



Fixed Parameter Tractability and Polynomial Time Results for the Synthesis of b -bounded Petri Nets

Ronny Tredup^(✉)

Institut für Informatik, Theoretische Informatik, Universität Rostock,
Albert-Einstein-Straße 22, 18059 Rostock, Germany
ronny.tredup@uni-rostock.de

Abstract. Synthesis for a type τ of Petri nets is the problem of finding, for a given transition system (TS, for short) A , a Petri net N of this type whose state graph is isomorphic to A if such a net exists. The decision version of this search problem, called τ -feasibility, asks if, for a given TS A , there exists a Petri net N of type τ with a state graph isomorphic to A . In this case, A is called τ -feasible. A 's feasibility is equivalent to fulfilling two so-called *separation properties*. In fact, a transition system A is τ -feasible if and only if it satisfies the type related *state separation property* (SSP) and *event state separation property* (ESSP). Both properties, SSP and ESSP, define decision problems. In this paper, we introduce for $b \in \mathbb{N}$ the type of restricted \mathbb{Z}_{b+1} -extended b -bounded P/T-nets and show that synthesis and deciding ESSP and SSP for this type is doable in polynomial time. Moreover, we demonstrate that, given a TS A , deciding if A has the SSP can be done in polynomial time for the types of (pure) \mathbb{Z}_{b+1} -extended b -bounded P/T-nets. Finally, we exhibit that deciding if a TS A is feasible or has the ESSP for the types of (pure) \mathbb{Z}_{b+1} -extended b -bounded P/T-nets is *fixed parameter tractable* if the number of occurrences of events is considered as parameter.

1 Introduction

Synthesis for a Petri net type τ (τ -synthesis, for short) is the task to find for a transition system A a Petri net N of type τ (τ -net, for short) with a state graph isomorphic to A . The associated decision version, which we call feasibility for τ (τ -feasibility, for short), asks whether there is a corresponding τ -net for the input A . If such a net exists then we call A τ -feasible. A 's feasibility is equivalent to fulfilling two so-called separation properties. More exactly, A is τ -feasible if and only if it has the *state separation property* and the *event state separation property* for τ (τ -SSP and τ -ESSP, for short) [5]. Both, τ -SSP and τ -ESSP, define decision problems asking whether the input A has the τ -SSP or the τ -ESSP, respectively.

Petri net synthesis has been investigated for many years and is applied in numerous fields. It yields implementations which are correct by design and allows

extracting concurrency and distributability information from sequential specifications as transition systems and languages [6,7]. Further application areas of Petri net synthesis currently cover, among others, the reconstruction of a model from its execution traces (process discovery), supervisory control for discrete event systems and the synthesis of speed-independent circuits [1,9,13].

A type of Petri nets is called *bounded* if there is a positive integer b which is not exceeded by the number of tokens on any place in every reachable marking. This paper deals with the computational complexity of synthesis, feasibility, SSP and ESSP for b -bounded Petri net types, that is, bounded Petri nets where b is predetermined.

In [3,5], Badouel et al. showed that Synthesis, feasibility, SSP and ESSP for the type of bounded and pure bounded place/transition nets (P/T-nets, for short) are solvable in polynomial time if no bound b is preselected. On the contrary, SSP, ESSP and feasibility are NP-complete for pure 1-bounded P/T-nets [4,12]. This remains true even for strongly restricted input transition systems [19,21]. In [18], we showed that feasibility, SSP and ESSP are NP-complete for (pure) b -bounded P/T-nets for arbitrary $b \in \mathbb{N}^+$.

In [16], Schmitt advanced the pure 1-bounded P/T-net type by an interaction between places and transitions simulating addition of integers modulo 2. This brings the complexity of synthesis, feasibility, ESSP and SSP for the resulting pure \mathbb{Z}_2 -extended 1-bounded P/T-nets down to polynomial time. On the contrary, we proved in [18] that extending (pure) b -bounded P/T-nets by \mathbb{Z}_{b+1} yields no tractable type if $b \geq 2$. In particular, feasibility and ESSP for (pure) \mathbb{Z}_{b+1} -extended b -bounded P/T-nets, $b \geq 2$, are NP-complete. We continued research on the impact of interactions on the computational complexity of synthesizing 1-bounded Petri nets in [20]. Here, we investigated 43 1-bounded types purely defined by interactions which they have or not. While for 37 of them synthesis is tractable, feasibility and ESSP for the remaining 7 are NP-complete.

Results of [2,8] show that putting restrictions on the sought nets's (syntactical) structure can have a positive impact on the complexity of synthesis. In particular, in [2], Agostini et al. proposed a polynomial time synthesis algorithm for Free-Choice Acyclic pure 1-bounded P/T-nets having applications in workflow models. Moreover, in [8], Best et al. showed that it suffices to check certain structural properties of the input A if the sought net is a pure b -bounded live marked graph. Whether A has these properties or not is decidable in polynomial time [14].

In this paper, we examine whether there are also types of b -bounded P/T-nets for which synthesis is tractable if $b \geq 2$. We affirm this question and propose the restricted \mathbb{Z}_{b+1} -extended b -bounded P/T-nets, $b \in \mathbb{N}$. This paper shows, that synthesis, feasibility, ESSP and SSP are solvable in polynomial time for this type. Moreover, our results prove that deciding whether a transition system (TS, for short) A has the SSP for the types of (pure) \mathbb{Z}_{b+1} -extended b -bounded P/T-nets, $b \in \mathbb{N}$, is also doable in polynomial time. Notice, that this discovers the first Petri net type where the provable computational complexity of SSP is different to ESSP and feasibility.

To decide whether a TS A is τ -feasible or has the τ -ESSP, where τ corresponds to (pure) \mathbb{Z}_{b+1} -extended b -bounded P/T-nets, $b \geq 2$, is NP-complete [18]. Hence, this problem is considered inherently hard to solve algorithmically. Consequently, one expects that every corresponding decision algorithm has an exponential running time if complexity is measured in terms of the input size of A only. In this paper, we analyze the computational complexity of feasibility and ESSP for these types in finer detail. To do so, we apply parameterization, a typical approach of modern complexity theory to tackle hard problems. The running time of parameterized algorithms is not only expressed in the input's size, but it also takes the parameters into account. The number k of *occurrences of events*, the maximum number of different transitions at which an event occur, is one of the most obvious parts of a TS which can be considered as a parameter. We show that feasibility and ESSP related to the types of (pure) \mathbb{Z}_{b+1} -extended b -bounded P/T-nets are only exponential in the size of k while polynomial in the size of the input. Hence, both problems are *fixed parameter tractable* if k is considered as parameter. This result could not be foreseen with certainty. In fact, in [19], we showed that feasibility, ESSP and SSP remain NP-complete for pure 1-bounded P/T-nets even if every event occurs at most *twice*. Hence, related to pure 1-bounded P/T-nets, these problems parameterized by k are not fixed parameter tractable as long as $P \neq NP$.

2 Preliminaries

A *transition system* (TS for short) $A = (S, E, \delta)$ consists of finite disjoint sets S of states and E of events and a partial *transition function* $\delta : S \times E \rightarrow S$. Usually, we think of A as an edge-labeled directed graph with node set S where every triple $\delta(s, e) = s'$ is interpreted as an e -labeled edge $s \xrightarrow{e} s'$, called *transition*. We say that an event e *occurs* at state s if $\delta(s, e) = s'$ for some state s' and abbreviate this with $s \xrightarrow{e}$. This notation is extended to words $w' = wa$, $w \in E^*$, $a \in E$ by inductively defining $s \xrightarrow{\varepsilon} s$ for all $s \in S$ and $s \xrightarrow{w'} s''$ if and only if $s \xrightarrow{w} s'$ and $s' \xrightarrow{a} s''$. If $w \in E^*$ then $s \xrightarrow{w}$ denotes that there is a state $s' \in S$ such that $s \xrightarrow{w} s'$. An *initialized* TS $A = (S, E, \delta, s_0)$ is a TS with an initial state $s_0 \in S$ where every state is *reachable*: $\forall s \in S, \exists w \in E^* : s_0 \xrightarrow{w} s$. The language of A is the set $L(A) = \{w \in E^* \mid s_0 \xrightarrow{w}\}$. In the remainder of this paper, if not explicitly stated otherwise, we assume all TSs to be initialized and we refer to the components of an (initialized) TS A consistently by $A = (S_A, E_A, \delta_A, s_{0,A})$.

The following notion of *types of nets* has been developed in [5]. It allows us to uniformly capture several Petri-net types in one general scheme. Every introduced Petri-net type can be seen as an instantiation of this general scheme. A type of nets τ is a TS $\tau = (S_\tau, E_\tau, \delta_\tau)$ and a Petri net $N = (P, T, f, M_0)$ of type τ , τ -net for short, is given by finite and disjoint sets P of places and T of transitions, an initial marking $M_0 : P \rightarrow S_\tau$, and a flow function $f : P \times T \rightarrow E_\tau$. The meaning of a τ -net is to realize a certain behavior by cascades of firing transitions. In particular, a transition $t \in T$ can fire in a marking $M : P \rightarrow S_\tau$

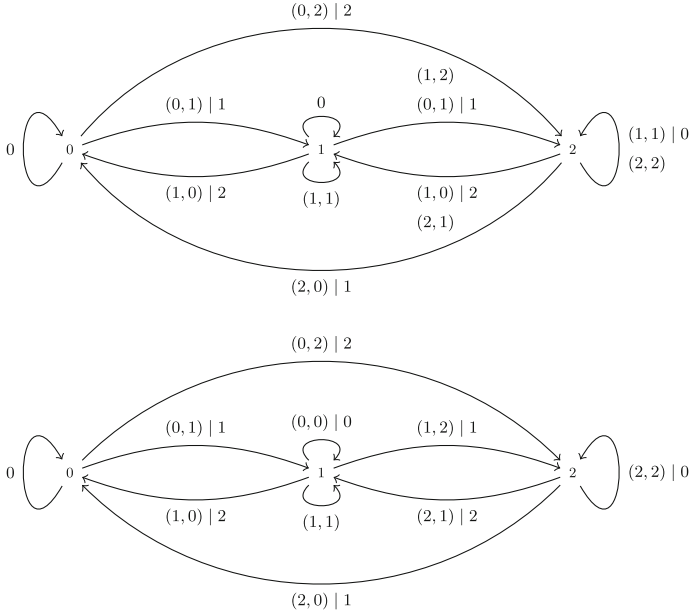


Fig. 1. Top: The type τ_2^2 . Edges with several labels represent different transitions. Discarding the (1, 1), (1, 2), (2, 1) and (2, 2) labeled transitions yields τ_3^2 . Bottom: The type τ_4^2 .

and thereby produces the marking $M' : P \rightarrow S_\tau$ if for all $p \in P$ the transition $M(p) \xrightarrow{f(p,t)} M'(p)$ exists in τ . This is denoted by $M \xrightarrow{t} M'$. Again, this notation extends to sequences $\sigma \in T^*$. Accordingly, $RS(N) = \{M \mid \exists \sigma \in T^* : M_0 \xrightarrow{\sigma} M\}$ is the set of all reachable markings of N . Given a τ -net $N = (P, T, f, M_0)$, its behavior is captured by the TS $A_N = (RS(N), T, \delta, M_0)$, called the state graph of N , where for every reachable marking M of N and transition $t \in T$ with $M \xrightarrow{t} M'$ the transition function δ of A_N is defined by $\delta(M, t) = M'$. The following types of b -bounded P/T-nets and pure b -bounded P/T-nets build the basis of the announced \mathbb{Z}_{b+1} -extensions:

0. The type of b -bounded P/T-nets is defined by $\tau_0^b = (\{0, \dots, b\}, \{0, \dots, b\}^2, \delta_{\tau_0^b})$ where for $s \in S_{\tau_0^b}$ and $(m, n) \in E_{\tau_0^b}$ the transition function is defined by $\delta_{\tau_0^b}(s, (m, n)) = s - m + n$ if $s \geq m$ and $s - m + n \leq b$, and undefined otherwise.
1. The type of pure b -bounded P/T-nets is a restriction of τ_0^b -nets that discards all events (m, n) from $E_{\tau_0^b}$ where both, m and n , are positive. To be exact, $\tau_1^b = (\{0, \dots, b\}, E_{\tau_0^b} \setminus \{(m, n) \mid 1 \leq m, n \leq b\}, \delta_{\tau_1^b})$, and for $s \in S_{\tau_1^b}$ and $e \in E_{\tau_1^b}$ we have $\delta_{\tau_1^b}(s, e) = \delta_{\tau_0^b}(s, e)$.

Having τ_0^b and τ_1^b , their following \mathbb{Z}_{b+1} -extension allows them to simulate the addition of integers modulo $b + 1$.

2. The type of \mathbb{Z}_{b+1} -extended b -bounded P/T -nets τ_2^b arises from τ_0^b such that, firstly, $E_{\tau_2^b}$ extends the event set $E_{\tau_0^b}$ by the elements $0, \dots, b$ and, secondly, the transition function $\delta_{\tau_2^b}$ extends $\delta_{\tau_0^b}$ by the addition of integers modulo $b+1$. More exactly, $\tau_2^b = (\{0, \dots, b\}, (E_{\tau_0^b} \setminus \{(0,0)\}) \cup \{0, \dots, b\}, \delta_{\tau_2^b})$ where for $s \in S_{\tau_2^b}$ and $e \in E_{\tau_2^b}$ we have that $\delta_{\tau_2^b}(s, e) = \delta_{\tau_0^b}(s, e)$ if $e \in E_{\tau_0^b}$ and $\delta_{\tau_2^b}(s, e) = (s + e) \bmod (b + 1)$ if $e \in \{0, \dots, b\}$.
3. The type of \mathbb{Z}_{b+1} -extended pure b -bounded P/T -nets is defined by $\tau_3^b = (\{0, \dots, b\}, (E_{\tau_1^b} \setminus \{(0,0)\}) \cup \{0, \dots, b\}, \delta_{\tau_3^b})$ where for $s \in S_{\tau_3^b}$ and $e \in E_{\tau_3^b}$ the transition function is given by $\delta_{\tau_3^b}(s, e) = \delta_{\tau_1^b}(s, e)$ if $e \in E_{\tau_1^b}$ and $\delta_{\tau_3^b}(s, e) = (s + e) \bmod (b + 1)$ if $e \in \{0, \dots, b\}$.

The new Petri net τ_4^b type arises as a restriction of τ_2^b :

4. The type of restricted \mathbb{Z}_{b+1} -extended b -bounded P/T -nets $\tau_4^b = (S_{\tau_2^b}, E_{\tau_2^b}, \delta_{\tau_4^b})$, $b \in \mathbb{N}^+$, origins from τ_2^b and has the same state set, $S_{\tau_4^b} = S_{\tau_2^b}$, and the same event set, $E_{\tau_4^b} = E_{\tau_2^b}$, but a restricted transition function $\delta_{\tau_4^b}$. In particular, the transition function $\delta_{\tau_4^b}$ restricts $\delta_{\tau_2^b}$ in way that for $s \in S_{\tau_4^b}$ and $(m, n) \in E_{\tau_4^b}$ we have that $\delta_{\tau_4^b}(s, (m, n)) = \delta_{\tau_2^b}(s, (m, n))$ if $s = m$ and, otherwise, if $s \neq m$ then $\delta_{\tau_4^b}(s, (m, n))$ remains undefined. Hence, every $(m, n) \in E_{\tau_4^b}$ occurs exactly once in τ_4^b . Furthermore, if $(s, e) \in \{0, \dots, b\}^2$ then $\delta_{\tau_4^b}(s, e) = \delta_{\tau_2^b}(s, e)$.

In the remainder of this paper we assume $\tau \in \{\tau_2^b, \tau_3^b, \tau_4^b\}$ and $b \in \mathbb{N}^+$, unless stated otherwise. Notice, that τ_4^1 coincides with Schmitt's type [16]. Figure 1 gives a graphical representation of τ_2^2, τ_3^2 and τ_4^2 . The following notion of τ -regions allows us, on the one hand, to define the type related ESSP and SSP and, on the other hand, to reveal in which way we are able to obtain a τ -net N for a given TS A if it exists. Figure 2 shows examples of all subsequently introduced terms.

If τ is a type of nets then a τ -region of a TS A is a pair of mappings (sup, sig) , where $sup : S_A \rightarrow S_\tau$ and $sig : E_A \rightarrow E_\tau$, such that, for each transition $s \xrightarrow{e} s'$ of A , we have that $sup(s) \xrightarrow{sig(e)} sup(s')$ is a transition of τ . If (sup, sig) is a τ -region of A then for $e \in E_A$ we define $sig^-(e) = m$, $sig^+(e) = n$ and $|sig(e)| = 0$ if $sig(e) = (m, n) \in E_\tau$ and, otherwise, $sig^-(e) = sig^+(e) = 0$ and $|sig(e)| = sig(e)$ if $sig(e) \in \{0, \dots, b\}$. Hence, by definition of τ , (sup, sig) is a τ -region if and only if $s \xrightarrow{e} s'$ entails $sup(s') = (sup(s) - sig^-(e) + sig^+(e) + |sig(e)|) \bmod (b + 1)$.

Two distinct states $s, s' \in S_A$ define an *SSP atom* (s, s') , which is said to be τ -solvable if there is a τ -region (sup, sig) of A such that $sup(s) \neq sup(s')$. An event $e \in E_A$ and a state $s \in S_A$ at which e does not occur, that is $\neg s \xrightarrow{e}$, define an *ESSP atom* (e, s) . The atom is said to be τ -solvable if there is a τ -region (sup, sig) of A such that $\neg sup(s) \xrightarrow{sig(e)}$. A τ -region solving an ESSP or a SSP atom (x, y) is a *witness* for the τ -solvability of (x, y) . A TS A has the

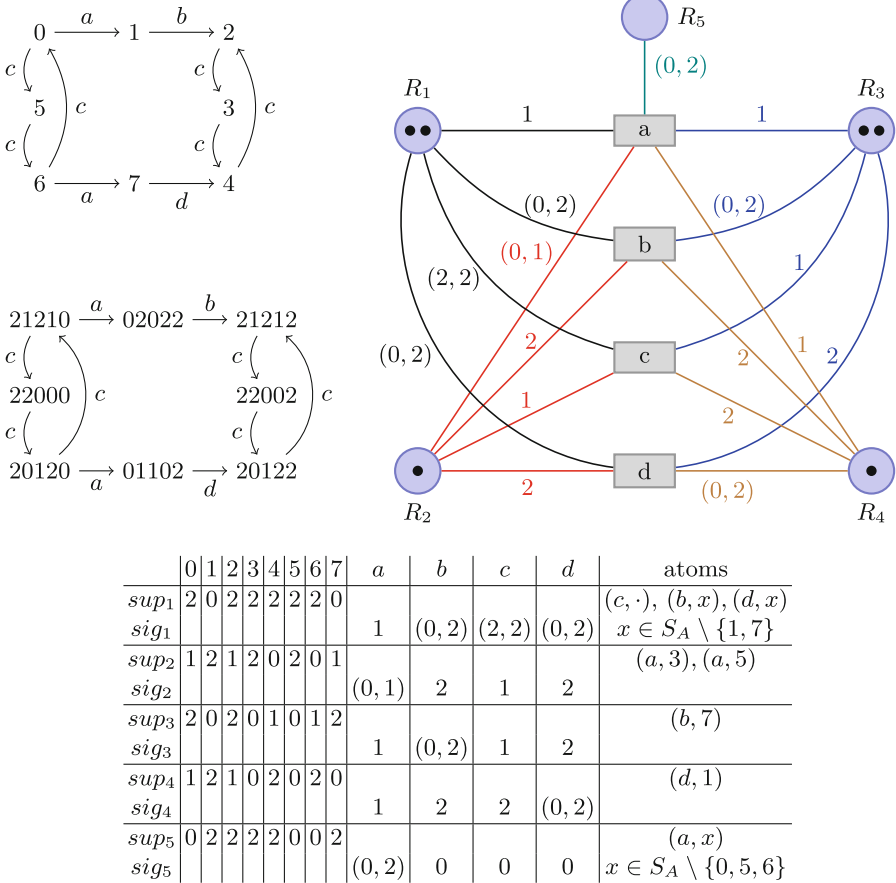


Fig. 2. Upper left, top: Input TS A . Bottom: The table depicts the set of τ_4^2 -regions $\mathcal{R} = \{R_1, R_2, R_3, R_4, R_5\}$ where $R_i = (sup_i, sig_i)$ for $i \in \{1, \dots, 5\}$. All regions of \mathcal{R} are also τ_2^2 -regions and all of $\mathcal{R} \setminus \{R_1\}$ are also τ_3^2 -regions. One verifies that \mathcal{R} contains a witness for every ESSP atom and every SSP atom of A . Hence, if $\tau \in \{\tau_2^2, \tau_4^2\}$ then A and the synthesized τ -net $N_A^{\mathcal{R}}$ has a state graph that is isomorphic to A . However, the set $\{R_2, \dots, R_5\}$ contains no witness for the solvability of $(c, 1)$ and R_1 is no τ_3^2 -region. The ESSP atom $(c, 1)$ is not τ_3^2 -solvable at all, hence, A is not τ_3^2 -feasible. Upper right: The graphical representation of the synthesized τ -net $N_A^{\mathcal{R}} = (\mathcal{R}, \{a, b, c\}, f, 21210)$, where $f(R_i, x) = sig_i(x)$ for every $x \in \{a, b, c\}$ and $M_0(R_1) \cdots M_0(R_5) = sup_1(0) \cdots sup_5(0)$. For readability, 0-labeled flow arcs for the representation of $f(R_5, x) = 0$ for $x \in \{b, c, d\}$ are neglected and flow arcs to the same place are drawn in the same color. Upper left, bottom: The state graph $A_{N_A^{\mathcal{R}}}$ of $N_A^{\mathcal{R}}$ where the reachable markings (states) are represented by 5-tupels $M(R_1) \cdots M(R_5)$. Obviously, $A_{N_A^{\mathcal{R}}}$ is isomorphic to A .

τ -ESSP (τ -SSP) if all its ESSP (SSP) atoms are τ -solvable. Naturally, A is said to be τ -feasible if it has the τ -ESSP and the τ -SSP.

The following fact is well known from [5, p. 161]: A set \mathcal{R} of τ -regions of A contains a witness for all ESSP and SSP atoms if and only if the *synthesized* τ -net $N_A^{\mathcal{R}} = (\mathcal{R}, E_A, f, M_0)$ has a state graph that is isomorphic to A . The flow function of $N_A^{\mathcal{R}}$ is defined by $f((sup, sig), e) = sig(e)$ and its initial marking is $M_0((sup, sig)) = sup(s_{0,A})$ for all $(sup, sig) \in \mathcal{R}, e \in E_A$. The regions of \mathcal{R} become places and the events of E_A become transitions of $N_A^{\mathcal{R}}$. Hence, for a τ -feasible TS A where \mathcal{R} is known, we can synthesize a net N with state graph isomorphic to A by constructing $N_A^{\mathcal{R}}$.

3 Polynomial Time Results

Theorem 1. *Solving τ_4^b -synthesis for a TS A or deciding if A has the τ_4^b -ESSP is doable in polynomial time. Moreover, for $\tau \in \{\tau_2^b, \tau_3^b, \tau_4^b\}$ one can decide in polynomial time whether a given TS A has the τ -SSP.*

The contribution of Theorem 1 is threefold. Firstly, in [18] it has been shown that deciding the τ -ESSP and τ -feasibility is a NP-complete problem for $\tau \in \{\tau_2^b, \tau_3^b\}$. Hence, by showing that deciding the τ -SSP for $\tau \in \{\tau_2^b, \tau_3^b\}$ is doable in polynomial time, Theorem 1 discovers the first Petri net types where the provable computational complexity of SSP is different to ESSP and feasibility.

In [16], Schmitt advanced pure 1-bounded P/T-nets by the additive group of integers modulo 2 and discovered a tractable superclass. In [18], we showed that lifting this approach to (pure) b -bounded P/T-nets where $b \geq 2$ do not lead to superclasses with a tractable synthesis problem. Thus, Theorem 1 proposes the first tractable type of b -bounded Petri nets, $b \geq 2$, so far. Finally, Theorem 1 gives us insight into which of the τ -net *properties*, $\tau \in \{\tau_0^b, \tau_1^b\}$, cause the synthesis' hardness. In particular, flow arc relations (events in τ) between places and transitions in a τ -net define conditions when a transition is able to fire. For example, if N is a τ -net with transition t and place p such that $f(p, t) = (1, 0)$ then the firing of t in a marking M requires $M(p) \geq 1$. By Theorem 1, the hardness of finding a τ -net N for A origins from the potential possibility of τ -nets to satisfy such conditions by multiple markings $M(p) \in \{1, \dots, b\}$. In fact, the definition of τ_4^b implies that $f(p, t) = (m, n)$ requires $M(p) = m$ for the firing of t and prohibits the possibility of multiple choices. By Theorem 1, this makes τ_4^b -synthesis tractable. It should be noted that the results of [4, 12] show that the restriction to "unambiguous markings" of p satisfying conditions defined by $f(p, t)$ does *not guarantee* tractability.

While the question of whether there are superclasses of τ_0^b, τ_1^b , $b \geq 2$, for which synthesis is doable in polynomial time remains unanswered, the following lemma shows that the type τ_4^b yields at least a tractable superclasses of Schmitt's type τ_4^1 [16]. In particular, if $b < b'$ then the class of τ_4^b -nets is strictly more comprehensive than the class of $\tau_4^{b'}$ -nets.

Lemma 1. *If $b < b' \in \mathbb{N}^+$ and if \mathcal{T} is the set of τ_4^b -feasible TSs and \mathcal{T}' the set of $\tau_4^{b'}$ -feasible TSs then $\mathcal{T} \subset \mathcal{T}'$.*

Proof. To proof the lemma, we consider a TS A which is $\tau_4^{b'}$ -feasible but not τ_4^b -feasible. Let A defined by $A = (\{s_0, \dots, s_{b'}\}, \{a\}, \delta, s_0)$ the TS with transition function $\delta(s_i, a) = s_{i+1}$ for $i \in \{0, \dots, b' - 1\}$ and $\delta(s_{b'}, a) = s_0$. By other words, A is a directed labeled cycle $s_0 \xrightarrow{a} \dots \xrightarrow{a} s_{b'} \xrightarrow{s} s_0$ where every transition is labeled by a . Notice, that A has no ESSP atom and, hence, the τ -ESSP for every type of nets. Consequently, A is τ -feasible if and only if it has the τ -SSP.

Assume, for a contradiction, that A is τ_4^b -feasible. By $b < b'$, A provides the SSP atom (s_0, s_{b+1}) and A 's τ_4^b -feasibility implies that there is a τ_4^b -region (sup, sig) solving it. If $sig(a) = (m, n)$ then $sup(s_1) = sup(s_0) - m + n \neq sup(s_0)$ and, by definition of τ_4^b , $\neg sup(s_1) \stackrel{(m, n)}{\rightarrow}$. This is a contradiction to $s_1 \xrightarrow{a}$. Hence, $sig(a) \in \{1, \dots, b\}$. By induction, $sup(s_{b+1}) = sup(s_0) + (b + 1) \cdot sig(a) = sup(s_0) \pmod{b + 1}$ implying $sup(s_{b+1}) = sup(s_0)$. Thus, (sup, sig) does not solve (s_0, s_{b+1}) , which proves that A not to be τ_4^b -feasible.

On the contrary, it is easy to see that the $\tau_4^{b'}$ -region (sup, sig) , which is defined by $sup(s_0) = 0$, $sig(e) = 1$ and $sup(s_{i+1}) = sup(s_i) + sig(a)$ for $i \in \{0, \dots, b' - 1\}$, solves every SSP atom of A . Hence, A is $\tau_4^{b'}$ -feasible. \square

3.1 Abstract Regions and Fundamental Cycles

Unless otherwise stated, in the remainder of this paper we assume that A is a (non-trivial) TS with at least two states, $|S_A| \geq 2$ and event set $E_A = \{e_1, \dots, e_n\}$. Recall that $\tau \in \{\tau_2^b, \tau_3^b, \tau_4^b\}$ and $b \in \mathbb{N}^+$.

The proof of Theorem 1 bases on a generalization of the approach used in [16] that reduces ESSP and SSP to systems of linear equations modulo $b + 1$. It exploits that the solvability of such systems is decidable in polynomial time which is the statement of the following lemma borrowed from [11]:

Lemma 2 ([11]). *If $A \in \mathbb{Z}_{b+1}^{k \times n}$ and $c \in \mathbb{Z}_{b+1}^k$ then deciding if there is an element $x \in \mathbb{Z}_{b+1}^n$ such that $Ax = c$ is doable in time $\mathcal{O}(nk \cdot \max\{n, k\})$.*

Essentially, our generalization composes for every ESSP atom and every SSP atom $\alpha = (x, y)$ of A , respectively, a system of equations modulo $b + 1$ which is solvable if and only if α is τ -solvable. Moreover, a solution of the corresponding system shall provide a τ -region of A that solves α . On the one hand, this approach ensures that having a solution for every system defined by single ESSP atoms and SSP atoms implies the τ -ESSP and τ -SSP for A , respectively. On the other hand, it provides a τ -solving region for every atom in question and, hence, a set \mathcal{R} of τ -regions that witnesses the τ -ESSP and τ -SSP of A . Thus, \mathcal{R} allows us to construct the synthesized net $N_A^{\mathcal{R}}$ with a state graph isomorphic to A . In the following, we establish the notions of *abstract regions* and *fundamental cycles* which make such a translation possible.

We proceed by deducing the notion of abstract regions. Our starting point is the goal to obtain regions (sup, sig) of A as solutions of linear equation systems modulo $b + 1$. By definition, (sup, sig) is a τ -region of A if and only if for every transition $s \xrightarrow{e} s'$ it is true that

$$sup(s') = (sup(s) - sig^-(e) + sig^+(e) + |sig(e)|) \pmod{b + 1} \tag{1}$$

Hence, installing for every transition $s \xrightarrow{e} s'$ the corresponding Eq. 1 yields a linear system of equations whose solutions are regions of A . If (sup, sig) is a solution of this system such that $sig(e) = (m, n) \in E_\tau \setminus \{0, \dots, b\}$ for $e \in E_A$ then, by definition, for every transition $s \xrightarrow{e} s'$ it has to be true that $m \leq sup(s)$ and $sup(s') - m + n \leq b$. Unfortunately, the conditions $m \leq sup(s)$ and $sup(s') - m + n \leq b$ can not be tested in the group \mathbb{Z}_{b+1} . To cope with this obstacle, we abstract from elements $(m, n) \in E_\tau$ by restricting to regions (solutions) that identify (m, n) with the unique element $x \in \{0, \dots, b\}$ such that $x = (n - m) \bmod (b + 1)$. This leads to the notion of *abstract* τ -regions. A τ -region (sup, sig) of A is called abstract if sig 's codomain restricts to $\{0, \dots, b\}$, that is, $sig : E_A \rightarrow \{0, \dots, b\}$. If (sup, sig) is an abstract region, then we call sig an *abstract signature*. For the sake of clarity, we denote abstract signatures by *abs* instead of *sig* and abstract regions by (sup, abs) instead of (sup, sig) .

By definition, two mappings $sup, abs : \{0, \dots, b\} \rightarrow \{0, \dots, b\}$ define an abstract τ -region if and only if for every transition $s \xrightarrow{e} s'$ of A it is true that

$$sup(s') = (sup(s) + abs(e)) \bmod (b + 1) \quad (2)$$

Obviously, for abstract regions Eq. 1 reduces to Eq. 2. Installing for every transition $s \xrightarrow{e} s'$ of A its corresponding Eq. 2 yields a system modulo $b + 1$ whose solutions are abstract regions. Uncomfortably, such systems require to deal with sup and abs simultaneously. It is better to first obtain abs independently of sup and then define sup with the help of abs . The following observations show how to realize this idea.

By induction and Eq. 2, one immediately obtains that (sup, abs) is an abstract region if and only if for every directed labeled path $p = s_{0,A} \xrightarrow{e'_1} \dots \xrightarrow{e'_m} s_m$ of A from the initial state $s_{0,A}$ to the state s_m the *path equation* holds:

$$sup(s_m) = (sup(s_{0,A}) + abs(e'_1) + \dots + abs(e'_m)) \bmod (b + 1) \quad (3)$$

To exploit Eq. 3 we, firstly, identify every abstract signature abs with the unique element $abs = (abs(e_1), \dots, abs(e_n)) \in \mathbb{Z}_{b+1}^n$. Secondly, we say that $\psi_{b+1}^p = (\#_{e_1}^p, \dots, \#_{e_n}^p) \in \mathbb{Z}_{b+1}^n$ is the Parikh-vector of p that counts the number $\#_{e_i}^p$ of occurrences of every event $e_i \in E_A$ on the path p modulo $(b + 1)$. Thirdly, for two elements $v, w \in \mathbb{Z}_{b+1}^n$ we define $v \cdot w = v_1 w_1 + \dots + v_n w_n$. As a result, considering p and abs as elements of \mathbb{Z}_{b+1}^n allows us to reformulate the path equation by $sup(s_m) = (sup(s_{0,A}) + \psi_{b+1}^p \cdot abs) \bmod (b + 1)$. Especially, if p, p' are two different paths from $s_{0,A}$ to s_m then $\psi_{b+1}^p \cdot abs = \psi_{b+1}^{p'} \cdot abs$. Thus, the support sup is fully determined by $sup(s_{0,A})$ and abs . By the validity of the path equation, every abstract signature abs implies $b + 1$ different abstract τ -regions of A , one for every $sup(s_{0,A}) \in \{0, \dots, b\}$. Altogether, we have argued that the challenge of finding abstract regions of A reduces to the task of finding A 's abstract signatures. In the following, we deduce the notion of fundamental cycles defined by chords of a spanning tree of A which enables us to find abstract

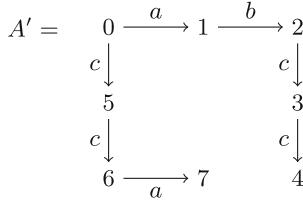


Fig. 3. A spanning tree A' of running example TS A introduced in Fig. 2. The unique Parikh vectors ψ_0, \dots, ψ_7 of A' (written as rows) are given by $\psi_0 = (0, 0, 0, 0), \psi_1 = (1, 0, 0, 0), \psi_2 = (1, 1, 0, 0), \psi_3 = (1, 1, 1, 0), \psi_4 = (1, 1, 2, 0), \psi_5 = (0, 0, 1, 0), \psi_6 = (0, 0, 2, 0)$ and $\psi_7 = (1, 0, 2, 0)$. The transitions $\delta_A(7, d) = 4, \delta_A(4, c) = 2$ and $\delta_A(6, c) = 0$ of A define the chords of A' . The corresponding fundamental cycles are given by $\psi_t = \psi_7 + (0, 0, 0, 1) - \psi_4 = (0, 2, 0, 1)$ and $\psi_{t'} = \psi_4 + (0, 0, 1, 0) - \psi_2 = (0, 0, 0, 0)$ and $\psi_{t''} = \psi_6 + (0, 0, 1, 0) - \psi_0 = (0, 0, 0, 0)$. Hence, if $abs = (x_a, x_b, x_c, x_d)$ then $\psi_t \cdot abs = 0 \cdot x_a + 2 \cdot x_b + 0 \cdot x_c + x_d = 2 \cdot x_b + x_d$. By $\psi_{t'} \cdot abs = \psi_{t''} \cdot abs = 0$ for every map abs , only the equation $2 \cdot x_b + x_d = 0$ contributes to the basic part of every upcoming system.

signatures. For readability, we often write $x = y_1 + \dots + y_\ell \pmod{b+1}$ instead of $x = (y_1 + \dots + y_\ell) \pmod{b+1}$

A *spanning tree* A' of A is a sub-transition system $A' = (S_{A'}, E_{A'}, \delta_{A'}, s_{0,A'})$ of A with a restricted transition function $\delta_{A'}$ such that, firstly, $\delta_{A'}(s, e) = s'$ entails $\delta_A(s, e) = s'$ and, secondly, for every $s \in S_{A'}$ there is *exactly* one path $p = s_{0,A'} \xrightarrow{e_1} \dots \xrightarrow{e_m} s$ in A' . By other words, the underlying undirected graph of A' is a tree in the common graph-theoretical sense. Every transition $s \xrightarrow{e} s'$ of A which is not in A' is called a *chord* (of A'). The chords of A' are exactly the edges that induce a cycle in A' 's underlying undirected graph. This gives rise to the following notion of fundamental cycles. For $e_i \in \{e_1, \dots, e_n\}$ we define $1_i = (x_1, \dots, x_n)^t \in \mathbb{Z}_{b+1}^n$, where $x_j = 1$ if $j = i$ and, else $x_j = 0$. If $t = s \xrightarrow{e_i} s'$ is a chord of A' then there are *unique* paths p from $s_{0,A}$ to s and p' from $s_{0,A}$ to s' in A' such that t corresponds to the unique element $\psi_t = \psi_{b+1}^p + 1_i - \psi_{b+1}^{p'} \in \mathbb{Z}_{b+1}^n$, called the *fundamental cycle* of t .

The following lemma teaches us how to use fundamental cycles to generate abstract signatures of A :

Lemma 3. *If A' is a spanning tree of a TS A with chords t_1, \dots, t_k then $abs \in \mathbb{Z}_{b+1}^n$ is an abstract signature of A if and only if $\psi_{t_i} \cdot abs = 0$ for all $i \in \{1, \dots, k\}$. Two different spanning trees A' and A'' provide equivalent systems of equations.*

Proof. We start with proving the first statement. *If:* Let $abs \in \mathbb{Z}_{b+1}^n$ such that $\psi_{t_i} \cdot abs = 0$ for all $i \in \{1, \dots, k\}$ and $sup(s_{0,A}) \in \{0, \dots, b\}$. If $s \in S_{A'}$ then there is a unique path $p = s_{0,A} \xrightarrow{e'_1} \dots \xrightarrow{e'_m} s_m = s$ in A' from $s_{0,A}$ to s . By defining $sup(s) = sup(s_{0,A}) + \psi_{b+1}^p \cdot abs$ we obtain inductively that every transition $s \xrightarrow{e} s'$ of A' satisfies $sup(s') = sup(s) + abs(e)$. It remains to prove that this definition is consistent with the remaining transitions of A , the chords of A' . Let $t = s \xrightarrow{e} s'$

be a chord of A' and let $p = s_{0,A} \xrightarrow{e'_1} \dots \xrightarrow{e'_m} s$ and $p' = s_{0,A} \xrightarrow{e''_1} \dots \xrightarrow{e''_l} s'$ be the unique paths from $s_{0,A}$ to s and s' in A' , respectively. By $\text{sup}(s) = \text{sup}(s_{0,A}) + \psi_{b+1}^p \cdot \text{abs}$ and $\text{sup}(s') = \text{sup}(s_{0,A}) + \psi_{b+1}^{p'} \cdot \text{abs}$ we have that

$$\begin{aligned}
0 &= \psi_t \cdot \text{abs} && \iff \\
0 &= (-\psi_{b+1}^{p'} + 1_i + \psi_{b+1}^p) \cdot \text{abs} && \iff \\
0 &= -\psi_{b+1}^{p'} \cdot \text{abs} + \text{abs}(e) + \psi_{b+1}^p \cdot \text{abs} && \iff \\
\psi_{b+1}^{p'} \cdot \text{abs} &= \text{abs}(e) + \psi_{b+1}^p \cdot \text{abs} && \iff \\
\text{sup}(s_{0,A}) + \psi_{b+1}^{p'} \cdot \text{abs} &= \text{sup}(s_{0,A}) + \psi_{b+1}^p \cdot \text{abs} + \text{abs}(e) && \iff \\
\text{sup}(s') &= \text{sup}(s) + \text{abs}(e)
\end{aligned}$$

where $0 = \psi_t \cdot \text{abs}$ is true by assumption. Hence, abs is an abstract signature of A and the proof shows how to get a corresponding abstract region (sup, abs) of A .

Only-if: If abs is an abstract region of A then we have $\text{sup}(s') = \text{sup}(s) + \text{abs}(e)$ for every transition in A . Hence, if $t = s \xrightarrow{e} s'$ a chord of a spanning tree A' of A then working backwards the equivalent equalities above proves $\psi_t \cdot \text{abs} = 0$.

The second statement is implied by the first: If A' , A'' are two spanning trees of A with fundamental cycles $\psi_{t_1}^{A'}, \dots, \psi_{t_k}^{A'}$ and $\psi_{t'_1}^{A''}, \dots, \psi_{t'_k}^{A''}$, respectively, then we have for $\text{abs} \in \mathbb{Z}_{b+1}^n$ that $\psi_{t_i}^{A'} \cdot \text{abs} = 0, i \in \{1, \dots, k\}$ if and only if abs is an abstract signature of A if and only if $\psi_{t'_i}^{A''} \cdot \text{abs} = 0, i \in \{1, \dots, k\}$. \square

In the following, justified by Lemma 3, we assume A' to be a fixed spanning tree of A with chords t_1, \dots, t_k . By $M_{A'}$ we denote the system of equations $\psi_{t_i} \cdot \text{abs} = 0, i \in \{1, \dots, k\}$. Moreover, by ψ_s we abridge for $s \in S_A$ the Parikh-vector ψ_{b+1}^p of the unique path $s_{0,A} \xrightarrow{e'_1} \dots \xrightarrow{e'_m} s$ in A' . A spanning tree of A is computable in polynomial time: As δ_A is a function, A has at most $|E||S_A|^2$ transitions and A' contains $|S_A| - 1$ transitions. Thus, by $2 \leq |S_A|$, A' has at most $|E||S_A|^2 - 1$ chords. Consequently, a spanning tree A' of A is computable in time $\mathcal{O}(|E||S_A|^3)$ [17].

To get polynomial time solvable systems of equations, we have restricted ourselves to equations like Eq. 2. This restriction results in the challenge to compute abstract signatures of A . By Lemma 3, abstract signatures of A are solutions of $M_{A'}$. An (abstract) τ -region (sup, abs) of A arises from abs by defining $\text{sup}(s_{0,A})$ and $\text{sup}(s) = \text{sup}(s_{0,A}) + \psi_s \cdot \text{abs}, s \in S(A)$. However, if (s, s') is a SSP atom of A then $\text{sup}(s) \neq \text{sup}(s')$ is not implied. Moreover, by definition, to τ -solve ESSP atoms (e, s) we need (concrete) τ -regions (sup, sig) such that $\text{sig} : E_A \rightarrow E_\tau$. The next section shows how to extend $M_{A'}$ to get τ -solving regions.

3.2 The Proof of Theorem 1

This section shows how to extend $M_{A'}$ for a given (E)SSP atom α to get a system M_α , whose solution yields a region solving α if there is one.

If α is an SSP atom (s, s') then we only need to assure that the (abstract) region (sup, abs) built on a solution of $M_{A'}$ satisfies $sup(s) \neq sup(s')$. By $sup(s) = sup(s_{A,0}) + \psi_s \cdot abs$ and $sup(s') = sup(s_{A,0}) + \psi_{s'} \cdot abs$, it is sufficient to extend $M_{A'}$ in a way that ensures $\psi_s \cdot abs \neq \psi_{s'} \cdot abs$. The next lemma proves this claim.

Lemma 4. *If $\tau \in \{\tau_2^b, \tau_3^b, \tau_4^b\}$ then a τ -SSP atom (s, s') of A is τ -solvable if and only if there is an abstract signature abs of A with $\psi_s \cdot abs \neq \psi_{s'} \cdot abs$.*

Proof. *If:* If abs is an abstract signature with $\psi_s \cdot abs \neq \psi_{s'} \cdot abs$ then the τ -region (sup, abs) with $sup(s_{0,A}) = 0$ and $sup(s) = \psi_s \cdot abs$ satisfies $sup(s) \neq sup(s')$. *Only-if:* If (sup, sig) is a τ -region then we obtain a corresponding abstract τ -region (sup, abs) as defined in Lemma 6. Clearly, abs is an abstract signature and satisfies the path equations. Consequently, by $sup(s_0) + \psi_s \cdot abs = sup(s) \neq sup(s') = sup(s_0) + \psi_{s'} \cdot abs$, we have that $\psi_s \cdot abs \neq \psi_{s'} \cdot abs$. \square

The next lemma applies Lemma 4 to get a polynomial time algorithm which decides the τ -SSP if $\tau \in \{\tau_2^b, \tau_3^b, \tau_4^b\}$.

Lemma 5. *If $\tau \in \{\tau_2^b, \tau_3^b, \tau_4^b\}$ then to decide whether a TS A has the τ -SSP is doable in time $\mathcal{O}(|E_A|^3 \cdot |S_A|^6)$.*

Proof. If $\alpha = (s, s')$ is a SSP atom of A then the (basic) part $M_{A'}$ of M_α consists of at most $|E| \cdot |S_A|^2 - 1$ equations for the fundamental cycles. To satisfy $\psi_s \cdot abs \neq \psi_{s'} \cdot abs$, we add the equation $(\psi_s - \psi_{s'}) \cdot abs = q$, where initially $q = 1$, and get (the first possible) M_α . A solution of M_α provides an abstract region satisfying $\psi_s \neq \psi_{s'}$. By Lemma 4, this proves the solvability of α . If M_α is not solvable then we modify M_α to M'_α simply by incrementing q and try to solve M'_α . Either we get a solution or we modify M'_α to M''_α by incrementing q again. By Lemma 4, if (s, s') is solvable then there is a $q \in \{1, \dots, b\}$ such that the corresponding (modified) system has a solution. Hence, after at most b iterations we can decide whether (s, s') is solvable or not. Consequently, we have to solve at most b linear systems with at most $|E_A| \cdot |S_A|^2$ equations for (s, s') . The value b is not part of the input. Thus, by Lemma 2, this is doable in $\mathcal{O}(|E_A|^3 \cdot |S_A|^4)$ time. We have at most $|S_A|^2$ different SSP atoms to solve. Hence, we can decide the τ -SSP in time $\mathcal{O}(|E_A|^3 \cdot |S_A|^6)$. \square

As a next step, we let $\tau = \tau_4^b$ and prove the polynomial time decidability of τ -ESSP. But before that we need the following lemma that tells us how to obtain abstract regions from (concrete) regions:

Lemma 6. *If (sup, sig) is a τ -region of a TS A then we obtain a corresponding abstract τ -region (sup, abs) by defining abs for $e \in E_A$ as follows: If $sig(e) = (m, n)$ then $abs(e) = -m + n \bmod (b + 1)$ and, otherwise, if $sig(e) \in \{0, \dots, b\}$ then $abs(e) = sig(e)$.*

Proof. We have to show that $s \xrightarrow{e} s'$ in A entails $sup(s) \xrightarrow{abs(e)} sup(s')$ in τ . If $abs(e) = sig(e) \in \{0, \dots, b\}$ this is true as (sup, sig) is a τ -region.

If $\text{sig}(e) = (m, n)$ then, by definition, we have $\text{sup}(s') = \text{sup}(s) - m + n \bmod (b + 1)$ implying $\text{sup}(s') - \text{sup}(s) = -m + n \bmod (b + 1)$. By $\text{abs}(e) = -m + n \bmod (b + 1)$ and symmetry, we get $-m + n = \text{abs}(e) \bmod (b + 1)$ and, by transitivity, we obtain $\text{sup}(s') - \text{sup}(s) = \text{abs}(e) \bmod (b + 1)$ which implies $\text{sup}(s') = \text{sup}(s) + \text{abs}(e) \bmod (b + 1)$. Thus $\text{sup}(s) \xrightarrow{\text{abs}(e)} \text{sup}(s')$. \square

Let α be an ESSP atom (e, s) and let s_1, \dots, s_k be the sources of e in A . By definition, a τ -region (sup, sig) solves α if and only if $\text{sig}(e) = (m, n)$ and $\neg \text{sup}(s) \xrightarrow{\text{sig}(e)}$ for a $(m, n) \in E_\tau$. By definition of τ , every element $(m, n) \in E_\tau$ occurs at exactly one state in τ and this state is m . Hence, $\text{sup}(s_1) = \dots = \text{sup}(s_k) = m$ and $\text{sup}(s) \neq m$. We base the following lemma on this simple observation. It provides necessary and sufficient conditions that an *abstract* region must fulfill to imply a *solving* (concrete) region.

Lemma 7. *Let $\tau = \tau_4^b$ and A be a TS and let $s_1 \xrightarrow{e} s'_1, \dots, s_k \xrightarrow{e} s'_k$ be the e -labeled transitions in A , that is, if $s' \in S_A \setminus \{s_1, \dots, s_k\}$ then $\neg s' \xrightarrow{e}$. The atom (e, s) is τ -solvable if and only if there is an event $(m, n) \in E_\tau$ and an abstract region (sup, abs) of A such that the following conditions are satisfied:*

1. $\text{abs}(e) = -m + n \bmod (b + 1)$,
2. $\psi_{s_1} \cdot \text{abs} = m - \text{sup}(s_{A,0}) \bmod (b + 1)$,
3. $(\psi_{s_1} - \psi_{s_i}) \cdot \text{abs} = 0 \bmod (b + 1)$ for $i \in \{2, \dots, k\}$,
4. $(\psi_{s_1} - \psi_s) \cdot \text{abs} \neq 0 \bmod (b + 1)$.

Proof. If: Let (sup, abs) be an abstract region that satisfies the conditions 1–4. We obtain a τ -solving region (sup, sig) with (the same support and) the signature sig defined by $\text{sig}(e') = \text{abs}(e')$ if $e' \neq e$ and $\text{sig}(e') = (m, n)$ if $e' = e$. To argue that (sup, sig) is a τ -region we have to argue that $q \xrightarrow{e'} q'$ in A implies $\text{sup}(q) \xrightarrow{\text{sig}(e')} \text{sup}(q')$. As (sup, abs) is an abstract region this is already clear for transitions $q \xrightarrow{e'} q'$ where $e' \neq e$. Moreover, (sup, abs) satisfies $\psi_{s_1} \cdot \text{abs} = m - \text{sup}(s_{A,0}) \bmod (b + 1)$ and the path equation holds, that is, $\text{sup}(s_1) = \text{sup}(s_{A,0}) + \psi_{s_1} \cdot \text{abs} \bmod (b + 1)$ which implies $\text{sup}(s_1) = m$. Consequently, by definition of τ , we have $\text{sup}(s_1) \xrightarrow{(m, n)} n$ in τ . Furthermore, by $\text{abs}(e) = -m + n \bmod (b + 1)$ we have $m + \text{abs}(e) = n \bmod (b + 1)$. Hence, by $\text{sup}(s_1) \xrightarrow{\text{abs}(e)} \text{sup}(s'_1)$, we conclude $\text{sup}(s'_1) = n$ and, thus, $\text{sup}(s_1) \xrightarrow{(m, n)} \text{sup}(s'_1)$. By $(\psi_{s_1} - \psi_{s_i}) \cdot \text{abs} = 0 \bmod (b + 1)$ for $i \in \{2, \dots, k\}$, we obtain that $\text{sup}(s_1) = \dots = \text{sup}(s_k) = m$. Therefore, similar to the discussion for $s_1 \xrightarrow{e} s'_1$, we obtain by $\text{sup}(s_i) \xrightarrow{\text{abs}(e)} \text{sup}(s'_i)$ that the transitions $\text{sup}(s_i) \xrightarrow{(m, n)} \text{sup}(s'_i)$ are present in τ for $i \in \{2, \dots, k\}$. Consequently, (sup, sig) is a τ -region.

Finally, by $(\psi_{s_1} - \psi_s) \cdot \text{abs} \neq 0 \bmod (b + 1)$, have that $\text{sup}(s_1) \neq \text{sup}(s)$ and, thus, $\neg \text{sup}(s) \xrightarrow{\text{sig}(e)}$. This proves (e, s) to be τ -solvable by (sup, sig) .

Only-if: Let (sup, sig) be a τ -region that solves (e, s) implying, by definition, $\neg \text{sup}(s) \xrightarrow{\text{sig}(e)}$. We use (sup, sig) to define a corresponding abstract τ -region

(sup, abs) in accordance to Lemma 6. If $sig(e) \in \{0, \dots, b\}$ then $sup(s) \xrightarrow{sig(e)}$, a contradiction. Hence, it is $sig(e) = (m, n) \in E_\tau$ such that $sup(s_i) \xrightarrow{(m, n)}$ for $i \in \{1, \dots, k\}$ and $\neg sup(s) \xrightarrow{(m, n)}$. This immediately implies $sup(s) \neq sup(s_1)$ and, hence, $(\psi_{s_1} - \psi_s) \cdot abs \neq 0 \pmod{b+1}$. By $sup(s_i) \xrightarrow{(m, n)} sup(s'_i)$ and definition of τ , we have that $sup(s_i) = m$ and $sup(s'_i) = n$ for $i \in \{1, \dots, k\}$ implying $(\psi_{s_1} - \psi_{s_i}) \cdot abs = 0 \pmod{b+1}$ for $i \in \{2, \dots, k\}$. Moreover, by $sup(s_1) \xrightarrow{abs(e)} sup(s'_1)$ we have $abs(e) = sup(s'_1) - sup(s_1) \pmod{b+1}$. Hence, it is $abs(e) = -m + n \pmod{b+1}$. Finally, by the path equation, we have $sup(s_1) = sup(s_{A,0}) + \psi_{s_1} \cdot abs \pmod{b+1}$ which with $sup(s_1) = m$ implies $\psi_{s_1} \cdot abs = m - sup(s_{A,0}) \pmod{b+1}$. This proves the lemma. \square

The next lemma's proof exhibits a polynomial time decision algorithm for the τ_4^b -ESSP: Given a TS A and a corresponding ESSP atom α , the system $M_{A'}$ is extended to a system M_α . If M_α has a solution abs then it implies a region (sup, abs) satisfying the conditions of Lemma 9. By Lemma 9, this implies α 's solvability. Reversely, by Lemma 9, if α is solvable then there is an abstract region (sup, abs) which satisfies the conditions (1–4). The abstract signature abs is the solution of a corresponding equation system M_α . Hence, we get a solvable M_α if and only if α is solvable. We argue that the number of possible systems is bounded polynomially in the size of A . The solvability of every system is also decidable in polynomial time. Consequently, by the at most $|E_A| \cdot |S_A|$ ESSP atoms to solve, this yields the announced decision procedure.

Lemma 8. *If a TS A has the τ_4^b -ESSP is decidable in time $\mathcal{O}(|E_A|^4 \cdot |S_A|^5)$.*

Proof. To estimate the computational complexity of deciding the τ_4^b -ESSP for A observe that A has at most $|S_A| \cdot |E_A|$ ESSP atoms to solve. Hence, the maximum costs of deciding the τ_4^b -ESSP for A equals $|S_A| \cdot |E_A|$ -times the maximum effort for a single atom.

To decide the τ -solvability of a single ESSP atom (e, s) , we compose systems in accordance to Lemma 7. The maximum costs can be estimated as follows: The (basic) part $M_{A'}$ of M_α has at most $|E_A| \cdot |S_A|^2$ equations. Moreover, e occurs at most at $|S_A| - 1$ states. This makes at most $|S_A|$ equations to ensure that e 's sources will have the same support, the third condition of Lemma 7. According to the first and the second condition, we choose an event $(m, n) \in E_\tau$, a value $sup(s_{A,0}) \in \{0, \dots, b\}$, define $abs(e) = -m + n \pmod{b+1}$ and add the corresponding equation $\psi_{s_1} \cdot abs = m - sup(s_{A,0})$. For the fourth condition we choose a fixed value $q \in \{1, \dots, b\}$ and add the equation $(\psi_{s_1} - \psi_s) \cdot abs = q$. Hence, the system has at most $2 \cdot |E_A| \cdot |S_A|^2$ equations.

By Lemma 2, one checks in time $\mathcal{O}(|E_A|^3 \cdot |S_A|^4)$ if such a system has a solution. Notice, we use that $2 \cdot |E_A| \cdot |S_A|^2 = \max\{|E_A|, 2 \cdot |E_A| \cdot |S_A|^2\}$. There are at most $(b+1)^2$ possibilities to choose a corresponding $(m, n) \in E_\tau$ and only $b+1$ possible values for x and for q , respectively. Hence, for a fixed atom (e, s) , we have to solve at most $(b+1)^4$ such systems and b is not part of the input. Consequently, we can decide in time $\mathcal{O}(|E_A|^3 \cdot |S_A|^4)$ if (e, s) is solvable.

A provides at most $|S_A| \cdot |E_A|$ ESSP atoms. Hence, the τ_4^b -ESSP for A is decidable in time $\mathcal{O}(|E_A|^4 \cdot |S_A|^5)$. \square

The following lemma completes the proof of Theorem 1 and shows that τ_4^b -synthesis is doable in polynomial time.

Corollary 1. *To construct for a TS A a τ_4^b -net N with a state graph A_N isomorphic to A if it exists is doable in time $\mathcal{O}(|E_A|^3 \cdot |S_A|^5 \cdot \max\{|E_A|, |S_A|\})$.*

Proof. By [5], if \mathcal{R} is a set of regions of A containing for each ESSP and SSP atom of A a solving region, respectively, then the τ -net $N_A^{\mathcal{R}} = (\mathcal{R}, E_A, f, M_0)$, where $f((sup, sig), e) = sig(e)$ and $M_0((sup, sig)) = sup(s_{0,A})$ for $(sup, sig) \in \mathcal{R}, e \in E_A$, has a state graph isomorphic to A . Hence, the corollary follows from Lemmas 5 and 8. \square

3.3 Examples

We pick up our running example TS A introduced in Fig. 2 and its spanning tree A' presented in Fig. 3. We present two steps of the method given by Lemma 8 for the type τ_4^2 and check τ_4^2 -solvability of the ESSP atom $(c, 1)$.

For a start, we choose $(m, n) = (0, 1)$ and $sup(0) = 0$ and determine $abs(c) = -0 + 1 = 1$ which yields $abs = (x_a, x_b, 1, x_d)$. We have to add $\psi_0 \cdot abs = m - sup(0) = 0$ which, by $\psi_0 = (0, 0, 0, 0)$, is always true and do not contribute to the system. Moreover, for $i \in \{0, 2, 3, 4, 5, 6\}$, we add the equation $(\psi_0 - \psi_i) \cdot abs = 0$. We have $\psi_0 - \psi_6 = (0, 0, -2, 0)$ and $(0, 0, -2, 0) \cdot abs = 0 \cdot x_a - 0 \cdot x_b - 2 \cdot 0 \cdot x_d = 0$ yields a contradiction. Hence, $(c, 1)$ is not solvable by a region (sup, sig) where $sup(0) = 0$ and $sig(c) = (0, 1)$. Similarly, we obtain that the system corresponding to $sup(0) \in \{1, 2\}$ and $sig(c) = (0, 1)$ is also not solvable.

For another try, we choose $(m, n) = (2, 2)$ and $sup(0) = 2$. In accordance to the first and the second condition of Lemma 7 this determines $abs = (x_a, x_b, 0, x_d)$ and yields the equation $\psi_0 \cdot abs = m - sup(0) = 2 - 2 = 0$ which is always true. For the fourth condition, we pick $q = 2$ and add the equation $(\psi_0 - \psi_1) \cdot abs = 2 \cdot x_a = 2$. Finally, for the third condition, we add for $i \in \{0, 2, 3, 4, 5, 6\}$ the equation $(\psi_0 - \psi_i) \cdot abs = 0$ and obtain the following system of equations modulo $(b + 1)$:

$$\begin{array}{rclcl}
 \psi_t \cdot abs & = & 2 \cdot x_b & + x_d & = 0 \\
 (\psi_0 - \psi_1) \cdot abs & = & 2 \cdot x_a & & = 2 \\
 (\psi_0 - \psi_2) \cdot abs & = & 2 \cdot x_a & + 2 \cdot x_b & = 0 \\
 (\psi_0 - \psi_3) \cdot abs & = & 2 \cdot x_a & + 2 \cdot x_b & + 2 \cdot 0 & = 0 \\
 (\psi_0 - \psi_4) \cdot abs & = & 2 \cdot x_a & + 2 \cdot x_b & + 1 \cdot 0 & = 0 \\
 (\psi_0 - \psi_5) \cdot abs & = & & & 2 \cdot 0 & = 0 \\
 (\psi_0 - \psi_6) \cdot abs & = & & & 1 \cdot 0 & = 0
 \end{array}$$

This system is solvable by $abs = (1, 2, 0, 2)$. We construct a region in accordance to the proof of Lemma 7: By $sup(0) = 2$ we obtain $sup(1) = 2 + \psi_1 \cdot abs = 2 + (1, 0, 0, 0) \cdot (1, 2, 0, 2) = 0$. Similarly, by $sup(i) = 2 + \psi_i \cdot abs$ for $i \in \{2, \dots, 7\}$ we obtain $sup(2) = sup(3) = sup(4) = sup(5) = sup(6) = 2$ and $sup(7) = 0$. Hence, by defining $sig(c) = (2, 2)$, $sig(a) = 1$, $sig(b) = 2$ and $sig(d) = 2$ we obtain a fitting τ_4^b -region (sup, sig) that solves $(c, 1)$.

A closer look shows, that this support equals sup_1 which is presented in Fig. 2 and allows the signature sig_1 , hence, $(sup, sig_1) = (sup_1, sig_1)$. The τ_4^b -region (sup, sig_1) solves a lot of further ESSP and SSP atoms. This observation reveals a first possible improvement of the method introduced by Lemma 8 and suggest, given a solution abs , to map as many events of A to a signature different from $0, \dots, b$ as possible.

4 Fixed Parameter Tractability Results

Classical complexity theory measures the computational complexity of decision problems only in the size of the input. In [18], we showed that deciding if a TS A is τ -feasible or has the τ -ESSP, respectively, is NP-complete for $\tau \in \{\tau_2^b, \tau_3^b\}$. Thus, both problems are intractable from the perspective of classical complexity. Unfortunately, measuring the complexity purely in the size of A tells us nothing about the “source” of this negative result. On the contrary, parameterized complexity, developed by Downey and Fellows [10], allows us to study in which way different parameters of a TS A influence the complexity. This makes a finer analysis possible. Moreover, if we find a parameter, typically small on input instances of real-world applications, then algorithms, exponential in the size of the parameter but polynomial in the size of A , may work well in practice.

Formally, we say that a (decision) problem P is *fixed parameter tractable* with respect to parameter k if there exists an algorithm that solves P in time $\mathcal{O}(f(k)n^c)$, where f is some computable function, n is the size of the input and c is a constant independent from parameter k .

Let A be a TS and let for $e \in E_A$ the set $S_e = \{s \in S_A \mid s \xrightarrow{e}\}$ containing the states of A at which e occur. The (maximum) *number of occurrences of events* is defined by $k = \max\{|S_e| \mid e \in E_A\}$. In [19] it has been shown that deciding τ_1^1 -feasibility and τ_1^1 -ESSP is NP-complete even if $k = 2$. If there is a $\mathcal{O}(f(k)|A|^c)$ -time algorithm for these problems then, for $k = 2$, it runs in polynomial time in A 's size. This is because $f(2)$ is a constant. Thus, τ_1^1 -feasibility and τ_1^1 -ESSP, parameterized by k , are not fixed parameter tractable as long as $P \neq NP$.

On the contrary, the main result of this paper discovers that τ -ESSP and τ -feasibility parameterized by k are fixed parameter tractable. This reveals, that the number of occurrences of events is a *structural property of the input A* that makes τ -ESSP and τ -feasibility problems inherently hard to solve.

Theorem 2. *Let $\tau \in \{\tau_2^b, \tau_3^b\}$ and let A be a TS system with number of occurrences of events k . The τ -ESSP and the τ -feasibility are fixed parameter tractable with respect to parameter k .*

Given an ESSP atom α of a TS A , the following lemma provides conditions which an abstract τ -region of A satisfies if and only if α is τ -solvable. Moreover, it teaches us how to gain a corresponding τ -solving region from an abstract region satisfying the conditions.

Lemma 9. *Let $\tau \in \{\tau_2^b, \tau_3^b\}$, let (e, s) be an ESSP atom of A and let s_1, \dots, s_k be the sources of e in A , that is, $s_i \xrightarrow{e}$ for $i \in \{1, \dots, k\}$ and if $s' \in S_A \setminus \{s_1, \dots, s_k\}$ then $\neg s' \xrightarrow{e}$.*

The ESSP atom (e, s) is τ -solvable if and only if there is an event $(m, n) \in E_\tau$ and an abstract region (sup, abs) of A that satisfies the following conditions:

1. $abs(e) = -m + n \bmod (b + 1)$,
2. $\psi_{s_i} \cdot abs = sup(s_i) - sup(s_{A,0}) \bmod (b + 1)$ and $m \leq sup(s_i) \leq b + m - n$ for $i \in \{1, \dots, k\}$,
3. $\psi_s \cdot abs = sup(s) - sup(s_{A,0}) \bmod (b + 1)$ and $0 \leq sup(s) \leq m - 1$ or $b + m - n + 1 \leq sup(s) \leq b$

Proof. *If:* Let (sup, abs) be an abstract τ -region of A satisfying (1)–(3). We get a τ -solving region (sup, sig) as follows: For $e' \in E_A$ we define $sig(e') = abs(e')$ if $e' \neq e$ and, otherwise, we set $sig(e') = (m, n)$ if $e' = e$. Firstly, we show that (sup, sig) is a region and, secondly, we argue that it τ -solves (e, s) .

We have to show, that $q \xrightarrow{e'} q'$ in A implies $sup(q) \xrightarrow{sig(e')} sup(q')$ in τ . If $e' \neq e$, then this is true by (sup, abs) being a τ -region. It remains to show that $s_i \xrightarrow{e} s'_i$ implies $sup(s_i) \xrightarrow{(m, n)} sup(s'_i)$ for $i \in \{0, \dots, k\}$. By $m \leq sup(s_i) \leq b + m - n$ and the definition of τ , there is an $s_\tau \in \{0, \dots, b\}$ with $sup(s_i) \xrightarrow{(m, n)} s_\tau$. This implies $s_\tau = sup(s_i) - m + n \bmod (b + 1)$. The assumption $abs(e) = -m + n \bmod (b + 1)$ yields $s_\tau = sup(s_i) + abs(e) \bmod (b + 1)$. Hence, we have that $sup(s_i) + abs(e) = sup(s_i) - m + n \bmod (b + 1)$. By $sup(s_i) \xrightarrow{abs(e)} sup(s'_i)$ we get $sup(s'_i) = sup(s_i) + abs(e) \bmod (b + 1)$ such that $sup(s'_i) = sup(s_i) - m + n \bmod (b + 1)$. Consequently, $s_\tau = sup(s'_i)$ implying $sup(s_i) \xrightarrow{(m, n)} sup(s'_i)$ making (sup, sig) a τ -region.

Moreover, by $0 \leq sup(s) \leq m - 1$ or $b + m - n + 1 \leq sup(s) \leq b$ we have that $\neg sup(s) \xrightarrow{(m, n)}$ such that (sup, sig) τ -solves (e, s) .

Only-If: Let (sup, sig) be a τ -region that solves (e, s) . In accordance to Lemma 6, we define the τ -abstract region (sup, abs) originating from (sup, sig) . We argue that (sup, abs) satisfies the conditions (1)–(3).

As (sup, sig) τ -solves (e, s) there is an event $(m, n) \in E_\tau$ such that $sup(s_i) \xrightarrow{(m, n)}$, $i \in \{1, \dots, k\}$, and $\neg sup(s) \xrightarrow{(m, n)}$. By abs 's definition, $abs = -m + n \bmod (b + 1)$ implying the first condition. Moreover, (sup, abs) satisfies the path equation. Hence, we have $sup(s_i) = sup(s_{A,0}) + \psi_{s_i} \cdot abs \bmod (b + 1)$ implying $\psi_{s_i} \cdot abs = sup(s_i) - sup(s_{A,0}) \bmod (b + 1)$ for $i \in \{1, \dots, k\}$. Furthermore, by $sup(s_i) \xrightarrow{(m, n)}$ and τ 's definition, we have $m \leq sup(s_i) \leq b + m - n$. Thus, the second condition is satisfied. Similarly, the path equation implies $\psi_s \cdot abs = sup(s) - sup(s_{A,0}) \bmod (b + 1)$ and, by $\neg sup(s) \xrightarrow{(m, n)}$, we obtain

$0 \leq \text{sup}(s) \leq m - 1$ or $b + m - n + 1 \leq \text{sup}(s) \leq b$. Hence, the third condition is also true. \square

The following lemma shows that deciding the τ -ESSP for A is only exponential in parameter k but polynomial in the size of the input.

Lemma 10. *If $\tau \in \{\tau_2^b, \tau_3^b\}$, then to decide for a k -fold TS A whether it has the τ -ESSP is possible in time $\mathcal{O}((b + 1)^{k+4} \cdot |E_A|^4 \cdot |S_A|^5)$.*

Proof. An ESSP atom (e, s) of A is a τ -solvable if and only if there is an abstract region (sup, abs) of A that satisfying the conditions of Lemma 9. Using Lemma 9, to decide the solvability of (e, s) we iteratively construct systems of linear equations M_α . There is an abstract region (sup, abs) , fulfilling the conditions, if and only if at least one M_α is solvable by abs . A single system to be computed modulo $b + 1$ is obtained as follows:

Firstly, it implements the basic part $M_{A'}$ requiring at most $|E| \cdot |S_A|^2 - 1$ equations.

Secondly, we choose an event $(m, n) \in E_A$ and a value $\text{sup}(s_{A,0}) \in \{0, \dots, b\}$ and, in accordance to Lemma 9.1, set $\text{abs}(e) = -m + n \pmod{b + 1}$. Thus, the number of unknown becomes $|E_A| - 1$.

Thirdly, in accordance to Lemma 9.2, we choose for every source s' of e in A ($s' \xrightarrow{e}$) a value $\text{sup}(s')$ satisfying $m \leq \text{sup}(s') \leq b + m - n$. After that we add the equation $\psi_{s'} \cdot \text{abs} = \text{sup}(s') - \text{sup}(s_{A,0})$. By definition of k , there are at most k sources of e . This yields at most k additional equations.

Finally, we choose $\text{sup}(s)$ such that $0 \leq \text{sup}(s) \leq m - 1$ or $b + m - n + 1 \leq \text{sup}(s) \leq b$, respectively. Then we add the equation $\psi_s \cdot \text{abs} = \text{sup}(s) - \text{sup}(s_{A,0})$. Now, a solution satisfies the condition of Lemma 9.3.

Altogether, by Lemma 9, this defines a fitting system whose solvability proves the τ -solvability of (e, s) . Moreover, the system has at most $|E_A| \cdot |S_A|^2 + k \leq 2 \cdot |E_A| \cdot |S_A|^2$ equations.

We estimate how many such systems must be maximally resolved for a single atom: By definition of τ , we have at most $(b + 1)^2$ possible choices for $(m, n) \in \{0, \dots, b\}^2$, and at most $b + 1$ different values for $\text{sup}(s_{A,0}) \in \{0, \dots, b\}$, respectively. Furthermore, having (m, n) and $\text{sup}(s_{A,0})$ already chosen, there are at most $b + 1$ possible choices for $\text{sup}(s)$ with $0 \leq \text{sup}(s) \leq m - 1$ or $b + m - n + 1 \leq \text{sup}(s) \leq b$. Similarly, for every source s' of e we have at most $b + 1$ choices for $\text{sup}(s')$ with $m \leq \text{sup}(s') \leq b + m - n$. By definition of k this makes at most $(b + 1)^k$ different possible choices for the sources of k . Altogether, we have at most $(b + 1)^{k+4}$ possibilities to define a system of linear equations whose solvability implies the τ -solvability of (e, s) . Moreover, each system has at most size $2 \cdot |E_A| \cdot |S_A|^2$.

Hence, by Lemmas 2 and 9 and $|E_A| \leq |E_A| \cdot |S_A|^2$, we can decide in time $\mathcal{O}((b + 1)^{k+4} \cdot |E_A|^3 \cdot |S_A|^4)$ if the atom (e, s) is τ -solvable. Consequently, by the at most $|E_A| \cdot |S_A|$ different ESSP atoms of A , we can decide whether A has the τ -ESSP in $\mathcal{O}((b + 1)^{k+4} \cdot |E_A|^4 \cdot |S_A|^5)$ time. \square

If $\tau \in \{\tau_2^b, \tau_3^b\}$ then, by Lemmas 5 and 9, deciding if a TS A has the τ -SSP and the τ -ESSP is doable in time $\mathcal{O}(|E_A|^3 \cdot |S_A|^6)$ and $\mathcal{O}((b+1)^{k+4} \cdot |E_A|^4 \cdot |S_A|^5)$, respectively. Thus, the following corollary is justified and completes the proof of Theorem 2.

Corollary 2. *If $\tau \in \{\tau_2^b, \tau_3^b\}$ then to decide if a TS A has the τ -feasibility is doable in time $\mathcal{O}((b+1)^{k+4} \cdot |E_A|^3 \cdot |S_A|^5 \cdot \max\{|E_A|, |S_A|\})$.*

5 Conclusion

In this paper, we investigate the computational complexity of synthesis, feasibility, ESSP and SSP for several types of b -bounded P/T-nets, $b \in \mathbb{Z}_{b+1}$. We introduce the new Petri net type of restricted \mathbb{Z}_{b+1} -extended b -bounded P/T-nets and show that for this type synthesis and all corresponding decision problems are solvable in polynomial time. Moreover, we show that SSP is decidable in polynomial time for the types of (pure) \mathbb{Z}_{b+1} -extended b -bounded P/T-nets. Finally, we prove that feasibility and ESSP for (pure) \mathbb{Z}_{b+1} -extended b -bounded P/T-nets are fixed parameter tractable if the (maximum) number of occurrences of events is considered as parameter.

It remains for future work to search for other parameters that makes feasibility for Petri net types fixed parameter tractable. Moreover, the question whether there are tractable superclasses of (pure) b -bounded P/T-nets is still open. One might also investigate the computational complexity for other Petri nets related synthesis problems: The exact complexity status of synthesis up to *language equivalence* is unknown. In [3], Badouel et al. proposed an algorithm that requires exponential space. Another open question has been stated in [15]: Schlachter et al. suggested to characterize the complexity of synthesis for b -bounded P/T-nets from *modal* transitions systems. Here, the task is to find, for a given modal TS M , a Petri net N that implements M . So far, we are at least aware of some (new) lower and upper bounds:

Conjecture 1. Let $b \geq 2$. Deciding, for a given TS A , if there is a (pure) b -bounded P/T-net N such that its state graph has the same language as A is NP-hard. Moreover, the problem is in PSPACE. To decide for a given modal TS M if there exists a (pure) b -bounded P/T-net N that implements M is NP-hard.

Acknowledgements. I would like to thank Uli Schlachter for his helpful remarks and for simplifying the proof of Lemma 1. Also, I'm thankful to the anonymous reviewers for their valuable comments.

References

1. van der Aalst, W.M.P.: Process Mining - Discovery, Conformance and Enhancement of Business Processes. Springer, Heidelberg (2011). <https://doi.org/10.1007/978-3-642-19345-3>
2. Agostini, A., De Michelis, G.: Improving flexibility of workflow management systems. In: van der Aalst, W., Desel, J., Oberweis, A. (eds.) Business Process Management. LNCS, vol. 1806, pp. 218–234. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45594-9_14
3. Badouel, E., Bernardinello, L., Darondeau, P.: Polynomial algorithms for the synthesis of bounded nets. In: Mosses, P.D., Nielsen, M., Schwartzbach, M.I. (eds.) CAAP 1995. LNCS, vol. 915, pp. 364–378. Springer, Heidelberg (1995). https://doi.org/10.1007/3-540-59293-8_207
4. Badouel, E., Bernardinello, L., Darondeau, P.: The synthesis problem for elementary net systems is NP-complete. Theor. Comput. Sci. **186**(1–2), 107–134 (1997). [https://doi.org/10.1016/S0304-3975\(96\)00219-8](https://doi.org/10.1016/S0304-3975(96)00219-8)
5. Badouel, E., Bernardinello, L., Darondeau, P.: Petri Net Synthesis. Texts in Theoretical Computer Science. An EATCS Series. Springer, Heidelberg (2015). <https://doi.org/10.1007/978-3-662-47967-4>
6. Badouel, E., Caillaud, B., Darondeau, P.: Distributing finite automata through Petri net synthesis. Formal Asp. Comput. **13**(6), 447–470 (2002). <https://doi.org/10.1007/s001650200022>
7. Best, E., Darondeau, P.: Petri net distributability. In: Clarke, E., Virbitskaite, I., Voronkov, A. (eds.) PSI 2011. LNCS, vol. 7162, pp. 1–18. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29709-0_1
8. Best, E., Devillers, R.: Characterisation of the state spaces of live and bounded marked graph Petri nets. In: Dediu, A.-H., Martín-Vide, C., Sierra-Rodríguez, J.-L., Truthe, B. (eds.) LATA 2014. LNCS, vol. 8370, pp. 161–172. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-04921-2_13
9. Cortadella, J., Kishinevsky, M., Kondratyev, A., Lavagno, L., Yakovlev, A.: A region-based theory for state assignment in speed-independent circuits. IEEE Trans. CAD Integr. Circ. Syst. **16**(8), 793–812 (1997). <https://doi.org/10.1109/43.644602>
10. Downey, R.G., Fellows, M.R.: Fundamentals of Parameterized Complexity. Texts in Computer Science. Springer, London (2013). <https://doi.org/10.1007/978-1-4471-5559-1>
11. Goldmann, M., Russell, A.: The complexity of solving equations over finite groups. Inf. Comput. **178**(1), 253–262 (2002). <https://doi.org/10.1006/inco.2002.3173>
12. Hiraishi, K.: Some complexity results on transition systems and elementary net systems. Theor. Comput. Sci. **135**(2), 361–376 (1994). [https://doi.org/10.1016/0304-3975\(94\)90112-0](https://doi.org/10.1016/0304-3975(94)90112-0)
13. Holloway, L.E., Krogh, B.H., Giua, A.: A survey of Petri net methods for controlled discrete event systems. Discret. Event Dyn. Syst. **7**(2), 151–190 (1997). <https://doi.org/10.1023/A:1008271916548>
14. Schlachter, U.: (2019, private correspondence)
15. Schlachter, U., Wimmel, H.: k-bounded Petri net synthesis from modal transition systems. In: CONCUR. LIPIcs, vol. 85, pp. 6:1–6:15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2017). <https://doi.org/10.4230/LIPIcs.CONCUR.2017.6>

16. Schmitt, V.: Flip-flop nets. In: Puech, C., Reischuk, R. (eds.) STACS 1996. LNCS, vol. 1046, pp. 515–528. Springer, Heidelberg (1996). https://doi.org/10.1007/3-540-60922-9_42
17. Tarjan, R.E.: Finding optimum branchings. *Networks* **7**(1), 25–35 (1977). <https://doi.org/10.1002/net.3230070103>
18. Tredup, R.: Hardness results for the synthesis of b -bounded Petri nets. In: Donatelli, S., Haar, S. (eds.) PETRI NETS 2019. LNCS, vol. 11522, pp. 127–147. Springer, Cham (2019)
19. Tredup, R., Rosenke, C.: Narrowing down the hardness barrier of synthesizing elementary net systems. In: CONCUR. LIPIcs, vol. 118, pp. 16:1–16:15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2018). <https://doi.org/10.4230/LIPIcs.CONCUR.2018.16>
20. Tredup, R., Rosenke, C.: The complexity of synthesis for 43 Boolean Petri net types. In: Gopal, T.V., Watada, J. (eds.) TAMC 2019. LNCS, vol. 11436, pp. 615–634. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-14812-6_38
21. Tredup, R., Rosenke, C., Wolf, K.: Elementary net synthesis remains NP-complete even for extremely simple inputs. In: Khomenko, V., Roux, O.H. (eds.) PETRI NETS 2018. LNCS, vol. 10877, pp. 40–59. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-91268-4_3