Fundamenta Informaticae 137 (2015) 61–86
DOI 10.3233/FI-2015-1170
IOS Press

Complexity Results for Problems of Communication-Free Petri Nets and Related Formalisms

Ernst W. Mayr*, **Jeremias Weihmann**
Institut für Informatik
Technische Universität München
Garching, Germany
mayr@in.tum.de; weihmann@in.tum.de

Abstract. We investigate several computational problems of communication-free Petri nets, and develop very efficient (mostly linear time) algorithms for different 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 part, we use our results for cf-PNs to give linear time algorithms for related problems of context-free (commutative) grammars, and, in turn, use known results for such grammars to give a **coNEXPTIME**-upper bound for the equivalence problem of cf-PNs.

1. Introduction

Communication-free Petri nets (cf-PNs [13], also known as BPP-Petri nets) are characterized by the simple topological constraint that each transition has exactly one input place (connected by an arc with multiplicity 1). There are several reasons why it is insightful to investigate this class. Studying nontrivial subclasses of Petri nets helps understanding the dynamics of Petri nets in general, which could finally lead to a primitive recursive algorithm for the reachability problem of general Petri nets (a non primitive recursive algorithm was given by Mayr [24]). Furthermore, this Petri net class is closely related to both Basic Parallel Processes (BPP, see [4, 5]), a subclass of Milner's process algebra (PA, which is also called Calculus of Communicating Systems, CCS, see [30]), and to context-free commutative grammars (CFCG, see [6, 9, 17]).

*Address for correspondence: Institut für Informatik, Technische Universität München, Boltzmannstr. 3, 85748 Garching, Germany

The strong topological constraint on cf-PNs limits the computational power of these nets in the sense that they are unable to model synchronizing actions since the fireability of a transition only depends on exactly one place. Esparza [9] showed that the reachability problem of cf-PNs is, nevertheless, **NP**-hard. Furthermore, he showed that the problem is in **NP**. Both results together yield an alternative proof for the **NP**-completeness of the uniform word problem for context-free commutative grammars (as shown earlier by Huynh [17]). Another proof for membership in **NP**, based on canonical firing sequences, was given by Yen [35].

For the equivalence problem of cf-PNs, which asks, given two cf-PNs, whether both have the same set of reachable markings, a **coNEXPTIME**-upper bound is implied by the same upper bound for the equivalence problem of CFCGs which was shown by Huynh [18]. His approach is based on the observation that the language of such a grammar is a semilinear set which has a semilinear set representation (SLSR) consisting of small linear set representations with nice properties. This result can be found in Corollary 6.1 of Section 6. The algorithm presented by Yen [35] for computing SLSRs of reachability sets of cf-PNs, which has been used to obtain another (weaker) upper bound of this problem, contains a gap that was fixed in a preliminary version of this paper [26]. Later, Yen [36] gave a proof for $\mathbf{\Pi}_2^p$-hardness for the equivalence problem, by reducing the $\mathbf{\Pi}_2^p$-complete problem $\overline{\mathbf{\Sigma}_2^p\text{-3-SAT}}$ to it. The latter denotes the complement of $\mathbf{\Sigma}_2^p$-3-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. Fribourg and Olsén [11] showed that also the set $\{(\mu, \mu') \mid \mu \to \mu'\}$ (i. e., all pairs of markings μ, μ' such that μ' can be reached from μ) of an unmarked cf-PN N is semilinear and an SLSR of this set is effectively constructible. We remark that one can improve on this result by using a time- or space-bounded algorithm (e. g., that of [26] or an adaptation of the algorithm given in [17]) to compute an SLSR of the reachability set of the modified input net, where we have additional places on which the initial marking of the original net is produced by additional transitions and subsequently stored. Kopczyński and To [21] showed that the equivalence problem of CFCGs with a fixed number of terminal symbols is $\mathbf{\Pi}_2^p$-complete. They used this result to obtain $\mathbf{\Pi}_2^p$-membership[1] for the equivalence problem of cf-PNs with a fixed number of places.

Our paper is structured as follows. In Section 2, we introduce the mathematical notation and concepts used in this paper (except for commutative grammars which are introduced later in Section 6).

In Section 3, we collect a number of useful observations about cf-PNs to establish the foundations for the later sections. This section also contains a first immediate result, namely, that the zero-reachability problem of cf-PNs is decidable in linear time.

For some notions of boundedness and liveness of BPPs ([22, 27, 28], also see [29]), polynomial time algorithms are already known. In addition to these, we also investigate a number of other variations of the boundedness and the covering problems for cf-PNs in Section 4, and variations of the liveness problem for cf-PNs in Section 5. For two new variants of the boundedness problem, and for the covering problem, we show **NP**-completeness. Most of the remaining problems can be decided very efficiently

[1]The actual theorem claims $\mathbf{\Pi}_2^p$-completeness but there is no simple argument why $\mathbf{\Pi}_2^p$-hardness for the equivalence problem of CFCGs with a fixed number of terminals implies $\mathbf{\Pi}_2^p$-hardness for the equivalence problem of cf-PNs with a fixed number of places since variables of grammars are usually also modeled by places. As a consequence, the words produced by the grammar do not correspond to the markings reachable in the net. A similar gap is contained in [35] in the argument for the (unrestricted) equivalence problem of cf-PNs, however, Yen [36] later presented a valid proof.

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 (e. g., finiteness of context-free grammars, see [10]). These results can be found in Section 6.

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, which are those Petri nets in which each arc from a place to a transition has multiplicity 1 and in which no transition can decrease the number of tokens of a place with more than one outgoing arc. Both are defined structurally in a similar way, both have **NP**-complete reachability problems ([9, 15, 19]), polynomial time algorithms for the boundedness problem (see this paper and [1, 16]), the RecLFS problem (which asks, given a Parikh vector, if is is enabled, see [9, 15]), the liveness problem (see this paper and [1]), and $\mathbf{\Pi}_2^p$-hard equivalence problems (even $\mathbf{\Pi}_2^p$-complete for conflict-free Petri nets, see [15, 36]).

In [25], the authors investigated a generalization of cf-PNs, called gcf-PNs (also known as join-free Petri nets), characterized by the sole topological constraint that each transition has *at most* one incoming arc. In particular, all arcs are also allowed to have multiplicities other than 1. It turns out that almost all problems considered (e. g., RecLFS, reachability, boundedness, covering) are **PSPACE**-complete. The large gap regarding the complexity between many problems for cf-PNs and corresponding problems for gcf-PNs motivates further investigation of how generalizing classes of Petri nets regarding their arc multiplicities influences the complexity of analogous decision problems. A number of other extensions of cf-PNs were investigated in [3]. For some of these extensions, the reachability problem remains **NP**-complete, while for others it turns out to be not decidable at all.

2. Preliminaries

2.1. Basic Notation and Concepts

Throughout this paper, 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 or a sequence v. The notation v_i is reserved for indexed elements of a set (e. g., $p_1, p_2, \ldots$). The set of all integers, all nonnegative integers, and all positive integers are denoted by $\mathbb{Z}$, $\mathbb{N}$, and $\mathbb{N}_{>0}$, respectively, while $[a, b] := \{a, a+1, \ldots, b\} \subsetneq \mathbb{Z}$, and $[k] = [1, k] \subsetneq \mathbb{N}_{>0}$. For two vectors $u, v \in \mathbb{Z}^k$, we write $u \geq v$ if $u_{[i]} \geq v_{[i]}$ for all $i \in [k]$, and $u > v$ if $u \geq v$ and $u_{[i]} > v_{[i]}$ for some $i \in [k]$. When k is understood, $\vec{a}$ denotes, for a number $a \in \mathbb{Z}$, the k-dimensional vector with $\vec{a}_{[i]} = a$ for all $i \in [k]$.

Throughout this paper we use a succinct encoding scheme. Every number is encoded in binary representation. A vector of $\mathbb{N}^k$ is encoded as a k-tuple. If we regard a tuple as an input, then it is encoded as a tuple of the encodings of the particular components. For an object x, size(x) denotes the encoding size of x under this encoding scheme. We remark that the input size of a problem instance consists of the encodings of all entities that are declared as being "given" in the respective problem statement. As our model of computation, we will use the RAM with logarithmic word size.

2.2. Petri Nets

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 $F : (P \times T) \cup (T \times P) \to \mathbb{N}$ is a *flow function*. Throughout this paper, n and m will always refer to the number of places resp. transitions of the Petri net under consideration. 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 context of an m-dimensional vector v and a transition t.

A *marking* μ (of N) is a vector of $\mathbb{N}^n$. A pair (N, μ_0) such that μ_0 is a marking of N is called a *marked Petri net*, and μ_0 is called its *initial marking*. We will omit the term "marked" if the presence of a certain initial marking is clear from the context. A place p (set $S \subseteq P$, 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.

For a transition $t \in T$, ${}^\bullet t$ ($t^\bullet$, resp.) is the *preset* (*postset*, resp.) of t and denotes the set of all places p such that $F(p, t) > 0$ ($F(t, p) > 0$, resp.). Analogously, the sets ${}^\bullet p$ and $p^\bullet$ of transitions are defined for the places $p \in P$. A Petri net (P, T, F) is a *communication-free Petri net* (*cf-PN*) if $|{}^\bullet t| = 1$ and $F(p, t) = 1$ for all $t \in T$ and $p \in {}^\bullet t$. In the case of cf-PNs, we occasionally abuse notation by identifying ${}^\bullet t$ with its unique element, for the sake of improved readability.

A Petri net naturally corresponds to a directed bipartite graph with arcs from P to T and vice versa such that there is an arc from $p \in P$ to $t \in T$ (from t to p, resp.) labeled with w if $0 < F(p, t) = w$ (if $0 < F(t, p) = w$, resp.). The label of an arc is called its *multiplicity*. If a Petri net is visualized, places are usually drawn as circles and transitions as bars. If the Petri net is marked by μ, then, for each place p, the circle corresponding to p contains $\mu(p)$ so-called *tokens*.

For a Petri net $N = (P, T, F)$ and a marking μ of N, a transition $t \in T$ can be *applied* at μ producing a vector $\mu' \in \mathbb{Z}^n$ with $\mu'(p) = \mu(p) - F(p, t) + F(t, p)$ for all $p \in P$. The 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 marking μ if t is enabled and applied at μ. If t is fired at μ, then the resulting vector μ' is a marking, and we write $\mu \xrightarrow{t} \mu'$. Intuitively, if a transition is fired, it first removes $F(p, t)$ tokens from p and then adds $F(t, p)$ tokens to p.

An element σ of T^* is called a *transition sequence*, and $|\sigma|$ denotes its length. An element of T^ω is an infinitely long sequence consisting of transitions and is called an ω*-transition sequence*. We write $\sigma_{[i..j]}$ for the infix $\sigma_{[i]} \cdot \sigma_{[i+1]} \cdots \sigma_{[j]}$, and $\sigma_{[..i]}$ for the prefix of length i of σ, i. e., $\sigma_{[..i]} = \sigma_{[1..i]}$. A *Parikh vector* Φ (also known as *firing count vector*) is simply an element of $\mathbb{N}^m$. The *Parikh map* $\Psi : T^* \to \mathbb{N}^m$ 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. Moreover, we write, abusing notation, $t \in \Phi$ if $\Phi(t) > 0$, and $t \in \sigma$ if $t \in \Psi(\sigma)$.

The *displacement* $\Delta : \mathbb{N}^m \to \mathbb{Z}^n$ maps Parikh vectors $\Phi \in \mathbb{N}^m$ onto the change of tokens at the places $p_1, \ldots, p_n$ when applying transition sequences with Parikh image Φ. That is, we have $\Delta(\Phi)(p) = \sum_{t \in T} \Phi(t) \cdot (F(t, p) - F(p, t))$ for all places p. Accordingly, we define the displacement $\Delta(\sigma)$ of a transition sequence σ by $\Delta(\sigma) := \Delta(\Psi(\sigma))$.

Note that $\mu \xrightarrow{t} \mu'$ iff t is enabled at μ and $\mu + \Delta(t) = \mu'$. 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 $\mu \xrightarrow{\sigma_{[..|\sigma|-1]}} \mu'' \xrightarrow{\sigma_{[|\sigma|]}} \mu'$ 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.) is *enabled at* μ or *enabled in* (N, μ) (where

(N, μ) is the Petri net under consideration), and *leads from μ to μ'*, and write $\mu \xrightarrow{\sigma}$ ($\mu \xrightarrow{\Phi}$, resp.) if we are not particularly interested in the marking μ'. Analogously, an ω-transition sequence is *enabled at μ* if each finite prefix is enabled at μ.

For a marked Petri net (N, μ_0), we call a transition sequence that is enabled at μ_0 a *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* $\mathcal{R}(N, \mu_0)$ consists of all markings that are reachable in (N, μ_0). We say that a marking μ can be *covered* if there is a reachable marking $\mu' \geq \mu$.

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 can be fired at least once 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 Petri net is encoded as an enumeration of places $p_1, \ldots, p_n$ and transitions $t_1, \ldots, t_m$ followed by an enumeration of the arcs with their respective arc multiplicities.

2.3. Graph-Theoretic Concepts for Petri Nets

Sometimes it is convenient to only consider those places and transitions that are relevant w. r. t. a given set of transitions or a Parikh vector. For a Petri net $\mathcal{P} = (P, T, F, \mu_0)$, and a set D of transitions, the Petri net $\mathcal{P}[D]$ consists of all transitions $t \in D$, all places $p \in \bigcup_{t \in D}({}^\bullet t \cup t^\bullet)$, and the flow function F and initial marking μ_0 restricted to these subsets of transitions and places. For a Parikh vector Φ we define $\mathcal{P}[\Phi] := \mathcal{P}[\{t \mid t \in \Phi\}]$.

In the case of cf-PNs the *strongly connected components* (*SCCs*) are also of major interest. The directed acyclic graph obtained by shrinking all SCCs to super nodes while maintaining the arcs between distinct SCC as arcs between the corresponding super nodes is called the *condensation (of the graph)*. Since Petri nets and their condensations are directed graphs, the term *path* will always refer to a simple directed path, i. e., a sequence $(v_1, \ldots, v_\ell)$ of different nodes such that (v_i, v_{i+1}), $i \in [\ell - 1]$, is an arc. Likewise, *cycle* refers to a simple directed cycle, i. e., a sequence $(v_1, \ldots, v_\ell = v_1)$ of nodes such that (v_i, v_{i+1}), $i \in [\ell - 1]$, is an arc and $v_i \neq v_j$ for all different $i, j \in [\ell - 1]$. We call an SCC a *top component* (*bottom component*) if it has no incoming arcs (outgoing arcs, resp.) 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).

The reason why SCCs are important is the following. Boundedness and liveness properties are closely related to the existence of loops which increase the number of tokens at certain places or which ensure that certain transitions can be fired over and over again. Loops, on the other hand, always contain a cycle if each transition has an incoming arc (which is always the case for cf-PNs). Furthermore, in cf-PNs, using each transition of a cycle exactly once constitutes a loop. In order to fire along a cycle of a cf-PNs it is sufficient and necessary that at least one place of the cycle contains at least one token since each transition of the cycle depends only on its input place which is also part of the cycle. Moreover, tokens can freely be transferred within SCCs of a cf-PN, and from an SCC C_1 to an SCC C_2 if and only if $C_1 \geq C_2$. Hence, to check if a cycle can be marked (and if the corresponding loop can be fired), we only need to investigate the SCCs of the net. If the cycle is contained in an SCC C_2 and some SCC $C_1 \geq C_2$ is marked, then we immediately know that the cycle can be marked.

An important concept in the analysis of Petri nets are traps. A subset $Q \subseteq P$ of places is a *trap* if, for all $t \in T$, ${}^\bullet t \cap Q \neq \emptyset$ implies $t^\bullet \cap Q \neq \emptyset$, i. e., 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 $R \subseteq P$, 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.

3. Fundamental Observations

Some observations about cf-PNs are needed at several occasions. Hence, we first collect and prove them in this section.

Lemma 3.1. ([9], 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 lemma is equivalent to Theorem 3.1 of [9]. 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. $\square$

Lemma 3.2. 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 $\mu_0 \xrightarrow{\sigma_{[1]}} \mu_1 \xrightarrow{\sigma_{[2]}} \ldots \xrightarrow{\sigma_{[k]}} \mu_k$. 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. $\square$

The next lemma states that loops of cf-PNs can be decomposed into subloops with certain nice properties.

Lemma 3.3. Let $\Phi \in \mathbb{N}^m$ 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, for each $i \in [k]$, ϑ_i denote the Parikh vector with $\vartheta_i(t) = \Phi(t)$ for all $t \in C_i$ and $\vartheta_i(t) = 0$ for all $t \notin C_i$. 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, and C_1 is the only top component of $\Phi[\vartheta_1]$. Therefore, ϑ as defined above always exists. 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 $\Phi_1, \ldots,$ Φ_k as given in the lemma.

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 $\mathcal{P}[\vartheta_C]$. The Parikh vector ϑ_C must be a loop since otherwise $\Phi - \Phi_1$ would not 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 arc.) 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 is not chosen maximally since $\vartheta + \vartheta_C$ is larger with respect to the above properties, a contradiction. $\square$

An example for this decomposition is illustrated in Figure 1. Using these observations, we can show the following lemma.

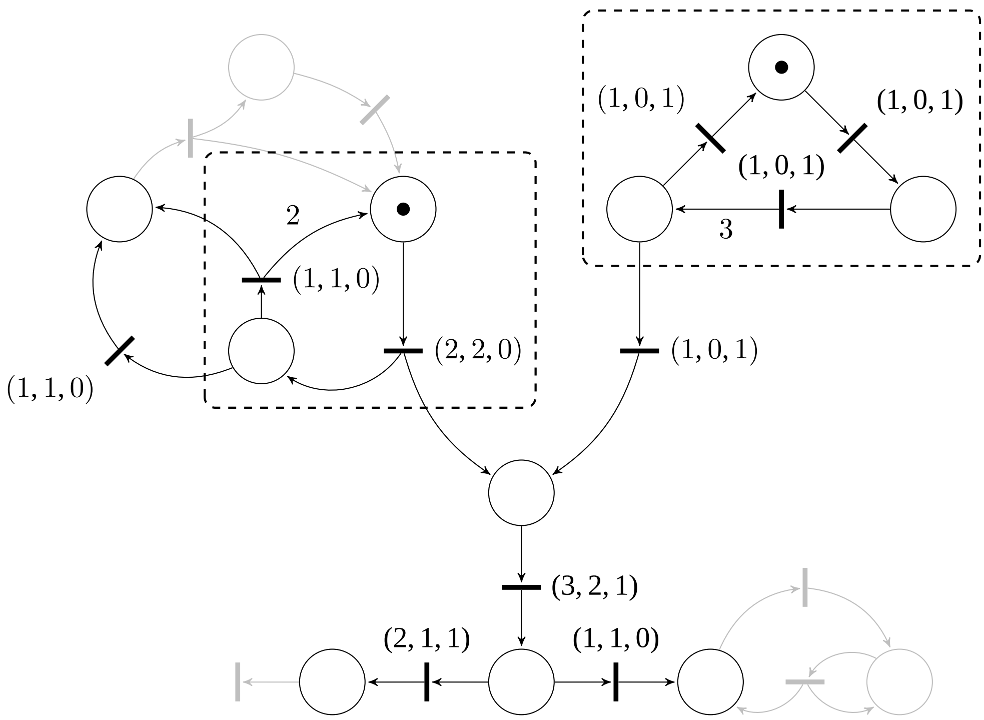

Figure 1. This figure illustrates the loop decomposition of Lemma 3.3 applied to a Petri net $\mathcal{P}$ and a loop Φ. Places and transitions that are not part of $\mathcal{P}[\Phi]$ are shown in gray. The top components of $\mathcal{P}[\Phi]$ are surrounded by dashed rectangles. The labels of the transitions t are triples $(\Phi(t), \Phi_1(t), \Phi_2(t))$, where Φ_1 and Φ_2 are the subloops of the decomposition of $\Phi = \Phi_1 + \Phi_2$. 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.

Lemma 3.4. 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 transitions sequences $\sigma_1, \ldots, \sigma_{k+1}$, loops $\vartheta_1, \ldots, \vartheta_k$, $k \leq n$, and a numbering of the top components of $\mathcal{P}$ 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 $\mathcal{P}[\vartheta]$, 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 3.3 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 Lemma 3.1 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 3.2 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 Lemma 3.1, 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. □

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 3.5. 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:
Consider a function $f_R : 2^R \to 2^R$ that takes subsets $S \subseteq R$ to either S if S is a trap or to a proper subset of S containing the maximum trap of S otherwise. Then, the maximum trap Q of R is the (unique) greatest fixed point of f_R. In particular, we can obtain Q by initializing Q by R and then iteratively setting $Q \leftarrow f_R(Q)$ until Q does not change anymore. We describe a procedure computing Q that uses this approach.

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 would not 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, where we assume w. l. o. g. that $P = \{p_1, \ldots, p_n\}$ and $T = \{t_1, \ldots, t_m\}$. 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 is 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.

We remark that this kind of computation using counters and "reversed" lists to obtain efficient algorithms for computational problems is a well known standard technique (see, e. g., [2, 8]). □

Lemma 3.6. 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 $R_{|R|} \subsetneq \ldots \subsetneq R_1$, 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 do not remove tokens from places outside of R, the displacement of the firing sequence at these places is nonnegative. □

Lemma 3.7. 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*) $\Rightarrow$ (*b*)": There is nothing to show. "(*b*) $\Rightarrow$ (*c*)": If Q is marked, then $R \supseteq Q$ will always be marked, regardless of the transitions fired. "(*c*) $\Rightarrow$ (*a*)": Notice that $R \setminus Q$ does not 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 3.6, 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 do not exist in $\mathcal{P}'$. Hence, the marking μ reached by σ in $\mathcal{P}$ satisfies the properties of (*a*). □

Lemma 3.8. 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 3.5, we find in linear time the maximum trap Q of R and check if it is empty. By Lemma 3.7, this is the case if and only if R can be emptied. $\Box$

These observations are already sufficient to show that the zero-reachability problem which, in general, is as hard as the reachability problem (see [12]), is decidable in linear time for cf-PNs.

Theorem 3.9. Given a cf-PN $\mathcal{P} = (P, T, F, \mu_0)$, we can decide in linear time if the zero marking $\vec{0}$ is reachable.

Proof:
We simply apply Lemma 3.8 to $\mathcal{P}$ and P. $\Box$

4. Boundedness Problems for Communication-Free Petri Nets

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 4.1. 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 [23] (which states that if two sets are simultaneously ω-unbounded on the same sequence σ, then their union is simultaneously ω-unbounded on a subsequence of σ) 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 universal 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 various problems of CFCGs in a very illustrative way.

4.1. Concepts of Non-Simultaneous 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.

Lemma 4.2. 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 $\Phi_1, \Phi_2, \ldots$ such that $\Delta(\Phi_i)(p) < \Delta(\Phi_{i+1})(p)$, $i \in \mathbb{N}_{>0}$. Consider the induced infinite sequence of vectors such that the i-th vector is $(\Phi_i, \mu_0 + \Delta(\Phi_i)) \in \mathbb{N}^{m+n}$. By the well-known Dickson's Lemma [7], there are $i_1, i_2 \in \mathbb{N}_{>0}$ with $i_1 < i_2$ and $(\Phi_{i_1}, \mu_0 + \Delta(\Phi_{i_1})) \leq (\Phi_{i_2}, \mu_0 + \Delta(\Phi_{i_2}))$ (also see, e. g., Lemma 4.1. of [20]). 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 3.4 to Φ, ϑ and some firing sequence σ having Parikh image Φ. Let σ_1, …, σ_{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 Lemma 3.1 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 $t \in C_2^\tau$ 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 does not 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 total number of tokens at C_2^τ, a contradiction to τ being a loop (remember that we are in the case where no transition of C_2^τ increases the total number of tokens at C_2^τ). Therefore, $t^\bullet \cap C_2^\tau \neq \emptyset$, and we obtain $\sum_{q \in t^\bullet \cap (C_2^\tau \cup C_3^\tau)} F(t, q) \geq 2$. Now, let C_i for $i \in [2, 4]$ be the SCC of $\mathcal{P}$ containing C_i^τ, and observe that C_1, …, 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 2. It is well known that, in general, a Petri net is unbounded if and only if there is a reachable marking μ and a positive loop τ enabled at μ (see [20]). By Lemma 4.2, an analogue observation can be made for single places of a cf-PN. In general Petri nets, however, the latter is not true. In Petri net (*b*) of Figure 2, 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 [23]) this concept does not 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 2.

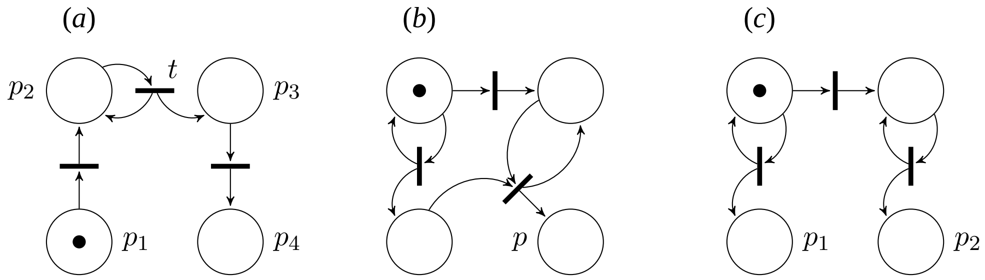

Figure 2. The cf-PN (*a*) illustrates Lemma 4.2, where p_4 is unbounded since $C_1 := \{p_1\}$, $C_2 := \{p_2, t\}$, $C_3 := \{p_3\}$, and $C_4 := \{p_4\}$ satisfy the properties of the lemma. In Petri net (*b*), place p is unbounded but not ω-unbounded. In cf-PN (*c*), $\{p_1,p_2\}$ is simultaneously unbounded but not simultaneously ω-unbounded.

We can use the characterization provided by Lemma 4.2 to give efficient algorithms for certain boundedness problems.

Theorem 4.3. Given a cf-PN $\mathcal{P} = (P, T, F, \mu_0)$, we can find in linear time all places that are (ω-) unbounded.

Proof:
Using Tarjan's modified depth-first search [34], 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 4.2. (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, which asks if all places of a given cf-PN are bounded.

Corollary 4.4. The boundedness problem for cf-PNs is decidable in linear time.

Proof:
We use Theorem 4.3 to check if there is no unbounded place. □

Theorem 4.5. 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 of $\mathcal{P}'$, 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 does not contain transitions incident to Q. μ is also reachable in $\mathcal{P}$ by σ. By Lemma 3.7, 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 3.5 and Theorem 4.3, finding Q, checking if Q is empty, and computing $\mathcal{P}'$ and U' can be performed in linear time. □

We remark that it was shown in [22] that boundedness of BPPs can be decided in polynomial time. Our results imply a linear time algorithm for all BPPs in *standard form* (see [5]).

Theorem 4.6. 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 4.2 and Theorem 4.3. 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 4.2. To find these SCCs in linear time, we use a similar scheme as in the proof of Theorem 4.3. 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 4.2, 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 3.7, we only have to determine if the maximum trap of R is marked. By Lemma 3.5, this can be done in linear time. □

A simple characterization of universally unbounded places is also possible.

Lemma 4.7. 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:
"⇒": We prove the contrapositive, i. e., we show that p is not universally unbounded if (*a*), (*b*) or (*c*) does not hold. If p is not persistently unbounded, then, by definition, p is not universally unbounded. Hence, assume that (*a*) holds. Now assume that (*b*) does not 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.

"⇐": 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 an infix $\sigma_{[i+1..j]}$ 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 arc.) 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 4.8. 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 4.7. Property (*a*) can be checked in linear time using the algorithm described at Theorem 4.6. Property (*b*) can trivially be checked in linear time. Now we show, how we can test for property (*c*). Let $\mathcal{P}'$ be the cf-PN resulting from $\mathcal{P}$ by removing all transitions $t \in {}^\bullet p$. A cycle of $\mathcal{P}$ does not 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 universal 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: Their token numbers can never decrease. This does not hold in general. Figure 3 illustrates such an example. An interesting 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?

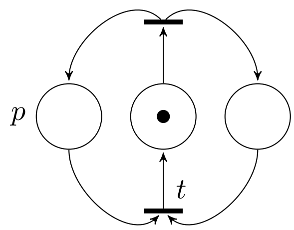

Figure 3. p is universally unbounded but $\Delta(t)(p) < 0$.

4.2. Simultaneous 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 simultaneous unboundedness, the simultaneous ω-unboundedness, and the covering problems for cf-PNs.

Definition 4.9.

- Problem `SU`: Given a cf-PN $\mathcal{P} = (P, T, F, \mu_0)$ and a subset $S \subseteq P$ of places, is S simultaneously unbounded?
- Problem `SIU`: Given a cf-PN $\mathcal{P} = (P, T, F, \mu_0)$ and a subset $S \subseteq P$ of places, is S simultaneously ω-unbounded?
- Problem `covering`: Given a cf-PN $\mathcal{P}$ and a marking μ of $\mathcal{P}$, is there a reachable marking $\mu' \geq \mu$?

In order to prove the next theorems, we need some observations. The following lemma states that each loop of a Petri net can be decomposed into subloops with small sums of components. Note that, in contrast to Lemma 3.3, this lemma is valid for every Petri net.

Lemma 4.10. Let $N = (P, T, F)$ be a Petri net with n places and m transitions, and let W be the largest arc multiplicity of N. Then, there is a finite set $\mathcal{H}(N) = \{\Phi_1, \ldots, \Phi_k\} \subsetneqq \mathbb{N}^m$ of loops of N such that each loop of $\mathcal{H}(N)$ consists of at most $(1 + (n + m)W)^n$ transitions, and such that, for each loop Φ of N, there are $a_1, \ldots, a_k \in \mathbb{N}$ 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 system matrix of $\mathcal{P}$, i. e., the i-th column of D equals $\Delta(t_i)$. Consider the system $D\Phi \geq \vec{0}$ of linear diophantine inequalities. The set $L \subsetneqq \mathbb{N}^m$ of nontrivial nonnegative integral solutions of this system equals the set of loops having at least one transition. Now, consider the system $(D, -I_n)y = 0$ having the set $L' \subsetneqq \mathbb{N}^{m+n}$ of nontrivial nonnegative integral solutions where I_n is the $n \times n$-identity matrix.

By Theorem 1 of [32], this system has a set $\mathcal{H}(D, -I_n)$ of minimal solutions (called the Hilbert basis) with the following properties:

(*a*) each element of L' is a linear combination of the elements of $\mathcal{H}(D, -I_n)$ with nonnegative integral coefficients, and

(*b*) each element of $\mathcal{H}(D, -I_n)$ has a component sum of at most $(1 + (m + n)W)^n$.

Let $y \in L'$, and a and b be the projection of y onto the first m and the last n components, respectively. We observe that a is a loop of N with $\Delta(a) = b$. Moreover, the set of projections of the elements of L' onto the first m components equals L. The properties (*a*) and (*b*) for L' imply analogue properties for L. This concludes the proof. □

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 [35]. 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.)

Lemma 4.11. 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 $\|\mu_0\|_1 \cdot n(nW)^n + m \cdot (1+(n+m)W)^n \leq c^{\text{size}(\mathcal{P})}$,

(*c*) each τ_i, $i \in [\ell]$, is a positive loop of length at most $(1 + (n + m)W)^n$, 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 4.10 to (P, T, F) and ϑ, and obtain, for some k (and an appropriate naming of the elements of $\mathcal{H}(P, T, F)$), coefficients $a_i \in \mathbb{N}_{>0}$ 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 $\|\vartheta_i\|_1 \leq (1 + (n + m)W)^n$ and cannot be decomposed into nontrivial loops any further. Note that, by this and Lemma 3.3, $\mathcal{P}[\vartheta_i]$, $i \in [k]$, 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 $\mu_0 + \Delta(\Phi) \geq \vec{0}$, Lemma 3.1 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 $|\bar{\xi}| \leq \|\Phi\|_1 + m \cdot \max_{j\in[r]}|\vartheta_{i_j}| \leq \|\mu_0\|_1 \cdot n(nW)^n + m \cdot (1 + (n + m)W)^n$.

By Lemma 3.1 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. $\square$

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 4.12. `SU`, `SIU`, and `covering` are **NP**-complete, even if we restrict the input 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:
For two problems A and B let $A \prec_{\log} B$ denote the existence of a logspace many-one reduction from A to B. We first show the **NP**-hardness of `SU` and `SIU` by showing `3-SAT` $\prec_{\log}$ `SU` and `3-SAT` $\prec_{\log}$ `SIU`. 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 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 4. A similar reduction was used by Esparza [9] to show the **NP**-hardness of the reachability problem for cf-PNs. Note that the cf-PN produced by this reduction satisfies the additional constraints ("even if...") of the lemma.

Next, we show that `covering` $\in$ **NP** by reducing `covering` in logspace to the reachability problem. We obtain a cf-PN $\bar{\mathcal{P}}$ from $\mathcal{P}$ by adding, for each $p \in P$, a transition t_p having $F(p, t_p) = 1$. The new transitions can be used to remove an arbitrary number of tokens from any place. We observe that a marking μ can be covered in $\mathcal{P}$ if and only if μ can be covered in $\bar{\mathcal{P}}$ if and only if μ is reachable in $\bar{\mathcal{P}}$.

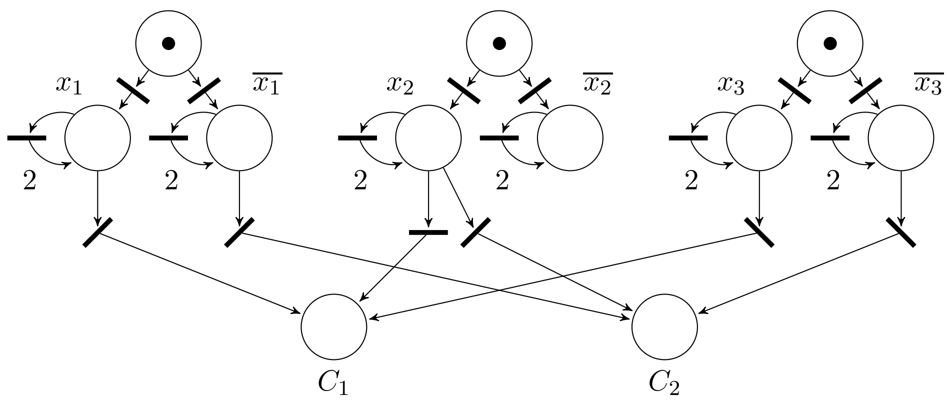

Figure 4. The formula $C_1 \wedge C_2 := (x_1 \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3})$ can be satisfied if and only if $\{C_1, C_2\}$ is simultaneously (ω-)unbounded.

Now, we show that SU $\in$ **NP** and that `covering` is **NP**-hard by showing SU $\prec_{\log}$ `covering`, even if the input for `covering` satisfies the additional constraints of the lemma. Let $\mathcal{P}$ be part of the input for SU, and let $\bar{\mathcal{P}}$ be as defined above. Let W be the largest arc multiplicity of $\bar{\mathcal{P}}$. Define the marking μ by $\mu(p) = \mu_0(p) + c^{\text{size}(\bar{\mathcal{P}})}W + 1$ if $p \in S$, and $\mu(p) = 0$ otherwise, where c is the constant of Lemma 4.11. Note that μ has polynomial encoding size (in particular, size$(\bar{\mathcal{P}})$ is polynomial in size$(\mathcal{P})$).

Assume that S is simultaneously unbounded in $\mathcal{P}$. Then μ can be covered in $\bar{\mathcal{P}}$. Now, assume that μ can be covered in $\bar{\mathcal{P}}$, i. e., μ is reachable in $\bar{\mathcal{P}}$. In accordance with Lemma 4.11, 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 $\bar{\mathcal{P}}$ leading from μ_0 to μ. Then, $|\bar{\xi}| \le c^{\text{size}(\bar{\mathcal{P}})}$ which implies that each prefix of $\bar{\xi}$ can increase the number of tokens at a place by at most $c^{\text{size}(\bar{\mathcal{P}})}W$. 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 $\bar{\nu}$ of $\bar{\mathcal{P}}$ 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 $\bar{\nu}(p) \ge (\mu_0 + \Delta(\bar{\xi}))(p) + k \ge k$ for all $p \in S$ (since $\mu_0 + \Delta(\bar{\xi}) \ge \vec{0}$). By removing all occurrences of the new transitions (which can only remove tokens from the net, and are part of $\bar{\mathcal{P}}$ 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 $\nu(p) \ge k$ for all $p \in S$. Therefore, S is simultaneously unbounded in $\mathcal{P}$. Note that `covering` is **NP**-hard even if we only allow inputs satisfying the additional constraints of this lemma since the reduction from 3-SAT to SU given above produces cf-PNs $\mathcal{P}$ satisfying these constraints, implying that also $\bar{\mathcal{P}}$ satisfies them.

It remains to be shown that SIU $\in$ **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 4.2. 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 $\Delta(\sigma_k)(p) \ge k$ 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 $\mu_0 + \Delta(\sigma_{i_k}) \le \mu_0 + \Delta(\sigma_{i_{k+1}})$ for all $k \in \mathbb{N}_{>0}$. 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 $k \in \mathbb{N}_{>0}$ 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 4.10 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 the encoding size of Φ^* is polynomial since each subloop Φ_i contains at most $(1 + (n+m)W)^n$ transitions (i. e., an at most exponential amount), and there are at most $((1 + (n+m)W)^n + 1)^n$ of such subloops (i. e., again an at most exponential amount).

Since $\mathcal{P}[\Phi] = \mathcal{P}[\Phi^*]$, Lemma 3.1 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 algorithm is correct since μ^* and Φ^* as defined above together with a certificate of polynomial length that μ^* can be covered is a certificate of polynomial length that S is simultaneously ω-unbounded. 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 [15] for conflict-free Petri nets.

5. Liveness Problems for Communication-free Petri Nets

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

Definition 5.1. 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 is not 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 $L_0, \ldots, L_4$-liveness are referred to in [31]. L_5-liveness is to our knowledge a new concept, corresponding to our new notion of universally unboundedness. When modeling a system by a Petri net where transitions correspond to actions within the system, an L_5-live transition would correspond to an action that must necessarily be executed over and over again when executing actions of the system. A test for L_5-liveness for specific transitions could be used to ensure that a process like, for instance, a garbage collecting action is periodically executed in the system.

Notice, that L_i-liveness implies L_j-liveness, where $5 \geq i \geq j \geq 1$. Using the results of Section 4, we can efficiently solve many decision problems involving these notions of liveness. We use the Parikh extensions of cf-PNs to carry out the reductions.

Definition 5.2. (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 $F^\Psi(t, p_t^\Psi) = 1$.

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., $\mu(p_t^\Psi) = \Psi(\sigma)(t)$. More precisely, the projection of μ onto the new places equals $\Psi(\sigma)$, while the projection of μ onto the places of the original net $\mathcal{P}$ equals the marking that is reached by firing σ in $\mathcal{P}$. We remark that the concept of the Parikh extension is closely related to the concept of extended Parikh maps used in [23] and [15] for persistent and conflict-free Petri nets.

Theorem 5.3. 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 $\mathcal{P}$. 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 Lemma 3.1). 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 4.3. 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 4.2) iff t_t is L_3-live. Hence, we simply apply the algorithm of Theorem 4.3 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 5.4. 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 4.6 and Theorem 4.8 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 [27] 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 [5]). 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 5.5. The partial deadlock reachability problem, i. e., the problem of deciding, given a cf-PN $\mathcal{P} = (P, T, F, \mu_0)$ and a set $T' \subseteq T$ of transitions, if there is a reachable marking μ at which no transition of T' is enabled, is decidable in linear time.

Proof:
By Lemma 3.8, we can check in linear time if there is a marking at which $R := \bigcup_{t \in T'} {}^\bullet t$ is empty. □

Theorem 5.6. The partial livelock reachability problem, i. e., the problem of deciding, given a cf-PN $\mathcal{P} = (P, T, F, \mu_0)$ and a set $T' \subseteq T$ of transitions, if there is a reachable marking μ such that no marking reachable from μ enables a transition $t \in T'$, is decidable in linear time.

Proof:
We introduce a counting place p and an arc 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 4.6, this can be decided in linear time. □

6. 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. In turn, we use results for the equivalence problem of context-free commutative grammars to obtain results for the corresponding problem of cf-PNs.

We assume that the reader is familiar with the basic concepts of context-free grammars (CFGs). Commutative grammars were introduced by Crespi-Reghizzi and Mandrioli [6] 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 [17]. A *commutative grammar* (CG) 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. A CG is a context-free CG (CFCG) if all productions are of the form $A \to u$ where $A \in V_N$ and $u \in (V_N \cup V_T)^{\circledast}$.

If $V_N = \{v_1, \ldots, v_p\}$ and $V_T = \{u_1, \ldots, u_q\}$, then, by fixing an order on the symbols, a *commutative word* $w \in (V \cup T)^{\circledast}$ can be identified with vectors of $\mathbb{N}^{p+q}$, and express productions in terms of such vectors. Productions and commutative words are succinctly encoded. For two words $w_1, w_2 \in \mathbb{N}^{p+q}$, we write $w_1 \Rightarrow w_2$ if there is a production $(a, b) \in \mathbb{N}^{2(p+q)}$ with $w_1 - a \geq \vec{0}$ and $w_1 - a + b = w_2$. The language $L(G)$ of a CG G is $\{w \in V_T^{\circledast} \mid s \stackrel{*}{\Rightarrow} w\}$, where $\stackrel{*}{\Rightarrow}$ is the transitive closure of $\Rightarrow$. In context of the fixed order of terminal symbols, $L(G)$ can be recognized as a subset of $\mathbb{N}^q$.

We now define the *canonical Petri net* of a CFCG $G = (V_N, V_T, P, s)$. 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 $F(p_i, t) = k_i$, $F(t, p_i) = \ell_i$, and $F(t, q_j) = \ell'_j$. The initial marking μ_0 of the canonical Petri net 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 by the grammar, and not only those consisting only of terminal symbols. The canonical Petri net of a CFCG or a CFG is a cf-PN.

Similarly, we can define the *canonical CFCG G* of a cf-PN $\mathcal{P}$, where each place is represented by both a variable and a terminal symbol. Each transition is represented by a production, and the initial marking is represented by the corresponding commutative word. Additionally, for each place there is a transition which replaces the corresponding variable by the corresponding terminal symbol. For this grammar, we observe $L(G) = \mathcal{R}(\mathcal{P})$.

We remark that the results of this section for problems of CFCG can also be obtained by making use of techniques and results of the rich theory on CFGs (see, e. g., [33]). However, one must be careful since the encoding of productions of CFCGs is succinct, i. e., the encoding size of productions is logarithmic in the number of symbols on each side of the production, where, for CFGs, the encoding size is at least linear in the number of symbols. Any reduction or argument must take this difference between (non-commutative) grammars and commutative grammars into consideration. Using the formalism of Petri nets, we can easily bypass this obstacle and provide efficient and illustrative algorithms for our problems of interest.

Huynh [18] showed that the equivalence problem of CFCGs (which asks if two CFCGs produce the same language) is in **coNEXPTIME**, the complement of **NEXPTIME**. Using the canonical CFCGs for two input cf-PNs, we can reduce the equivalence problem of cf-PNs in logarithmic space to that problem, and thus obtain the same upper bound. Together with the result of Yen [36] that the equivalence problem of cf-PNs is $\mathbf{\Pi}_2^p$-hard, we obtain the following result.

Corollary 6.1. The equivalence problem of cf-PNs, i. e., the problem of deciding, given two cf-PNs $\mathcal{P}_1$ and $\mathcal{P}_2$, if $\mathcal{R}(\mathcal{P}_2) = \mathcal{R}(\mathcal{P}_2)$ holds, is $\mathbf{\Pi}_2^p$-hard and in **coNEXPTIME**.

Esparza et al. [10] 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 [14] 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 6.2. 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 $\mathcal{P} = (S, T, F, \mu_0)$ be the canonical cf-PN of G, 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 4.5, 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 [14] for this problem is that it does not require the grammar being in Chomsky normal form. In [10], the authors also provided linear time algorithms for the emptiness problem and the problem of finding nullable variables of context-free grammars (also see [33]). Our results provide alternative linear time algorithms for these problems as well as for corresponding problems of context-free *commutative* grammars.

Theorem 6.3. Given a context-free commutative or non-commutative grammar $G = (V_N, V_T, P, s)$, we can decide in linear time if $L(G)$ is empty.

Proof:
Let $\mathcal{P} = (S, T, F, \mu_0)$ be the canonical cf-PN of G. Then, $L(G) = \emptyset$ if and only if each word produced by the grammar contains a variable if and only if S_{V_N} cannot be emptied. By Lemma 3.8, this can be checked in linear time. □

Theorem 6.4. Given a context-free commutative or non-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 $\mathcal{P} = (S, T, F, \mu_0)$ be the canonical cf-PN of G, Q be the maximum trap of S, and let μ_v for $v \in V_N$ denote the marking that contains exactly one token at the 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, T, F, μ_v). By Lemma 3.7, this is the case if and only if $p_v \notin Q$. By Lemma 3.5, we can compute Q in linear time. After that, we simply collect all variables that correspond to places outside of Q. □

7. Conclusion

We investigated several boundedness and liveness problems for cf-PNs. For some of them, as well as for the covering problem, we showed **NP**-completeness. For the remaining problems, we achieved polynomial time algorithms, most of them even having linear time. Using our results for cf-PNs, we gave linear time algorithms for several problems in related areas (BPPs, CFGs and CFCGs). Conversely, we used a result on CFCGs to give a **coNEXPTIME**-upperbound for the equivalence problem of cf-PNs. Many open problems and directions for future research still exist. We name just a few of them.

- The lower bound ($\mathbf{\Pi}_2^p$-hardness) and the upper bound (**coNEXPTIME**) of the equivalence problem for cf-PNs or CFCGs do not match. How can the gap be closed? Are these problems complete for some known complexity class? A similar question applies to the equivalence problem of cf-PNs with a fixed number of places.

- Under which restrictions of the set S are the problems SU and SIU decidable in polynomial time?
- How do the complexities of the reachability and the problems considered in this paper behave in case of different generalizations and extensions of cf-PNs? One instance of such an extension are cf-PNs where the arcs from places to transitions are allowed to be inhibitor arcs. This class was first considered in [3]. However, an upper bound for the reachability problem was found only for a restricted subclass.

Acknowledgments. We thank Javier Esparza for many helpful discussions, as well as Hsu-Chun Yen for sending us his proof for the lower bound of the equivalence problem. We are grateful to the anonymous referees whose comments greatly helped us to improve the presentation of this paper.

References

[1] Alimonti, P., Feuerstein, E., Nanni, U.: 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)*, vol. 583 of *Lecture Notes in Computer Science*, Springer, 1992, 1–14.

[2] Arnold, A., Crubille, P.: A linear algorithm to solve fixed-point equations on transition systems, *Information Processing Letters*, **29**(2), 1988, 57–66.

[3] Chen, C.-L., Wang, S., Yen, H.-C.: Reachability analysis of variants of communication-free Petri nets., *IEICE Transactions on Information and Systems*, **92**(3), 2009, 377–388.

[4] Christensen, S.: Distributed bisimularity is decidable for a class of infinite state-space systems, in: *Proceedings of the 3rd International Conference on Concurrency Theory (CONCUR'92)*, vol. 630 of *Lecture Notes in Computer Science*, Springer, 1992, 148–161.

[5] Christensen, S., Hirshfeld, Y., Moller, F.: Bisimulation equivalence is decidable for basic parallel processes, in: *Proceedings of the 4th International Conference on Concurrency Theory (CONCUR'93)*, vol. 715 of *Lecture Notes in Computer Science*, Springer, 1993, 143–157.

[6] Crespi-Reghizzi, S., Mandrioli, D.: Commutative grammars, *CALCOLO*, **13**(2), 1976, 173–189.

[7] Dickson, L. E.: Finiteness of the odd perfect and primitive abundant numbers with n distinct prime factors, *American Journal of Mathematics*, **35**(4), 1913, 413–422.

[8] Dowling, W. F., Gallier, J. H.: Linear-time algorithms for testing the satisfiability of propositional horn formulae, *The Journal of Logic Programming*, **1**(3), 1984, 267–284.

[9] Esparza, J.: Petri nets, commutative context-free grammars, and basic parallel processes, *Fundamenta Informaticae*, **31**(1), 1997, 13–25.

[10] Esparza, J., Rossmanith, P., Schwoon, S.: A uniform framework for problems on context-free grammars, *Bulletin of the EATCS*, **72**, 2000, 169–177.

[11] Fribourg, L., Olsén, H.: A decompositional approach for computing least fixed-points of datalog programs with z-counters, *Constraints*, **2**(3-4), 1997, 305–335.

[12] Hack, M.: The recursive equivalence of the reachability problem and the liveness problem for Petri nets and vector addition systems, *IEEE Conference Record of the 15th Annual Symposium on Switching and Automata Theory*, 1974.

[13] Hirshfeld, Y.: Petri nets and the equivalence problem, in: *Selected Papers of the 7th Workshop on Computer Science Logic (CSL'93)*, vol. 832 of *Lecture Notes in Computer Science*, Springer, 1994, 165–174.

[14] Hopcroft, J. E., Ullman, J. D.: *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley, 1979.

[15] Howell, R. R., Rosier, L. E.: Completeness results for conflict-free vector replacement systems, *Journal of Computer and System Sciences*, **37**(3), 1988, 349–366.

[16] Howell, R. R., Rosier, L. E., Yen, H.-C.: An $O(n^{1.5})$ algorithm to decide boundedness for conflict-free vector replacement systems, *Information Processing Letters*, **25**(1), 1987, 27–33.

[17] Huynh, D. T.: Commutative grammars: The complexity of uniform word problems, *Information and Control*, **57**(1), 1983, 21–39.

[18] Huynh, D. T.: The complexity of equivalence problems for commutative grammars, *Information and Control*, **66**(1–2), 1985, 103–121.

[19] Jones, N. D., Landweber, L. H., Lien, Y. E.: Complexity of some problems in Petri nets, *Theoretical Computer Science*, **4**(3), 1977, 277–299.

[20] Karp, R. M., Miller, R. E.: Parallel program schemata, *Journal of Computer and System Sciences*, **3**(2), 1969, 147–195.

[21] Kopczyński, E., To, A. W.: Parikh images of grammars: Complexity and applications, *Proceedings of the 25th Annual IEEE Symposium on Logic in Computer Science (LICS)*, July 2010.

[22] Kučera, A.: 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*, vol. 1180 of *Lecture Notes in Computer Science*, Springer, 1996, 111–122.

[23] Landweber, L. H., Robertson, E. L.: Properties of conflict-free and persistent Petri nets, *Journal of the ACM*, **25**(3), 1978, 352–364.

[24] Mayr, E. W.: An algorithm for the general Petri net reachability problem, *Proceedings of the 13th ACM Symposium on Theory of Computing (STOC'81)*, ACM, 1981.

[25] Mayr, E. W., Weihmann, J.: Completeness results for generalized communication-free Petri nets with arbitrary edge multiplicities, in: *Proceedings of the 7th International Workshop on Reachability Problems*, vol. 8169 of *Lecture Notes in Computer Science*, Springer, 2013, 209–221.

[26] Mayr, E. W., Weihmann, J.: Results on equivalence, boundedness, liveness, and covering problems of BPP-Petri nets, in: *Proceedings of the 34th International Conference on Application and Theory of Petri Nets (ICATPN'13)*, vol. 7927 of *Lecture Notes in Computer Science*, Springer, 2013, 70–89.

[27] Mayr, R.: Tableau methods for PA-processes, in: *Proceedings of the 1997 International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX'97)*, vol. 1227 of *Lecture Notes in Computer Science*, Springer, 1997, 276–290.

[28] Mayr, R.: *Decidability and Complexity of Model Checking Problems for Infinite-State Systems*, Ph.D. Thesis, Technische UniversitÃd't MÃijnchen, 1998.

[29] Mayr, R.: On the complexity of bisimulation problems for basic parallel processes, in: *Proceedings of the 27th International Colloquium on Automata, Languages and Programming (ICALP'00)*, vol. 1853 of *Lecture Notes in Computer Science*, Springer, 2000, 329–341.

[30] Milner, R.: *Communication and concurrency*, Prentice Hall, 1995.

[31] Murata, T.: Petri nets: Properties, analysis and applications, *Proceedings of the IEEE*, **77**(4), 1989, 541–580.

[32] Pottier, L.: Minimal solutions of linear diophantine systems: bounds and algorithms, in: *Proceedings of the 4th International Conference on Rewriting Techniques and Applications (RTA'91)*, vol. 488 of *Lecture Notes in Computer Science*, Springer, 1991, 162–173.

[33] Sippu, S., Soisalon-Soininen, E.: *Parsing Theory*, vol. 15 of *EATCS Monographs on Theoretical Computer Science*, Springer, 1988.

[34] Tarjan, R.: Depth-first search and linear graph algorithms, *SIAM Journal on Computing*, **1**(2), 1972, 146–160.

[35] Yen, H.-C.: On reachability equivalence for BPP-nets, *Theoretical Computer Science*, **179**, 1997, 301–317.

[36] Yen, H.-C.: *Private Communication*, 2013.