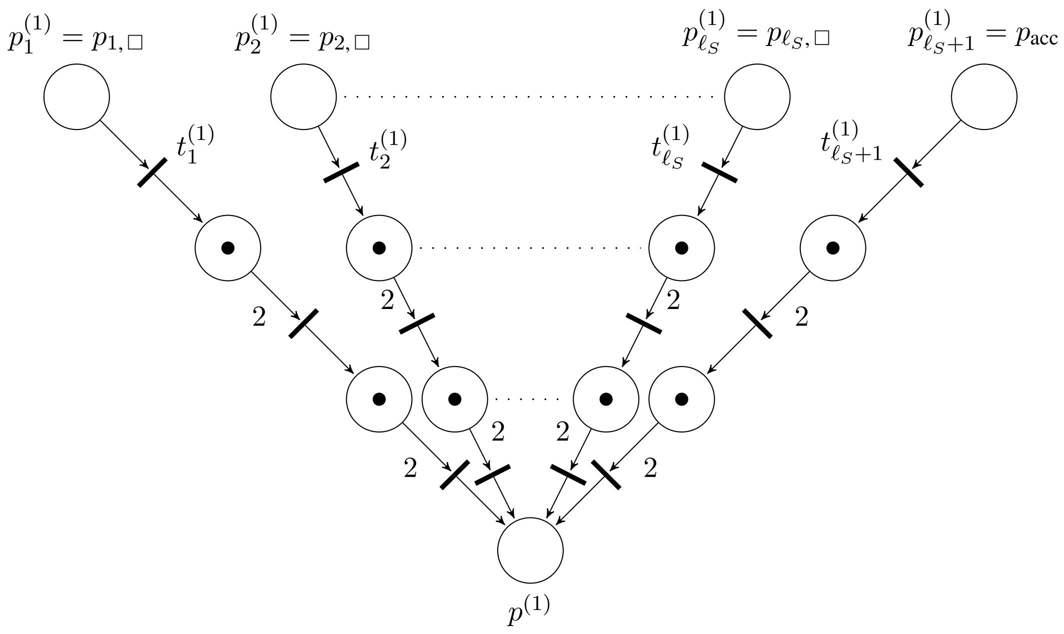


Figure 6.6: Instance G_1 of gadget $G_{\ell_S+1}^{(1)}$

Assume that $\mathcal{P}_1$ was constructed from $\mathcal{P}_{\text{gss}}$. As argued above, μ_α cannot have more than 3 tokens at any place $p_i^{(1)}(G_1)$, $i \in [\ell_S + 1]$. Hence, there is at most one occurrence of each $t_i^{(1)}(G_1)$, $i \in [\ell_S + 1]$, in β which implies $\mu'(p^{(1)}(G_1)) \leq \ell_S + 1$. By this observation, the following are equivalent.

- $\mu'(p^{(1)}(G_1)) \geq \ell_S + 1$,
- $\mu'(p^{(1)}(G_1)) = \ell_S + 1$,
- β contains each transition of gadget G_1 exactly once,
- $\mu' = \mu$,
- μ_α restricted to the places of $\mathcal{P}_{\text{gss}}$ equals μ_{end}, and
- M accepts x.

Furthermore, if M accepts x, then, by the fact that M is in standard form, and by Lemmata 6.3, 6.4, and 6.5, there is a firing sequence of length at most $2\ell_T$ leading to μ_{acc}. This proves the lemma for this case.

Now, assume that $\mathcal{P}_1$ was constructed from $\mathcal{P}_{\text{gcf}}$ or $\mathcal{P}_{\text{igcf}}$. The total number of tokens cannot change by firing a transition of $\mathcal{P}_{\text{gcf}}$ or $\mathcal{P}_{\text{igcf}}$. This means that the total number of tokens at μ_α is $\ell_S + 1$. This together with the upper bound for $\mu'(p^{(1)})$ implies an analogous equivalence like before which proves the lemma. $\Box$

We now modify the net $\mathcal{P}_1$ even further to obtain the Petri net $\mathcal{P}_2$. Let k denote the smallest power of 2 which is at least $\ell_S + 1$. We add an instance G_2 of gadget $G_{\ell_S+1}^{(2)}$ to the net which consists of a single path $(p^{(1)}(G_1) = p_{\text{ld}\,k}^{(2)}, t_{\text{ld}\,k}^{(2)}, p_{\text{ld}\,k-1}^{(2)} \ldots, t_1^{(2)}, p_0^{(2)})$ such that all edges to transitions have multiplicity 2, the edges emanating from transitions have multiplicity 1, and place $p_i^{(2)}$, $i \in [\text{ld}\,k]$, contains one token at the initial marking if the i-th least digit of the binary representation of $k - (\ell + 1)$ is 1, and no token otherwise. We remind the reader that $\ell_S \geq 3$ such that we do not need to consider the special case where $\ell_S + 1 = 1$. An example is illustrated in Figure 6.7.

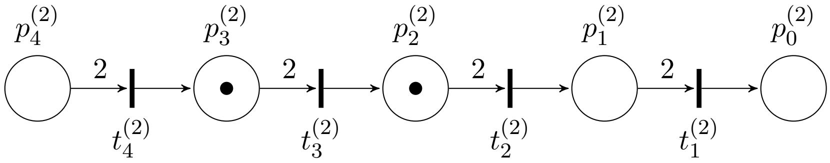


Figure 6.7: Example of gadget G_2 for the case $\ell_S + 1 = 10$, where $k - (\ell_S + 1) = 6 = 110_2$

Lemma 6.7. *Let $\mu :\triangleq p_0^{(2)}$ be a marking of $\mathcal{P}_2$. The following are equivalent:*

- *μ is reachable in $\mathcal{P}_2$,*
- *μ is reachable in $\mathcal{P}_2$ by some firing sequence that contains every transition t of every gadget $G_{i,d}$, $i \in [\ell_S]$, $d \in \delta$, at most $2\ell_T$ times,*
- *some marking μ' with $\mu(p_0^{(2)}) \geq 1$ is reachable in $\mathcal{P}_2$, and*
- *M accepts x.*

Proof. $p_0^{(2)}$ can be marked if and only if at least $\ell_S + 1$ tokens can be transferred to $p^{(1)}(G_1)$. By Lemma 6.6, the claim follows. □

Let $\mathcal{P}_3 = (P, T, F, \mu_0)$ be the Petri net that is obtained from $\mathcal{P}_2$ by adding, for each place p of $\mathcal{P}_2$, a distinguished transition $t_p^{(3)}$ with $p_0^{(2)} \xrightarrow{t_p^{(3)}} 2p$, as well as a distinguished transition $t^{(3)}$ with $p_0^{(2)} \xrightarrow{t^{(3)}} 0$, which can be used to remove tokens from $p_0^{(2)}$. This is the final Petri net we use to show **PSPACE**-hardness for the problems of interest.

Lemma 6.8. *The following problems are* **PSPACE***-hard in the strong sense, even if restricted to either a. o. gss-PNs, a. o. gcf-PNs, or a. o. igcf-PNs:*

- *zero-reachability,*
- *reachability,*
- *boundedness,*
- *SU,*
- *SIU, and*
- *covering.*

Proof. We first observe that $\mathcal{P}_3$ satisfies the restrictions of the lemma. Furthermore, $\mathcal{P}_3$ can be constructed in time polynomial in the size $|x|$ of the input for the decision problem L. We consider the problems one after another.

zero-reachability/reachability: The tokens of $\mathcal{P}_3$ can only be completely removed by transition $t^{(3)}$. By Lemma 6.7, $t^{(3)}$ can only fire if the marking $\mu :\triangleq p_0^{(2)}$ is reachable. On the other hand, if μ is reachable, then firing $t^{(3)}$ once at μ yields the empty marking. Therefore, by the same lemma, the empty marking is reachable if and only if M accepts x.

boundedness/SU/SIU: First note that, by Lemma 6.2 and by the fact that $\mathcal{P}_{\text{gcf}}$ and $\mathcal{P}_{\text{igcf}}$ are conservative, $\mathcal{P}_3$ is bounded if $p_0^{(2)}$ cannot be marked. On the other hand, each place of $\mathcal{P}_3$ is unbounded if $p_0^{(2)}$ can be marked. By Lemma 6.7, $p_0^{(2)}$ can be marked if and only if M accepts x.

covering: By Lemma 6.7, the marking which has one token at $p_0^{(2)}$ and is empty at all other places can be covered if and only if M accepts x. □

By Lemma 2.20, the RecLFS problem cannot be **PSPACE**-hard in the strong sense if **NP** $\neq$ **PSPACE**. Similarly to Theorem 5.12 for conservative Petri nets, we can, however, show that it's **PSPACE**-hard (in the weak sense) and subsequently **PSPACE**-complete.

Theorem 6.9. *The RecLFS problem is* **PSPACE**-*complete, even if restricted to either a. o. gss-PNs, a. o. gcf-PNs, or a. o. igcf-PNs.*

Proof. By Lemma 2.19, the RecLFS problem is in **PSPACE**. To prove **PSPACE**-hardness, we use the same strategy as in Theorem 5.12 for ordinary 1-conservative Petri nets. Let Φ be a Parikh vector of $\mathcal{P}_3$ defined as follows:

- $\Phi(t) = 2\ell_T$ for all transitions t of the gadgets $G_{i,d}$, $i \in [\ell_S]$, $d \in \delta$,
- $\Phi(t) = 1$ for each transition t of the gadget G_1 (see Figure 6.6),
- $\Phi(t)$ for a transition t of the path (see Figure 6.7) is chosen in such a way that firing these transitions according to Φ transfers all $\ell_S + 1$ tokens at the input place of the path as well as all remaining tokens on the path to the output place $p_0^{(2)}$ of the path,
- $\Phi(t_p^{(3)}) = 2\ell_T \cdot |\delta|$ for all places p of the gadgets $G_{i,d}$, $i \in [\ell_S]$, $d \in \delta$, and
- $\Phi(t_{p_0^{(2)}}^{(3)}) = \ell_S \cdot |\delta| \cdot \ell_T \cdot |\delta|$.

If M does not accept x, then, by Lemma 6.7, the transition $t_{p_0^{(2)}}^{(3)}$ can never be fired, and Φ is not enabled. If M accepts x, then, by the same lemma, there is a Parikh vector $\Phi' \leq \Phi$ with $\Phi'(t_p^{(3)}) = 0$ for all $p \in P$, leading to a marking μ at which $p_0^{(2)}$ is marked. Now, $\Phi(t_p^{(3)})$ is, for each $p \in P$, large enough to flood the places of the gadgets $G_{i,d}$, $i \in [\ell_S]$, $d \in \delta$, with enough tokens to enable the remaining Parikh vector $\Phi - \Phi'$ at μ. Besides $\mathcal{P}_3$, also Φ can be constructed in polynomial time in $|x|$ from the fixed Turing machine M and the input x. □

As the last problem of this section, we consider the liveness problem. The difficulty is that the Petri nets $\mathcal{P}_{\text{gss}}$, $\tilde{\mathcal{P}}_{\text{gss}}$, $\mathcal{P}_{\text{gcf}}$, and $\mathcal{P}_{\text{gcf}}$ exhibit the possibility to make "mistakes", to fire "wrong" transitions leading to deadlocks (i. e., markings at which no transition is enabled), even if M accepts x. In the case of gcf-PNs, we can address this problem relatively easily.

Lemma 6.10. *M accepts x if and only if some marking μ with $\mu(p_{acc}) \geq 1$ is reachable in $\mathcal{P}_{gcf}$.*

Proof. By Lemma 6.4, the implication "⇒" immediately follows. The proof for the implication "⇐" is similar to that of Lemma 6.4. Assume that some marking at which p_{acc} is marked is reachable in $\mathcal{P}_{\text{gcf}}$. Let σ be some shortest firing sequence leading to such a marking μ. σ must contain a transition $t^{\text{c}}(G^{\text{gcf}}_{i,d})$ for some $i \in [\ell_S]$, $d \in \delta$ since we know that the last transition of σ is such a transition which additionally marks p_{acc}. Similarly to the proof of Lemma 6.4 (induced by the proof of Lemma 6.3), we can permute σ to a firing sequence α such that $\alpha_{[..3]}$ leads to a configuration marking: We pick the first occurrence of a transition $t^{\text{c}}(G^{\text{gcf}}_{i,d})$, $i \in [\ell_S]$, $d \in \delta$, and shift it together with $t^{\Gamma}_{\text{in}}(G^{\text{gcf}}_{i,d})$ and $t^{Q}_{\text{in}}(G^{\text{gcf}}_{i,d})$, which must precede $t^{\text{c}}(G^{\text{gcf}}_{i,d})$ in σ to enable it, to the front of the sequence. By repeatedly applying this argument, we find a permutation β of σ that leads to μ_{acc}. By Lemma 6.4, M accepts x. □

We remark that we could show an analogue lemma also for gss-PNs, whose proof would be similar to that of Lemma 6.3. This would simplify the proofs for almost all propositions of Lemma 6.8 since we could skip the constructions involving the gadgets G_1 and G_2, and only use a construction similar to that used to obtain $\mathcal{P}_3$ from $\mathcal{P}_2$. However, using this by itself is insufficient to show **PSPACE**-hardness for the zero-reachability problem. Recall that the asset of $\mathcal{P}_2$ is not only that we have the place $p^{(2)}_0$ which is an indicator place for acceptance of x by M, but also that there is a marking at which all places except $p^{(2)}_0$ are unmarked which is reachable if and only if M accepts x. The same cannot be said about $\mathcal{P}$ and p_{acc}. Therefore, $p^{(2)}_0$ has stronger properties than p_{acc} which we need for the zero-reachability problem. We continue with the discussion of the liveness problem of gcf-PNs.

Lemma 6.11. *The liveness problem of gcf-PNs is* **PSPACE***-hard in the strong sense, even if restricted to a. o. gcf-PNs.*

Proof. Let $\mathcal{P}'_{\text{gcf}} = (P', T', F', \mu'_0)$ denote the Petri net obtained in the following way from $\mathcal{P}_{\text{gcf}} = (P, T, F, \mu_0)$. Let k denote the smallest power of two that is at least $\ell_S + 1$. As in the construction of $\mathcal{P}_2$, we add a path $(p^{(2)}_{\text{ld}\,k}, t^{(2)}_{\text{ld}\,k}, p^{(2)}_{\text{ld}\,k-1} \ldots, t^{(2)}_1, p^{(2)}_0)$ whose initial marking is defined in such a way that the path can reach the marking at which $p^{(2)}_0$ contains one token and all other places of the path are unmarked if and only if at least $\ell + 1$ tokens from outside of the path are transferred to $p^{(2)}_{\text{ld}\,k}$ (under the premise that no tokens from outside of the path are transferred to some other place of the path). See again Figure 6.7 for an example. Now, we add the following transitions:

- for each place $p \in P$, a transition t with $p \overset{t}{\mapsto} p^{(2)}_{\text{ld}\,k}$,
- a transition t_{reset} with $p^{(2)}_0 \overset{t_{\text{reset}}}{\longmapsto} \mu'_0$, and
- for each place $p \in P'$, a transition t_p with $p_{\text{acc}} \overset{t_p}{\mapsto} p$.

If M does not accept x, then, by Lemma 6.10 and by the observation that even the new transitions can only remove tokens from the original places or mark them with μ'_0 again, p_{acc} can never be marked. (We remark that the original net $\mathcal{P}_{\text{gcf}}$ is 1-conservative, and we can therefore not use the original transitions to "cheat" by increasing the number of tokens and then using the path to obtain a marking that properly covers the initial marking.) This means that the transition $t_{p_{\text{acc}}}$ can never be fired, and $\mathcal{P}'_{\text{gcf}}$ is not live. On the other hand, if M accepts x, then, by the same lemma, p_{acc} can be marked. Consequently, for each reachable marking μ, there is a transition sequence containing only the new transitions that leads from μ to some marking $\mu' \geq \mu'_0$. This implies that, for each reachable marking μ, there is a transition sequence leading to some marking μ' at which p_{acc} is marked. Furthermore, we can use the new transitions at μ' to increase the number of tokens at any place of the net. Hence, for each reachable marking μ and each transition t, there is a reachable marking μ' such that t is enabled at μ', i. e., $\mathcal{P}'_{\text{gcf}}$ is live. Note that $\mathcal{P}'_{\text{gcf}}$ can be constructed in time polynomial in $|x|$ from the fixed Turing machine M and the input x. □

6.2 Canonical firing sequences and SLSRs of reachability sets

This section is dedicated to derive enabled canonical permutations for firing sequences of gcf-PNs such that each marking obtained when firing the permutation has polynomial encoding size in the size of the Petri net and the end marking. This allows us to apply our framework of Section 4 to obtain canonical firing sequences, and SLSRs of reachability sets. The canonical firing sequences of gcf-PNs can also be used to obtain canonical firing sequences for igcf-PNs. This allows an application of our framework also for the class of igcf-PNs. In the next section, we use the results of this section to obtain complexity results for our problems of interest.

In the context of gcf-PNs, we use the following variation of the Parikh map: For a transition sequence σ of a gcf-PN, $\Psi_{\text{first}}(\sigma) \in \{0,1\}^m$ is the 0-1-vector such that, for all transitions t, $\Psi_{\text{first}}(\sigma)(t) = 1$ if and only if $t \in \sigma$ and ${}^\bullet t' \neq {}^\bullet t$ for all transitions $t' \in \sigma$ in front of the first occurrence of t in σ.

As our first step, we collect a number of fundamental observations. The implication of the first one is that we can push certain transitions of a firing sequence to the front, obtaining again a firing sequence.

Lemma 6.12. *Let σ be a firing sequence of a gcf-PN (N, μ_0). If a transition $t \in \Psi_{\mathit{first}}(\sigma_{[i+1..]})$ is enabled at $\mu_0 + \Delta(\sigma_{[..i]})$, then $\sigma_{[..i]} \cdot t \cdot (\sigma_{[i+1..]} \mathbin{\dot{-}} t)$ is a firing sequence.*

Proof. Assume ${}^\bullet t \neq \emptyset$. Let $\sigma_{[j]}$, $j \geq i+1$, be the first occurrence of t in $\sigma_{[i+1..]}$, and assume that t is enabled at $\mu_0 + \Delta(\sigma_{[..i]})$. Then, t is enabled at $\mu_0 + \Delta(\sigma_{[..j-1]})$ since, by the choice of t, $\Delta(\sigma_{[..j-1]})({}^\bullet t) \geq \Delta(\sigma_{[..i]})({}^\bullet t)$. Furthermore, since ${}^\bullet\sigma_{[j-1]} \neq {}^\bullet t$, $\sigma_{[j-1]}$ is enabled at $\mu + \Delta(\sigma_{[..j-2]} \cdot t)$. If ${}^\bullet t = \emptyset$, a similar argumentation can be applied. By iteratively performing pairwise switches, we obtain the lemma. □

We will use this observation mainly in situations where we want to generate a firing sequence that contains only transitions that consume tokens from places with a large number of tokens. The following second observation is that a Parikh vector is enabled at some initial marking if this marking and the end marking is large enough at all places contained in the preset of the transitions used by the Parikh vector. We show it by using a permutation procedure nested within an induction.

Lemma 6.13. *Let (P, T, F) be a gcf-PN, σ a transition sequence, and μ, μ' markings with $\mu + \Delta(\sigma) = \mu'$ and $\mu(p)$, $\mu'(p) \geq W$ for all $p \in {}^\bullet\sigma$. Then, there is a permutation of σ enabled at μ (and leading to μ').*

Proof. For the empty sequence ϵ and all markings $\mu = \mu'$, the claim holds. Let σ be a transition sequence of length $k > 0$ and μ, μ' be markings satisfying the requirements.

Assume, the claim holds for all transition sequences of length less than k and for all markings satisfying the requirements. If ${}^\bullet\sigma = \emptyset$, then σ is enabled at μ, thus assume ${}^\bullet\sigma \neq \emptyset$. We initialize $\tilde{\sigma} \leftarrow \epsilon$ as the empty sequence, and $\bar{\sigma} \leftarrow \sigma$. As long as $|\tilde{\sigma}| < k$ and there is a place $p \in {}^\bullet\bar{\sigma}$ with $\Delta(\tilde{\sigma})(p) \geq 0$, we choose a transition $t \in \bar{\sigma}$ with ${}^\bullet t = p$, and set $\tilde{\sigma} \leftarrow \tilde{\sigma} \cdot t$ as well as $\bar{\sigma} \leftarrow \bar{\sigma} \mathbin{\dot{-}} t$. At the end of this procedure $\tilde{\sigma}$ is a nonempty transition sequence (since ${}^\bullet\sigma \neq \emptyset$), and enabled at μ. If $\tilde{\sigma}$ has length k, then we are finished.

Otherwise, we have $0 < |\tilde{\sigma}| < k$. Then, $\tilde{\sigma}$ satisfies $\Delta(\tilde{\sigma})(p) \in [-W, -1]$ for all $p \in {}^\bullet\tilde{\sigma}$. Consider the sequence $\bar{\sigma}$, and let $\mu_{\bar{\sigma}} := \mu + \Delta(\bar{\sigma})$. Since $|\bar{\sigma}| < k$, and $\mu_{\bar{\sigma}}(p) = (\mu + \Delta(\tilde{\sigma} \cdot \bar{\sigma}) - \Delta(\tilde{\sigma}))(p) = (\mu' - \Delta(\tilde{\sigma}))(p) > \mu'(p) \geq W$ for all $p \in {}^\bullet\bar{\sigma}$, we can apply the induction hypothesis to $\bar{\sigma}$, μ and $\mu_{\bar{\sigma}}$. Hence, we find $\mu \xrightarrow{\bar{\sigma}'} \mu_{\bar{\sigma}}$ for some permutation $\bar{\sigma}'$ of $\bar{\sigma}$. In addition to $\mu_{\bar{\sigma}}(p) \geq W$ for all $p \in {}^\bullet\bar{\sigma}$, we have $\mu_{\bar{\sigma}}(p) = \mu(p) \geq W$ for all $p \in {}^\bullet\sigma \setminus {}^\bullet\bar{\sigma}$, and thus $\mu_{\bar{\sigma}}(p) \geq W$ for all $p \in {}^\bullet\tilde{\sigma}$. By applying the induction hypothesis to $\tilde{\sigma}$, $\mu_{\bar{\sigma}}$ and μ', we obtain $\mu_{\bar{\sigma}} \xrightarrow{\tilde{\sigma}'} \mu'$ for some permutation $\tilde{\sigma}'$ of $\tilde{\sigma}$. Therefore, the permutation $\bar{\sigma}' \cdot \tilde{\sigma}'$ of σ is enabled at μ and leads from μ to μ'. □

This observation is very useful in situations where we have a firing sequence σ and a large marking μ, and want to argue that some permutation of σ is enabled at μ. We remark that this sufficient condition is a crucial tool for obtaining permutations which only touch space-bounded markings when being fired, and which are needed to satisfy the prerequisites of the framework. The reason is that, when the marking is small, then, on one hand, we have no much choice in choosing the next transition from a given sequence, and, on the other hand, we have no problem regarding the number of tokens anyway. However, when the marking is already large, we don't want to increase the number of tokens further by firing transitions. Instead, we can use this sufficient condition, which ensures that the sequence of interest is enabled.

The next observation is that a large enough increase of the number of tokens at some place by a firing sequence with certain properties implies that the sequence contains a positive loop.

Lemma 6.14. *Let $\mathcal{P} = (P, T, F, \mu_0)$ be a gcf-PN, and $B \subseteq P$ a subset of places. Further, let σ be a nonempty firing sequence of length $k > 0$ such that, for markings μ_i defined by $\mu_0 \xrightarrow{\sigma_{[1]}} \mu_1 \ldots \mu_{k-1} \xrightarrow{\sigma_{[k]}} \mu_k$, the following properties hold:*

(*a*) ${}^\bullet\sigma \subseteq B$, *and* $|{}^\bullet\sigma_{[i]}| = 1$ *for all* $i \in [k]$,

(*b*) $\mu_{i-1}({}^\bullet\sigma_{[i]}) = \max(\mu_{i-1}, B)$ *for all* $i \in [k]$ *(i. e., each transition removes tokens from a place of B with the maximum number of tokens), and*

(*c*) $\max(\mu_k, B) > \max(\mu_0, B) + 2|B|W$.

Then, for some $i \in [k]$, the suffix $\sigma_{[i..]}$ is a positive loop for some place of B.

Proof. Consider the disjoint intervals $[\max(\mu_0, B) + 2\ell W - 2W + 1, \max(\mu_0, B) + 2\ell W]$, $\ell \in [|B|]$. Since $\max(\mu_k, B)$ is outside of all of these intervals, at least one of these intervals, denoted by $[a, b]$, must satisfy $\mu_k(p) \notin [a, b]$ for all $p \in B$. Let $i \in [0, k-1]$ be the smallest index such that $\max(\mu_j, B) \geq a + W$ for all $j \in [i, k]$. (Note that this index exists since $\max(\mu_{k-1}, B) \geq \max(\mu_k, B) - W \geq b - W + 1 = a + W$.) We observe $\max(\mu_i, B) \leq b$ since, by the choice of i, $\max(\mu_{i-1}, B) \leq a + W - 1 = b - W$. Now, for all $p \in B$ having a $j \in [i, k-1]$ with $\mu_j(p) \in [a, b]$, we observe $\mu_k(p) > b$, and therefore $\mu_i(p) < \mu_k(p)$, since, by the choice of i, the token numbers of these places can leave the interval $[a, b]$ only by crossing the border b. This occurs for at least one place of B. For all places which do not have such a j, we observe $\mu_i(p) \leq \mu_k(p)$ since the remaining sequence leading from μ_i to μ_k does not remove tokens from them. Therefore, $\sigma_{[i+1..]}$ is a positive loop for some place of B. □

We will use this observation in a situation, where we actually know that a certain sequence does not contain a positive loop for a subset $B \subseteq P$, and want to argue that the sequence does not increase the number of tokens at places of B too much.

As our next step, we show how a given firing sequence of a gcf-PN can be permuted in such a way that each prefix of the permutation leads to a marking that has polynomial encoding size. This is the most important construction of this section, ensuring that we can apply our framework to gcf-PNs. Our goal is to show that we can permute a given firing sequence σ into a firing sequence φ such that $\max(\mu_0, \varphi)$ is small. However, a possibly surprising idea in the construction is the following: We not only aim to keep the number of tokens at all places small but we also aim to ensure that a place with a large number of tokens does not lose too many tokens, when other big places continue to be big. That is, we want to trap the number of tokens of big places within an interval whose smaller boundary is considerably larger than the maximum number of tokens at the initial and the end marking (and whose size depends on the number of big places) to keep these places simultaneously big. At first sight, it may seem counter-intuitive that we also need this lower bound for the number of tokens at such places. The reason behind this idea is to prevent that the number of tokens get out of control during long sequences that push tokens back and forth between two big places such that these places alternate between being small and being big, while the total number of tokens continually increases. Such a behavior would prevent us from being able to appropriately use the observations made above.

Lemma 6.15. *There is a constant c such that, for each gcf-PN $\mathcal{P} = (P, T, F, \mu_0)$ with $n > 0$ and each firing sequence σ, leading from μ_0 to some marking μ, there is an enabled permutation φ of σ with* $\max(\mu_0, \varphi) \leq (nmW + \max(\mu_0) + \max(\mu) + 1)^{c(n+m)}$.

Proof. Let $\mathcal{P} = (P, T, F, \mu_0)$ be a gcf-PN, and σ a firing sequence leading to some marking μ. We first observe that for $m = 0$ or $W = 0$, μ_0 is the only reachable marking, and $\max(\mu_0, \varphi) = \max(\mu_0) \leq (nmW + \max(\mu_0) + \max(\mu) + 1)^{c(n+m)}$ holds for any $c \geq 1$. Hence, in addition to the prerequisite $n > 0$, we assume $m > 0$ and $W > 0$.

We define two special levels $\ell_{\text{big}} := \max\{W, \max(\mu_0), \max(\mu_\sigma) + 1\}$ and $\ell_{\text{fire}} := \ell_{\text{big}} + W$. Additionally, for $i \in [0, n]$, we define the levels $\ell_i := \ell_{\text{fire}} + W + i \cdot (\max\{(1 + (n+m)W)^{n+m}, 2n\} + 1)W$. A place p is *big* at a marking μ' if $\mu'(p) \geq \ell_{\text{big}}$, and *firing* if $\mu'(p) \geq \ell_{\text{fire}}$.

Consider the following invariants for two transition sequences α and β:

(*i*) $\alpha \cdot \beta$ is a permutation of σ with $\mu_0 \xrightarrow{\alpha} \mu_\alpha \xrightarrow{\beta} \mu$ for some marking μ_α,

(*ii*) $\max(\mu_0, \alpha) \leq \ell_n$, and

(*iii*) if there are $b \geq 1$ big places at μ_α, then $\max(\mu_\alpha) \leq \ell_{b-1}$.

For the empty sequence $\alpha = \epsilon$ and $\beta = \sigma$, these invariants are obviously satisfied. Assume $|\alpha| < |\sigma|$, and that α and β satisfy the invariants. We show how to extend α at the end by a sequence α_{ext} to a longer transition sequence $\alpha_{\text{new}} = \alpha \cdot \alpha_{\text{ext}}$, and how to obtain a corresponding sequence β_{new} such that α_{new} and β_{new} again satisfy the invariants.

First, we consider the case that there are no places which are firing at μ_α. Then, we set $\alpha_{\text{new}} := \alpha \cdot \beta_{[1]}$, and $\beta_{\text{new}} := \beta_{[2..]}$. α_{new} and β_{new} obviously satisfy property (*i*). For each place p that is big at $\mu_\alpha + \Delta(\beta_{[1]})$, we have $(\mu_\alpha + \Delta(\beta_{[1]}))(p) \leq \mu_\alpha(p) + W < \ell_{\text{fire}} + W = \ell_0$. Hence, the properties (*ii*) and (*iii*) are also satisfied.

Next, we consider the case that there are places which are firing at μ_α. The following procedure is divided into several steps, which are illustrated on a high level in Figure 6.8.

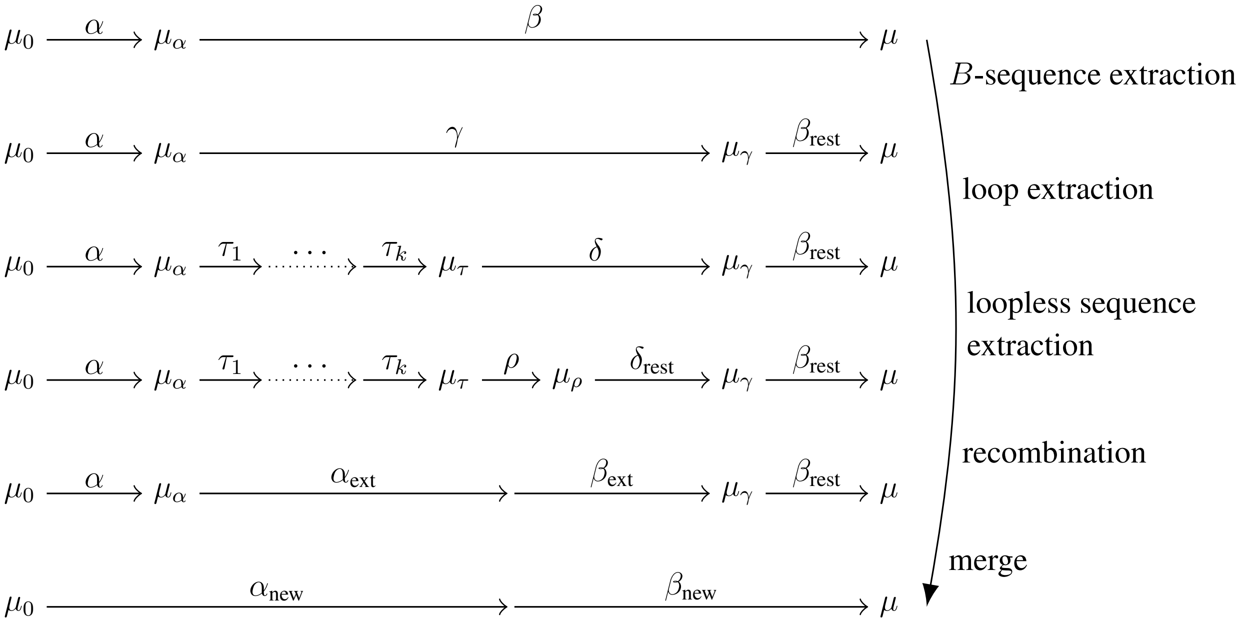


Figure 6.8: The markings obtained when firing α are small. *B-sequence extraction*: Specific transitions removing tokens from big places are cut out of β and pasted as a sequence γ. *Loop extraction*: A maximal set of short loops $\tau_1, \ldots, \tau_k$ maximizing their total length is extracted from γ. *Loopless sequence extraction*: A specific loopless sequence ρ is extracted from δ. *Recombination*: Specific sequences α_{ext}, β_{ext} with $\alpha_{\text{ext}} \cdot \beta_{\text{ext}} = \tau_1 \cdots \tau_k \cdot \rho \cdot \delta_{\text{rest}}$ and $|\alpha_{\text{ext}}| \in [1, |\tau_1 \cdots \tau_k \cdot \rho|]$ are defined. *Merge*: We set $\alpha_{\text{new}} := \alpha \cdot \alpha_{\text{ext}}$ and $\beta_{\text{new}} := \beta_{\text{ext}} \cdot \beta_{\text{rest}}$. The markings obtained when firing α_{new} are small. We made progress.

B-sequence extraction. Let B be the set of places which are big at μ_α and $b := |B| \geq 1$ their number. Upper and lower bounds for the number of tokens of a big place $p^* \in B$ as functions of time are illustrated in (*a*) of Figure 6.9.

We initialize an empty transition sequence $\gamma \leftarrow \epsilon$, as well as $\beta_{\text{rest}} \leftarrow \beta$. As long as there is a place $p \in B$ which is firing at $\mu_\alpha + \Delta(\gamma)$, we select the transition $t \in \Psi_{\text{first}}(\beta_{\text{rest}})$ with $p = {}^\bullet t$, and set $\gamma \leftarrow \gamma \cdot t$, as well as $\beta_{\text{rest}} \leftarrow \beta_{\text{rest}} \mathbin{\dot{-}} t$. Notice that t must exist since, by $(\mu_\alpha + \Delta(\gamma))(p) > \mu(p)$, β_{rest} contains a transition that reduces the number of tokens at p.

Since each transition of γ removes tokens only from a place with at least $\ell_{\text{fire}} = \ell_{\text{big}} + W$ tokens, at least ℓ_{big} tokens remain at this place after having fired the transition. Therefore, we observe $\ell_{\text{big}} \leq \min(\mu_\alpha, \gamma, B)$.

Furthermore, γ is nonempty since μ_α has a firing place. Let $\mu_\gamma := \mu_\alpha + \Delta(\gamma)$. Part (*b*) of Figure 6.9 illustrates the current situation. By Lemma 6.12, $\gamma \cdot \beta_{\text{rest}}$ is enabled at μ_α. In total, we observe

$$\mu_0 \xrightarrow{\alpha} \mu_\alpha \xrightarrow{\gamma} \mu_\gamma \xrightarrow{\beta_{\text{rest}}} \mu, \quad \text{and} \quad \ell_{\text{big}} \leq \min(\mu_\alpha, \gamma, B).$$

Loop extraction. In the following, we will continue to work only with the sequence γ. The important property of γ is that every place of ${}^\bullet\gamma \subseteq B$ is big at μ_α. The sequence α_{ext}, which we want to construct, will be created from transitions of γ.

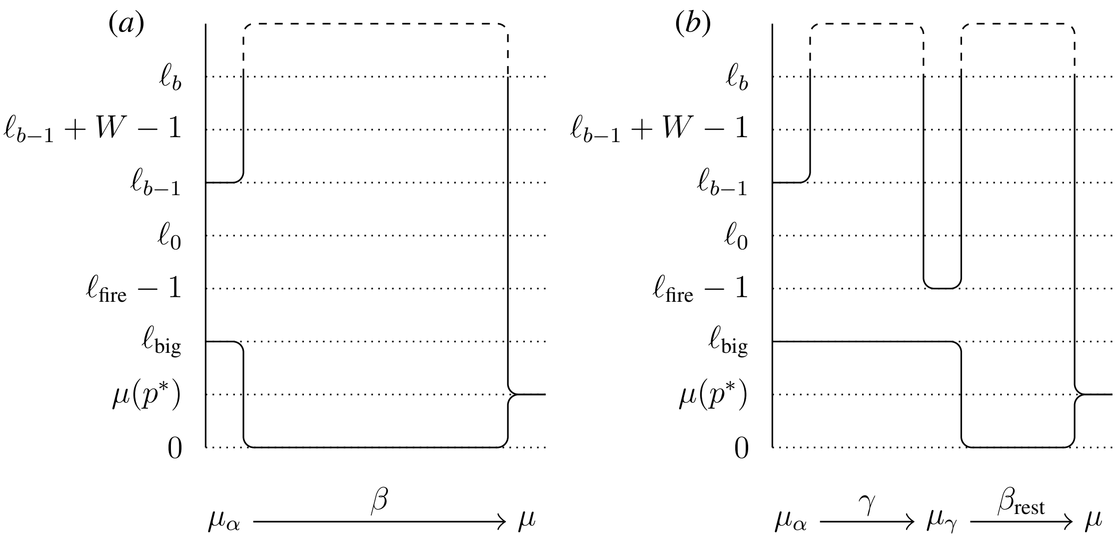


Figure 6.9: (a) and (b) illustrate the development of the number of tokens at a place p^* which is big at μ_α before and after the B-*sequence extraction*-step of the permutation procedure described in Lemma 6.15. The number of tokens at p^* is bounded from above and below by the respective curves. The number of big places at μ_α is b. Dashed lines symbolize that the number of tokens can become arbitrarily big.

Consider a loop $\Phi \leq \Psi(\gamma)$ with maximal component sum, i. e., there is no loop $\Phi' \leq \Psi(\gamma)$ with $\sum_{t \in T} \Phi'(t) > \sum_{t \in T} \Phi(t)$. Using Lemma 2.18, we decompose Φ into small loops $\Phi_1, \ldots, \Phi_k$, each with component sum at most $(1 + (n+m)W)^{n+m}$. Since $\mu_\alpha(p) \geq W$ for all $p \in B$, and ${}^\bullet t \in B$ for all $j \in [k]$ and $t \in \Phi_j$ (by the construction of γ), we can use Lemma 6.13 to find transition sequences $\tau_1, \ldots, \tau_k$ with $\Psi(\tau_j) = \Phi_j$, $j \in [k]$, such that $\tau := \tau_1 \cdots \tau_k$ is enabled at μ_α. Let the markings $\nu_0, \nu_1, \ldots, \nu_k$ be defined by $\mu_\alpha \xrightarrow{\tau_1 \cdots \tau_i} \nu_i$, $i \in [0, k]$. (In particular, this means $\nu_0 = \mu_\alpha$ and $\nu_k = \mu_\tau$)

Now, we show the important observation $\Delta(\tau)(p) < W$ for each $p \in B$. Assume for the sake of contradiction that $\Delta(\tau)(p) \geq W$ for some place $p \in B$. Then, by the maximality of $\Phi = \Psi(\tau)$, $\Psi(\gamma) - \Psi(\tau)$ does not contain a transition t with $p = {}^\bullet t$. Therefore, $\Delta(\gamma)(p) = \Delta(\tau)(p) + \Delta(\Psi(\gamma) - \Psi(\tau))(p) \geq W$. But then, $\mu_\alpha(p) + \Delta(\gamma)(p) \geq \ell_{\text{big}} + W = \ell_{\text{fire}}$, a contradiction to the fact that no place of B is firing at μ_γ.

Since all τ_j, $j \in [k]$, are loops, we obtain $\Delta(\tau_1 \cdots \tau_j)(p) \leq W$ for all $p \in B$ and $j \in [k]$. Furthermore, $|\tau_j| < (1 + (n+m)W)^{n+m}$ implies that $\max(\vec{0}, \tau_j)$, the maximum displacement over all places and prefixes of τ_j, is at most $(1 + (n+m)W)^{n+m} W$. We obtain $\max(\nu_{j-1}, \tau_j, B) \leq \ell_{b-1} + W + (1 + (n+m)W)^{n+m} W \leq \ell_b$ for all $j \in [k]$. Our observations imply the first important intermediate result of the proof:

$$\max(\mu_\alpha, \tau, B) \leq \ell_b, \quad \text{and} \quad \min(\nu_j, B) \geq \ell_{\text{big}} \text{ for all } j \in [k].$$

We now consider $\gamma \mathbin{\dot{-}} \tau$. Since $\min(\mu_\tau, B) \geq \min(\mu_\alpha, B) \geq \ell_{\text{big}} \geq W$, $\min(\mu_\gamma, B) \geq \ell_{\text{big}} \geq W$ and ${}^\bullet(\gamma \mathbin{\dot{-}} \tau) \subseteq B$, we can apply Lemma 6.13 to $\gamma \mathbin{\dot{-}} \tau$. This yields a permutation δ of $\gamma \mathbin{\dot{-}} \tau$ that is enabled at μ_τ. The current situation is illustrated in Figure 6.10.

Loopless sequence extraction. We initialize another empty transition sequence $\rho \leftarrow \epsilon$, as well as $\delta_{\text{rest}} \leftarrow \delta$. As long as there is a place of B which is firing at $\mu_\tau + \Delta(\rho)$, we select a place $p \in B$

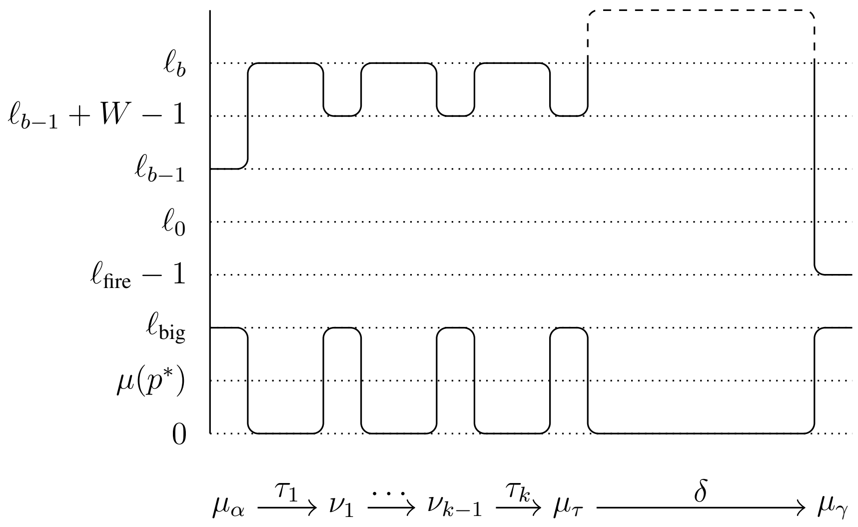


Figure 6.10: The development of the number of tokens at a place p^* which is big at μ_α after the *loop extraction*-step

with $\max(\mu_\tau + \Delta(\rho), B) = (\mu_\tau + \Delta(\rho))(p)$ and the transition $t \in \Psi_{\text{first}}(\delta_{\text{rest}})$ with $p = {}^\bullet t$, and set $\rho \leftarrow \rho \cdot t$, as well as $\delta_{\text{rest}} \leftarrow \delta_{\text{rest}} \mathbin{\dot{-}} t$. It is important to note the difference of this selection procedure compared to the one before. Here, we select a place of B with the largest number of tokens. Also note that ρ is nonempty since some place of B is firing at μ_α and therefore also firing at $\mu_\tau \geq \mu_\alpha$. Let $\mu_\rho := \mu_\tau + \Delta(\rho)$. By construction of ρ, we have $\min(\mu_\rho, B) \geq \ell_{\text{big}}$. By Lemma 6.12, we find $\mu_\alpha \xrightarrow{\tau} \mu_\tau \xrightarrow{\rho} \mu_\rho \xrightarrow{\delta_{\text{rest}}} \mu_\gamma$.

We observe $\max(\mu_\tau, B) = \max(\mu_\alpha + \Delta(\tau), B) \leq \max(\mu_\alpha, B) + W \leq \ell_{b-1} + W$. To see this, assume for the sake of contradiction that $\max(\mu_\tau, \rho, B) > \ell_b$. Then, there is an $i \in [|\rho|]$ such that $\max(\mu_\tau + \Delta(\rho_{[..i]}), B) > \ell_b \geq \ell_{b-1} + W + 2nW \geq \max(\mu_\tau, B) + 2nW$. But then, Lemma 6.14 implies that ρ contains a positive loop, a contradiction to the maximality of $\Phi = \Psi(\tau)$. Therefore, $\max(\mu_\tau, \rho, B) \leq \ell_b$.

Our observations can now be summarized as our second important intermediate result:

$$\mu_\alpha \xrightarrow{\tau \cdot \rho} \mu_\rho \xrightarrow{\delta_{\text{rest}}} \mu_\gamma, \quad |\tau \cdot \rho| > 0, \quad \max(\mu_\alpha, \psi, B) \leq \ell_b, \quad \text{and}$$
$$\ell_{\text{big}} \leq \min(\mu_\rho, B) \leq \max(\mu_\rho, B) < \ell_{\text{fire}}.$$

The current situation is illustrated in Figure 6.11.

Recombination and merge. In the last step, we consider three cases. In the first case, there is a smallest $i \in [k]$ such that some place $p \in P \setminus B$ is big at ν_i. Since $\nu_j(p') < \ell_{\text{big}}$ and $\max(\vec{0}, \tau_j) \leq (1 + (n+m)W)^{n+m}W$ for all $p' \in [P \setminus B]$ and $j \in [i-1]$, we find $\max(\mu_\alpha, \tau_1 \cdots \tau_i, P \setminus B) \leq \ell_{\text{big}} + (1 + (n+m)W)^{n+m}W \leq \ell_b$. We define $\alpha_{\text{ext}} := \tau_1 \cdots \tau_i$, $\beta_{\text{ext}} := \tau_{i+1} \cdots \tau_k \cdot \rho \cdot \delta_{\text{rest}}$, and accordingly $\alpha_{\text{new}} := \alpha \cdot \alpha_{\text{ext}}$ as well as $\beta_{\text{new}} := \beta_{\text{ext}} \cdot \beta_{\text{rest}}$. By the observations made above, and by the fact that there are at least $b+1$ places that are big at ν_i, α_{new} and β_{new} satisfy (*i*)–(*iii*).

In the second case, the first case doesn't apply, and there is a smallest $i \in [|\rho|]$ such that some place $p \in P \setminus B$ is big at $\mu_\tau + \Delta(\rho_{[..i]})$. An analogue argument as in the first case shows that $\max(\mu_\alpha, \tau, P \setminus B) \leq \ell_b$. Moreover, by the choice of i, $\max(\mu_\tau, \rho_{[..i]}, P \setminus B) \leq \ell_{\text{big}} + W \leq \ell_0$. We define $\alpha_{\text{ext}} := \tau \cdot \rho_{[..i]}$, $\beta_{\text{ext}} := \rho_{[i+1..]} \cdot \delta_{\text{rest}}$, and α_{new}, β_{new} as before. Again, our observations and the

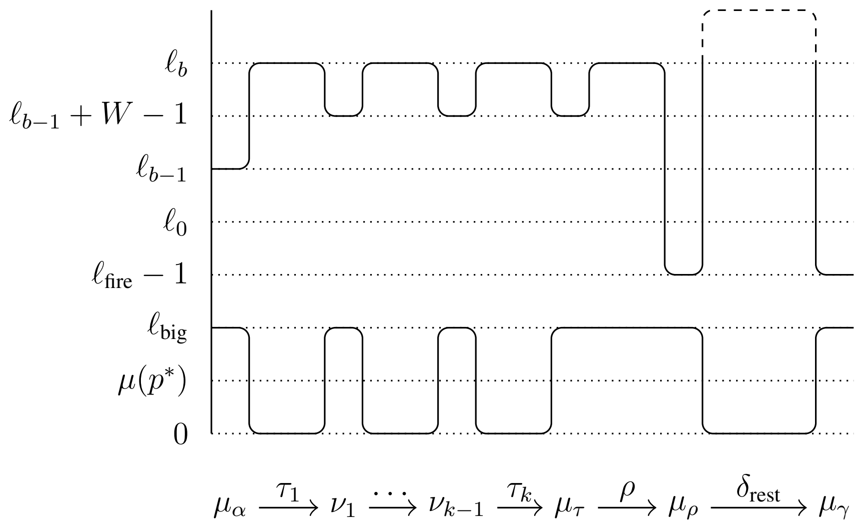


Figure 6.11: The development of the number of tokens at a place p^* which is big at μ_α after the *loopless sequence extraction*-step

fact that there are at least $b+1$ places that are big at $\mu_\tau + \Delta(\rho_{[..i]})$ imply that α_{new} and β_{new} satisfy (*i*)–(*iii*).

In the third case, neither the first nor the second case applies. Then, analogue arguments as before imply $\max(\mu_\alpha, \tau \cdot \rho, P \setminus B) \leq \ell_b$. Furthermore, by the observations made above, we find $\max(\mu_\rho) < \ell_{\text{fire}} < \ell_0$. We define $\alpha_{\text{ext}} := \tau \cdot \rho$, $\beta_{\text{ext}} := \delta_{\text{rest}}$, and α_{new}, β_{new} as before. Then, α_{new} and β_{new} satisfy (*i*)–(*iii*).

Since $|\alpha_{\text{new}}| > |\alpha|$ holds in each of these cases, we made progress. By iteratively applying this procedure, we obtain a permutation φ of σ such that $\mu_0 \xrightarrow{\varphi} \mu$ and $\max(\mu_0, \varphi) \leq \ell_n$, i. e., all markings obtained when firing φ contain at most ℓ_n tokens at each place. Since n, m, $W > 0$, we observe that each summand of

$$\ell_n = \max\{W, \max(\mu_0), \max(\mu_\sigma) + 1\} + 2W + n \cdot (\max\{(1 + nW + mW)^{n+m}, 2n\} + 1)W$$

is at most $(nmW + \max(\mu_0) + \max(\mu) + 1)^{d(n+m)}$ for some constant d. Hence, we can find a constant $c \in \mathbb{N}$ with $\ell_n \leq (nmW + \max(\mu_0) + \max(\mu) + 1)^{c(n+m)}$. □

The existence of canonical permutations as of Lemma 6.15 is the focal point for the application of our framework developed in Chapter 4.

Lemma 6.16. *There is a constant c such that the class of gcf-PNs is simple structurally f-f-canonical, where $f(n, m, W, K) = (nmW + K + 1)^{cn(n+m)}$.*

Proof. Let $\mathcal{P} = (P, T, F, \mu_0)$ be a gcf-PN with $n > 0$ places. First note that the wipe-extension of $\mathcal{P}$ is a gcf-PN itself, which has $n' := n$ places, $m' := m + n$ transitions, maximal edge multiplicity $W' := \max\{W, 1\}$, and initial marking $\mu'_0 := \mu_0$. Assume first that m, $W > 0$. Let σ be a firing sequence of the wipe-extension, leading to $\vec{0}$. By Lemma 6.15, there is a permutation φ of σ satisfying

$$\begin{aligned} &\max(\mu'_0, \varphi) \leq (n'm'W' + \max(\mu'_0) + 1)^{c(n'+m')} \\ &= (n(m+n)\max\{W, 1\} + \max(\mu_0) + 1)^{c(2n+m)} \leq (nmW + \max(\mu_0) + 1)^{d(n+m)}, \end{aligned}$$

where c is the constant of Lemma 6.15, and $d > 1$ is some appropriately chosen larger constant. Note that if one of the values m, W is 0, then $\max(\mu) \leq \max(\mu_0) \leq (nmW + \max(\mu_0) + 1)^{d(n+m)}$ for each marking μ reachable in the wipe-extension. Hence, we can apply Theorem 4.2 for the function $\tilde{f}$ satisfying $\tilde{f}(n, m, W, K) = (nmW + K + 1)^{d(n+m)}$. By this theorem, the class of gcf-PNs is f-f-canonical for some function f satisfying $f(n, m, W, K) = (nmW + K + 1)^{d'n(n+m)} \geq \left((nmW + K + 1)^{d(n+m)} + 1\right)^{2n}$, where d' is some constant. Furthermore, the class of gcf-PNs is closed under a change of initial markings, and we can check in polynomial space if a Petri net is a gcf-PN. Hence, the lemma follows. □

The definition of simple structurally f-g-canonical classes is given in Definition 4.1. For convenience, we specify (a slight simplification of) the corresponding canonical firing sequences in the following corollary.

Corollary 6.17. *There is a constant $c \in \mathbb{N}$ such that, for each gcf-PN $\mathcal{P} = (P, T, F, \mu_0)$ and each reachable marking μ of $\mathcal{P}$, there are $k \in [0, \max(\mu) + u]$ and transition sequences ξ, $\bar{\xi}$, α_1, …, α_{k+1}, τ_1, …, τ_k with the following properties, where $u = (nmW + \max(\mu_0) + 1)^{cn(n+m)}$:*

- *(a) $\xi = \alpha_1 \cdot \tau_1 \cdot \alpha_2 \cdot \tau_2 \cdots \tau_k \cdot \alpha_{k+1}$ is a firing sequence of length at most $(k + 1)u$ leading from μ_0 to μ,*
- *(b) $\bar{\xi} = \alpha_1 \cdot \alpha_2 \cdots \alpha_{k+1}$ is a firing sequence of length at most u, and*
- *(c) each τ_i, $i \in [k]$, is a positive loop of length at most u enabled at some marking μ^* with $\max(\mu^*) \leq u$ and $\mu^* \leq \mu_0 + \Delta(\alpha_1 \cdot \alpha_2 \cdots \alpha_i)$.*

Proof. This follows immediately from Lemma 6.16 and Definition 4.1. Note that the constant c of this corollary is chosen to be larger than the constant c of Lemma 6.16 in order to simplify some terms (e. g., the upper bound of the interval for k). □

As mentioned in Chapter 4, we call the sequence $\bar{\xi}$ the *backbone* of the canonical sequence under consideration. Furthermore, we call ξ *positive canonical sequence* since it incorporates positive loops.

For some problems, we need another form of canonical firing sequence, which we construct in the following. In contrast to those of Corollary 6.17, they will not incorporate positive loops but negative loops instead. The stepping stone for the construction is a procedure that, given a firing sequence (satisfying some conditions), permutes it in such a way that we obtain a firing sequence which, among other things, has a suffix that is a negative loop.

Lemma 6.18. *Let (P, T, F, μ_0) be a gcf-PN, and σ a firing sequence leading to some marking μ_σ such that $\mu_\sigma(p) \geq W$ for all $p \in {}^\bullet\sigma$, and $\mu_0(p^*) \geq \mu_\sigma(p^*) + (\delta + 3) \cdot W$ for some place $p^* \in {}^\bullet\sigma$, where $\delta = (\max(\mu_\sigma) + W + 1)^{n-1}$. Then, there are transition sequences α, τ and a marking $\mu^* \in \{0, W\}^n$ such that*

- *(a) $\alpha \cdot \tau$ is a permutation of σ with $\mu_0 \xrightarrow{\alpha \cdot \tau} \mu_\sigma$,*
- *(b) $(\mu_0 + \Delta(\tau)) \xrightarrow{\alpha} \mu_\sigma$,*
- *(c) τ is a negative loop with $\Delta(\tau)(p^*) \in [-(nmW + 1)^{n+m}, -1]$ and $\Delta(\tau)(p) = 0$ for all $p \neq p^*$,*

(*d*) $|\tau| \leq (nmW+1)^{n+m}$, *and*

(*e*) $(\mu^* - \Delta(\tau)) \xrightarrow{\tau} \mu^*$, *where* $\mu^*(p) = W$ *if* $\mu_\sigma(p) \geq W$, *and* $\mu^*(p) = 0$ *otherwise.*

Proof. In the following, we will construct transition sequences α', τ' satisfying (*a*) and (*b*) (replace α and τ by α' and τ' there) as well as the property

(*c'*) $\Delta(\tau)(p^*) \in [-\delta \cdot W, -1]$ and $\Delta(\tau)(p) = 0$ for all $p \in P \setminus \{p^*\}$.

Property (*c'*) is weaker than (*c*) in the sense that (*c*) implies (*c'*) but not the other way round.

Our first step to this end consists of finding transition sequences α_0, φ_0, ..., φ_δ and markings ν_0, ..., ν_δ such that, for appropriately defined integral levels $\ell_0 > \ldots > \ell_{\delta+1}$, the following properties are satisfied for all $i \in [0, \delta]$:

(*i*) $\nu_0 \xrightarrow{\varphi_0 \cdots \varphi_i} \nu_i$,

(*ii*) $\nu_i(p) \in [0, \max(\mu_\sigma) + W]$ for all $p \in P \setminus \{p^*\}$, and

(*iii*) $\nu_i(p^*) \in [\ell_{i+1} + 1, \ell_i]$.

Moreover, these transition sequences shall be defined in such a way that we can use them later to construct α' and τ'. The general idea how we find these markings and sequences is illustrated in Figure 6.12.

We first show how to construct ν_0. We initialize $\alpha_0 \leftarrow \epsilon$ as the empty sequence and $\psi \leftarrow \sigma$. As long as there is a $t \in \Psi_{\text{first}}(\psi)$ such that ${}^\bullet t \neq p^*$ and $(\mu_0 + \Delta(\alpha_0))({}^\bullet t) > \mu_\sigma({}^\bullet t) + W$, we set $\alpha_0 \leftarrow \alpha_0 \cdot t$ and $\psi \leftarrow \psi \mathbin{\dot{-}} t$. (Note that this means that we extend α_0 by firing t at marking $\mu_0 + \Delta(\alpha_0)$.) We stop, when no such t exists (any more). After that, we define $\nu_0 := \mu_0 + \Delta(\alpha_0)$ and $\varphi_0 := \epsilon$. (The sequence φ_0 is a dummy sequence to avoid a special case when constructing the remaining sequences φ_i, $i > 0$.) By Lemma 6.12, we observe $\mu_0 \xrightarrow{\alpha_0} \nu_0 \xrightarrow{\psi} \mu_\sigma$.

Next, we define the levels ℓ_0, ..., $\ell_{\delta+1}$ by $\ell_i := \nu_0(p^*) - i \cdot W$, $i \in [0, \delta+1]$. Note that properties (*i*)–(*iii*) are satisfied for $i = 0$.

We proceed by recursively defining the remaining transition sequences and markings. For all $i \in [\delta]$, let φ_i and ν_i be recursively defined by $\varphi_i := \texttt{getSubSeq}(i)$ and $\nu_i := \nu_0 + \Delta(\varphi_0 \cdots \varphi_i)$. Furthermore, we define $\sigma_{\text{rest}} := \sigma \mathbin{\dot{-}} (\alpha_0 \cdot \varphi_0 \cdots \varphi_\delta)$. Lemma 6.12 immediately implies that $\alpha_0 \cdot \varphi_0 \cdots \varphi_\delta \cdot \sigma_{\text{rest}}$ is a firing sequence which ensures property (*i*) for all $i \in [\delta]$. Using induction on $i \in [0, \delta]$, it is not hard to show properties (*ii*)–(*iii*) for the remaining sequences and markings: Let $i \in [\delta]$, and assume that (*ii*)–(*iii*) hold for step $i-1$. Property (*ii*) and $\nu_i(p^*) \leq \ell_i$ of (*iii*) directly follow from the definition of Function getSubSeq. Furthermore, $\nu_{i-1}(p^*) \geq \ell_i + 1$ holds by the induction hypothesis.

Funktion getSubSeq(iteration i)

$\varphi \leftarrow$ empty sequence
$\psi_{\text{rest}} \leftarrow \psi \mathbin{\dot{-}} (\varphi_0 \cdots \varphi_{i-1})$
while $\exists t \in \Psi_{\textit{first}}(\psi_{\textit{rest}}) : [(p^* = {}^\bullet t$ **and** $(\nu_{i-1} + \Delta(\varphi))(p^*) > \ell_i)$ **or**
$(\exists p \in P \setminus \{p^*\} : p = {}^\bullet t$ **and** $(\nu_{i-1} + \Delta(\varphi))(p) > \mu_\sigma(p) + W)]$ **do**
  $\varphi \leftarrow \varphi \cdot t$
  $\psi_{\text{rest}} \leftarrow \psi_{\text{rest}} \mathbin{\dot{-}} t$
return φ

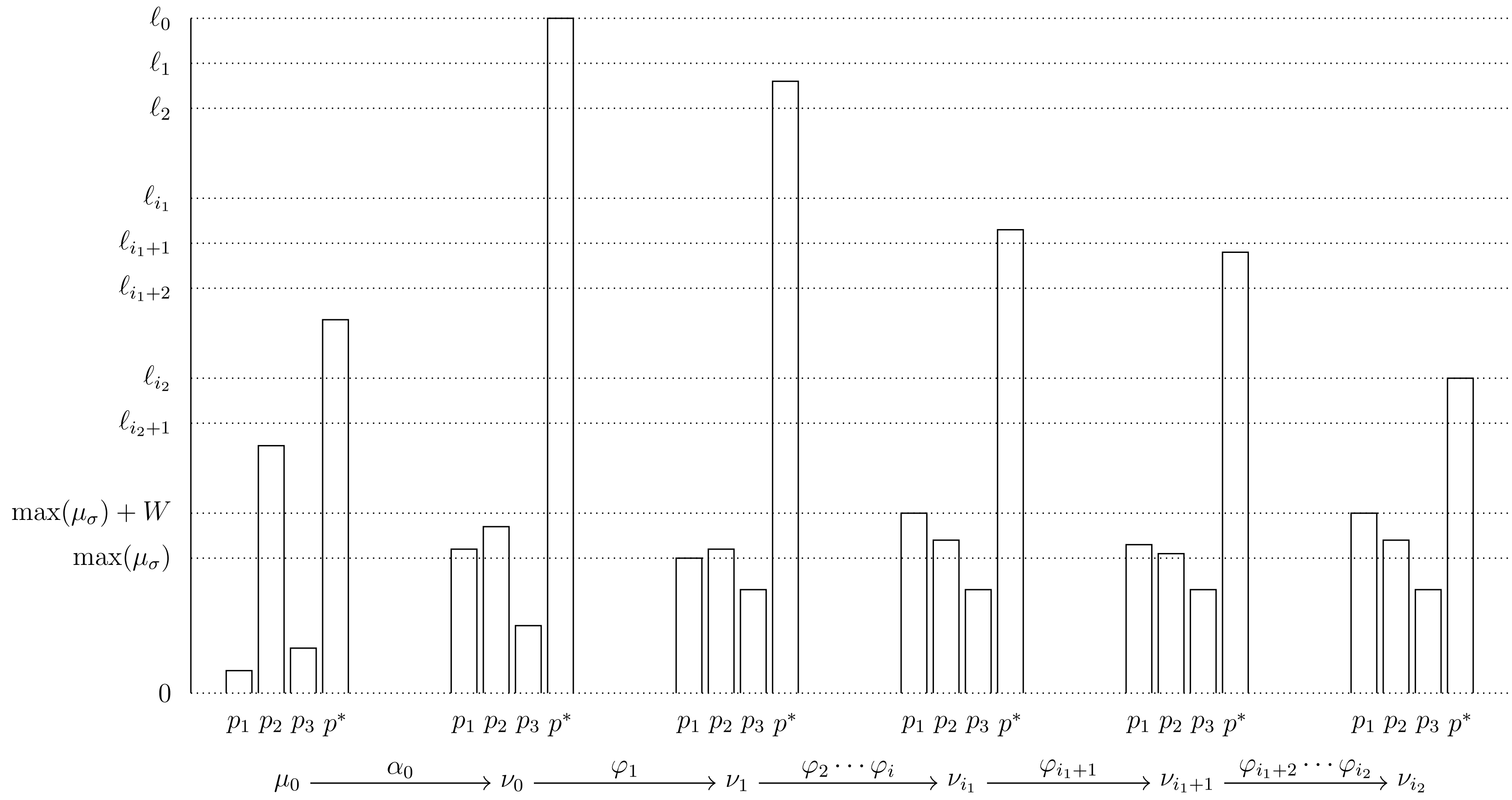


Figure 6.12: The sequences generated in the following consist of transitions of the original sequence σ such that each occurrence of a transition of σ is used at most once in these sequences. Starting from μ_0, we generate α_0 which leads to a marking ν_0 such that $\nu_0(p) \leq \max(\mu_\sigma)(p) + W$ for all $p \in P \setminus \{p^*\}$, and $\Delta(\alpha_0)(p^*) \geq 0$. From there, we define levels ℓ_i, $i \in [0, \delta + 1]$, and generate sequences φ_i, $i \in [\delta]$, such that, when encountering the respective corresponding markings ν_i, the respective number of tokens at each place $p \in P \setminus \{p^*\}$ is still trapped within the interval $[0, \max(\mu_\sigma) + W]$, and such that the number of tokens at p^* is strictly decreasing, where $\nu_i(p^*) \in [\ell_{i+1} + 1, \ell_i]$. By constructing a sufficient number of these sequences and markings, we find $i_1 < i_2$ such that $\nu_{i_1}(p) = \nu_{i_2}(p)$ for all $p \in P \setminus \{p^*\}$. The sequence $\varphi_{i_1+1} \cdots \varphi_{i_2} =: \tau''$ is then a negative loop for p^*. Permuting τ'' in an appropriate way yields the negative loop τ'.

Subsequently, one iteration of the while-loop of Function getSubSeq cannot decrease the number of tokens at p^* below $\ell_{i+1} + 1$, which implies $\nu_{i+1}(p^*) \geq \ell_{i+1} + 1$. Thus, (*iii*) holds.

We continue the proof by defining α' and τ'. By (*ii*), there are at most δ different projections of the $\delta + 1$ markings ν_i, $i \in [0, \delta]$, onto the places $p \in P \setminus \{p^*\}$. Hence, there are i_1, $i_2 \in [0, \delta]$, $i_1 < i_2$, such that $\nu_{i_1}(p) = \nu_{i_2}(p)$ for all $p \in P \setminus \{p^*\}$. Let $\alpha_1 := \varphi_0 \cdots \varphi_{i_1}$, $\tau'' := \varphi_{i_1+1} \cdots \varphi_{i_2}$, $\alpha_2 := \varphi_{i_2+1} \cdots \varphi_\delta \cdot \sigma_{\text{rest}}$, $\alpha_{0,1} := \alpha_0 \cdot \alpha_1$, $\alpha_{1,2} := \alpha_1 \cdot \alpha_2$, and $\alpha' := \alpha_0 \cdot \alpha_1 \cdot \alpha_2$, see (*a*) of Figure 6.13. (In this illustration, certain sequences are indicated to be enabled at certain markings. This follows from the definition of the sequences and from our observations made above, in particular property (i).)

(*a*)

$$\mu_0 \xrightarrow{\alpha_0} \nu_0 \xrightarrow{\alpha_1 = \varphi_0 \cdots \varphi_{i_1}} \nu_{i_1} \xrightarrow{\tau'' = \varphi_{i_1+1} \cdots \varphi_{i_2}} \nu_{i_2} \xrightarrow{\varphi_{i_2+1} \cdots \varphi_\delta} \nu_\delta \xrightarrow{\sigma_{\text{rest}}} \mu_\sigma$$

$$\nu_{i_2} \xrightarrow{\alpha_2 = \varphi_{i_2+1} \cdots \varphi_\delta \cdot \sigma_{\text{rest}}} \mu_\sigma$$

(*b*)

$$\mu_0 \xrightarrow{\alpha' = \alpha_0 \cdot \alpha_1 \cdot \alpha_2} \mu_0 + \Delta(\alpha') \xrightarrow{\tau'} \mu_\sigma$$

(*c*)

$$\mu_0 + \Delta(\tau') \xrightarrow{\alpha' = \alpha_0 \cdot \alpha_1 \cdot \alpha_2} \mu_\sigma$$

(*d*)

$$\forall i \in [q]: \;\; \mu_\sigma - \Delta(\tau_i) \xrightarrow{\tau_i} \mu_\sigma$$

$$\mu_\sigma - \Delta(\alpha) = \mu_0 + \Delta(\tau_j) \xrightarrow{\alpha = \alpha_0 \cdot \alpha_1 \cdot \alpha_2 \cdot \tau_1 \cdots \tau_{j-1} \cdot \tau_{j+1} \cdots \tau_q} \mu_\sigma$$

$$\mu_0 \xrightarrow{\alpha = \alpha_0 \cdot \alpha_1 \cdot \alpha_2 \cdot \tau_1 \cdots \tau_{j-1} \cdot \tau_{j+1} \cdots \tau_q} \mu_0 + \Delta(\alpha) \xrightarrow{\tau = \tau_j} \mu_\sigma$$

Figure 6.13: Observations for different steps during the construction of negative canonical firing sequences of gcf-PNs

Property (*iii*) ensures $\Delta(\tau'')(p^*) \in [-(i_2 - i_1 + 1)W, \, -1]$, and therefore that τ'' satisfies (*c*') (where τ of (*c*') is identified with τ'').. This together with the assumption of this lemma implies $(\nu_0 + \Delta(\alpha_{1,2}))(p) \geq \mu_\sigma(p) \geq W$ for all $p \in {}^\bullet\tau''$. Therefore, we can apply Lemma 6.13 to the markings $\nu_0 + \Delta(\alpha_{1,2})$ (which is none other than $\mu_0 + \Delta(\alpha')$), μ_σ, and the sequence τ'' to obtain a permutation τ' of τ'' which is enabled at $\mu_0 + \Delta(\alpha')$. Note that also τ' satisfies (*c*') (where τ of (*c*') is identified with τ'). Next we observe that α_2 is enabled at ν_{i_1} since α_2 is enabled at ν_{i_2} and $\nu_{i_1} \geq \nu_2$

holds. Consequently, $\alpha' \cdot \tau'$ is a permutation of σ enabled at μ_0, and (*a*) follows, see (*b*) of Figure 6.13.

To show (*b*) for α', it's only necessary to consider $\alpha_{0,1}$ since we already know that $\nu_{i_2} \xrightarrow{\alpha_2} \mu_\sigma$. Moreover, it's sufficient to verify that, for all $j \in [0, |\alpha_{0,1}|-1]$ and $p \in P$, $(\mu_0+\Delta(\tau')+\Delta((\alpha_{0,1})_{[..j]}))(p) \geq \min\{W, (\mu_0+\Delta((\alpha_{0,1})_{[..j]}))(p)\}$ holds since this implies that the marking $(\mu_0+\Delta(\tau')+\Delta((\alpha_{0,1})_{[..j]}))$ enables the next transition $(\alpha_{0,1})_{[j+1]}$. Since, by (*c'*), τ' does not change the number of tokens at some place other than p^*, this sufficient condition remains to be shown for p^*, which we do now.

We first consider α_0. Since $\mu_0(p^*) + \Delta(\tau')(p^*) \geq W$ and $p^* \notin {}^\bullet\alpha_0$, we find $(\mu_0 + \Delta(\tau') + \Delta((\alpha_0)_{[..j]}))(p^*) \geq W$ for all $j \in [0, |\alpha_0| - 1]$.

Next, we consider $\alpha_1 = \varphi_1 \cdots \varphi_{i_1}$. By (*iii*), we find

$$\begin{aligned}\nu_i(p^*) &\geq \ell_{i+1} + 1 > \ell_{i_1} = \ell_{i_1+1} + W = \nu_0(p^*) - (i_1 + 1) \cdot W + W \geq \mu_0(p^*) - (i_1 + 1) \cdot W + W \\ &\geq \mu_\sigma(p^*) + (\delta + 3) \cdot W - (i_1 + 1) \cdot W + W = \mu_\sigma(p^*) + (\delta - i_1 + 1) \cdot W + 2W \\ &\geq (\delta - i_1 + 1) \cdot W + 3W\end{aligned}$$

for all $i \in [0, i_1 - 1]$ (remember for this sequence of inequalities the prerequisite $\mu_\sigma(p^*) \geq W$). By property (*iii*) and the definition of Function `getSubSeq`, we observe $\Delta((\varphi_i)_{[..j]})(p^*) \geq -2W$ for all $i \in [0, \delta]$, $j \in [0, |\varphi_i|]$. Thus, $(\nu_i + \Delta((\varphi_{i+1})_{[..j]}))(p^*) \geq (\delta - i_1 + 1) \cdot W + W$ for all $i \in [0, i_1 - 1]$ and $j \in [0, |\varphi_{i+1}|]$. Adding the displacement of τ' yields $(\nu_i + \Delta(\tau') + \Delta((\varphi_{i+1})_{[..j]}))(p^*) \geq -(i_2 - i_1 + 1)W + (\delta - i_1 + 1) \cdot W + W \geq W$ for all $i \in [0, i_1 - 1]$ and $j \in [0, |\varphi_{i+1}|]$. This implies $(\nu_0 + \Delta(\tau') + \Delta((\alpha_1)_{[..j]}))(p^*) \geq W$ for all $j \in [0, |\alpha_1| - 1]$. In total, (*b*) follows for α', see (*c*) of Figure 6.13.

So far, we have shown that α' and τ' satisfy (*a*), (*b*), and (*c'*), but not necessarily (*c*). We now define sequences α and τ satisfying (*a*)–(*c*). The following observations are illustrated in (*d*) of Figure 6.13. We apply Lemma 2.18 to the inverse net of $\mathcal{P}$ and $\Psi(\tau')$, and obtain nonpositive loops $\Phi_1, \ldots, \Phi_q$ with $\min(\Phi_i) \geq -(1+(n+m)W)^{n+m} \geq -(nmW+1)^{c(n+m)}$ for all $i \in \ell$ and some constant $c \in \mathbb{N}$, and $\Psi(\tau') = \sum_{i=1}^{q} \Phi_i$. By property (*c'*) and by the fact that $(\mu_\sigma - \Delta(\Phi_i))(p) \geq \mu_\sigma(p) \geq W$ holds for all $p \in {}^\bullet\sigma \supseteq {}^\bullet\tau_i$ and $i \in [q]$, we can apply Lemma 6.13 to find nonpositive loops $\tau_1, \ldots, \tau_q$ with $\Psi(\tau_i) = \Phi_i$ and $\mu_\sigma - \Delta(\tau_i) \xrightarrow{\tau_i} \mu_\sigma$ for all $i \in [q]$ such that, for some $j \in [q]$, $\tau_j =: \tau$ is a negative loop satisfying (*c*) and (*d*). Correspondingly, we define $\alpha := \alpha' \cdot \tau_1 \cdots \tau_{j-1} \cdot \tau_{j+1} \cdots \tau_q$. Note that $\alpha \cdot \tau$ is a firing sequence since α' is enabled at $\mu_\sigma - \Delta(\alpha')$ (i. e., property (*b*)) and each τ_i, $i \in [q]$, is a nonpositive loop enabled at $\mu_\sigma - \Delta(\tau_i)$. Hence, α and τ satisfy (*a*)–(*e*). □

We obtain the following canonical firing sequence by first permuting a given firing sequence appropriately, and then iteratively applying Lemma 6.18 as well as Corollary 6.17. Since the most characteristic property of this canonical sequence is that almost all transitions are contained in short negative loops, we call it *negative canonical sequence*.

Lemma 6.19. *There is a constant $c \in \mathbb{N}$ such that, for each gcf-PN $\mathcal{P} = (N, \mu_0)$ and each marking μ reachable in $\mathcal{P}$, there are $k \in [0, n \cdot \max(\mu_0)]$, transition sequences α, β, $\tau_1, \ldots, \tau_k$, and a marking $\mu^* \in \{0, W\}^n$ such that*

(*a*) *$\alpha \cdot \tau_1 \cdots \tau_k \cdot \beta$ is a firing sequence leading from μ_0 to μ,*

(*b*) *$\alpha \cdot \beta$ is enabled at $(\mu - \Delta(\alpha \cdot \beta))$ with $|\alpha \cdot \beta| \leq (nmW + \max(\mu) + 1)^{cn^2(n+m)}$,*

(*c*) *$\mu_0 + \Delta(\alpha \cdot \tau_1 \cdots \tau_k) \geq \mu^*$, and* (*d*) *each* τ_i, $i \in [k]$, *is a negative loop with*

(*i*) $\Delta(\tau_i)(p^*) \in [-(nmW+1)^{(n+m)},\ -1]$ *for some place* p^* *and* $\Delta(\tau_i)(p) = 0$ *for all* $p \neq p^*$,

(*ii*) $|\tau_i| \leq (nmW+1)^{(n+m)}$, *and*

(*iii*) $(\mu^* - \Delta(\tau_i)) \xrightarrow{\tau_i} \mu^*$.

Proof. As usual, we assume n, m, $W > 0$ since the lemma holds otherwise. Let μ be a reachable marking, and ρ be a firing sequence leading to μ.

We want to show the lemma by repeatedly applying Lemma 6.18. However, this lemma requires that the sequence σ which we are applying the lemma to satisfies $\mu_\sigma(p) \geq W$ for all $p \in {}^\bullet\sigma$. Therefore, we extract such a sequence from ρ as follows.

First, we initialize $\sigma \leftarrow \epsilon$ and $\rho_{\text{rest}} \leftarrow \rho$. As long as there is a transition $t \in \Psi_{\text{first}}(\rho_{\text{rest}})$ with $(\mu_0 + \Delta(\sigma))({}^\bullet t) \geq 2W$, we set $\sigma \leftarrow \sigma \cdot t$ and $\rho_{\text{rest}} \leftarrow \rho_{\text{rest}} \mathbin{\dot{-}} t$. At the end of this procedure, we define $\mu_\sigma := \mu_0 + \Delta(\sigma)$. It is easy to see that σ has the desired property.

Now, we iteratively apply Lemma 6.18 as long as possible, using $(P, T, F, \mu_0 + \Delta(\tau_1 \cdots \tau_{i-1}))$, α_{i-1}, and μ_σ in the i-th iteration, where τ_i and α_i (with $\alpha_0 = \sigma$) denote the sequences resulting from the i-th application of the lemma. Let $\ell \leq n \cdot \max(\mu_0)$ denote the number of applications of Lemma 6.18, and μ^* the marking defined in Lemma 6.18. We remark that α_ℓ is the "residual" sequence which remains of σ after we have extracted the negative loops $\tau_1, \ldots, \tau_\ell$. We first observe that, by Lemma 6.18, the negative loops τ_i, $i \in [\ell]$, satisfy property (*d*).

Next, we find $(\mu_\sigma - \Delta(\alpha_\ell)) \xrightarrow{\alpha_\ell} \mu_\sigma \geq \mu^*$. Furthermore, $\max(\mu_\sigma) \leq \max(\mu) + 2W$ holds. To see this, assume $\max(\mu_\sigma) > \max(\mu) + 2W$ for the sake of contradiction. Then, there is a transition $t \in \Psi_{\text{first}}(\rho_{\text{rest}})$ with $(\mu_0 + \Delta(\sigma))({}^\bullet t) \geq 2W$ which, by the construction of σ, cannot be the case. By this, we find

$$\begin{aligned}\max(\mu_\sigma - \Delta(\alpha_\ell)) &\leq \max(\mu_\sigma) + (\delta + 3) \cdot W \\ &\leq \max(\mu_\sigma) + ((\max(\mu_\sigma) + W + 1)^{n-1} + 3) \cdot W \\ &\leq (\max(\mu) + 2W)^{an}\end{aligned}$$

for some constant $a \in \mathbb{N}$. We apply Corollary 6.17 to the markings $(\mu_\sigma - \Delta(\alpha_k))$ and μ_σ, and obtain a transition sequence α (i.e., sequence ξ of Corollary 6.17) with $(\mu_\sigma - \Delta(\alpha_\ell)) = (\mu_\sigma - \Delta(\alpha)) \xrightarrow{\alpha} \mu_\sigma$. The sequence α is not necessarily a permutation of α_ℓ but has the same displacement. In particular, we have $\mu_0 \xrightarrow{\alpha} (\mu_0 + \Delta(\alpha)) \xrightarrow{\tau_1 \cdots \tau_\ell} \mu_\sigma \geq \mu^*$, meaning that (*c*) is satisfied. Let c denote the constant of Corollary 6.17 (the constant c of the Lemma we are proving at the moment is somewhat larger than the constant c of Corollary 6.17). Furthermore, let u and k be the numbers of this particular application of Corollary 6.17. Using the bounds given above, we find

$$\begin{aligned}|\alpha| \leq (k+1)u &\leq \left(\max(\mu_\sigma) + (nmW + \max(\mu_\sigma - \Delta(\alpha_\ell)) + 1)^{cn(n+m)}\right)^2 \\ &\leq \left((\max(\mu) + 2W) + (nmW + (\max(\mu) + 2W)^{an} + 1)^{cn(n+m)}\right)^2 \\ &\leq (nmW + \max(\mu) + 1)^{dn^2(n+m)}\end{aligned}$$

for some constant d. Last, we apply Corollary 6.17 to the markings μ_σ and μ, yielding a transition sequence β (i.e., sequence ξ of Corollary 6.17) with $\mu_\sigma \xrightarrow{\beta} \mu$. Let u, k the numbers of this particular application of Corollary 6.17. Then, we observe

$$\begin{aligned}|\beta| &\le (k+1)u \le \big(\max(\mu) + (nmW + \max(\mu_\sigma) + 1)^{cn(n+m)}\big)^2 \\ &\le \big(\max(\mu) + (nmW + \max(\mu) + 2W + 1)^{cn(n+m)}\big)^2 \\ &\le (nmW + \max(\mu) + 1)^{dn(n+m)}\end{aligned}$$

for some constant $d \in \mathbb{N}$. The observations above imply $(\mu_\sigma - \Delta(\alpha)) \xrightarrow{\alpha} \mu_\sigma \xrightarrow{\beta} \mu$ as well as $\mu_0 \xrightarrow{\alpha} (\mu_0 + \Delta(\alpha)) \xrightarrow{\tau_1 \cdots \tau_k} \mu_\sigma \xrightarrow{\beta} \mu$. This together with the bounds on $|\alpha|$ and $|\beta|$ ensures properties (*a*) and (*b*). □

By reversing negative canonical firing sequences of gcf-PNs, we obtain positive canonical firing sequences of igcf-PNs.

Corollary 6.20. *There is a constant $c \in \mathbb{N}$ such that, for each igcf-PN $\mathcal{P} = (N, \mu_0)$ and each marking μ reachable in $\mathcal{P}$, there are $k \in [0, n \cdot \max(\mu)]$, transition sequences α, β, $\tau_1, \ldots, \tau_k$, and a marking $\mu^* \in \{0, W\}^n$ such that*

(*a*) *$\alpha \cdot \tau_1 \cdots \tau_k \cdot \beta$ is a firing sequence leading from μ_0 to μ,*

(*b*) *$\alpha \cdot \beta$ is enabled at μ_0 with $|\alpha \cdot \beta| \le (nmW + \max(\mu_0) + 1)^{cn^2(n+m)}$,*

(*c*) *$\mu_0 + \Delta(\alpha) \ge \mu^*$, and*

(*d*) *each τ_i, $i \in [k]$, is a positive loop with*

- (*i*) *$\Delta(\tau_i)(p^*) \in [1, (nmW+1)^{c(n+m)}]$ for some place p^* and $\Delta(\tau_i)(p) = 0$ for all $p \neq p^*$,*
- (*ii*) *$|\tau_i| \le (nmW + 1)^{c(n+m)}$, and*
- (*iii*) *τ_i is enabled at μ^*.*

Proof. Lemma 6.19 implies the corollary by considering the inverse net of $\mathcal{P}$, as well as μ as the initial marking and μ_0 as the end marking. □

Corollary 6.21. *There is a constant $c \in \mathbb{N}$ such that the class of igcf-PNs is simple structurally f-g-canonical, where $f(n, m, W, K) = (nmW + K + 1)^{cn^2(n+m)}$ and $g(n, m, W, K) = 2$.*

Using our framework of Chapter 4 and the canonical firing sequences given here, we obtain constructions for SLSRs of the reachability sets of gcf-PNs and igcf-PNs.

Lemma 6.22. *Given a (generalized) communication-free Petri net $\mathcal{P} = (P, T, F, \mu_0)$, we can construct an SLSR of $\mathcal{R}(\mathcal{P})$ in (doubly, resp.) exponential time.*

Proof. For cf-PNs, the claim follows from Theorem 3.14. We now consider the class of gcf-PNs. By Lemma 6.16, this class is f-f-canonical, where $f(n, m, W, K) = (nmW + K + 1)^{cn(n+m)}$ for some constant c. Obviously, it is also simple structurally f-f-canonical. Hence, by Theorem 4.4, there is a polynomial p such that we can construct, for a gcf-PN $\mathcal{P}$, a SLSR of $\mathcal{R}(\mathcal{P})$ in time polynomial in

$$\begin{aligned}
&(\max(\mu_0) + 2f(n, m, W, \max(\mu_0))W)^{f(n, m, W, \max(\mu_0))n} \cdot \\
&\quad \cdot 2^{p(\mathrm{size}(\mathcal{P})) + n\,\mathrm{ld}(f(n,m,W,\max(\mu_0)+2f(n,m,W,\max(\mu_0))W))} + r \\
&\overset{\mathrm{P}}{\leq} 2^{2^{p(\mathrm{size}(\mathcal{P}))}} \cdot 2^{p(\mathrm{size}(\mathcal{P})) + n\,\mathrm{ld}(2^{p(\mathrm{size}(\mathcal{P}))})} \overset{\mathrm{P}}{\leq} 2^{2^{p(\mathrm{size}(\mathcal{P}))}}.
\end{aligned}$$

□

This construction is reminiscent of that given in Section 3.2 for SLSRs of reachability sets of cf-PNs which uses results of Yen [Yen97], and yields SLSRs of single exponential encoding size. The difference in the encoding sizes between these SLSRs for cf-PNs and the SLSRs of Lemma 6.22 for gcf-PNs does not result from the slight differences between the canonical firing sequences of cf-PNs (given in [Yen97]) and gcf-PNs themselves (in fact, our canonical sequence can also be used to generate the semilinear representations for cf-PNs in single exponential time). Rather, it results from the following.

For cf-PNs, we used that each loop that is intermediately enabled by some backbone can be partitioned into suitable loops which are intermediately enabled by every other backbone with the same Parikh image. Therefore, it is sufficient to only consider one of these backbones. This results in a single exponential number of relevant backbones, and therefore in a single exponential number of linear sets, each of single exponential size. However, the same strategy fails in the case of gcf-PNs since the order of the transitions is much more relevant for gcf-PNs than for cf-PNs: firing transitions in a certain order can intermediately enable loops that cannot be partitioned further and that are not intermediately enabled by firing the same transitions in some other order. This is illustrated in Figure 6.14. Hence, to improve the upper bound on the size of the SLSRs for the reachability sets of gcf-PNs, some other or a refined approach will have to be found.

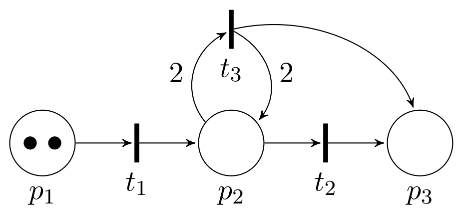


Figure 6.14: The firing sequences $t_1t_1t_2t_2$ and $t_1t_2t_1t_2$ have the same Parikh image but only the first sequence intermediately enables the positive loop t_3.

Lemma 6.23. *Given a igcf-PN $\mathcal{P} = (P, T, F, \mu_0)$, we can construct an SLSR of $\mathcal{R}(\mathcal{P})$ in exponential time.*

Proof. By Corollary 6.20, the class of igcf-PNs is simple structurally f-g-canonical for functions f and g with $f(n, m, W, K) = (nmW + K + 1)^{cn^2(n+m)}$ for some constant c and $g(n, m, W, K) = 2$. Hence, by Theorem 4.4, there is a polynomial p such that we can construct, for an igcf-PN $\mathcal{P}$, a SLSR of $\mathcal{R}(\mathcal{P})$ in time polynomial in

$$\begin{aligned}
&(\max(\mu_0) + 2f(n,m,W,\, \max(\mu_0))W)^{g(n,m,W,\, \max(\mu_0))n} \cdot \\
&\quad \cdot 2^{p(\text{size}(\mathcal{P}))+n\,\mathrm{ld}(f(n,m,W,\, \max(\mu_0)+2f(n,m,W,\, \max(\mu_0))W))} + r \\
&\overset{\mathrm{P}}{\leq} 2^{p(\text{size}(\mathcal{P}))} \cdot 2^{p(\text{size}(\mathcal{P}))+n\,\mathrm{ld}(2^{p(\text{size}(\mathcal{P}))})} \overset{\mathrm{P}}{\leq} 2^{p(\text{size}(\mathcal{P}))}.
\end{aligned}$$

□

6.3 Complexity results

In this section, we bring together the lower bounds discovered in Section 6.1 and the upper bounds provided by applying our framework of Chapter 4. For most of our problems of interest this yields **PSPACE**-completeness. We remark that **PSPACE**-completeness of the RecLFS-problem has already been shown in Theorem 6.9.

Theorem 6.24. *The following problems for gcf-PNs and igcf-PNs are* **PSPACE***-complete in the strong sense, even if restricted to either a. o. gss-PNs, or a. o. gcf-PNs, or a. o. igcf-PNs:*

- *zero-reachability,*
- *reachability,*
- *boundedness, and*
- *covering.*

The liveness problem of gcf-PNs is **PSPACE***-complete in the strong sense, even if restricted to a. o. gcf-PNs. The liveness problem of igcf-PNs is in* **PSPACE***.*

Proof. The lower bound for the zero-reachability, the boundedness, and the covering problems are shown in Lemma 6.8. Let f be a function defined by $f(n,m,W,K) = (nmW + K + 1)^{cn^2(n+m)}$. Then, by Lemma 6.16 and Corollary 6.21, the class consisting of all gcf-PNs and igcf-PNs is simple structurally f-f-canonical. By Theorem 4.3, the reachability, and the covering problems of gcf-PNs and igcf-PNs are decidable in space polynomial in

$$\text{size}(\mathcal{P}) + \text{size}(\mu) + n\,\mathrm{ld}\, f(n,m,W,\, \max(\mu_0)) + r \overset{\mathrm{P}}{\leq} \text{size}(\mathcal{P}) + \text{size}(\mu).$$

Similarly, the boundedness problem is decidable in space polynomial in

$$\text{size}(\mathcal{P}) + n\,\mathrm{ld}\, f(n,m,W,\, \max(\mu_0)) + r \overset{\mathrm{P}}{\leq} \text{size}(\mathcal{P}).$$

By Theorem 4.5, the liveness problem of gcf-PNs and igcf-PNs is decidable in space polynomial in

$$\text{size}(\mathcal{P}) + n\,\mathrm{ld}(f(n,m,W,\, \max(\mu_0) + f(n,m,W,\, \max(\mu_0))W)) + r \overset{\mathrm{P}}{\leq} \text{size}(\mathcal{P}).$$

The lower bound for the liveness problem of gcf-PNs is shown in Lemma 6.11. □

Theorem 6.25. *The containment and the equivalence problems of gcf-PNs and igcf-PNs are* **PSPACE***-hard in the strong sense, even if restricted to either a. o. gss-PNs, a. o. gcf-PNs, or a. o. igcf-PNs. Furthermore, these problems are decidable in doubly exponential space for gcf-PNs, and in exponential space for igcf-PNs.*

Proof. We first show **PSPACE**-hardness for these problems. Consider the wipe-extension $\widehat{\mathcal{P}}_3$ of the Petri net $\mathcal{P}_3 = (P, T, F, \mu_0)$ constructed in Section 6.1. Observe that M accepts x if and only if every marking of $\widehat{\mathcal{P}}_3$ is reachable in $\widehat{\mathcal{P}}_3$. We construct a gss-PN $\mathcal{P} = (P, T, F, \mu_0)$ with the same places as $\widehat{\mathcal{P}}_3$ and, for each place $p \in P$, two transitions t_p and t'_p for each place $p \in P$ such that $0 \xmapsto{t_p} p$ and $p \xmapsto{t'_p} 0$. Note that every marking of $\widehat{\mathcal{P}}_3$ is reachable in $\mathcal{P}$. Hence, M accepts x if and only if $\mathcal{R}(\mathcal{P}) \subseteq \mathcal{R}(\widehat{\mathcal{P}}_3)$ if and only if $\mathcal{R}(\mathcal{P}) = \mathcal{R}(\widehat{\mathcal{P}}_3)$. Now, we prove the upper bounds for these problems. As before, let f be a function defined by $f(n, m, W, K) = (nmW + K + 1)^{cn^2(n+m)}$. Then, by Lemma 6.16 and Corollary 6.21, the class consisting of all gcf-PNs and igcf-PNs is simple structurally f-f-canonical. By Theorem 4.6 and Lemma 6.16, there are polynomials p, p' such that the equivalence and containment problems of gcf-PNs are decidable in space polynomial in

$$\begin{aligned}&(K + 2u_1K)^{u_2n} \cdot 2^{p(s+n\,\mathrm{ld}(u_3))} + r\\ \overset{\mathrm{P}}{\leq}\, &(K + 2f(n, m, W, K)K)^{f(n,m,W,K)n} \cdot 2^{p(s+n\,\mathrm{ld}\, f(n,m,W,K+2f(n,m,W,K)W))} \overset{\mathrm{P}}{\leq} 2^{2^{p'(s)}}.\end{aligned}$$

In case of igcf-PNs, Theorem 4.6 and Corollary 6.21 imply the existence of polynomials p, p' such that the equivalence and containment problems are decidable in space polynomial in

$$\begin{aligned}&(K + 2u_1K)^{u_2n} \cdot 2^{p(s+n\,\mathrm{ld}(u_3))} + r\\ \overset{\mathrm{P}}{\leq}\, &(K + 2f(n, m, W, K)K)^{2n} \cdot 2^{p(s+n\,\mathrm{ld}\, f(n,m,W,K+2f(n,m,W,K)W))} \overset{\mathrm{P}}{\leq} 2^{p'(s)}.\end{aligned}$$

□

We now consider home spaces of gcf-PNs and igcf-PNs. Using the SLSRs of their reachability sets provided by our framework, we can show the following results.

Theorem 6.26. *Given a (generalized) communication-free Petri net $\mathcal{P}$, we can*

(*a*) *determine if $\mathcal{P}$ has a finite home space,*

(*b*) *determine a minimal finite home space if a finite home space exists, and*

(*c*) *determine an SLSR of a home space,*

all in (doubly, resp.) exponential space.

Proof. We first prove that a gcf-PN $\mathcal{P} = (P, T, F, \mu_0)$ has a finite home space consisting of markings whose components are at most exponential in the size of $\mathcal{P}$, provided $\mathcal{P}$ has a finite home space at all. To this end, assume that $\mathcal{P}$ has a minimal finite home space $\mathcal{HS}$.

The general idea is: If there is a large marking in the home space, then the positive loops contained in the canonical firing sequence leading to this marking imply the existence of an even larger reachable marking. Then, the existence of negative loops in the canonical firing sequence leading from such a very large reachable marking to a marking of the home space implies that, by using these negative loops, also the large marking in the home space can appropriately be decreased, yielding another home space with some smaller marking than the original home space.

Let $x := \max\{\max(\mu) \mid \mu \in \mathcal{HS}\}$ be the maximum component over all markings of $\mathcal{HS}$. Assume that there is a marking $\mu^* \in \mathcal{HS}$ such that, for some place p^*, $\mu^*(p^*) > \mu_0(p^*) + (nmW + \max(\mu_0) + 1)^{cn(n+m)} \cdot W$, where c is the constant of Corollary 6.17. Since, by the minimality of $\mathcal{HS}$, μ^* is reachable, the same lemma implies that there are sequences α, β, τ such that τ is a positive loop for p^* and $\mu_0 \xrightarrow{\alpha\cdot\tau\cdot\beta} \mu^*$. Therefore, p^* is unbounded, ensuring the existence of a reachable marking μ with $\mu(p^*) > x + (nmW + x + 1)^{cn^2(n+m)}$, where c is the constant of Lemma 6.19. Then, by the same lemma, there is a marking $\mu' \in \mathcal{HS}$ (with $\max(\mu') \leq x$) such that $\mu \xrightarrow{\alpha\cdot\tau\cdot\beta} \mu'$, where τ is a negative loop for p^*. For the following argumentation, μ and μ' aren't needed anymore. Important is only the existence of some negative loop (i. e., τ) that decreases the number of tokens at p^*.

We shift our attention back to μ^*. Let ρ be a transition sequence enabled at μ^* that is obtained by greedily firing as many transitions of τ as possible. Let ν^* be the marking reached by firing ρ at μ^*. Then, we observe $\nu^*(p^*) < \mu^*(p^*)$ and $\nu^*(p) \leq \max\{\mu^*(p), W\}$ for all $p \neq p^*$. To see this, assume for the sake of contradiction that $\nu^*(p^*) \geq \mu^*$ or $\nu^*(p) > \max\{\mu^*(p), W\}$ for some $p \neq p^*$. If $\nu^*(p^*) \geq \mu^*$, then $\tau \mathbin{\dot{-}} \rho$ contains a transition t with ${}^\bullet t = p^*$ since τ is a negative loop for p^*. Obviously, t can be fired at ν^*, contradicting the maximality of ρ. If $\nu^*(p) > \max\{\mu^*(p), W\}$, then an analogous argument holds.

Therefore, we can replace μ^* by ν^* to obtain another home space. Iteratively applying this argument yields a home space $\mathcal{HS}^*$ such that $\max(\mu) \leq \mu_0(p^*) + (nmW + \max(\mu_0) + 1)^{cn(n+m)} \cdot W$ for all $\mu \in \mathcal{HS}^*$. In other words, if $\mathcal{P}$ has a finite home space, then it has a finite home space which consists of at most exponentially many markings (more precisely, at most $(\mu_0(p^*) + (nmW + \max(\mu_0) + 1)^{cn(n+m)} \cdot W + 1)^n$), each of polynomial encoding size.

All such sets of markings that are candidates for minimal finite home spaces can be enumerated in exponential space. We now show how to decide for such a candidate set M if it is a minimal home space. To determine whether M is minimal, we simply test for all μ, $\mu' \in M$ if $\mu \to \mu'$. If this is the case for such a pair of markings, then we discard μ since $M \setminus \{\mu\}$ is also a home space and therefore M is not minimal. By Theorem 6.24 and the bounds on the encoding sizes of μ and μ', this test can be performed in polynomial space in size($\mathcal{P}$) (not including the space needed to store M, which is (at most) exponential in size($\mathcal{P}$)). Now, assume that M is minimal in this sense, and we want to determine if it is a home space. Let $(\bar{N}, \mu_0)$ denote the inverse net of $\mathcal{P}$, and let $\mathcal{SL}_{(\bar{N},M)}$ be the SLSR which is obtained by combining the SLSRs of all reachability sets $\mathcal{R}(\bar{N}, \mu)$, $\mu \in M$. By Lemma 6.23, $\mathcal{SL}_{(\bar{N},M)}$ can be constructed in exponential time. The set $\boldsymbol{\mathcal{SL}}_{(\bar{N},M)}$ contains all markings μ' for which a marking $\mu \in M$ with $\mu' \to \mu$ exists, i. e., all markings from which we can reach a marking of M in $\mathcal{P}$. Let $\mathcal{SL}_{\mathcal{R}(\mathcal{P})}$ be the SLSR of $\mathcal{R}(\mathcal{P})$ as of Lemma 6.22. $\mathcal{SL}_{\mathcal{R}(\mathcal{P})}$ can be constructed in exponential time if $\mathcal{P}$ is a cf-PN, and in doubly exponential time if $\mathcal{P}$ is a gcf-PN.

Our candidate M is a home space if and only if $\boldsymbol{\mathcal{SL}}_{\mathcal{R}(\mathcal{P})} \subseteq \boldsymbol{\mathcal{SL}}_{(\bar{N},M)}$ (we remark that an analogue observation regarding Petri nets and their inverse nets in the special case of home states was made by Melinte et al. [Mel+02]). By Corollary 2.5, this condition can be checked in polynomial space in size($\mathcal{SL}_{\mathcal{R}(\mathcal{P})}$) + size($\mathcal{SL}_{(\bar{N},M)}$), i. e., in exponential space respectively doubly exponential space in size($\mathcal{P}$). Moreover, if no candidate M is a finite home space, then $\mathcal{P}$ doesn't have one. In this case, $\mathcal{SL}_{\mathcal{R}(\mathcal{P})}$ is an SLSR of the trivial home space $\mathcal{R}(\mathcal{P})$. □

It was shown by Frutos Escrig and Johnen [FEJ89] that the following problem is decidable: Given a Petri net $\mathcal{P}$, and an SLSR $\mathcal{SL}$ consisting of LSRs having the same periods, is $\boldsymbol{\mathcal{SL}}$ a home space of $\mathcal{P}$? Note that this problem generalizes the home space recognition problem as defined in Theorem 5.14. The following theorem improves on this bound in case of igcf-PNs. In particular, it does not require any restrictions on the given SLSR.

Theorem 6.27. *Given an inverse (generalized) communication-free Petri net $\mathcal{P} = (P, T, F, \mu_0)$ and an SLSR $\mathcal{SL} = \bigodot_{i=1}^{k} \mathcal{L}(\zeta_i, \Pi_i)$, we can determine in (doubly, resp.) exponential space if $\boldsymbol{\mathcal{SL}}$ is a home space of $\mathcal{P}$.*

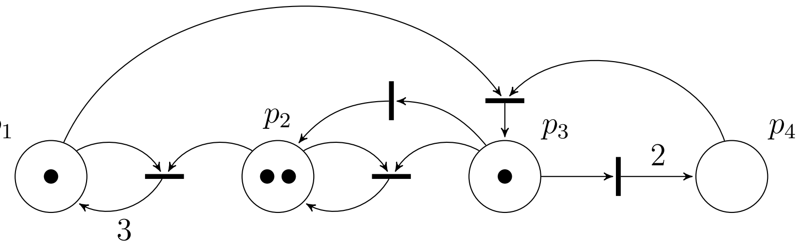
Proof. The proof idea is straightforward. Given the inverse (generalized) communication-free Petri net $\mathcal{P}$, we first show that we can construct, in (doubly, resp.) exponential time, a SLSR $\mathcal{SL}_M$ of the set M which consists of all markings from which we can reach some marking of $\boldsymbol{\mathcal{SL}}$. Given $\mathcal{SL}_M$, we ask if $\boldsymbol{\mathcal{SL}}_{\mathcal{R}(\mathcal{P})} \subseteq \boldsymbol{\mathcal{SL}}_M$, where $\mathcal{SL}_{\mathcal{R}(\mathcal{P})}$ denotes the SLSR of $\mathcal{R}(\mathcal{P})$ given in Lemma 6.23. This is the case if and only if $\boldsymbol{\mathcal{SL}}$ is a home space. Since, by Lemma 6.23, $\mathcal{SL}_{\mathcal{R}(\mathcal{P})}$ can be constructed in exponential time, and the containment problem of SLSRs is, by Corollary 2.5, decidable in polynomial space, the theorem then follows.

We now present the construction for $\mathcal{SL}_M$. Let $N := (P, T, F')$ be the unmarked (generalized) cf-PN that is the inverse Petri net of $\mathcal{P}$. The important observation is that $M = \bigcup_{\mu \in \boldsymbol{\mathcal{SL}}} \mathcal{R}(N, \mu)$. Hence, we extend N to a (generalized) cf-PN $\mathcal{P}' = (P', T', F', \mu'_0)$ by adding control places and transitions such that the original places of $\mathcal{P}'$ can be marked with each marking $\mu \in \boldsymbol{\mathcal{SL}}$ by some firing sequence. Starting from such a marking, the original transitions can be applied to reach any marking of M for the original places P. In particular, our construction will ensure that only these markings can be reached.

The following construction is illustrated in Figure 6.15. We introduce a new place q_{init}, as well as, for each LSR $\mathcal{L}(\zeta_i, \Pi_i)$,

- a place q_i,
- a transition r_i with $q_{\text{init}} \overset{r_i}{\longmapsto} q_i + \sum_{p \in P} \zeta_i(p) \cdot p$, and
- for each period $\pi \in \Pi_i$, a transition r_π with $q_i \overset{r_\pi}{\longmapsto} q_i + \sum_{p \in P} \pi(p) \cdot p$.

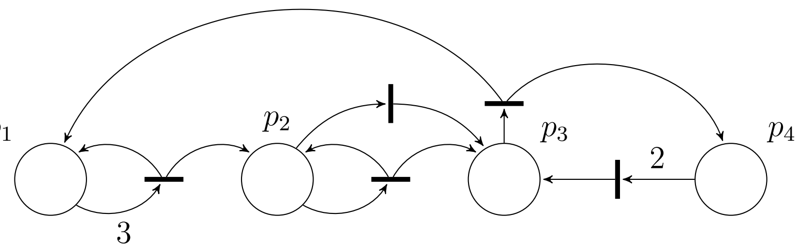
(*a*)



(*b*)

$$\mathcal{SL} = \mathcal{L}(\zeta_1, \{\pi_{1,1}, \pi_{1,2}\}) \odot \mathcal{L}(\zeta_2, \{\pi_{2,1}\}), \text{ where}$$
$$\zeta_1 = (1,2,3,4), \pi_{1,1} = (7,1,3,0), \pi_{1,2} = (3,1,1,2)$$
$$\zeta_2 = (3,2,5,6), \pi_{2,1} = (2,8,2,3)$$

(*c*)



(*d*)

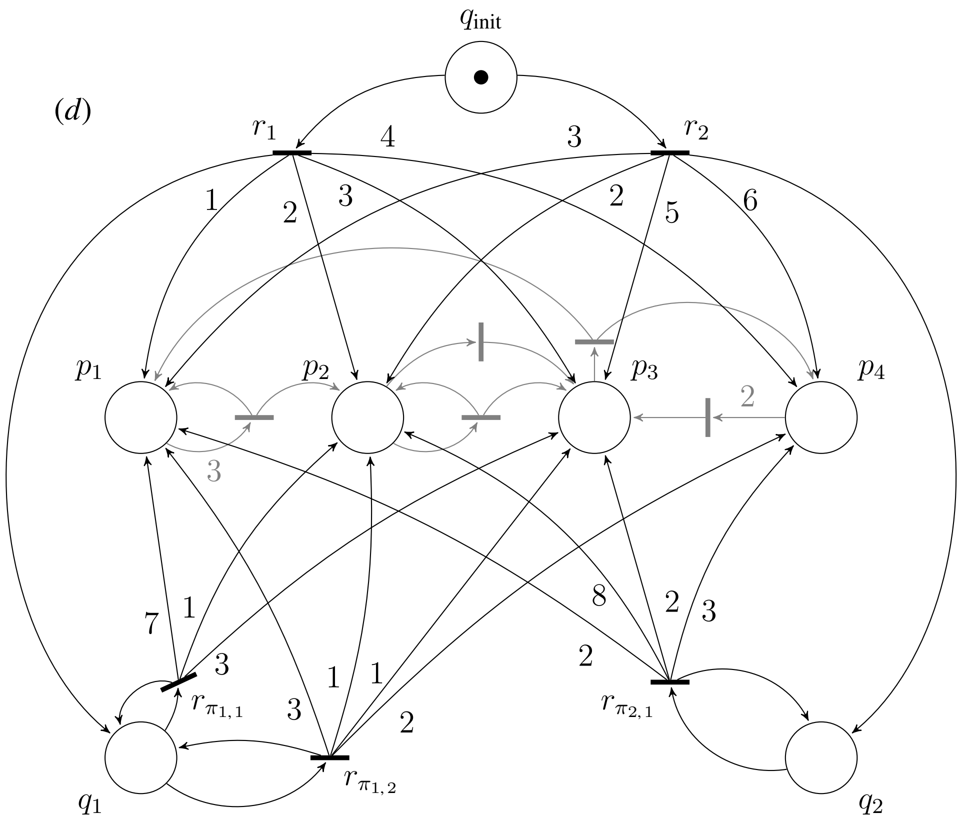


Figure 6.15: (*a*) illustrates an igcf-PN $\mathcal{P}$, (*b*) an SLSR $\mathcal{SL}$, (*c*) the inverse net N of $\mathcal{P}$ (which is a gcf-PN), and (*d*) the gcf-PN $\mathcal{P}'$ as defined in the proof of Theorem 6.27, where, for the sake of clarity, the original transitions of N are gray and the new transitions are black.

The initial marking is $\mu'_0 :\triangleq q_{\text{init}}$. Note that, even though a transition r_p could add more tokens to the original places after some original transitions of T have already been fired, we can w. l. o. g. assume all occurrences of r_p took place before all occurrences of original transitions. We conclude that the projection of all reachable markings of $\mathcal{P}'$ to the set P of original places equals M. Thus, we compute a SLSR of $\mathcal{R}(\mathcal{P}')$ and project it to the original places of P (i. e., to the first n components), yielding a SLSR of M. If $\mathcal{P}$ is an inverse (generalized) cf-PN, then $\mathcal{P}'$ is a (generalized, resp.) cf-PN, and, by Lemma 6.22, $\mathcal{SL}_M$ can be computed in (doubly, resp.) exponential time. □

Even though both theorems refer to (seemingly) very similar problems, it is an open problem how to prove an analogon of Theorem 6.26 for igcf-PNs or an analogon of Theorem 6.27 for gcf-PNs.

6.4 Exponent-sensitive grammars

We refer to Section 2.5 for the definition of commutative grammars. Based on our results for gcf-PNs, we can give complexity results for the uniform word problem of a new class of commutative grammars we call exponent-sensitive commutative grammars (ESCGs). An ESCG is commutative grammar (V_N, V_T, P, s) such that $P \subset V_N^{\oplus} \times (V_N \cup V_T)^{\circledast}$. In other words, productions of an ESCG can substitute many occurrences of a single variable $x \in V_N$ at once by variables and terminal symbols. Hence, in a certain sense, ESCGs are not entirely context-free. Note, though, that the families of ESCGs and of context-sensitive commutative grammars (CSCGs) are incomparable: ESCGs allow productions substituting a commutative word by some shorter commutative word, which is not allowed for CSCGs, whereas productions of CSCGs can substitute a word consisting of different variables, which is not allowed for ESCGs. We can show that the uniform word problem of ESCGs (see Definition 2.3) has the same complexity as the reachability problem of gcf-PNs.

Theorem 6.28. *The uniform word problem of ESCGs is* **PSPACE***-complete in the strong sense.*

Proof. Using canonical Petri nets of ESCGs and canonical commutative grammars of gcf-PNs (see Section 2.5), we observe that the uniform word problem of ESCGs and the reachability problem of gcf-PNs are reducible to each other in polynomial time. Thus, Theorem 6.24 implies that the uniform word problem of ESCGs is **PSPACE**-complete in the strong sense. □

We now introduce the corresponding non-commutative analogon of ESCGs called exponent-sensitive grammars (ESGs). A grammar $G = (V_N, V_T, P, s)$ is exponent-sensitive if $P \subset \{x^+ \mid x \in V_N\} \times (V_N \cup V_T)^*$. We say that G is in ESG normal form if, for each production $p \in P$, there are $A \in V_T$ and B, $C \in V_T \cup V_N$ such that p is one of the productions $A \to BC$, $AA \to B$, or $A \to \epsilon$. Note that similarly to ESCGs and CSCGs, the classes of ESGs and context-sensitive grammars are incomparable. In the following, $L(G) = \{w \in V_T^* \mid s \to^* w\}$ denotes the language generated by G.

Theorem 6.29. *For each grammar G, we can, in polynomial time, compute a grammar G' in ESG normal form such that $L(G) = L(G')$.*

Proof. Let $G = (V_N, V_T, P, s)$ be an unrestricted grammar such that, w. l. o. g., each production has at most two symbols on each side, and the symbols on the left-hand side are only variables. In addition, we assume w. l. o. g. that we have no productions of the forms $AB \to C$, $AB \to \epsilon$ and $A \to C$ since these can appropriately be substituted by a set of new productions of the form $AB \to CD$ and $A \to \epsilon$

using additional variables. Therefore, we only have to show how to replace productions of the form $AB \to CD$ by productions allowed for grammars in ESG normal form.

Assume that there is a production $p = AB \to CD \in P$. In the following we show how to obtain a grammar $\bar{G} = (\bar{V}, \Sigma, P \setminus \{p\} \cup P_p, s)$ (i. e., we obtain $\bar{G}$ from G by replacing p by a set of productions P_p) generating the same language as G such that $V \subset \bar{V}$, and each production of P_p is either allowed for grammars in ESG normal form or has the form $A'A' \to B'C'D'E'$ or $A'A' \to \epsilon$. Productions of the additional forms are used for convenience only and can easily be replaced by productions having the desired form.

Let P_p consist of the productions $A \to LX$, $B \to XR$, $XX \to LCDR$, $LL \to \epsilon$, and $RR \to \epsilon$, where L, R, X are new variables not contained in V. We first show that two nonempty maximal subsequences of the form $L^*X^*R^*$ (i. e., not properly contained in some subsequence of this form) cannot overlap. To this end, assume that there is a substring $\alpha\beta\gamma$ with $|\beta| > 0$ such that both $\alpha\beta$ and $\beta\gamma$ are of this form. Then $|\alpha| > 0$ or $|\gamma| > 0$. Assume $|\alpha| > 0$. If $\beta_{[1]} = L$, then α is of the form L^* and $\beta\gamma$ of the form $L^*X^*R^*$, a contradiction to $\beta\gamma$ being maximal. If $\beta_{[1]} = X$, then α is of the form L^*X^* and $\beta\gamma$ of the form X^*R^*, again a contradiction. If $\beta_{[1]} = R$, then α is of the form $L^*X^*R^*$ and $\beta\gamma$ of the form R^*, a contradiction. The case $|\gamma| > 0$ is analogous.

In the following, we write X_L (X_R, resp.) for a symbol X produced by a production $A \to LX$ ($B \to XR$, resp.). We now prove by induction the structural invariant that, for each word w generated by a sequence of productions of $\bar{G}$, each of its nonempty maximal subsequences $L^pX^*R^q$ has the form

(*i*) L^pR^q where p and q are even,

(*ii*) $L^pX_LR^q$ where p is odd and q is even,

(*iii*) $L^pX_RR^q$ where p is even and q is odd, or

(*iv*) $L^pX_LX_RR^q$ where both p and q are odd.

This is obviously true for the word consisting only of the start symbol s. Assume it for all words obtained by a sequence of $k-1$ productions of $\bar{G}$. If the k-th production $\bar{p}$ doesn't change the structure of nonempty maximal subsequences of the form $L^*X^*R^*$, then the invariant holds for the word resulting from this k-th production. Therefore, we assume that $\bar{p}$ changes this structure. We first consider the case $\bar{p} \notin P_p$. Keep in mind that $\bar{p}$ cannot operate on the symbols L, R or X. Therefore, $\bar{p}$ can change the structure only by gluing two nonempty maximal subsequences to a longer subsequence by deleting symbols between them. One could think of a structural change by first making two such subsequences neighbors resulting in again two nonempty maximal subsequences, where one sequence is longer than the sequence it stems from, and the other one is shorter. However, this is prevented by the maximality condition and the fact that two maximal subsequences cannot overlap. By carefully investigating all 16 possibilities of merging two such sequences of the form (*i*)–(*iv*), we find that the invariant holds after the merge.

We list the different possibilities how two such subsequences can merge in Table 6.1. The properties given in the fourth column holds under the premise that the invariant holds before the application of the production and both subsequences are merged resulting in a structural change. These properties imply that the invariant holds again after the merge. The case (*iii*)&(*iii*) is symmetrical to (*ii*)&(*ii*), (*iv*)&(*i*) is symmetrical to (*i*)&(*iv*), and all other cases involving (*iv*)-sequences are impossible since the resulting sequence would contain two X separated by a nonempty string of the form L^+ or R^+ which is incompatible with the form $L^*X^*R^*$.

Left sequence		Right sequence		Resulting sequence		Properties
Type	Form	Type	Form	Type	Form	
(*i*)	L^pR^q	(*i*)	$L^{p'}R^{q'}$	(*i*) (*i*)	$L^pR^{q+q'}$ or $L^{p+p'}R^{q'}$	p, q, p', q' even, ($q = 0$ or $p' = 0$)
(*i*)	L^pR^q	(*ii*)	$L^{p'}X_LR^{q'}$	(*ii*)	$L^{p+p'}X_LR^{q'}$	p, q' even, p' odd, $q = 0$
(*i*)	L^pR^q	(*iii*)	$L^{p'}X_RR^{q'}$	(*iii*)	$L^{p+p'}X_RR^{q'}$	p, p' even, q' odd, $q = 0$
(*i*)	L^pR^q	(*iv*)	$L^{p'}X_LX_RR^{q'}$	(*iv*)	$L^{p+p'}X_LX_RR^{q'}$	p even, $q = 0$ p', q' odd
(*ii*)	$L^pX_LR^q$	(*i*)	$L^{p'}R^{q'}$	(*ii*)	$L^pX_LR^{q+q'}$	q, q' even, p odd, $p' = 0$
(*ii*)	$L^pX_LR^q$	(*ii*)	$L^{p'}X_LR^{q'}$	impossible since		p' odd
(*ii*)	$L^pX_LR^q$	(*iii*)	$L^{p'}X_RR^{q'}$	(*iv*)	$L^pX_LX_RR^{q'}$	p, q' odd, $q = p' = 0$
(*iii*)	$L^pX_RR^q$	(*i*)	$L^{p'}R^{q'}$	(*iii*)	$L^pX_RR^{q+q'}$	p, q' even, q odd, $p' = 0$
(*iii*)	$L^pX_RR^q$	(*ii*)	$L^{p'}X_LR^{q'}$	impossible since		q, p' odd

Table 6.1: The possibilities in which two nonempty maximal subsequences of the form $L^*X^*R^*$ can merge by a production that deletes all symbols between the two subsequences.

Next, consider the case $\bar{p} \in P_p$. The production $A \to LX$ either glues two (possibly empty) (*i*)-sequences by substituting the separating A, resulting in a (*ii*)-sequence, or glues a (possibly empty) (*i*)-sequence and a (*iii*)-sequence, resulting in a (*iv*)-sequence. This conserves the structural invariant. Equivalently, the production $B \to XR$ conserves the structural invariant. The production $XX \to LCDR$ splits a (*iv*)-sequence into two (*i*)-sequences which then are separated by CD. The productions $LL \to \epsilon$ and $RR \to \epsilon$ transform, for some j, a (j)-sequence into some other (possibly empty) (j)-sequence. Altogether, we find that the structural invariant is satisfied for every sequence of productions of $\bar{G}$.

Consider a sequence of words $\bar{w}_0, \ldots, \bar{w}_k \in (\bar{V} \cup \Sigma)^*$ where $\bar{w}_0 = s$ and $\bar{w}_i$ is produced by applying a production of $\bar{G}$ to $\bar{w}_{i-1}$. For each word $\bar{w}_i$ we obtain the word w_i by deleting all occurrences of L and R, and substituting each X_L (i. e., X resulting from a production $A \to LX$) by A and each X_R by B.

In the following, we show that, for each i, either $w_i = w_{i-1}$, or w_i can be obtained by applying a production of G to w_{i-1}. If we obtain $\bar{w}_i$ from $\bar{w}_{i-1}$ by applying a production of $P \setminus \{p\}$, then we can simply apply the same production to w_{i-1} to obtain w_i. If $\bar{w}_i$ results from applying $A \to LX$, $B \to XR$, $LL \to \epsilon$, or $RR \to \epsilon$, then $w_i = w_{i-1}$. The crucial case is, when $XX \to LCDR$ is applied to $\bar{w}_{i-1}$. Then, we can apply $AB \to CD$ to w_{i-1} to obtain w_i. The reason is that our structural invariant implies that the first X must be an X_L and the second must be an X_R, corresponding to AB in w_{i-1}.

This means, not only can productions of G be simulated by productions of $\bar{G}$ but also a sequence of productions of $\bar{G}$ leading to a word consisting only of terminal symbols can be simulated by a sequence of productions of G. In other words, G and $\bar{G}$ generate the same language. Therefore, we can replace each production of the form $AB \to CD$ by a set of productions as defined above without changing the language of the grammar. To finally obtain a grammar with the desired form, we replace, in the obvious way, the productions of the form $AA \to BCDE$ and $AA \to \epsilon$ by a set of productions which are allowed for grammars in ESG normal form. It is not hard to see that we can, in polynomial time, transform a grammar into a grammar in ESG normal form as described in this proof. □

Theorem 6.30. *The uniform word problem of exponent-sensitive grammars is undecidable.*

Proof. By Theorem 6.29, the word problem of general grammars, which is undecidable, and the word problem of exponent-sensitive grammars are recursively equivalent. □

Table 6.2 summarizes the complexities for the (uniform) word problem of different classes of grammars that are of particular interest in the context of ESGs and ESCGs.

<table>
<tr><th colspan="2">Class</th><th>(Uniform) word problem</th><th>References</th></tr>
<tr><td rowspan="2">context-free</td><td>non-commutative</td><td>P-complete</td><td>[Coc69; JL76; Kas65; Sip97; You67]</td></tr>
<tr><td>commutative</td><td>NP-complete</td><td>[Esp97; Huy83]</td></tr>
<tr><td rowspan="2">context-sensitive</td><td>non-commutative</td><td>PSPACE-complete</td><td>[Kur64]</td></tr>
<tr><td>commutative</td><td>PSPACE-complete</td><td>[Huy83]</td></tr>
<tr><td rowspan="2">exponent-sensitive</td><td>non-commutative</td><td>undecidable</td><td rowspan="2">this thesis</td></tr>
<tr><td>commutative</td><td>PSPACE-complete</td></tr>
<tr><td>semi-groups</td><td>commutative</td><td>EXPSPACE-complete</td><td>[MM82]</td></tr>
</table>

Table 6.2: The complexities of some (uniform) word problems

7 Generalized conflict-free Petri nets

In this chapter, we investigate generalized conflict-free Petri nets (gcnf-PNs). The relationship of the classes defined in the following is illustrated in Figure 7.1. A Petri net $\mathcal{P} = (P, T, F, \mu_0)$ is a

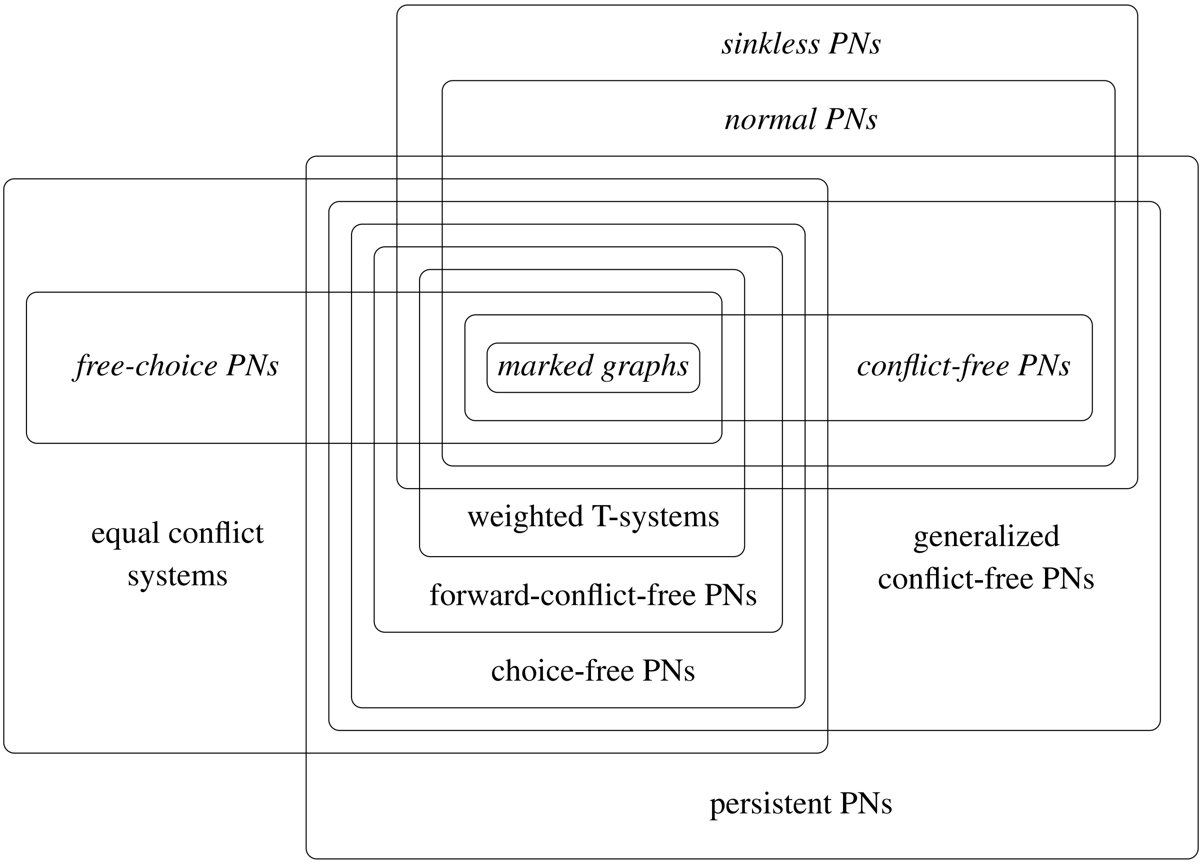


Figure 7.1: This class diagram illustrates the relationships between classes of Petri nets that are closely related to generalized conflict-free Petri nets. Classes written in italics consist of (forward-) ordinary Petri nets.

generalized conflict-free Petri net (gcnf-PN) if $\Delta(t)(p) \geq 0$ for all places $p \in P$ with $|p^\bullet| > 1$ and transitions $t \in p^\bullet$. In other words, a transition cannot decrease the number at a place p if p has more than one outgoing edge. The class of gcnf-PNs contains the very closely related class of choice-free Petri nets (not to be confused with free-choice Petri nets) which are those Petri nets without decisions. A subclass of choice-free Petri nets is the class of forward-conflict-free Petri nets which are those gcnf-PNs, where each place has *exactly* one outgoing edge. At the end of this chapter, we will argue that the reachability problem for gcnf-PNs can be reduced in polynomial time to the reachability problem of backward-ordinary forward-conflict-free Petri nets with edge multiplicities of [2]. Thus, results of this restricted class can also be of relevance for gcnf-PNs.

The class of gcnf-PNs is the natural generalization of the class of conflict-free Petri nets (cnf-PNs) which are exactly the ordinary gcnf-PNs. We remark that the results for cnf-PNs presented in the following still hold if we allow arbitrary edge multiplicities for edges from transitions to places. This is either clear from the proofs of these results, or is immediately implied by the fact that most proofs were presented for conflict-free VRSs, whose Petri net counterparts are actually *forward-ordinary* gcnf-PNs.

The theory on conflict-free Petri nets is rich. They were first defined by Crespi-Reghizzi and Mandrioli [CRM75][1] (in the formalism of VASs) who showed that the reachability problem of cnf-PNs is decidable. In the same paper, the authors showed that zero-loops contained in a firing sequence of cnf-PNs can be shifted to the end of the sequence, a result that we will use later in this chapter. A subclass of cnf-PNs, called *marked graphs*[2] (also known as T-systems), was introduced by Genrich [Gen71] and Holt and Commoner [HC70] (see also [Com+71]), and has been the focus of much research. Marked graphs are cnf-PNs, with $|{}^\bullet p| = |p^\bullet| = 1$ for all of their places p (sometimes they are defined by the less restrictive constraints $|{}^\bullet p|, |p^\bullet| \leq 1$). A comprehensive survey about marked graphs can be found in [Mur89], Chapter VII. Jones et al. [Jon+77] showed that the reachability problem of cnf-PNs is **NP**-hard. Later, Howell and Rosier [HR88] (implicitly) showed that the RecLFS problem of cnf-PNs is decidable in polynomial time. By using this in combination with the observation that reachable markings are reachable by Parikh vectors of polynomial encoding size, the authors obtained **NP**-completeness for the reachability problem. Howell et al. [How+89] used similar techniques to prove **NP**-membership for the promise problem variation of the reachability problem for normal and sinkless Petri nets, which are both superclasses of cnf-PNs, first defined by Yamasaki [Yam84]. A general approach for classes of Petri nets with simple circuits, including conflict-free Petri nets, was given by Yen and Yu [YY03], yielding an alternative proof for **NP**-completeness of the respective reachability (promise) problems.

Landweber and Robertson [LR78] showed that the boundedness problem of cnf-PNs is decidable in exponential time. This bound was improved by Howell et al. [How+87] to quadratic time[3], and finally by Alimonti et al. [Ali+92] to linear time. Similarly, Howell and Rosier [HR89] showed that the liveness problem of cnf-PNs is decidable in quadratic time, and obtained **P**-completeness for this problem. Alimonti et al. [Ali+92] improved this bound to linear time. For the equivalence problem of cnf-PNs, Howell and Rosier [HR88] showed $\mathbf{\Pi}_2^p$-completeness. Some more results on cnf-PNs can be found in [Bes+07; Yen98; Yen99; Yen02; Yen+93]

Even though the property of being forward-ordinary is (for obvious reasons) almost never explicitly mentioned in the proofs of these results, it is apparent that the arguments break down very quickly when this property is not implicitly assumed. Moreover, it's not obvious how to adapt these arguments for gcnf-PNs. Some results for choice-free or forward-conflict-free Petri nets were presented by Amer-Yahia et al. [AY+99], Amer-Yahia and Zerhouni [AYZ99], and Teruel et al. [Ter+97]. The class of weighted T-systems, a subclass of gcnf-PNs consisting of Petri nets with the topology of marked graphs and arbitrary edge multiplicities, was investigated by Teruel et al. [Ter+92]. Teruel and Silva [TS93; TS94; TS96] also obtained results for equal conflict systems, a natural generalization of free-choice Petri nets[4]. Another very well-known superclass of gcnf-PNs is the class of persistent Petri nets. A Petri net is persistent if, for all reachable markings μ and two different transitions t, t', $\mu \xrightarrow{t}$ and $\mu \xrightarrow{t'}$ implies $\mu \xrightarrow{t \cdot t'}$. In other words, the firing of a transition does not disable other enabled transitions. It's not hard to see that gcnf-PNs (cnf-PNs, resp.) are exactly those Petri net that are persistent Petri nets (ordinary persistent Petri nets, resp.) for every initial marking. We remark that in early literature, every Petri net was assumed to be ordinary. Therefore, persistent Petri nets were introduced as a class of ordinary Petri nets. However, most results for ordinary persistent Petri nets

[1] They originally used the more restrictive constraint that each place has *exactly* one outgoing edge.

[2] Curiously enough, marked graphs were also called conflict-free Petri nets at the beginning.

[3] It was shown that the boundedness problem of conflict-free VRS V is decidable in time $\mathcal{O}(\text{size}(V)^{1.5})$. The larger exponent for cnf-PNs is due to the different encoding schemes for the problem instances [Ali+92].

[4] We refer to the introductions of Chapters 3 and 6 for more information about equal conflict systems and free-choice Petri nets, and their relationship to our classes of interest.

also hold in the non-ordinary case since arguments involving persistent Petri nets usually don't use any (implicit or explicit) assumptions about edge multiplicities. Furthermore, a little bit later, results were usually formulated for persistent VASs or VRSs who correspond to persistent Petri nets with arbitrary edge multiplicities. Hence, we assume that persistent Petri nets are defined as a class of Petri nets with arbitrary edge multiplicities. Some of the most important results on persistent Petri nets can be found in [Gra80; LR78; May81b; Mül80]. Unfortunately, from an analytical point of view, persistent Petri nets provide almost as hard obstacles than Petri nets in general. Via SLSRs of the reachability sets of persistent Petri nets, which can be constructed as shown by [Gra80; May81b; Mül80], most classical problems are decidable. However, no primitive recursive upper bound for the size of these SLSRs is known. Consequently, the known bounds for the reachability and other problems don't substantially improve on the bounds for the respective problems of Petri nets in general.

Similarly, the complexities of many classical problems of gcnf-PNs, in particular those problems investigated in this work, are still unknown. In particular, the best bounds we know for the reachability problem of gcnf-PNs are that the problem is **NP**-hard (due to the **NP**-hardness of this problem for cnf-PNs) and decidable (due to the fact that the reachability problem of persistent or, alternatively, general Petri nets is decidable).

In the following, we try to narrow this gap and provide a starting point for future research. We show that the RecLFS problem is in **coNP**. Using this result, it's not hard to show that the reachability problem is in Σ_2^p.

Lemma 7.1 ([LR78], Lemma 3.1)**.** *Let σ and ρ be firing sequences in a persistent Petri net. Then there is a firing sequence β such that $\Psi(\beta) = \max(\Psi(\sigma), \Psi(\rho))$. Moreover, β may be constructed so that $\beta = \sigma \cdot (\rho \dot{-} \sigma)$.*

Lemma 7.2. *Let $\mathcal{P} = (P, T, F, \mu_0)$ be a generalized conflict-free Petri net, Φ be an enabled Parikh vector, and $\Phi' \leq \Phi$ be a nonpositive loop. Then it holds $\mu_0 \xrightarrow{\Phi - \Phi'} \mu_1 \xrightarrow{\Phi'} \mu_2$ for some markings μ_1, μ_2.*

Proof. This follows from the proof of a similar lemma of Crespi-Reghizzi and Mandrioli [CRM75] for conflict-free Petri nets and zero-loops Φ'. Note that the proof doesn't make use of the property that cnf-PNs are forward-ordinary. The proof also doesn't fully use that Φ' is a zero-loop, but only that Φ' is a nonpositive loop. We remark that the paper contains a small mistake. This was fixed by Theil [The06] in her diploma thesis. In her proof, also gcnf-PNs were considered. □

Lemma 7.3. *Let Φ be a Parikh vector of a persistent Petri net. Then, Φ is* not *enabled if and only if there is an enabled Parikh vector $\Phi' < \Phi$ leading to a marking at which no transition $t \in \Phi - \Phi'$ is enabled.*

Proof. "⇒": If each enabled Parikh vector $\Phi' < \Phi$ leads to a marking at which some transition $t \in \Phi - \Phi'$ is enabled, then we iteratively construct a sequence $\vec{0} = \Phi_0, \Phi_1, \ldots, \Phi_{|\Phi|_1} = \Phi$ of enabled Parikh vectors, where $t \in \Phi_i - \Phi_{i-1}$ is some transition enabled at the marking reached by Φ_i.

"⇐": Assume for the sake of contradiction that Φ is enabled. Then, there are firing sequences σ, ρ with $\Psi(\sigma) = \Phi'$ and $\Psi(\rho) = \Phi$. Hence, by Lemma 7.1, the sequence $\rho \dot{-} \sigma$ (respective its Parikh image $\Phi - \Phi' > \vec{0}$) is enabled at the marking reached by σ (respective Φ'), a contradiction. □

In contrast to persistent Petri nets, the condition that Φ' must be enabled to serve as a "stopper" isn't necessary in case of gcnf-PNs.

Lemma 7.4. *Let Φ be a Parikh vector of a gcnf-PN $\mathcal{P} = (N, \mu_0)$. Then, Φ is* not *enabled if and only if there is a Parikh vector $\Phi' < \Phi$ (regardless whether enabled or not) such that no transition $t \in \Phi - \Phi'$ is enabled at $\mu_0 + \Delta(\Phi')$.*

Proof. "$\Rightarrow$": This follows directly from Lemma 7.3.

"$\Leftarrow$": Let $\mu' := \mu_0 + \Delta(\Phi')$. If Φ' is enabled, then Lemma 7.3 implies that Φ is not enabled. Thus, assume that Φ' is not enabled. Then, again by Lemma 7.3, there is an enabled Parikh vector $\Phi'' < \Phi'$ such that no transition of $\Phi' - \Phi''$ is enabled at the marking μ'' reached by Φ''. Consider a transition $t \in \Phi - \Phi''$.

Case 1: $\Phi''(t) = \Phi'(t) < \Phi(t)$. Since $\Phi'' < \Phi'$ and no transition other than t can decrease the number of tokens at a place $p \in {}^\bullet t$, we find $\Delta(\Phi'')(p) \leq \Delta(\Phi')(p)$, and therefore $\mu'(p) \geq \mu''(p)$. Taking into consideration that t is not enabled at μ', t is also not enabled at μ''.

Case 2: $\Phi''(t) < \Phi'(t) \leq \Phi(t)$. By assumption, t is not enabled at μ''.

In total, no transition $t \in \Phi - \Phi''$ is enabled at the marking μ'' reached by $\Phi'' < \Phi$. Therefore, by Lemma 7.3, Φ is not enabled. □

Using this characterization of enabled Parikh vectors, we obtain the following bound for the RecLFS problem.

Theorem 7.5. *The RecLFS problem of generalized conflict-free Petri nets is in* **coNP**.

Proof. We show that the complement of the RecLFS problem is in **NP**. Assume, a Parikh vector Φ is not enabled. Then we can guess the Parikh vector $\Phi' < \Phi$ as defined at Lemma 7.4 and check if no transition $t \in \Phi - \Phi'$ is enabled at $\mu_0 + \Delta(\Phi')$, both in polynomial time. □

To use this theorem for the reachability problem, we show the following lemma which provides a bound on the size of minimal Parikh vectors leading to reachable markings.

Lemma 7.6. *Each reachable marking μ of any gcnf-PN $\mathcal{P} = (P, T, F, \mu_0)$ is reachable by a Parikh vector with component sum at most $(2 + mW + \max(\mu_0) + \max(\mu))^{2n+m}$.*

Proof. W. l. o. g., we assume n, m, $W > 0$ and $\mu_0 \neq \mu$. Let μ be reachable by some Parikh vector Φ'. Let $\Phi \leq \Phi'$ be the Parikh vector of Lemma 2.17. Since $\Phi' - \Phi$ is a zero-loop (and therefore a nonpositive loop), Lemma 7.2 implies that Φ is enabled. □

By combining the result on RecLFS for gcnf-PNs and the bound provided in the last lemma, we obtain the following result for the reachability problem of gcnf-PNs.

Theorem 7.7. *The reachability problem of gcnf-PNs is* **NP***-hard and in* $\mathbf{\Sigma}_2^p$.

Proof. Let $\mathcal{P} = (P, T, F, \mu_0)$ be a generalized conflict-free Petri net and μ a marking of $\mathcal{P}$. Assume that μ is reachable in $\mathcal{P}$. By Lemma 7.6, there is a Parikh vector Φ with polynomial encoding size, leading from μ_0 to μ. A polynomial time NDTM M can guess Φ, check if $\mu_0 + \Delta(\Phi) = \mu$, and use an oracle to determine if Φ is enabled. If this is the case, then M accepts, and rejects otherwise. Since RecLFS is in **coNP**, the reachability problem of generalized conflict-free Petri nets is in Σ_2^p. The **NP**-hardness follows from the fact that the reachability problem of conflict-free vector addition systems is **NP**-complete [HR88]. □

We remark that, by Theorem 3.11, the reachability problem is **NP**-hard, even if restricted to Petri nets that are ordinary, cycle-free, conflict-free, and communication-free at the same time, and restricted to the end marking $\vec{1}$.

As promised at the beginning of the chapter, we conclude the chapter by showing that it's sufficient to investigate a restricted subclass of gcnf-PNs to obtain insight into gcnf-PNs in general. Possible implications of this observation are discussed in Chapter 9.

Lemma 7.8. *The reachability problem of gcnf-PNs can be reduced in polynomial time to the reachability problem of backward-ordinary forward-conflict-free Petri nets with edge multiplicities of* $[2]$.

Proof. Consider the reduction rules illustrated in Figure 7.2. It's not hard to see that a reduction as

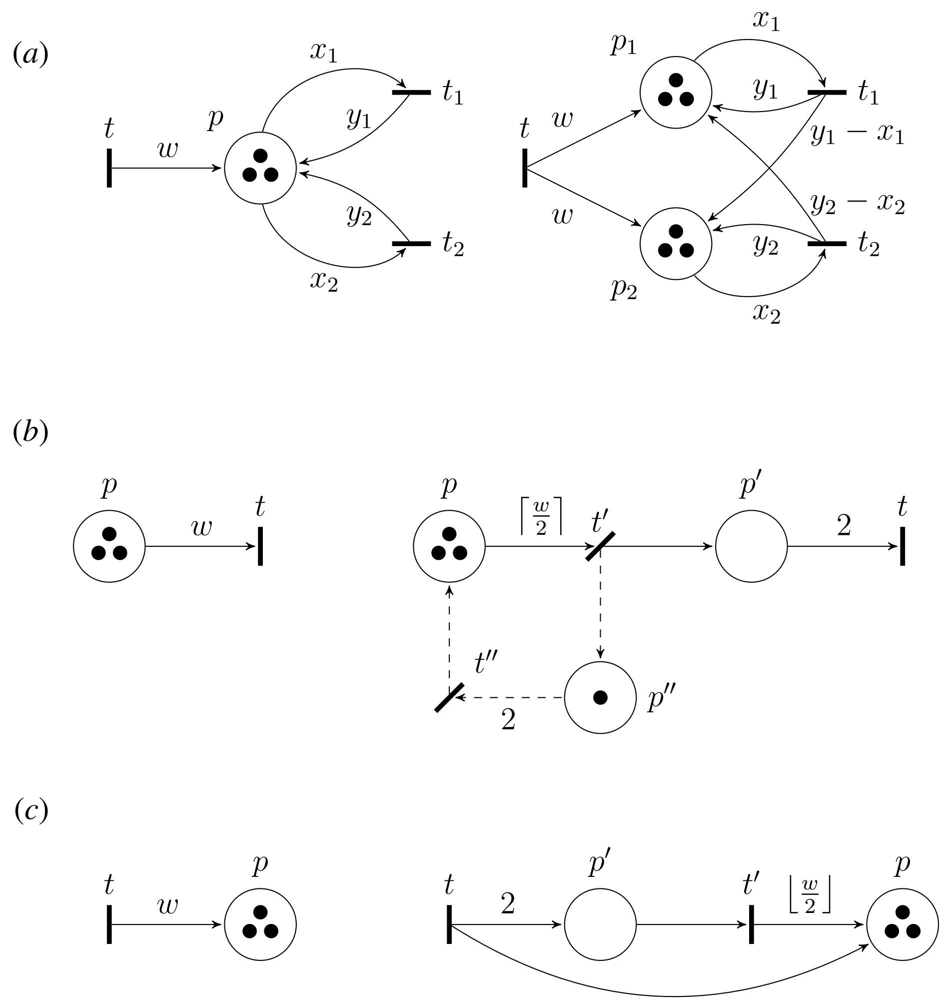


Figure 7.2: (a)–(c) illustrate reduction rules for transforming a generalized communication-free Petri net into a choice-free Petri net with edge multiplicities out of $[2]$. Dashed edges only exist if w is odd.

proposed in the lemma can be performed by using these rules and introducing artificial input places for transitions without incoming edges. For the sake of completeness, we formally prove it in the following. If we have a place p with $k > 1$ outgoing edges, leading to transitions $t_1, \ldots, t_k$, then we apply rule (*a*) which splits p into k copies $p_1, \ldots, p_k$ of it, such that, for each $i \in [k]$, place p_i has an outgoing edge to t_i. Each p_i behaves exactly the same as place p in the original net. In particular, a transition sequence is a firing sequence of the old net if and only if it's a firing sequence of the new net. Hence, we can reduce the reachability problem for gcnf-PNs to the reachability problem of choice-free Petri nets by applying this rule at most nm times.

Next, consider a choice-free Petri net $\mathcal{P} = (P, T, F, \mu_0)$ with a place p, a transition t, and an edge with multiplicity $w > 2$ from p to t. We assume that w is odd since, in case of an even w, we can apply an analogue but simpler argument. Both p and t can have more incident edges but they are not relevant for the following. Consider the reduction rule illustrated in (*b*). This rule replaces edge (p, t) by a small subnet consisting of places p', p'' and transitions t', t'', where p'' is initially marked by one token. Let $\mathcal{P}'$ be the resulting choice-free Petri net. Let μ be the end marking of interest for $\mathcal{P}$ and $\mu' :\triangleq \mu + p''$ be the corresponding end marking for $\mathcal{P}'$. If μ is reachable in $\mathcal{P}$, then μ' is reachable in $\mathcal{P}'$ since each application of t in $\mathcal{P}$ can be simulated by the transition sequence $t' \cdot t'' \cdot t' \cdot t$ in $\mathcal{P}'$. Now, assume μ' is reachable in $\mathcal{P}'$ by a firing sequence σ. If σ doesn't contain t', then it induces a firing sequence of $\mathcal{P}$ leading to μ. Hence, assume that t' is contained in σ. Then, we observe $0 = \Delta(\sigma)(p') = \Delta(\sigma)(p'')$ which is equivalent to $\Psi(\sigma)(t) = \Psi(\sigma)(t'') = 2 \cdot \Psi(\sigma)(t')$. We now permute σ in the following way. We shift the first occurrence of t, which must be preceded by at least two occurrences of t', to the position right after the second occurrence of t'. Then, we shift the first occurrence of t'' to the position right in front of the second occurrence of t'. Last, we shift the first occurrence of t' to the position right in front of the first occurrence of t''. Note that each step of this procedure results in a firing sequence. Let φ be the firing sequence resulting from the last step. The sequence φ contains a subsequence $t' \cdot t'' \cdot t' \cdot t$ which is not preceded by any of the transitions t, t', or t''. In particular, the marking reached by the prefix ending with this subsequence corresponds (in the sense above) to a marking of $\mathcal{P}$. By iterating this argument for the corresponding suffix, we find that μ is reachable in $\mathcal{P}$. We can use this rule to reduce the reachability problem of choice-free Petri nets to the reachability problem of choice-free Petri nets whose edges from places to transitions have multiplicity 1 or 2. The number of applications of the rule is bounded by $m \cdot \mathrm{ld}(2W)$ which is polynomial in the size of $\mathcal{P}$.

Next, consider a choice-free Petri net $\mathcal{P} = (P, T, F, \mu_0)$ with a transition t, a place p, and an edge with multiplicity $w > 2$ from t to p. Consider the reduction rule illustrated in (*c*). This rule replaces edge (t, p) by a small subnet consisting of a place p', and transitions t'. Let $\mathcal{P}'$ be the resulting choice-free Petri net. Let μ be the end marking of interest for $\mathcal{P}$ and $\mu' :\triangleq \mu$ be the corresponding end marking for $\mathcal{P}'$. If μ is reachable in $\mathcal{P}$, then μ' is reachable in $\mathcal{P}'$ since each application of t in $\mathcal{P}$ can be simulated by the transition sequence $t \cdot t'$ in $\mathcal{P}'$. Now, assume μ' is reachable in $\mathcal{P}'$ by a firing sequence σ. If σ doesn't contain t, then it is firing sequence of $\mathcal{P}$ leading to μ. If it does contain t, then the sequence after the first occurrence of t must contain two occurrences of t'. Both can be shifted to the position behind t's first occurrence. The resulting sequence is a firing sequence, whose shortest prefix containing $t \cdot t' \cdot t'$ leads to a marking that corresponds to a marking of $\mathcal{P}$. Iterating this argument shows that μ is reachable in $\mathcal{P}$. We can use this rule to reduce the reachability problem of choice-free Petri nets whose edges from places to transitions have multiplicity 1 or 2 to the reachability problem of choice-free Petri nets for which all edges have multiplicity 1 or 2. The number of applications of the rule is bounded by $m \cdot \mathrm{ld}(2W)$ which is polynomial in the size of $\mathcal{P}$.

The nets resulting from the last reduction can be transformed into forward-conflict-free Petri nets by introducing, for each transition t without incoming edge, a marked place p and a cycle (p, t, p) whose edges have multiplicity 1. $\square$

8 Ring Petri nets

In this chapter, we investigate ring Petri nets. A Petri net is a ring Petri net (ring-PN) if it consists of exactly one circuit. Ring-PNs are not only gcf-PNs but also gcnf-PNs. A ring-PN whose edge multiplicities are powers of the same number $k \in \mathbb{N}$ is called k-multiplicity-regular.

Our two main motivations to investigate ring-PNs are as follows. Insight into this class could help to close the gap between the respective lower and upper bounds of the RecLFS and reachability problems of gcnf-PNs. Due to Lemma 7.8, we only need to consider gcnf-PNs with edge multiplicities 1 or 2 for the reachability problem. Furthermore, **PSPACE**-completeness of various problems has been obtained for almost ordinary gss-PNs in Chapter 6. However, we have been unable to adapt the proofs for **PSPACE**-hardness for gss-PNs with edge multiplicities 1 or 2. Results for Ring-PNs could help to narrow the gap between the lower bound (**NP**-hard, see Theorem 3.26) and the upper bound (**PSPACE**, see Theorem 6.24) for these restricted gss-PNs.

The main result of this chapter is that, in k-multiplicity-regular ring-PNs, many problems are decidable in polynomial time, while they are in **NP** or **coNP** for ring-PNs in general. Note that ring-PNs with edge multiplicities 1 or 2 are 2-multiplicity-regular, and all circuits within the Petri nets of the classes mentioned above are such ring-PNs. Whether this result can be applied for the purpose mentioned earlier is subject to further research.

Ring-PNs were previously investigated by Lien [Lie76b] and Teruel et al. [Ter+92]. Chrzastowski-Wachtel and Raczunas [CWR93] considered liveness properties of conservative ring-PNs.

To avoid unnecessarily confusing modulo expressions, we assume that throughout this subsection the index $n + 1$ denotes the index 1. Furthermore, we assume w. l. o. g. that a ring-PN contains, for all $i \in [n]$, the edges (p_i, t_i) and (t_i, p_{i+1}) with multiplicities w_i^- and w_i^+, respectively. Figure 8.1 illustrates an example.

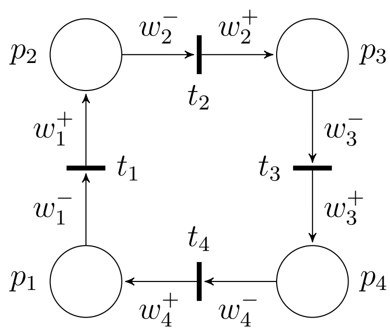


Figure 8.1: A ring-PN with four places and transitions

We first collect a number of observations.

Corollary 8.1. *Let $\mathcal{P} = (P, T, F, \mu_0)$ be a ring-PN, Φ be an enabled Parikh vector, and $\Phi' \leq \Phi$ be a loop. Then it holds $\mu_0 \xrightarrow{\Phi'} \mu_1 \xrightarrow{\Phi-\Phi'} \mu_2$ for some markings μ_1, μ_2.*

Proof. We simply apply Lemma 7.2 to Φ, Φ', and the inverse Petri net of $\mathcal{P}$, using $\mu_0 + \Delta_{\mathcal{P}}(\Phi)$ as its initial marking. □

Through the remainder of this chapter, let D denote the displacement matrix of the ring-PN under consideration, and let $L_{>0} := \{x \in \mathbb{Q}^m_{\geq 0} \mid Dx > \vec{0}\}$, $L_{=0} := \{x \in \mathbb{Q}^m_{>0} \mid Dx = \vec{0}\}$, and $L_{<0} := \{x \in \mathbb{Q}^m_{\geq 0} \mid Dx < \vec{0}\}$. Note that $\vec{0}$ is not contained in any of these sets.

Lemma 8.2. *If $L_{>0}$ ($L_{<0}$, resp.) is nonempty, then, for each $i \in [n]$, there is a vector $x \in L_{>0}$ ($x \in L_{<0}$, resp.) such that $Dx = e_i$ ($Dx = -e_i$, resp.).*

Proof. Let $x_0 \in L_{>0}$. We construct the vectors $x_k \in L_{>0}$, $k \in [m-1]$, recursively by $x_k := x_{k-1} + \frac{(Dx_{k-1})_{[i+k]}}{w^-_{i+k}} \cdot e_{i+k}$. Then, there is a $c \in \mathbb{Q}_{>0}$ such that $cx_{m-1} \in L_{>0}$ and $Dcx_{m-1} = e_i$. Analogously, we can prove the lemma for $L_{<0}$. □

Lemma 8.3. *Exactly one of the sets $L_{>0}$, $L_{=0}$, $L_{<0}$ is nonempty. Furthermore, if $L_{=0}$ is nonempty, then $L_{=0} = \{cx \mid c \in \mathbb{Q}_{>0}\}$ for each $x \in L_{=0}$.*

Proof. If there are $x \in L_{>0}$, and $y \in L_{=0}$, then there is a vector $z = x - cy \in L > 0$ for some $c \in \mathbb{Q}_{>0}$ such that $z_{[i]} = 0$ and $z_{[i+1]} > 0$ for some index i. However, then $(Dz)_{[i+1]} < 0$, a contradiction. Similarly, $L_{<0}$ and $y \in L_{=0}$ cannot simultaneously be nonempty. Now, assume for the sake of contradiction that $|L_{>0}| > 0$ and $|L_{<0}| > 0$. By Lemma 8.2, there are $x \in L_{>0}$ and $y \in L_{<0}$ such that $Dx = e_1$ and $Dy = -e_1$. Therefore, $D(x + y) = \vec{0}$ which is impossible as shown before. Furthermore, it follows immediately from Theorems 3.1 and 3.2 of [Ter+92] that at least one of the sets $L_{>0}$, $L_{=0}$, $L_{<0}$ is nonempty. (This can also easily be shown in a similar fashion as the statement of Lemma 8.2.) □

Lemma 8.4. *Let $\mathcal{P} = (P, T, F, \mu_0)$ be a ring-PN, and μ be a marking. Then, we can determine in polynomial time if there is a Parikh vector Φ with $\mu_0 + \Delta(\Phi) = \mu$. Furthermore, if such a Parikh vector exists, then we can compute in polynomial time a Parikh vector Φ such that μ is reachable if and only if Φ is enabled and leads to μ.*

Proof. Consider the system $Dx = \mu - \mu_0$ with the set $L := \{x \in \mathbb{Q}^m_{\geq 0} \mid Dx = \mu - \mu_0\}$ of nonnegative rational valued solutions. Using linear programming, we can find in polynomial time a solution of L, provided L is nonempty. If L is empty, then μ is not reachable.

Assume that there is exactly one such solution $\Phi \in L$. If $\Phi \notin \mathbb{N}^m_0$, then there is no Parikh vector satisfying $\mu_0 + \Delta(\Phi) = \mu$, and we are finished. If $\Phi \in \mathbb{N}^m_0$, then it is a Parikh vector with the properties of the lemma.

Assume now that there are infinitely many solutions. Let $x, y \in L$ be different solutions. Then, for $z := x - y \neq \vec{0}$, we have $Dz = 0$. Furthermore, either $z \in \mathbb{Q}^m_{>0}$ or $z \in \mathbb{Q}^m_{<0}$ holds since otherwise there would be an index i such that $(Dz)_{[i]} > 0$ or $(Dz)_{[i]} < 0$. Hence, $L_{=0}$ is nonempty, and, by Lemma 8.3, $L_{=0} = \{c\pi \mid c \in \mathbb{Q}_{>0}\}$ for some vector $\pi \in L_{=0}$. Therefore, there is a unique vector $\zeta \in L$ such that $\zeta \leq x$ for all $x \in L$, and $L = \{\zeta + c\pi \mid c \in \mathbb{Q}_{>0}\}$.

The vectors ζ and π can be determined in polynomial time with linear programming (we choose π as the vector minimizing the 1-norm over all vectors of the set $\{x \in \mathbb{Q}^m_{\geq 0} \mid Dx = \vec{0}, \|x\|_1 \geq 1\}$). Let $\Phi \in L$ with be the unique Parikh vector (i. e., $\Phi \in \mathbb{N}^m$) such that $\Phi' > \Phi$ for all other Parikh vectors $\Phi' \in L$. Using ζ and π, Φ can be found in polynomial time.

We now show that Φ satisfies the properties of the lemma. If μ is not reachable, then Φ certainly is not enabled. Assume that μ is reachable by a Parikh vector $\Phi' \in L$. Then, $\Phi' - \Phi$ is a zero-loop (which is a multiple of π). By Lemma 7.2, Φ is also enabled (and leads to μ). □

These observations enable us to show the following bound for the reachability problem of ring-PNs.

Theorem 8.5. *The reachability problem of ring-PNs is in* **coNP**.

Proof. We deterministically compute in polynomial time the Parikh vector Φ of Lemma 8.4, and then ask an oracle if Φ is enabled. The end marking of interest is reachable if and only if Φ exists and the answer of the oracle is positive. Since, by Theorem 7.5, RecLFS of conflict-free Petri nets is in **coNP**, the theorem follows. □

We remark that, as was shown in the proof of the last theorem, the complexity of the reachability problem of ring-PNs is the same as the complexity of the RecLFS problem of ring-PNs.

For k-multiplicity-regular ring-PNs, however, we can solve the RecLFS problem in polynomial time. The basis for this observation is provided in the next two lemmata.

Lemma 8.6. *Let $\mathcal{P}$ be a k-multiplicity-regular ring-PN with a loop Φ'. Then, there are a transition t and a loop Φ with $\Phi(t) = 1$ which can be computed in polynomial time.*

Proof. Let, w. l. o. g., $t_1 = \arg\min_{t' \in T}\{\Phi'(t')\}$, $\Phi'' := \frac{1}{\Phi'(t_1)} \cdot \Phi'$ (i. e., we scale Φ'), and $\Phi_1 := e_1$. Further, let Φ_ℓ for $\ell \in [2, m]$ be recursively defined by $\Phi_\ell := \Phi_{\ell-1} + \frac{\Delta(\Phi_{\ell-1})(p_\ell)}{w_\ell^-} \cdot e_\ell$. By construction, $\Delta(\Phi) = ce_1$, where, by Lemma 8.3, $c \geq 0$.

We first show by induction that $\Phi(t) \geq \Phi''(t)$ for all $t \in T$. For t_1, we observe $\Phi(t_1) = \Phi''(t_1)$. Let $\ell \in [2, m]$, and let $\Phi(t_{\ell-1}) \geq (\Phi''(t_{\ell-1})$. Then, $\Phi(t_{\ell-1}) \cdot \Delta(t_{\ell-1})(t_{\ell-1}{}^\bullet) \geq \Phi''(t_{\ell-1}) \cdot \Delta(t_{\ell-1})(t_{\ell-1}{}^\bullet)$, and therefore $\Phi(t_\ell) \geq \Phi''(t_\ell)$. This implies $\Phi \geq \vec{1}$.

We now show that, for each t, $\Phi(t)$ is a power of k. For t_1, we have $\Phi(t_1) = 1 = k^0$ by definition. Now, assume that, for some $\ell \in [2, m]$, $\Phi(t_{\ell-1}) = k^a$ for some $a \in \mathbb{Z}$. Then, $\Phi(t_\ell) = \frac{\Delta(\Phi)(t_{\ell-1}{}^\bullet)}{w_\ell^-} = \frac{k^a \cdot w_{i+\ell-1}^+}{w_{i+\ell}^-} = \frac{k^a \cdot k^b}{k^c}$ for some $b, c \in \mathbb{N}_0$.

Since each component of Φ is a power of k and is at least 1, we find $\Phi \in \mathbb{N}$. Therefore, Φ is a loop. The transition t and the loop Φ can be found by considering all t and computing the corresponding Parikh vector Φ as shown above. Since $\max(\Phi)$ is exponentially bounded, Φ can be computed in polynomial time. □

Lemma 8.7 ([HI88; Mur89]). *Let Φ be a Parikh vector of a cycle-free Petri net $\mathcal{P} = (P, T, F, \mu_0)$. Then, Φ is enabled if and only if $\mu_0 + \Delta(\Phi) \geq 0$.*

A consequence of this lemma is that RecLFS for cycle-free Petri nets is decidable in polynomial time. We can now show a polynomial time upper bound for the RecLFS problem of k-multiplicity-regular ring-PNs.

Theorem 8.8. *The RecLFS problem of k-multiplicity-regular ring-PNs is decidable in polynomial time.*

Proof. Let $\mathcal{P} = (P, T, F, \mu_0)$ be a k-multiplicity-regular ring-PN, and Φ be a Parikh vector. If $\Phi(t) = 0$ for some transition t, then we find, by applying Lemma 8.7 to Φ and the net induced by Φ, that we can decide in polynomial time (in the encoding size of the net and the vector Φ) if Φ is enabled. Hence, assume $\Phi(t) > 0$ for all $t \in T$, and let t and ϑ be the transition and the loop of Lemma 8.6, which can be found in polynomial time. Assume w. l. o. g. $t = t_1$. We compute in polynomial time the unique maximal (w. r. t. $\geq$) enabled Parikh vector Φ' with $\Phi' \leq \Phi$ and $t_1 \notin \Phi'$. Let μ_1 be the marking reached by Φ', and let $\Phi_1 := \Phi - \Phi'$. By Lemma 7.1, $\mu_0 \xrightarrow{\Phi}$ if and only if $\mu_1 \xrightarrow{\Phi_1}$.

If t is not enabled at μ_1, then Φ is not enabled at μ. If, on the other hand, t is enabled, then ϑ is also enabled (which can easily be seen, for instance by applying Lemma 8.7 to $\mathcal{P}[\vartheta - e_1]$ and $\vartheta - e_1$). Consider the largest $c \in \mathbb{N}_0$ such that $c \cdot \vartheta \leq \Phi_1$, let μ_2 be the marking reached by firing $c \cdot \vartheta$ at μ_1, and let $\Phi_2 := \Phi_1 - c \cdot \vartheta$. By Lemma 7.1, $\mu_0 \xrightarrow{\Phi}$ if and only if $\mu_2 \xrightarrow{\Phi_2}$. Let $i \in [m]$ be the smallest index such that $\vartheta(t_i) \geq \Phi_2(t_i)$. Since t and therefore ϑ is enabled at μ_2, the Parikh vector $\Phi'' := \left(\sum_{j=1}^{i-1} \vartheta(t_j) \cdot e_j\right) + \max\{\vartheta(t_i), \Phi_2(t_i)\} \cdot e_i$ is enabled at μ_2. Let μ_3 be the marking reached by firing Φ'' at μ_2, and let $\Phi_3 := \Phi_2 - \Phi''$. By Lemma 7.1, $\mu \xrightarrow{\Phi}$ holds if and only if $\mu_3 \xrightarrow{\Phi_3}$. Note that we can determine Φ_3 in polynomial time. Now, observe that $\mathcal{P}[\Phi_3]$ is cycle-free. Therefore, by Lemma 8.7, we can determine in polynomial time if Φ_3 is enabled at μ_3. ☐

As already mentioned, for ring-PNs, the reachability problem has the same complexity as the RecLFS problem. The same holds for k-multiplicity-regular ring-PNs as is shown in the last theorem of this thesis.

Theorem 8.9. *The reachability problem of k-multiplicity-regular ring-PNs is decidable in polynomial time.*

Proof. Let $\mathcal{P} = (P, T, F, \mu_0)$ be a k-multiplicity-regular ring-PN, and μ be the end marking of interest. If $\mathcal{P}$ has no loop, then we consider the inverse net of $\mathcal{P}$ with initial marking μ and end marking μ_0 which, by Lemma 8.3, must have a loop. Hence, assume w. l. o. g. that $\mathcal{P}$ has a loop.

We first determine if there is a Parikh vector Φ with $\mu_0 + \Delta(\Phi)$. If this is not the case, then μ is not reachable. Otherwise, let Φ denote the Parikh vector of Lemma 8.4. By this Lemma, these steps can be performed in polynomial time. By Theorem 8.8, we can determine in polynomial time if Φ is enabled, where, by Lemma 8.4, Φ is enabled if and only if μ is reachable. ☐

9 Conclusion and outlook

Summary. In this thesis, we investigated several computational problems of different classes of Petri nets. We showed that many problems of cf-PNs can be solved in polynomial time, whose counterparts of gcf-PNs are **PSPACE**-complete. We presented a framework that, under the right circumstances, can be used to obtain upper bounds for many classical computational problems of restricted classes of Petri nets with arbitrary edge multiplicities. The core of the framework and its application is built on canonical permutations and firing sequences. Using permutation techniques, we showed that conservative Petri nets, gcf-PNs, igcf-PNs, and gss-PNs satisfy the requirements of the framework. For most of the problems under consideration the upper bounds found by applying the framework match the lower bounds, which were obtained by simulating **PSPACE**-Turing machines. Consequently, we obtained **PSPACE**-completeness for these problems. Furthermore, we discovered results for several problems involving home spaces, and for commutative and non-commutative grammars that are related to (generalized) communication-free Petri nets. In addition to the classes mentioned earlier, we investigated gcnf-PNs and ring-PNs.

Open problems. Some questions which are of relevance in the context of this thesis are still left open. A few have already been mentioned in the previous chapters. Here, we briefly discuss the more important or interesting ones.

Even though communication-free Petri nets are very well understood, no completeness-result has been found for the equivalence problem. Results of Yen [Yen13] and for CFCGs show that this problem is Π_2^p-hard and in **coNEXPTIME**. The equivalence problem of CFCGs exhibits the same gap between lower and upper bound. It's possible that we need stronger canonical firing sequences or stronger results about the semilinear set representation for the reachability set before we can close the gap.

On a similar note, the equivalence and containment problems of gcf-PNs and igcf-PNs are problems for which we didn't discover completeness-results. The upper bound of doubly exponential space for these problems in case of gcf-PNs is particularly large. Alternative canonical firing sequences with stronger properties (if they exist) could be useful to decrease the upper bound. For the liveness problem of igcf-PNs and gss-PNs, we showed membership in **PSPACE** but didn't find a matching lower bound. A straightforward adaption of our approach used for gcf-PNs is not possible since igcf-PNs are unable to use a single transition to mark the net with the initial marking again. For for many problems of gcf-PNs and igcf-PNs, we discovered completeness-results. Another open question is whether the upper bounds also hold for the corresponding classes in extended Petri net models. Furthermore, finding bounds for other problems (like those discussed by Hack [Hac79]) in case of gcf-PNs and igcf-PNs or for the home space problems mentioned in Section 6.3 is another open problem.

For the reachability problem of gcnf-PNs, we found the upper bound of Σ_2^p-membership which almost fits the lower bound of **NP**-hardness. The upper bound is larger than the upper bound for the reachability problem of cnf-PNs, which is **NP**-membership. This difference is a result of the difference in the upper bound of the respective RecLFS problems, which are membership in **coNP** and membership in **P**, respectively. While it is unclear whether arbitrary edge multiplicities actually increase the computational complexities of these problems, we have to deal with the fact that the established theory of cnf-PNs doesn't apply to gcnf-PNs anymore, if we want to close the gaps between lower and upper bounds. Our investigation of ring-PNs, a very restricted subclass of gcnf-

PNs, and the results of this investigation seem to imply that this is not a trivial matter. However, our observation that the reachability and the RecLFS problems of 2-multiplicity-regular ring-PNs are decidable in polynomial time could be useful for approaching these problems of gcnf-PNs since we only need to consider gcnf-PNs with edge multiplicities of [2], and all circuits within such a gcnf-PN are 2-multiplicity-regular ring-PNs.

The low complexity for the reachability and RecLFS problems of 2-multiplicity-regular ring-PNs could be useful for another open problem. We were able to obtain **PSPACE**-hardness of a number of problems for gss-PNs with edge multiplicities of [3]. However, the situation is unclear if we only consider gss-PNs with edge multiplicities of [2]. As before, all circuits within such a gss-PN are 2-multiplicity-regular ring-PNs. Assuming that problems of such gss-PNs have a lower complexity than **PSPACE**, an approach that decomposes firing sequences into fragments corresponding to such circuits and then applying bounds for 2-multiplicity-regular ring-PNs could possibly be fruitful.

A more general open problem is that of finding an improved upper bound for the reachability problem of persistent Petri nets. Howell et al. [How+93] showed that reachable markings in single-path Petri nets, which are persistent by nature, are reachable by canonical firing sequences with certain nice properties. Consequently, the reachability problem (and other problems) of this class are decidable in polynomial space. This result and the results of this thesis emphasize the power of permutation techniques and canonical firing sequences. It may be possible to apply similar approaches for persistent Petri nets.

List of Figures

List of Tables

Bibliography

[Ali+92] P. Alimonti, E. Feuerstein, and U. Nanni. Linear time algorithms for liveness and boundedness in conflict-free Petri nets. In: *Proceedings of the 1st Latin American Symposium on Theoretical Informatics (LATIN'92)*. Volume 583. Lecture Notes in Computer Science. Springer, 1992, pages 1–14. DOI: 10.1007/BFb0023812.

[AY+99] C. Amer-Yahia et al. Some subclasses of Petri nets and the analysis of their structural properties: a new approach. In: *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* 29.2 (1999), pages 164–172. DOI: 10.1109/3468.747851.

[AYZ99] C. Amer-Yahia and N. Zerhouni. Structure theory of choice-free Petri nets based on eigenvalues. In: *Journal of the Franklin Institute* 336.5 (1999), pages 833–849. DOI: 10.1016/S0016-0032(99)00008-3.

[AB09] S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.

[Bak73] H. Baker. *Rabin's proof of the undecidability of the reachability set inclusion problem of vector addition systems*. Computation Structures Group Memo 79. Cambridge, Massachusetts: Massachusetts Institute of Technology, Project MAC, 1973.

[Bes+07] E. Best et al. Separability in conflict-free Petri nets. In: *Revised Papers of the 6th International Andrei Ershov Memorial Conference on Perspectives of Systems Informatics (PSI'06)*. Volume 4378. Lecture Notes in Computer Science. Springer, 2007, pages 1–18. DOI: 10.1007/978-3-540-70881-0_1.

[Car+76] E. Cardoza, R. Lipton, and A. R. Meyer. Exponential space complete problems for Petri nets and commutative semigroups (preliminary report). In: *Proceedings of the 8th ACM Symposium on Theory of Computing (STOC'76)*. Hershey, Pennsylvania, USA: ACM, 1976, pages 50–54. DOI: 10.1145/800113.803630.

[Che+09] C.-L. Chen, S. Wang, and H.-C. Yen. Reachability analysis of variants of communication-free Petri nets. In: *IEICE Transactions on Information and Systems* 92.3 (2009), pages 377–388. URL: http://search.ieice.org/bin/summary.php?id=e92-d_3_377.

[Che+95] A. Cheng, J. Esparza, and J. Palsberg. Complexity results for 1-safe nets. In: *Theoretical Computer Science* 147 (1995), pages 117–136. DOI: 10.1016/0304-3975(94)00231-7.

[Chr92] S. Christensen. Distributed bisimularity is decidable for a class of infinite state-space systems. In: *Proceedings of the 3rd International Conference on Concurrency Theory (CONCUR'92)*. Volume 630. Lecture Notes in Computer Science. Springer, 1992, pages 148–161. DOI: 10.1007/BFb0084789.

[Chr+93] S. Christensen, Y. Hirshfeld, and F. Moller. Bisimulation equivalence is decidable for basic parallel processes. In: *Proceedings of the 4th International Conference on Concurrency Theory (CONCUR'93)*. Volume 715. Lecture Notes in Computer Science. Springer, 1993, pages 143–157. DOI: 10.1007/3-540-57208-2_11.

[CWR93] P. Chrzastowski-Wachtel and M. Raczunas. Liveness of weighted circuits and the diophantine problem of Frobenius. In: *Proceedings of the 9th International Conference on Fundamentals of Computation Theory (FCT'93)*. Volume 710. Lecture Notes in Computer Science. Springer, 1993, pages 171–180. DOI: 10.1007/3-540-57163-9_13.

[Coc69] J. Cocke. *Programming languages and their compilers: Preliminary notes*. Courant Institute of Mathematical Sciences, New York University, 1969.

[Com+71] F. Commoner et al. Marked directed graphs. In: *Journal of Computer and System Sciences* 5.5 (1971), pages 511–523. DOI: 10.1016/S0022-0000(71)80013-2.

[CRM75] S. Crespi-Reghizzi and D. Mandrioli. A decidability theorem for a class of vector-addition systems. In: *Information Processing Letters* 3.3 (1975), pages 78–80.

[CRM76] S. Crespi-Reghizzi and D. Mandrioli. Commutative grammars. English. In: *CALCOLO* 13.2 (1976), pages 173–189. DOI: 10.1007/BF02575679.

[Del+13] J.-M. Delosme, T. Hujsa, and A. Munier-Kordon. Polynomial sufficient conditions of well-behavedness for weighted join-free and choice-free systems. In: *Proceedings of the 13th International Conference on Application of Concurrency to System Design (ACSD'13)*. 2013, pages 90–99. DOI: 10.1109/ACSD.2013.12.

[DE95] J. Desel and J. Esparza. *Free Choice Petri Nets*. Cambridge University Press, 1995.

[Esp97] J. Esparza. Petri nets, commutative context-free grammars, and basic parallel processes. In: *Fundamenta Informaticae* 31.1 (1997), pages 13–25. DOI: 10.3233/FI-1997-3112.

[Esp98] J. Esparza. Reachability in live and safe free-choice Petri nets is NP-complete. In: *Theoretical Computer Science* 198 (1998), pages 211–224. DOI: 10.1016/S0304-3975(97)00235-1.

[EN94] J. Esparza and M. Nielsen. Decibility issues for Petri nets - a survey. In: *Journal of Information Processing and Cybernetics* 30.3 (1994), pages 143–160.

[Esp+00] J. Esparza, P. Rossmanith, and S. Schwoon. A uniform framework for problems on context-free grammars. In: *Bulletin of the EATCS* 72 (2000), pages 169–177. URL: http://www7.in.tum.de/um/bibdb/info/schwoon.ERS00.shtml.

[Eve+84] S. Even, A. L. Selman, and Y. Yacobi. The complexity of promise problems with applications to public-key cryptography. In: *Information and Control* 61.2 (1984), pages 159–173. DOI: 10.1016/S0019-9958(84)80056-X.

[FEJ89] D. de Frutos Escrig and C. Johnen. *Decidability of home space property*. Report LRI 503. Université de Paris-Sud, Centre d'Orsay, Laboratoire de Recherche en Informatique, 1989. URL: http://www.labri.fr/perso/johnen/pdf/rapLRI503.pdf.

[Gen71] H. J. Genrich. *Einfache nicht-sequentielle Prozesse*. Report 37. Bonn: Institut für Informationssystemforschung, Gesellschaft für Mathematik und Datenverarbeitung, 1971.

[GS66] S. Ginsburg and E. H. Spanier. Semigroups, Presburger formulas, and languages. In: *Pacific Journal of Mathematics* 16.2 (1966), pages 285–296. URL: http://projecteuclid.org/euclid.pjm/1102994974.

[Gol06] O. Goldreich. On promise problems: a survey. In: *Theoretical Computer Science*. Volume 3895. Lecture Notes in Computer Science. Springer, 2006, pages 254–290. DOI: 10.1007/11685654_12.

[Gra80] J. Grabowski. The decidability of persistence for vector addition systems. In: *Information Processing Letters* 11.1 (1980), pages 20–23. DOI: 10.1016/0020-0190(80)90026-5.

[Ha+12] L. M. Ha, P. V. Trung, and P. T. H. Duong. A polynomial-time algorithm for reachability problem of a subclass of Petri net and chip firing games. In: *Proceedings of the 2012 IEEE International Conference on Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF'12)*. 2012, pages 1–6. DOI: 10.1109/rivf.2012.6169852.

[Hac72] M. Hack. *Analysis of production schemata by Petri nets*. Computation Structures Group Note 17. Cambridge, Massachusetts: Massachusetts Institute of Technology, Project MAC, 1972.

[Hac73] M. Hack. *A Petri net version of Rabin's undecidability proof for vector addition systems*. Computation Structures Group Memo 94. Cambridge, Massachusetts: Massachusetts Institute of Technology, Project MAC, 1973.

[Hac74a] M. Hack. *Decision problems for Petri nets and vector addition systems*. Computation Structures Group Memo 95. Cambridge, Massachusetts: Massachusetts Institute of Technology, Project MAC, 1974.

[Hac74b] M. Hack. The recursive equivalence of the reachability problem and the liveness problem for Petri nets and vector addition systems. In: *IEEE Conference Record of the 15th Annual Symposium on Switching and Automata Theory*. 1974, pages 156–164. DOI: 10.1109/SWAT.1974.28.

[Hac76] M. Hack. The equality problem for vector addition systems is undecidable. In: *Theoretical Computer Science* 2.1 (1976), pages 77–95. DOI: 10.1016/0304-3975(76)90008-6.

[Hac79] M. Hack. *Decidability questions for Petri nets*. Garland, 1979.

[HI88] K. Hiraishi and A. Ichikawa. A class of Petri nets that a necessary and sufficient condition for reachability is obtainable. Japanese. In: *Transactions of the Society of Instrument and Control Engineers* 24.6 (1988), pages 635–40.

[HT01] K. Hiraishi and H. Tanaka. An algorithm for legal firing sequence problem of Petri nets based on partial order method. In: *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* E84.11 (2001), pages 2881–2884. URL: http://search.ieice.org/bin/summary.php?id=e84-a_11_2881.

[Hir94] Y. Hirshfeld. Petri nets and the equivalence problem. In: *Selected Papers of the 7th Workshop on Computer Science Logic (CSL'93)*. Volume 832. Lecture Notes in Computer Science. Springer, 1994, pages 165–174. DOI: 10.1007/BFb0049331.

[HC70] A. W. Holt and F. Commoner. Events and conditions. In: *Record of the Project MAC Conference on Concurrent Systems and Parallel Computation*. New York: ACM, 1970, pages 3–52.

[HP79] J. E. Hopcroft and J.-J. Pansiot. On the reachability problem for 5-dimensional vector addition systems. In: *Theoretical Computer Science* 8.2 (1979), pages 135–159. DOI: 10.1016/0304-3975(79)90041-0.

[HU79] J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, 1979.

[HR88] R. R. Howell and L. E. Rosier. Completeness results for conflict-free vector replacement systems. In: *Journal of Computer and System Sciences* 37.3 (1988), pages 349–366. DOI: 10.1016/0022-0000(88)90013-X.

[HR89] R. R. Howell and L. E. Rosier. Problems concerning fairness and temporal logic for conflict-free Petri nets. In: *Theoretical Computer Science* 64.3 (1989), pages 305–329. DOI: 10.1016/0304-3975(89)90053-4.

[How+87] R. R. Howell, L. E. Rosier, and H.-C. Yen. An $O(n^{1.5})$ algorithm to decide boundedness for conflict-free vector replacement systems. In: *Information Processing Letters* 25.1 (1 1987), pages 27–33. DOI: 10.1016/0020-0190(87)90089-5.

[How+89] R. R. Howell, L. E. Rosier, and H.-C. Yen. Normal and sinkless Petri nets. In: *Proceedings of the 1989 International Conference on Fundamentals of Computation Theory (FCT'89)*. Volume 380. Lecture Notes in Computer Science. Springer, 1989, pages 234–243. DOI: 10.1007/3-540-51498-8_22.

[How+93] R. R. Howell, P. Jancar, and L. E. Rosier. Completeness results for single-path Petri nets. In: *Information and Computation* 106.2 (1993), pages 253–265. DOI: 10.1006/inco.1993.1055.

[Huy80] D. T. Huynh. The complexity of semilinear sets. In: *Proceedings of the 7th International Colloquium on Automata, Languages and Programming (ICALP'80)*. Volume 85. Lecture Notes in Computer Science. Springer, 1980, pages 324–337. DOI: 10.1007/3-540-10003-2_81.

[Huy83] D. T. Huynh. Commutative grammars: the complexity of uniform word problems. In: *Information and Control* 57.1 (1983), pages 21–39. DOI: 10.1016/S0019-9958(83)80022-9.

[Huy84] D. T. Huynh. Deciding the inequivalence of context-free grammars with 1-letter terminal alphabet is Σ_2^p-complete. In: *Theoretical Computer Science* 33.2–3 (1984), pages 305–326. DOI: 10.1016/0304-3975(84)90092-6.

[Huy85] D. T. Huynh. The complexity of equivalence problems for commutative grammars. In: *Information and Control* 66.1–2 (1985), pages 103–121. DOI: 10.1016/S0019-9958(85)80015-2.

[Huy86] D. T. Huynh. A simple proof for the Σ_2^p upper bound of the inequivalence problem for semilinear sets. In: *Elektronische Informationsverarbeitung und Kybernetik* 22 (1986), pages 147–156.

[Imm88] N. Immerman. Nondeterministic space is closed under complementation. In: *SIAM Journal on Computing* 17.5 (1988), pages 935–938. DOI: 10.1137/0217058.

[JL76] N. D. Jones and W. T. Laaser. Complete problems for deterministic polynomial time. In: *Theoretical Computer Science* 3.1 (1976), pages 105–117. DOI: 10.1016/0304-3975(76)90068-2.

[Jon+77] N. D. Jones, L. H. Landweber, and Y. E. Lien. Complexity of some problems in Petri nets. In: *Theoretical Computer Science* 4.3 (1977), pages 277–299. DOI: 10.1016/0304-3975(77)90014-7.

[Kar84] N. Karmarkar. A new polynomial-time algorithm for linear programming. English. In: *Combinatorica* 4.4 (1984), pages 373–395. DOI: 10.1007/BF02579150.

[KM69] R. M. Karp and R. E. Miller. Parallel program schemata. In: *Journal of Computer and System Sciences* 3.2 (1969), pages 147–195. DOI: 10.1016/S0022-0000(69)80011-5.

[Kas65] T. Kasami. *An efficient recognition and syntax-analysis algorithm for context-free languages*. Technical report. Bedford, Massachusetts: Air Force Cambridge Research Lab, 1965.

[Kel72] R. M. Keller. *Vector replacement systems: a formalism for modeling asynchronous systems*. Technical Report 117. Princeton, New Jersey: Computer Science Laboratory, Princeton University, 1972.

[Kos82] S. R. Kosaraju. Decidability of reachability in vector addition systems (preliminary version). In: *Proceedings of the 14th ACM Symposium on Theory of Computing (STOC'82)*. San Francisco, California, USA: ACM, 1982, pages 267–281. DOI: 10.1145/800070.802201.

[Kuč96] A. Kučera. Regularity is decidable for normed PA processes in polynomial time. In: *Proceedings of the 16th Conference on Foundations of Software Technology and Theoretical Computer Science*. Volume 1180. Lecture Notes in Computer Science. Springer, 1996, pages 111–122. DOI: 10.1007/3-540-62034-6_42.

[Kur64] S.-Y. Kuroda. Classes of languages and linear-bounded automata. In: *Information and Control* 7.2 (1964), pages 207–223. DOI: 10.1016/S0019-9958(64)90120-2.

[Lam92] J. L. Lambert. A structure to decide reachability in Petri nets. In: *Theoretical Computer Science* 99.1 (1992), pages 79–104. DOI: dx.doi.org/10.1016/0304-3975(92)90173-D.

[LR78] L. H. Landweber and E. L. Robertson. Properties of conflict-free and persistent Petri nets. In: *Journal of the ACM* 25.3 (1978), pages 352–364. DOI: 10.1145/322077.322079.

[Ler09] J. Leroux. The general vector addition system reachability problem by presburger inductive invariants. In: *Proceedings of the 24th Annual IEEE Symposium on Logic In Computer Science (LICS'09)*. 2009, pages 4–13. DOI: 10.1109/LICS.2009.10.

[Ler11] J. Leroux. Vector addition system reachability problem: a short self-contained proof. In: *ACM SIGPLAN Notices* 46.1 (2011), pages 307–316. DOI: 10.1145/1925844.1926421.

[Lie72] Y. E. Lien. Study of theoretical and practical aspects of transition systems. PhD thesis. Berkely, California: University of California, 1972.

[Lie76a] Y. E. Lien. A note on transition systems. In: *Information Sciences* 10.4 (1976), pages 347–362. DOI: 10.1016/0020-0255(76)90054-2.

[Lie76b] Y. E. Lien. Termination properties of generalized Petri nets. In: *SIAM Journal on Computing* 5.2 (1976), pages 251–265. DOI: 10.1137/0205020.

[Lip76] R. J. Lipton. *The reachability problem requires exponential space*. Research Report 62. New Haven, Connecticut: Yale University, Department of Computer Science, 1976.

[May81a] E. W. Mayr. An algorithm for the general Petri net reachability problem. In: *Proceedings of the 13th ACM Symposium on Theory of Computing (STOC'81)*. Milwaukee, Wisconsin, United States: ACM, 1981, pages 238–246. DOI: 10.1145/800076.802477.

[May81b] E. W. Mayr. Persistence of vector replacement systems is decidable. In: *Acta Informatica* 15.3 (1981), pages 309–318. DOI: 10.1007/BF00289268.

[May84] E. W. Mayr. An algorithm for the general Petri net reachability problem. In: *SIAM Journal on Computing* 13.3 (1984), pages 441–460. DOI: 10.1137/0213029.

[MM81] E. W. Mayr and A. R. Meyer. The complexity of the finite containment problem for Petri nets. In: *Journal of the ACM* 28.3 (1981), pages 561–576. DOI: 10.1145/322261.322271.

[MM82] E. W. Mayr and A. R. Meyer. The complexity of the word problems for commutative semigroups and polynomial ideals. In: *Advances in Mathematics* 46.3 (1982), pages 305–329. DOI: 10.1016/0001-8708(82)90048-2.

[May97] R. Mayr. Tableau methods for PA-processes. In: *Proceedings of the 1997 International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX'97)*. Volume 1227. Lecture Notes in Computer Science. Springer, 1997, pages 276–290. DOI: 10.1007/BFb0027420.

[May98] R. Mayr. Decidability and Complexity of Model Checking Problems for Infinite-State Systems. PhD thesis. Technische Universität München, 1998.

[May00] R. Mayr. On the complexity of bisimulation problems for basic parallel processes. In: *Proceedings of the 27th International Colloquium on Automata, Languages and Programming (ICALP'00)*. Volume 1853. Lecture Notes in Computer Science. Springer, 2000, pages 329–341. DOI: 10.1007/3-540-45022-X_29.

[Mel+02] R. Melinte et al. The home marking problem and some related concepts. In: *Acta Cybernetica* 15.3 (2002), pages 467–478.

[Mil95] R. Milner. *Communication and concurrency*. Prentice Hall, 1995.

[Mül80] H. Müller. Decidability of reachability in persistent vector replacement systems. In: *Proceedings of the 9th Symposium on Mathematical Foundations of Computer Science (MFCS'80)*. Volume 88. Lecture Notes in Computer Science. Springer, 1980, pages 426–438. DOI: 10.1007/BFb0022522.

[MW96] K. Morita and T. Watanabe. The legal firing sequence problem of Petri nets with state machine structure. In: *Proceedings of the 1996 IEEE International Symposium on Circuits and Systems (ISCAS'96)*. Volume 3. 1996, pages 64–67. DOI: 10.1109/ISCAS.1996.541481.

[Mur77] T. Murata. State equation, controllability, and maximal matchings of Petri nets. In: *IEEE Transactions on Automatic Control* 22.3 (1977), pages 412–416. DOI: 10.1109/TAC.1977.1101509.

[Mur89] T. Murata. Petri nets: properties, analysis and applications. In: *Proceedings of the IEEE* 77.4 (1989), pages 541–580. DOI: 10.1109/5.24143.

[Par61] R. Parikh. *Language generating devices*. Quarterly Progress Report 60. Cambridge, Massachusetts: Massachusetts Institute of Technology, 1961, pages 199–212.

[Pet81] J. L. Peterson. *Petri net theory and the modeling of systems*. Prentice Hall, 1981.

[Pet62] C. A. Petri. Kommunikation mit Automaten. German. PhD thesis. Universität Hamburg, 1962. URL: http://edoc.sub.uni-hamburg.de/informatik/volltexte/2011/160/.

[Pot91] L. Pottier. Minimal solutions of linear diophantine systems: bounds and algorithms. In: *Proceedings of the 4th International Conference on Rewriting Techniques and Applications (RTA'91)*. Volume 488. Lecture Notes in Computer Science. Springer, 1991, pages 162–173. DOI: 10.1007/3-540-53904-2_94.

[Pre30] M. Presburger. Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt. In: *Sprawozdanie z I Kongresu matematyków krajów słowiańskich*. Warsaw, 1930, pages 92–101.

[Rac78] C. Rackoff. The covering and boundedness problems for vector addition systems. In: *Theoretical Computer Science* 6.2 (1978), pages 223–231. DOI: 10.1016/0304-3975(78)90036-1.

[RY86] L. E. Rosier and H.-C. Yen. A multiparameter analysis of the boundedness problem for vector addition systems. In: *Journal of Computer and System Sciences* 32.1 (1986), pages 105–135. DOI: 10.1016/0022-0000(86)90006-1.

[Sav70] W. J. Savitch. Relationships between nondeterministic and deterministic tape complexities. In: *Journal of Computer and System Sciences* 4.2 (1970), pages 177–192. DOI: 10.1016/S0022-0000(70)80006-X.

[Sip97] M. Sipser. *Introduction to the theory of computation*. PWS Publishing Company, 1997, pages I–XV, 1–396.

[Ste91] I. A. Stewart. *On the reachability problem for some classes of Petri nets*. Technical Report 357. Newcastle upon Tyne: Newcastle University, 1991. URL: http://www.cs.ncl.ac.uk/publications/trs/papers/357.pdf.

[Sze88] R. Szelepcsényi. The method of forced enumeration for nondeterministic automata. English. In: *Acta Informatica* 26.3 (1988), pages 279–284. DOI: 10.1007/BF00299636.

[TW99] S. Taoka and T. Watanabe. A linear time algorithm solving the legal firing sequence problem for a class of edge-weighted cactuses. In: *Proceedings of the 1999 IEEE International Conference on Systems, Man, and Cybernetics (SMC'99)*. Volume 3. 1999, pages 893–898. DOI: 10.1109/ICSMC.1999.823346.

[TW06] S. Taoka and T. Watanabe. Time complexity analysis of the legal firing sequence problem of Petri nets with inhibitor arcs. In: *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* E89.11 (2006), pages 3216–3226. DOI: 10.1093/ietfec/e89-a.11.3216.

[Tao+03] S. Taoka, S. Furusato, and T. Watanabe. A heuristic algorithm FSDC based on avoidance of deadlock components in finding legal firing sequences of Petri nets. In: *Proceedings of the 24th International Conference on Applications and Theory of Petri Nets (ICATPN'03)*. Volume 2679. Lecture Notes in Computer Science. Springer, 2003, pages 417–439. DOI: 10.1007/3-540-44919-1_26.

[Tar72] R. Tarjan. Depth-first search and linear graph algorithms. In: *SIAM Journal on Computing* 1.2 (1972), pages 146–160. DOI: 10.1137/0201010.

[TS93] E. Teruel and M. Silva. Liveness and home states in equal conflict systems. In: *Proceedings of the 14th International Conference on Application and Theory of Petri Nets (ICATPN'93)*. Volume 691. Lecture Notes in Computer Science. Springer, 1993, pages 415–432. DOI: 10.1007/3-540-56863-8_59.

[TS94] E. Teruel and M. Silva. Well-formedness of equal conflict systems. In: *Proceedings of the 15th International Conference on Application and Theory of Petri Nets (ICATPN'94)*. Volume 815. Lecture Notes in Computer Science. Springer, 1994, pages 491–510. DOI: 10.1007/3-540-58152-9_27.

[TS96] E. Teruel and M. Silva. Structure theory of equal conflict systems. In: *Theoretical Computer Science* 153 (1996), pages 271–300. DOI: 10.1016/0304-3975(95)00124-7.

[Ter+92] E. Teruel et al. On weighted T-systems. In: *Proceedings of the 13th International Conference on Application and Theory of Petri Nets (ICATPN'92)*. Volume 616. Lecture Notes in Computer Science. Springer, 1992, pages 348–367. DOI: 10.1007/3-540-55676-1_20.

[Ter+97] E. Teruel, J. M. Colom, and M. Silva. Choice-free Petri nets: a model for deterministic concurrent systems with bulk services and arrivals. In: *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* 27.1 (1997), pages 73–83. DOI: 10.1109/3468.553226.

[The06] C. Theil. Effizienz von Algorithmen in konfliktfreien und persistenten Petrinetzen. Diploma thesis. Garching: Technische Universität München, 2006.

[Wat+89a] T. Watanabe, Y. Mizobata, and K. Onaga. Legal firing sequence and related problems of Petri nets. In: *Proceedings of the 3rd International Workshop on Petri Nets and Performance Models (PNPM'89)*. 1989, pages 277–286. DOI: 10.1109/PNPM.1989.68561.

[Wat+89b] T. Watanabe, Y. Mizobata, and K. Onaga. Legal firing sequences and minimum initial markings for Petri nets. In: *Proceedings of the 1989 IEEE International Symposium on Circuits and Systems (ISCAS'89)*. 1989, pages 323–326. DOI: 10.1109/ISCAS.1989.100356.

[Yam84] H. Yamasaki. Normal Petri nets. In: *Theoretical Computer Science* 31.3 (1984), pages 307–315. DOI: 10.1016/0304-3975(84)90038-0.

[Yen97] H.-C. Yen. On reachability equivalence for BPP-nets. In: *Theoretical Computer Science* 179 (1997), pages 301–317. DOI: 10.1016/S0304-3975(96)00147-8.

[Yen98] H.-C. Yen. Priority conflict-free Petri nets. In: *Acta Informatica* 35.8 (1998), pages 673–688. DOI: 10.1007/s002360050138.

[Yen99] H.-C. Yen. Integer linear programming and the analysis of some Petri net problems. In: *Theory of Computing Systems* 32.4 (1999), pages 467–485. DOI: 10.1007/s002240000127.

[Yen02] H.-C. Yen. Sequential versus concurrent languages of labeled conflict-free Petri nets. In: *IEEE Transactions on Automatic Control* 47.7 (2002), pages 1158–1162. DOI: 10.1109/TAC.2002.800664.

[Yen13] H.-C. Yen. *Private Communication*. 2013.

[YY03] H.-C. Yen and L.-P. Yu. Petri nets with simple circuits. In: *Proceedings of the 9th Annual International Conference on Computing and Combinatorics (COCOON'03)*. Volume 2697. Lecture Notes in Computer Science. Springer, 2003, pages 149–158. DOI: 10.1007/3-540-45071-8_17.

[Yen+93] H.-C. Yen, B.-Y. Wang, and M.-S. Yang. A unified approach for reasoning about conflict-free Petri nets. In: *Proceedings of the 14th International Conference on Application and Theory of Petri Nets (ICATPN'93)*. Volume 691. Lecture Notes in Computer Science. Springer, 1993, pages 513–531. DOI: 10.1007/3-540-56863-8_64.

[You67] D. H. Younger. Recognition and parsing of context-free languages in time n^3. In: *Information and Control* 10.2 (1967), pages 189–208. DOI: 10.1016/S0019-9958(67)80007-X.

Index

List of Symbols

$v_{[i]}$	i-th component of vector/sequence/word v	7
v_i	i-th vector/sequence of an indexed set of vectors/sequences	7
$\mathbb{Z}$	integers	7
$\mathbb{N}_0$	nonnegative integers	7
$\mathbb{N}$	positive integers	7
$\mathbb{Q}$	rational numbers	7
$\mathbb{Q}_{\geq 0}$	nonnegative rational numbers	7
$\mathbb{Q}_{>0}$	positive rational numbers	7
$\mathbb{Q}_{<0}$	negative rational numbers	7
$\mathbb{R}$	real numbers	7
ld	binary logarithm	7
$[a, b]$	interval $\{a, a+1, \ldots, b\} \subsetneqq \mathbb{Z}$	7
$[a]$	interval $\{1, 2, \ldots, a\} \subsetneqq \mathbb{N}$	7
$u \geq v$	$u_{[i]} \geq v_{[i]}$ for all i	7
$u > v$	$u_{[i]} \geq v_{[i]}$ for all i and $u \neq v$	7
$\vec{a}$	vector with $\vec{a}_{[i]} = a$ for all i	7
$\lVert v \rVert_1$	1-norm of v	7
$\lVert v \rVert_\infty$	∞-norm of v	7
$\max(v)$	largest component of v	7
$\min(v)$	smallest component of v	7
$\lVert A \rVert_{1,\infty}$	maximum of all 1-norms of the columns of A	7
$\sigma \cdot \varphi$	concatenation of σ and φ	7
$\sigma_{[i..j]}$	subsequence/subword $\sigma_{[i]} \cdot \sigma_{[i+1]} \cdots \sigma_{[j]}$	7
$\sigma_{[..i]}$	prefix $\sigma_{[1]} \cdot \sigma_{[2]} \cdots \sigma_{[i]}$	7
$\sigma_{[i..]}$	suffix $\sigma_{[i]} \cdot \sigma_{[i+1]} \cdots$	7
$\lvert \sigma \rvert$	length of sequence/word σ	7
ϵ	empty sequence/word	7
$\sigma \leftarrow \varphi$	assignment of the value of sequence φ to σ	7
e_k	k-th standard unit vector	7
$f(.) \overset{\mathrm{P}}{\leq} g(.)$	f is polynomially bounded by g	7
$\mathrm{size}(x)$	encoding size of object x	11
$v(p)$	component of v corresponding to place or transition p	12
$\Psi(\sigma)$	Parikh image of σ	13
$\Delta(x)$	displacement of (Parikh) vector of transition sequence x	13
$t \in \Phi$	$\Phi(t) > 0$ for transition t and (Parikh) vector Φ	13
$t \in \sigma$	transition t is contained in sequence σ	13
$\mu \xrightarrow{\sigma} \mu'$	σ is firing sequence, leading from μ to μ'	13
$\mu \xrightarrow{\Phi} \mu'$	$\exists\, \sigma$ with $\Psi(\sigma) = \Phi$ and $\mu \xrightarrow{\sigma} \mu'$	13
$\mathcal{R}(N, \mu_0)$	reachability set of Petri net (N, μ_0)	14
$\max(\mu, \sigma, S)$	$\max\{(\mu + \Delta(\sigma_{[..i]}))(p) \mid i \in [0, \lvert\sigma\rvert], p \in S\}$	14
D	displacement matrix	14
${}^\bullet x$	preset of transition, transition sequence or place x	14

$x^\bullet$	postset of transition, transition sequence or place x	14
$\sigma \dot{-} \varphi$	the sequence obtained by deleting the transitions of φ greedily from the front of σ	14
$v :\triangleq \sum$	definition of a marking or (Parikh) vector v in terms of a sum of places or transitions	14
$v \triangleq \sum$	association of a vector v with a sum of places or transitions	14
$\overset{t}{\underset{F}{\mapsto}}$	compact description of $F(p,t)$ and $F(t,p)$ for all places p	14
$\mathcal{P}[T']$	the Petri net induced by Petri net $\mathcal{P}$ and a subset $T' \subseteq T$ of transitions	15
$\mathcal{P}[\Phi]$	the Petri net induced by Petri net $\mathcal{P}$ and Parikh vector Φ	15
$\widehat{\mathcal{P}}$	wipe extension of Petri net $\mathcal{P}$	15
$\mathcal{L}$	linear set representation	20
$\mathcal{SL}$	semilinear set representation	20
$\odot$	concatenation operator for semilinear set representations	20
I_m	$m \times m$ identity matrix	24
$\mathcal{P}^\Psi$	Parikh extension of Petri net $\mathcal{P}$	29
$\mathcal{P}^G_{M,x}$	Petri net for gadget G, **PSPACE**-TM M, and input string x	59
$\Psi_{\text{first}}(\sigma)$	0-1-vector with $\Psi_{\text{first}}(\sigma)(t) = 1$ iff the first transition t' in σ with $^\bullet t' = {}^\bullet t$ is t	79

Acronyms

a. o.	almost ordinary
BPP	basic parallel processes
cf-PN	communication-free Petri net
CFCG	context-free commutative grammar
CFG	context-free grammar
cnf-PN	conflict-free Petri net
DFS	depth first search
ESCG	exponent-sensitive commutative grammar
ESG	exponent-sensitive grammar
gcf-PN	generalized communication-free Petri net
gcnf-PN	generalized conflict-free Petri net
gss-PN	generalized S-system Petri net
igcf-PN	inverse generalized communication-free Petri net
LSR	linear set representation
NDTM	nondeterministic Turing machine
PN	Petri net
RecLFS	recognize legal firing sequence
SCC	strongly connected component
SIU	simultaneously-ω-unboundedness problem
SLSR	semilinear set representation
SU	simultaneously-unboundedness problem
TM	Turing machine
VAS	vector addition system
VRS	vector replacement system