Synthesis of Bounded Choice-Free Petri Nets

Eike Best[1] **and Raymond Devillers**[2]

1 **Department of Computing Science, Carl von Ossietzky Universität Oldenburg, Germany**
eike.best@informatik.uni-oldenburg.de

2 **Département d'Informatique, Université Libre de Bruxelles, Belgium**
rdevil@ulb.ac.be

Abstract

This paper describes a synthesis algorithm tailored to the construction of choice-free Petri nets from finite persistent transition systems. With this goal in mind, a minimised set of simplified systems of linear inequalities is distilled from a general region-theoretic approach, leading to algorithmic improvements as well as to a partial characterisation of the class of persistent transition systems that have a choice-free Petri net realisation.

1998 ACM Subject Classification F.1.1 Models of Computation

Keywords and phrases Choice-Freeness, Labelled Transition Systems, Persistence, Petri Nets, System Synthesis

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2015.128

1 Introduction, some examples, and basic notation

In system analysis, the main task is to examine a given system's properties by means of a behavioural description. By contrast, in system synthesis, the task is to construct – preferably automatically – an implementing system from a given behavioural specification. The benefit of such an approach is that a successfully synthesised system is "correct by design". There is no need to re-examine its behavioural properties, because they are known to hold by construction. If synthesis fails, this may also help to delineate the true reasons of the failure, paving the way to modifications of the given input behaviour allowing for a more successful subsequent synthesis.

Synthesis is being applied in many different areas (e.g., [11, 19]). In general, however, since behavioural descriptions may be extremely (even infinitely) large, synthesis algorithms could be impossible to obtain by theoretical undecidability [14], or at least be very time-consuming. Also, synthesis suffers from nondeterminism, since for a given behavioural specification, many different implementations may exist. Moreover, if there is a desire for an implementation to enjoy further properties, detecting the existence of a suitable one (if possible) tends to increase the difficulty of a synthesis problem.

We investigate a special, decidable instance of system synthesis. It is assumed that a behavioural specification is given in the form of a finite, edge-labelled transition system, or *lts*, for short. For example, we could be interested in the transition system TS_1 shown on the left-hand side of Figure 1. We shall be asking whether or not such an lts can be implemented by an unlabelled Petri net having a specific shape. The shape we shall be aiming at is *choice-freeness*, meaning that every place has at most one outgoing transition. For example, both Petri nets N_1 and N_1' shown in Figure 1 implement TS_1, in the sense that their reachability graphs are isomorphic to TS_1. However, N_1 is choice-free while N_1' is not.

26th International Conference on Concurrency Theory (CONCUR 2015).
Editors: Luca Aceto and David de Frutos Escrig; pp. 128–141

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

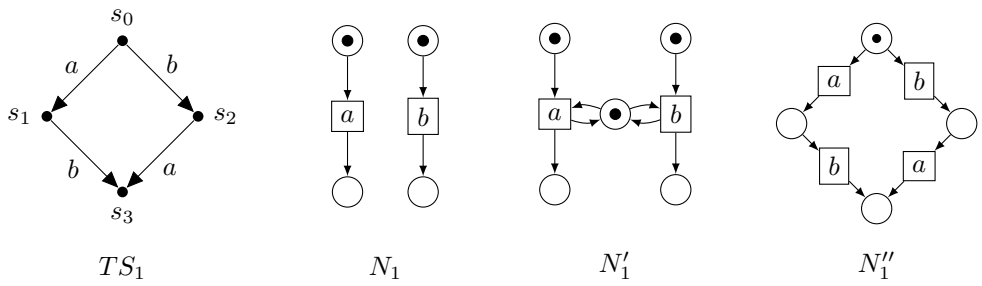

Figure 1 The lts TS_1 is solved by the Petri net N_1. It is also solved by N_1'. The net N_1'' is not (in this paper) accepted as a solution of TS_1 because its transitions are non-injectively labelled.

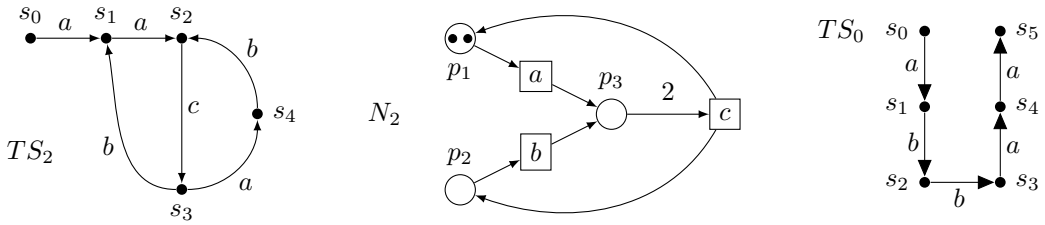

Figure 2 The net N_2 solves TS_2. No plain solution of TS_2 exists. No solution of TS_0 exists.

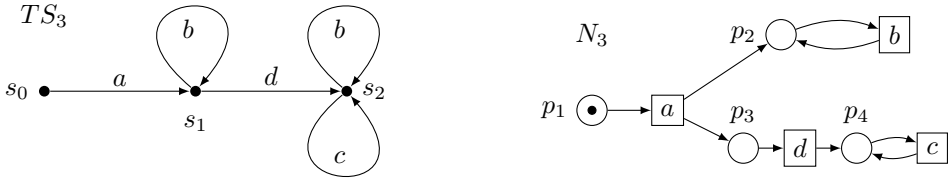

Figure 3 The Petri net N_3 solves the lts TS_3. No pure solution of TS_3 exists.

Being related to arbiter-freeness [16], choice-freeness is interesting in a digital design context [11]. Choice-free Petri nets are also precisely the class of nets allowing a fully distributed [1] implementation. The problem has been addressed and solved for special classes of choice-free nets in previous papers by the present authors, as follows: for connected marked graphs and T-systems in [5, 7]; for bounded, reversible choice-free nets (i.e., where it is always possible to come back to the initial state) in [6, 8]; and for connected, bounded, live choice-free nets (i.e., where no transition may become dead) in [9]. In the present paper, this framework will be generalised to bounded choice-free nets, also allowing for non-live transitions.

We shall be concerned with exact synthesis, disallowing that two or more transitions carry the same label. This excludes nets such as N_1'' in Figure 1 as implementations. Moreover, we shall take into consideration the full class of place/transition systems [18]. For example, the lts TS_2 depicted in Figure 2 can be solved by N_2 with an arc having weight 2 from p_3 to c, but not by any plain (meaning: having arc weights at most 1) Petri net. Similarly, the lts TS_3 shown in Figure 3 can be solved by N_3, but not by a pure (meaning: side-place free) Petri net. Observe that there are also specifications which cannot be implemented by any unlabelled Petri net, such as the lts TS_0 shown on the right-hand side of Figure 2 ([3]). The proofs of partial or full unsolvability are not hard and are left to the reader; [10] may help.

For easy reference, basic formal definitions are summarised in the remainder of this section. Important concepts with strong impact on the formal development of this paper will

be introduced in-line, that is, in the text explaining their relevance. To facilitate spotting them, such notions will be emphasised in *italic* at the point of their formal introduction.

▶ **Definition 1.1** (Basic notations and conventions used in this paper)**.** A *finite labelled transition system* with initial state is a tuple $TS = (S, \rightarrow, T, s_0)$ with nodes S (a finite set of states), edge labels T, edges $\rightarrow \subseteq (S \times T \times S)$, and an initial state $s_0 \in S$. A label t is enabled at $s \in S$, written formally as $s[t\rangle$, if $\exists s' \in S\colon (s,t,s') \in \rightarrow$, and backward enabled at s, written as $[t\rangle s$, if $\exists s' \in S\colon (s',t,s) \in \rightarrow$. A state s' is reachable from s through the execution of $\sigma \in T^*$, denoted by $s[\sigma\rangle s'$, if there is a directed path from s to s' whose edges are labelled consecutively by σ. The set of states reachable from s is denoted by $[s\rangle$. A (firing) sequence $\sigma \in T^*$ is allowed from a state s, denoted by $s[\sigma\rangle$, if there is some state s' such that $s[\sigma\rangle s'$. The language of TS is the set $L(TS) = \{\sigma \in T^* \mid s_0[\sigma\rangle\}$. Two lts $TS_1 = (S_1, \rightarrow_1, T, s_{01})$ and $TS_2 = (S_2, \rightarrow_2, T, s_{02})$ are language-equivalent if $L(TS_1) = L(TS_2)$, and isomorphic if there is a bijection $\zeta\colon S_1 \rightarrow S_2$ with $\zeta(s_{01}) = s_{02}$ and $(s,t,s') \in \rightarrow_1 \Leftrightarrow (\zeta(s),t,\zeta(s')) \in \rightarrow_2$, for all $s, s' \in S_1$.

An *initially marked Petri net* is denoted as $N = (P, T, F, M_0)$ where P is a finite set of places, T is a finite set of transitions, F is the flow function $F\colon ((P \times T) \cup (T \times P)) \rightarrow \mathbb{N}$ specifying the arc weights, and M_0 is the initial marking (where a marking is a mapping $M\colon P \rightarrow \mathbb{N}$, indicating the number of tokens in each place). N is plain if no arc weight exceeds 1; pure or side-place free if $\forall p \in P\colon (p^\bullet \cap {}^\bullet p) = \emptyset$, where $p^\bullet = \{t \in T \mid F(p,t){>}0\}$ and ${}^\bullet p = \{t \in T \mid F(t,p){>}0\}$; and CF (*choice-free* [12, 20]) or ON (place-output-nonbranching [8]) if $\forall p \in P\colon |p^\bullet| \leq 1$. A transition $t \in T$ is enabled at a marking M, denoted by $M[t\rangle$, if $\forall p \in P\colon M(p) \geq F(p,t)$. The firing of t leads from M to M', denoted by $M[t\rangle M'$, if $M[t\rangle$ and $M'(p) = M(p) - F(p,t) + F(t,p)$. This can be extended, as usual, to $M[\sigma\rangle M'$ for sequences $\sigma \in T^*$, and $[M\rangle$ denotes the set of markings reachable from M. The reachability graph $RG(N)$ of N is the labelled transition system with the set of vertices $[M_0\rangle$, initial state M_0, label set T, and set of edges $\{(M,t,M') \mid M, M' \in [M_0\rangle \wedge M[t\rangle M'\}$. If an lts TS is isomorphic to the reachability graph of a Petri net N, we say that N *solves* TS. All notions defined for labelled transition systems apply to Petri nets through their reachability graphs. ◀ 1.1

2 Basic synthesis of Petri nets, recapitulated

Certain classes of lts can be excluded from consideration up front. For instance, it might happen that in some transition system $TS = (S, \rightarrow, T, s_0)$, some state s is not reachable from the initial state s_0. Such an lts can never be solved by a Petri net, since the reachability graph is defined by adding all reachable markings (and no others). Hence we shall adopt, for TS, *total reachability*, meaning that $\forall s \in S\colon s \in [s_0\rangle$.

Nondeterminism can never occur in the reachability graph of a Petri net, because if $M[t\rangle M'$, then the successor marking M' is uniquely determined by M and t. Generalising this, let $\Psi(\sigma)$ denote the *Parikh vector* of a sequence $\sigma \in T^*$, i.e., the vector with index set T which returns the number of occurrences of $t \in T$ in σ, and call an lts *strongly deterministic* if, whenever $\Psi(\tau) = \Psi(\tau')$ and either $s[\tau\rangle s'$ and $s[\tau'\rangle s''$ or $s'[\tau\rangle s$ and $s''[\tau'\rangle s$, then $s' = s''$.

If some sequence τ is cyclic in the reachability graph of a Petri net, i.e., leads from some marking M to itself, $M[\tau\rangle M$, then (according to standard Petri net theory) any sequence which is Parikh-proportional to τ is also cyclic, at any marking at which it is enabled. Let us call an lts *strongly cycle-consistent* if the same property holds for it.

For the sake of brevity, let us call an lts *decent* if it is totally reachable, strongly deterministic, and strongly cycle-consistent. A non-decent lts has no possible Petri net

solution, and it can therefore be interesting to first check this structural constraint in order to avoid applying uselessly a costly regional analysis. This observation is exploited by some tools (e.g. [21]).

A very general (but also expensive) algorithm for synthesising a Petri net from a labelled transition system can be described as follows.

- For a given finite and decent lts $TS = (S, \rightarrow, T, s_0)$, we start constructing a net by letting it have T as its set of transitions, and no places. Such a net has the language T^*, which (normally) contains many more words than $L(TS)$. In order to exclude words disallowed in TS, and in order to guarantee a bijection between TS and the reachability graph of the hoped-for net N, a place will be introduced for every separation problem in TS as follows.
- An *event/state separation problem* consists of an ordered pair $(s,t) \in S \times T$ with $\neg(s[t\rangle)$. There are at most $|S| \cdot |T|$ such problems. For every event/state separation problem, N needs to have at least one place p such that $M(p) < F(p,t)$ for the marking M corresponding to state s, where $F(p,t)$ is the weight of the arc from p to t.
- A *state separation problem* consists of a pair of states $\{s, s'\}$ with $s \neq s'$. There are $\frac{1}{2} \cdot (|S| \cdot (|S|-1))$ such problems. For every state separation problem, N needs to contain at least one place p such that $M(p) \neq M'(p)$ for the markings M and M' corresponding to states s and s', respectively.
- The notion of a "place" is not known for TS. A *region* [2] of an lts $(S, \rightarrow, T, s_0)$ is a triple $(\mathbb{R}, \mathbb{B}, \mathbb{F}) \in (S \rightarrow \mathbb{N}, T \rightarrow \mathbb{N}, T \rightarrow \mathbb{N})$ such that for all $s[t\rangle s'$, $\mathbb{R}(s) \geq \mathbb{B}(t)$ and $\mathbb{R}(s') = \mathbb{R}(s) - \mathbb{B}(t) + \mathbb{F}(t)$. A region models a place p, in the sense that $\mathbb{B}(t)$ models $F(p,t)$, $\mathbb{F}$ models $F(t,p)$, and $\mathbb{R}(s)$ models the token count of p in the marking corresponding to s.
- A straightforward algorithm inspects every separation problem in turn and tries to solve a linear inequality system for it. The unknowns are the arc weights of a place p with respect to every transition in T, and the initial marking $M_0(p)$. The inequality system arises from the need to guarantee the region properties (giving rise to many inequalities), and from the need to guarantee a separation property (giving rise to one or two additional inequalities). If these systems are solvable for every separation problem, we find a net which is isomorphic to TS, otherwise such a net does not exist. If they are solvable for every event/state separation problem, then we can construct a Petri net which is language-equivalent to TS.

Thus, in general, we need to solve $O(|S|^2)$ inequality systems, each with more than $2 \cdot |T|$ unknowns. In the present paper, we ask whether a given finite, decent lts has a choice-free Petri net solution. If such a requirement is added, the algorithm could become more complex; but the aim of this paper is to demonstrate that, knowing what solutions we are looking for may also work the other way, namely focussing the search and speeding up the region-based general algorithm.

The reachability graphs of choice-free Petri nets necessarily satisfy an additional set of properties which are not shared by all finite labelled transition systems, even if they are decent. This excludes many decent ones from consideration. In the next Section 3, we shall gather a set of such properties. In Section 4, we describe how these properties and the special shape of the places (and regions) of a choice-free net can be exploited in order to simplify the region inequalities.

3 Persistent transition systems, small cycles, and CF Petri nets

The reachability graphs of choice-free Petri nets are persistent, and persistent lts enjoy a property of small cycles, as will be described next.

An lts $TS = (S, \to, T, s_0)$ is called *persistent* [17] if, whenever $s[a\rangle$ and $s[b\rangle$ with $a \neq b$, then also $s[ab\rangle s'$ and $s[ba\rangle s'$ for some common state s'. For example, all of the lts shown in figures 1–3 (including TS_0) are persistent. Any choice-free net $N = (S, T, F, M_0)$ is persistent, because if $a \neq b$ for $a, b \in T$, then there is no common pre-place p of a and b, i.e., for all $p \in P$, either $F(p, a) = 0$ or $F(p, b) = 0$, or both, which directly entails the persistence property, if we add the strong determinism of any Petri net.

The property of small cycles generalises the following observations. First, define a *home state* of TS to be a state $\widetilde{s} \in S$ which satisfies $\forall s \in [s_0\rangle \colon \widetilde{s} \in [s\rangle$. Finite persistent lts always have at least one home state (with an easy proof; see, e.g. corollary 2 of [4]). The sets of home states of TS_0, TS_1, TS_2 and TS_3 of figures 1–3 are, respectively, $\{s_5\}$, $\{s_3\}$, $\{s_1, s_2, s_3, s_4\}$, and $\{s_2\}$. Next, define a nontrivial cycle $s[\sigma\rangle s$ around a state $s \in [s_0\rangle$ to be *small* if there is no nontrivial cycle $s'[\sigma'\rangle s'$ with $s' \in [s_0\rangle$ and $\Psi(\sigma') \lneqq \Psi(\sigma)$, where $\lneqq = \ (\leq \cap \neq)$. For example, in TS_2, both $s_3[bac\rangle s_3$ and $s_3[abc\rangle s_3$, and also $s_1[acb\rangle s_1$ (and others) are small cycles, but $s_1[acabcb\rangle s_1$ (and others) are not. Notice that in TS_2, all small cycles have the same Parikh vector. In TS_3, by contrast, $s_1[b\rangle s_1$ is small, $s_2[b\rangle s_2$ is small, $s_2[c\rangle s_2$ is small, but $s_2[bc\rangle s_2$ is not. Notice that in TS_3, all small cycles either have the same Parikh vectors, or are label-disjoint, where two Parikh vectors are *label-disjoint* if their supports are disjoint, and the *support* of a Parikh vector $\Psi \colon T \to \mathbb{N}$ is defined as the set $\{t \in T \mid \Psi(t) > 0\}$.

This property is general, as follows.

▶ **Theorem 3.1** (Cycle decomposition at home states). *Let $TS = (S, \to, T, s_0)$ be a finite, decent, persistent lts. Then there exist a state $\widetilde{s} \in [s_0\rangle$ and a finite set $\mathcal{C} = \{\widetilde{s}[\rho_i\rangle\widetilde{s} \mid 1 \leq i \leq n\}$ of mutually label-disjoint small cycles around $\widetilde{s}$, with $n \leq |T|$, such that for any state $s \in [s_0\rangle$, the Parikh vector of any cycle $s[\rho\rangle s$ decomposes as $\Psi(\rho) = \sum_{i=1}^{n} k_i \cdot \Psi(\rho_i)$ for some $k_i \in \mathbb{N}$.*

Proof. From theorem 2 of [4], observing that the preconditions of this result are implied by finiteness, decency, and persistence, and keeping in mind that small cycles have been called hypersimple in [4]. ◀

In fact, the results of [4] also show that for $\widetilde{s}$, any home state can be chosen and that the set of Parikh vectors in $\mathcal{C}$ is independent of the choice of home state. For example, in TS_2, we may choose $\widetilde{s} = s_1$ and $\mathcal{C} = \{s_1[acb\rangle s_1\}$ with $|\mathcal{C}| = 1$, and in TS_3, $\widetilde{s} = s_2$ is the only possible choice, and we get $\mathcal{C} = \{s_2[b\rangle s_2, s_2[c\rangle s_2\}$ with $|\mathcal{C}| = 2$. In TS_0 and TS_1, $\widetilde{s} = s_5$ and $\widetilde{s} = s_3$ (respectively), and we have $\mathcal{C} = \emptyset$ in both cases. If a persistent Petri net $N = (S, T, F, M_0)$ is *bounded*, i.e., has a finite reachability graph $RG(N)$, then it satisfies the premises of the previous theorem. Hence the cycles of $RG(N)$ can be decomposed in the same way.

The decomposition theorem has implications for the distribution of live transitions in TS. A label $t \in T$ is *live* if $\forall s \in [s_0\rangle \exists s' \in [s\rangle \colon s'[t\rangle$. If TS is finite, decent, and persistent, then live transitions are exactly those that can be found in one of the cycles in $\mathcal{C}$. All others (for instance, a and d in TS_3, but also a and b in TS_0) can never again be executed, once a home state has been reached. We cast these notions in the following definition.

▶ **Definition 3.2** (The small cycles property $\mathbf{P}\{\Upsilon_1, \ldots, \Upsilon_n\}$). Let $TS = (S, \to, T, s_0)$ be an lts. For some $n \in \mathbb{N}$ and for all $1 \leq i \leq n$, let Υ_i be a function $\Upsilon_i \colon T \to \mathbb{N}$ such that these functions are mutually label-disjoint. TS satisfies property $\mathbf{P}\{\Upsilon_1, \ldots, \Upsilon_n\}$ iff $\{\Upsilon_1, \ldots, \Upsilon_n\}$ is the set of Parikh vectors of small cycles of TS. ◀ 3.2

If an lts TS satisfies $\mathbf{P}\{\Upsilon_1, \ldots, \Upsilon_n\}$, then we shall henceforth denote by T_i the support of Υ_i and by T_0 the set $T \setminus (T_1 \cup \ldots \cup T_n)$. By definition, these $n+1$ sets are mutually disjoint. In case TS is finite, decent, and persistent, the set on non-live transitions is exactly T_0. For example, TS_2 and TS_0 in Figure 2 satisfy $\mathbf{P}\{\Upsilon_1\}$ and $\mathbf{P}\emptyset$, respectively, where Υ_1 maps a, b and c to 1. Also, T_0 is $\emptyset$ in TS_2 and $\{a, b\}$ in TS_0. In Figure 3, TS_3 satisfies $\mathbf{P}\{\Upsilon_1, \Upsilon_2\}$ where $\Upsilon_1 = (a \mapsto 0, b \mapsto 1, c \mapsto 0, d \mapsto 0)$, $\Upsilon_2 = (a \mapsto 0, b \mapsto 0, c \mapsto 1, d \mapsto 0)$, and $T_0 = \{a, d\}$.

A coherent notion of distance between states can be introduced as follows.

▶ **Definition 3.3** (Modulo vectors and distances). Let $TS = (S, \to, T, s_0)$ be a finite, decent, and persistent lts satisfying $\mathbf{P}\{\Upsilon_1, \ldots, \Upsilon_n\}$.
For a natural T-vector Υ, let

$$\Upsilon \bmod \{\Upsilon_1, \ldots, \Upsilon_n\} \quad = \quad \Upsilon - \sum_{i \in \{1,\ldots,n\}} \left(\min_{t \in T_i} \left(\Upsilon(t) \div \Upsilon_i(t) \right) \right) \cdot \Upsilon_i$$

be the natural T-vector obtained by subtracting each of the vectors Υ_i as often as possible from Υ (in the formula, $\div$ denotes integer division). Let $r \in S$, $s \in [r\rangle$ and $r[\alpha\rangle s$ be a path of TS. Then $\Delta_{r,s} = \Psi(\alpha) \bmod \{\Upsilon_1, \ldots, \Upsilon_n\}$ is called the *distance* between r and s. ◀ 3.3

The following lemma shows that $\Delta_{r,s}$ does not depend on the path chosen between r and s.

▶ **Lemma 3.4** (istances are well-defined). *Let $TS = (S, \to, T, s_0)$ be a finite, decent, and persistent lts satisfying $\mathbf{P}\{\Upsilon_1, \ldots, \Upsilon_n\}$.*
Let $r[\sigma\rangle s$ and $r[\sigma'\rangle s$ be two paths between the same states $r, s \in [s_0\rangle$.
Then $\Psi(\sigma) \bmod \{\Upsilon_1, \ldots, \Upsilon_n\} = \Psi(\sigma') \bmod \{\Upsilon_1, \ldots, \Upsilon_n\}$.

For the proof of this lemma, we use *Keller's theorem* [15], a basic tool for analysing persistent systems. For sequences $\sigma, \tau \in T^*$, let $\tau \stackrel{\bullet}{-} \sigma$ denote the sequence left after erasing successively in τ the leftmost occurrences of all symbols from σ, read from left to right. Keller's theorem states that in a deterministic and persistent lts, if $s[\tau\rangle$ and $s[\sigma\rangle$ for some $s \in [s_0\rangle$, then $\Psi(\tau(\sigma \stackrel{\bullet}{-} \tau)) = \Psi(\sigma(\tau \stackrel{\bullet}{-} \sigma))$ and $s[\tau(\sigma \stackrel{\bullet}{-} \tau)\rangle \widehat{s}$ and $s[\sigma(\tau \stackrel{\bullet}{-} \sigma)\rangle \widehat{s}$ for some state $\widehat{s} \in [s_0\rangle$.

Proof. Applied to $r[\sigma\rangle s$ and $r[\sigma'\rangle s$, Keller's theorem yields $s[\sigma \stackrel{\bullet}{-} \sigma'\rangle \widehat{s}$ and $s[\sigma' \stackrel{\bullet}{-} \sigma\rangle \widehat{s}$. The definition of $\stackrel{\bullet}{-}$ implies that $\sigma \stackrel{\bullet}{-} \sigma'$ and $\sigma' \stackrel{\bullet}{-} \sigma$ are label-disjoint. Lemma 4 in [4] states that in a finite, deterministic, strongly cycle-consistent and persistent lts, two paths between two different states always have a common label. This implies $s = \widehat{s}$.

Thus, both $s[\sigma \stackrel{\bullet}{-} \sigma'\rangle s$ and $s[\sigma' \stackrel{\bullet}{-} \sigma\rangle s$ are cyclic, and by Theorem 3.1, both $\Psi(\sigma \stackrel{\bullet}{-} \sigma')$ and $\Psi(\sigma' \stackrel{\bullet}{-} \sigma)$ are linear combinations of $\Upsilon_1, \ldots, \Upsilon_n$. On T_0, they both must be null, so that $\Psi(\sigma)$ and $\Psi(\sigma')$ coincide on T_0. On each T_i with $i \in \{1, \ldots, n\}$, since $\sigma \stackrel{\bullet}{-} \sigma'$ and $\sigma' \stackrel{\bullet}{-} \sigma$ are label-disjoint, at least one of them must be the empty sequence; hence, on T_i, one of $\Psi(\sigma)$ or $\Psi(\sigma')$ must be greater or equal to the other, by a multiple of Υ_i. ◀

In Figure 4, TS_4 satisfies $\mathbf{P}\{\Upsilon_1, \Upsilon_2\}$ where $\Upsilon_1 = (a \mapsto 0, b \mapsto 1, c \mapsto 1, d \mapsto 0, e \mapsto 0) = \Psi(bc)$ and $\Upsilon_2 = (a \mapsto 0, b \mapsto 0, c \mapsto 0, d \mapsto 1, e \mapsto 1) = \Psi(de)$. There are two Parikh-incomparable paths $s_0[bac\rangle s$ and $s_0[dae\rangle s$, and no smaller ones from s_0 to s. Yet the distance $\Delta_{s_0,s}$ is uniquely defined as the Parikh vector $\Delta_{s_0,s} = (a \mapsto 1, b \mapsto 0, c \mapsto 0, d \mapsto 0, e \mapsto 0) = \Psi(a)$, and it can be obtained either by subtracting Υ_1 from $\Psi(bac)$ or by subtracting Υ_2 from $\Psi(dae)$.

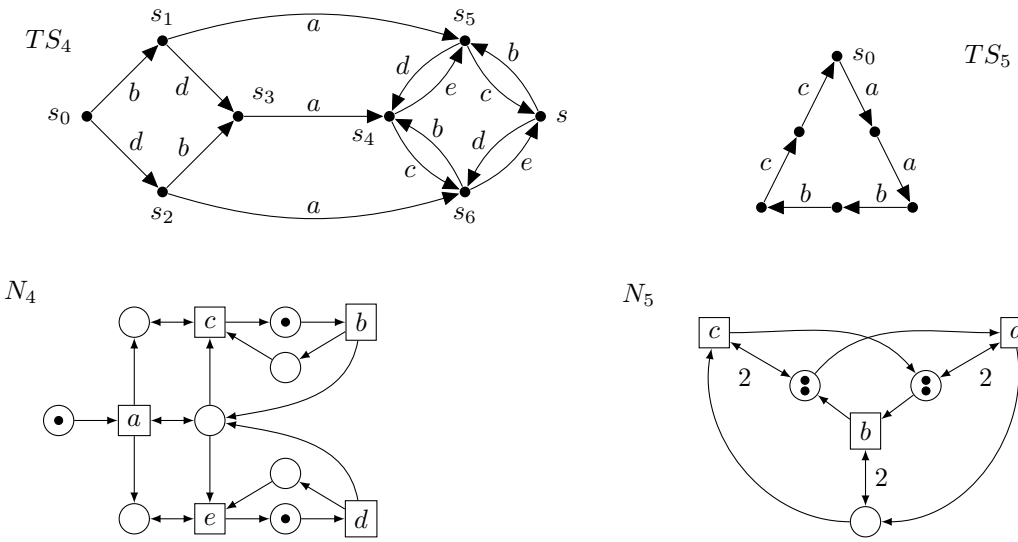

Figure 4 Upper part: Two lts which are PN-solvable but have no choice-free Petri net solutions. Lower part: Solutions N_4 of TS_4 (left-hand side) and N_5 of TS_5 (right-hand side).

The properties defined thus far (total reachability, determinism, cycle-consistency, persistence, and the small cycle property) are shared by all reachability graphs of bounded persistent Petri nets. The reachability graphs of bounded choice-free Petri nets enjoy further (stronger) properties, two of which, the prime cycle property and the distance property, are described in the remainder of this section. If any of these properties is violated for some lts, then it is certain that choice-free synthesis (to be defined in the next section) will fail for it.

▶ **Definition 3.5** (Prime cycles). Let $TS = (S, \to, T, s_0)$ be any lts and $s[\sigma\rangle s$ a cycle in it. This cycle is called *prime* if $\gcd\{\Psi(\sigma)(t) \mid t \in T\} = 1$ (where gcd denotes the greatest common divisor). ◀ 3.5

▶ **Lemma 3.6** (Prime cycle property). *In the reachability graph of a bounded choice-free net, all small cycles are prime.*

Proof. Lemma 16 in [20] states that in a choice-free net (P, T, F, M_0), if there is a T-semiflow X (i.e., a T-indexed vector $X \geq 0$ such that $\forall p \in P\colon \sum_{t \in T}(F(t,p) - F(p,t)) \cdot X(t) = 0$), and a firing sequence $M_0[\sigma\rangle M$ with $\Psi(\sigma) \geq X$, then it is possible to rearrange σ in such a way that $M_0[\sigma'\rangle M[\rho\rangle M$ with $\Psi(\rho) = X$ and $\Psi(\sigma'\rho) = \Psi(\sigma)$. This implies that at home states, non-prime cycles can be factored out into a nontrivial initial part followed by a prime cycle executing exactly X, and hence are not small. ◀

This is illustrated by TS_5 shown on the right-hand side of Figure 4. It is finite, decent, persistent, and satisfies $\mathbf{P}\{\Upsilon_1\}$ with $\Upsilon_1 = (a \mapsto 2, b \mapsto 2, c \mapsto 2)$. It is PN-solvable by N_5, as also shown in Figure 4, but, as a consequence of Lemma 3.6, may not be solved by a choice-free net.

▶ **Definition 3.7** (Parikh-minimal paths). Let $TS = (S, \to, T, s_0)$ be any lts, let $r \in S$, let $s \in [r\rangle$, and let $r[\sigma\rangle s$ be a path from r to s. The latter is called *Parikh-minimal* if there is no path $r[\sigma'\rangle s$ with $\Psi(\sigma') \lneqq \Psi(\sigma)$. ◀ 3.7

The following lemma implies that in choice-free nets, the distance $\Delta_{r,s}$ between two states r and s agrees with the minimal Parikh vector of any path from r to s. In particular, unlike in TS_4, there are no Parikh-incomparable Parikh-minimal paths between the same states.

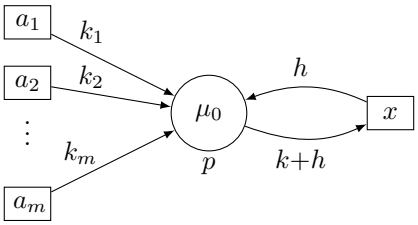

Figure 5 A general pure ($h = 0$) or non-pure ($h > 0$) choice-free place with initial marking μ_0.

▶ **Lemma 3.8** (Distance property). *In the reachability graph $RG(N)$ of a bounded choice-free net N, if $r[\sigma\rangle s$ is a Parikh-minimal path, then $\Psi(\sigma) = \Delta_{r,s}$. Also, if $q[\rho\rangle q$ is a small cycle in $RG(N)$, then $\Psi(\sigma) \not\geq \Psi(\rho)$.*

Proof. This again follows from an iterated application of Lemma 16 in [20]. ◀

For example, the lts TS_4 shown in Figure 4 is PN-solvable (by N_4, also shown in the figure), finite, decent, persistent, and satisfies $\mathbf{P}\{\Upsilon_1, \Upsilon_2\}$ as already mentioned, but may not be solved by a choice-free net, since, for instance, $s_0[bac\rangle s$ is Parikh-minimal but $\Psi(bac) \neq \Delta_{s_0,s} = \Psi(a)$; also, $\Psi(bac) \geq \Upsilon_1 = (a \mapsto 0, b \mapsto 1, c \mapsto 1, d \mapsto 0, e \mapsto 0)$.

4 Choice-free synthesis

In this section, it will be shown how the special form of the target places of a hoped-for choice-free Petri net solving a given lts can be exploited in order to reduce the number, and to simplify the shape, of the linear inequality systems that need to be solved. Concretely, let us consider an lts $TS = (S, \to, T, s_0)$ which is finite, decent, persistent, and satisfies $\mathbf{P}\{\Upsilon_1, \ldots, \Upsilon_n\}$. We shall analyse when and how one can synthesise a corresponding choice-free net $N = (P, T, F, M_0)$. As before, the set of transitions of N is the same as the set of labels of TS. Since N is intended to be choice-free, every place $p \in P$ has the general form shown in Figure 5, with $x \in T$ as its only outgoing transition, and $\{a_1, \ldots, a_m\} = T \setminus \{x\}$. All arc weight parameters $k, h, k_1, k_2, \ldots, k_m$ and the initial marking μ_0 of p are required to be semipositive, and they are the unknowns of the synthesis. If p is used for preventing x at some state s, i.e., for solving some event/state separation problem $\neg(s[x\rangle)$, we must have $k + h > 0$; but, for the time being, any of these parameters could also be zero.
Let $T = T_0 \uplus T_1 \uplus \ldots \uplus T_n$ be the partition of T induced by $\mathbf{P}\{\Upsilon_1, \ldots, \Upsilon_n\}$. For $0 \leq i \leq n$, we denote by $I_i = \{j \mid a_j \in T_i\}$ the indices of transitions a_j for which $\Upsilon_i(a_j) > 0$.
In the following, we shall also denote by ℓ the unique index such that $x \in T_\ell$.

Ensuring that cycles are preserved. Since the net effect of firing x on p is $-k = (-(k+h)+h)$ and all Parikh vectors in $\{\Upsilon_1, \ldots, \Upsilon_n\}$ are cyclic, we must have

$$\forall i \in \{1, \ldots, n\} : \quad \sum_{j \in I_i} k_j \cdot \Upsilon_i(a_j) \quad = \quad k \cdot \Upsilon_i(x) \tag{1}$$

ensuring that if every transition t is fired $\Upsilon_i(t)$ times, the marking on p is reproduced. Note that this implies $k \geq 0$, and even $k > 0$ unless all the k_j's for $j \in I_i$ are null.

If $x \in T_0$, i.e., $\ell = 0$, all the right-hand sides of (1) are null, so that all k_j for $j \notin I_0$ must be null too; in other words, if $p^\bullet \subseteq T_0$, then ${}^\bullet p \subseteq T_0$. Thus, if x is non-live, then all transitions in ${}^\bullet p$ are also non-live. If $x \in T_\ell$ for $\ell \in \{1, \ldots, n\}$, the right-hand sides of (1) are

null when $i \in \{1,\dots,n\} \setminus \{\ell\}$, so that all k_j for $j \notin I_0 \cup I_\ell$ must be null too; in other words, if $p^\bullet \subseteq T_\ell$, then ${}^\bullet p \subseteq T_0 \cup T_\ell$. Thus, if x is live and part of a small cycle, then all transitions in ${}^\bullet p$ are either non-live, or live and part of the same small cycle.

Ensuring that the marking on p does not prevent enabled transitions. By the shape of the place shown in Figure 5, by ${}^\bullet p \subseteq T_0 \cup T_\ell$, and by the firing rule, the marking of place p at (the marking corresponding to) an arbitrary state $r \in [s_0\rangle$ is

$$M_r(p) \;=\; \mu_0 + \sum_{j \in I_0 \cup I_\ell} k_j \cdot \Delta_{s_0,r}(a_j) - k \cdot \Delta_{s_0,r}(x) \tag{2}$$

This sum must always be nonnegative. Let us start by analysing what it means that p may never prevent any enabled x. Thus, consider an edge $r[x\rangle r'$ in TS. Then the marking $M_r(p)$ has to be at least $k+h$, and $M_{r'}(p)$ is at least h. More generally, if $l > 0$ and $r[x^l\rangle r'$, then we must have $M_r(p) \geq l \cdot k + h$, or equivalently, $M_{r'}(p) \geq h$. For this reason, when considering the marking of p at a state r with $r[x\rangle$, we first try to follow x-chains in forward direction as long as possible. If $r[x\rangle r'[x\rangle r''$ with $r \neq r'$, then by strong determinism, r'' is necessarily different from r and from r', so that x-chains starting with two different states can never hit an x-cycle, nor an x-branch, and necessarily have a unique last state. We may have infinite x-paths, but only in case $r[x\rangle r$ (like those for b or for c in Figure 3), where state r can never be left by an x-edge. Thus, we are interested in the following subset of states:

$$XNX(x) \;=\; \{r \in S \mid [x\rangle r \wedge \neg r[x\rangle\} \;\cup\; \{r \in S \mid r[x\rangle r\}$$

which either are produced by x but do not enable x, or have an x-loop. The above considerations amount to a proof of the following corollary:

▶ **Corollary 4.1** (XNX and enabling condition). $\big(\forall r \in S\colon r[x\rangle \Rightarrow M_r(p) \geq k+h\big) \iff \big(\forall r \in XNX(x)\colon M_r(p) \geq h\big)$.

In other words, we only need to require $M_r(p) \geq h$ for states $r \in XNX(x)$ in order to guarantee that place p allows x whenever it is enabled. But we can do more. Suppose that $r, r' \in XNX(x)$ and $r \in [r'\rangle$, and consider the case that $r'[\alpha\rangle r$ where α is x-free. Then α acts semipositively on p, and we only need to require $M_{r'}(p) \geq h$ on r', in order to have $M_r(p) \geq h$ on r. For this reason, before imposing the requirement $M_r(p) \geq h$ on some $r \in XNX(x)$, we may try to follow x-free backward chains starting at r, hoping to find some $r' \in XNX(x)$ such that imposing $M_{r'}(p) \geq h$ on r' implies $M_r(p) \geq h$ for r. In doing so, we may hit a cycle. However, by Theorem 3.1, such a cycle may not contain any transition from $T_0 \cup T_\ell$. This is because cycles may never contain transitions from T_0 anyway, and because, while a cycle might intersect T_ℓ if $\ell > 0$, it would then also have to contain x (but we only follow x-free backward chains). Thus, the cycle we may be hitting could at most be formed by transitions in $T \setminus (T_0 \cup T_\ell)$; however, this has no importance since – as we already know – such cycles do not modify the marking of p. Let us therefore consider the following equivalence relation between states q, q':

$$q \equiv_\ell q' \quad\iff\quad q[\beta\rangle q' \text{ and } q'[\beta'\rangle q \text{ with } \beta, \beta' \in (T \setminus (T_0 \cup T_\ell))^*$$

and let us define

$$MXNX(x) = \{r \in XNX(x) \mid \nexists r' \in XNX(x) : r' \not\equiv_\ell r \text{ and } r'[\alpha\rangle r \text{ with } \alpha \in (T \setminus \{x\})^+\}$$

i.e., we only consider states in $XNX(x)$ which do not lie x-freely after another (non-$\equiv_\ell$-equivalent) one. This set may contain many $\equiv_\ell$-equivalent states, but we need only keep one of them. Let us therefore choose some set

$$\boxed{mXNX(x) \subseteq MXNX(x) \text{ with a single representative of each } \equiv_\ell\text{-equivalent class in it}}$$

These considerations amount to a proof of

▶ **Corollary 4.2** ($mXNX$ and enabling condition). $\big(\forall r \in S\colon r[x\rangle \Rightarrow M_r(p) \geq k + h\big) \iff \big(\forall r \in mXNX(x)\colon M_r(p) \geq h\big)$.

In other words, we only need to require $M_r(p) \geq h$ for states $r \in mXNX(x)$ in order to guarantee that place p allows x whenever it is enabled. Combining Corollary 4.2 with the formula (2) relating any $M_r(p)$ to μ_0 yields the following set of constraints:

$$\forall r \in mXNX(x)\colon \quad \mu_0 \quad \geq \quad k \cdot \Delta_{s_0,r}(x) - \sum_{j \in I_0 \cup I_\ell} k_j \cdot \Delta_{s_0,r}(a_j) + h \tag{3}$$

Let us now re-examine the constraint that $\forall r \in [s_0\rangle\colon M_r(p) \geq 0$. By $r \in [s_0\rangle$, there is a path $s_0[\alpha\rangle r$. If α contains an x, let s be the visited state before the last x; from the previous constraints, $M_s(p) \geq k + h$ and $M_r(p) \geq h \geq 0$. If α contains no x, then it has a semipositive effect on p and thus, $\mu_0 \leq M_r(p)$. It is then enough to impose $\mu_0 \geq 0$ to get the desired property $M_r(p) \geq 0$. However, $\mu_0 \geq 0$ can always be ensured by adding, if necessary, an adequate shift to all markings of p, as well as to h.

In sum, requiring the non-negative solvability of (3) suffices in order to find parameters $k, h, k_1, \ldots, k_m, \mu_0$ in such a way that the corresponding region (and then, a corresponding place p with initial marking $M_0(p) = \mu_0$) allows all the paths that are possible in TS.

Ensuring that place p solves an event/state separation problem. For each state s not enabling x, there should be a place p which does not have enough tokens to allow to perform x when reaching the state corresponding to s. That is, in addition to the constraints derived in the previous section, p should satisfy $M_s(p) < k + h$, i.e., using (2) with $r = s$:

$$\mu_0 \quad < \quad k + h + k \cdot \Delta_{s_0,s}(x) - \sum_{j \in I_0 \cup I_\ell} k_j \cdot \Delta_{s_0,s}(a_j) \tag{4}$$

Again, it is possible to reduce the number of such inequalities. For any $\alpha \in (T \setminus \{x\})^*$ with $s'[\alpha\rangle s$, we have $M_{s'}(p) \leq M_s(p)$. Hence, if (4) is ascertained for s, it is no longer necessary to bother about s'. Again, we may have cycles of equivalent states allowing to progress indefinitely while staying in states non-enabling x. Note that, by persistence, if $s \equiv_\ell s'$, then $s[x\rangle \iff s'[x\rangle$; moreover, if $s[a\rangle r$ with $r \not\equiv_\ell s$, then $s'[a\rangle r'$ with $r \equiv_\ell r'$; i.e., $\equiv_\ell$-equivalent states behave equivalently, as far as (non-) enabling of x is concerned. Hence, it makes sense to define $MNX(x) = \{s \in S \mid \neg s[x\rangle \text{ and } \forall s[a\rangle r\colon (s \equiv_\ell r) \vee r[x\rangle\}$ and

$$\boxed{mNX(x) \subseteq MNX(x) \text{ with a single representative of each } \equiv_\ell\text{-equivalence class in it}}$$

i.e., we consider states not allowing x such that no non-equivalent successor still excludes performing x, and we keep one representative of each class. Thus, if the event/state separation problems can be solved for the states s in $mNX(x)$, then they are solved for all $s \in S$ with $\neg s[x\rangle$. Hence, for every $s \in mNX(x)$, we need to find a place satisfying (3) and (4).

Combining the constraints (3) and (4) allows to eliminate both μ_0 and h:

$$\boxed{\forall r \in mXNX(x)\colon 0 < k \cdot [1 + \Delta_{s_0,s}(x) - \Delta_{s_0,r}(x)] + \textstyle\sum_{j \in I_0 \cup I_\ell} k_j \cdot [\Delta_{s_0,r}(a_j) - \Delta_{s_0,s}(a_j)]} \tag{5}$$

input a finite, decent, persistent lts $TS = (S, \to, T, s_0)$ satisfying $\mathbf{P}\{\Upsilon_1, \ldots, \Upsilon_n\}$;
initially T is the set of transitions, and P is empty;
for every $x \in T_\ell$ $(0 \le \ell \le n)$ and $s \in mNX(x)$ **do**
 construct a set $mXNX(x)$ and the corresponding system (5/6);
 if there is no natural solution to this system **then**
  {**output** "TS is not CF-solvable, due to x, s, and system (5/6)"; **stop**};
 choose a set of natural numbers $(k, k_1, \ldots, k_m)$ satisfying (5), as well as (1) if $\ell \neq 0$,
  and compute h and μ_0;
 add to P a place as in Fig. 5, with weights $k_1, \ldots, k_m, k+h, h$, and initial marking μ_0;
end for
output "The net with transitions T and places P CF-solves TS".

Figure 6 An algorithm checking CF-solvability and constructing an adequate solution.

If the system (5) is solvable in the domain of natural numbers (with $k_j = 0$ if $j \notin I_0 \cup I_\ell$), let us define $\mu = \max\{k \cdot \Delta_{s_0,r}(x) - \sum_{j \in I_0 \cup I_\ell} k_j \cdot \Delta_{s_0,r}(a_j) \mid r \in mXNX(x)\}$. If $\mu \ge 0$, by choosing $h = 0$ and $\mu_0 = \mu$ we shall get a solution to the systems (3) and (4), with $\mu_0 \ge 0$. If $\mu < 0$, it is not possible to create a suitable pure place from this solution, but we may choose $h = -\mu$ and $\mu_0 = 0$ (realising the "adequate shift" referred to above), and we shall again get a solution to the systems (3) and (4), with $\mu_0 \ge 0$.

If $x \notin T_0$, the constraints (1) need to be fulfilled as well. Combining (1) and (5), we get

$$\begin{array}{l}\forall r \in mXNX(x):\\ 0 < \sum_{j \in I_0 \cup I_\ell} k_j \cdot [\Upsilon_\ell(a_j) \cdot (1 + \Delta_{s_0,s}(x) - \Delta_{s_0,r}(x)) - \Upsilon_\ell(x) \cdot (\Delta_{s_0,s}(a_j) - \Delta_{s_0,r}(a_j))]\end{array} \quad (6)$$

If the system (6) is solvable in the domain $\mathbb{N}$, it is also possible to find a natural solution to both (1) and (5), by choosing a suitable value for k using (1), and, if necessary, multiplying the solution found by a common factor. Then we may choose h and μ_0 as described above.

Figure 6 summarises the resulting algorithm, where by "system (5/6)" we mean "system (5)" if $x \in T_0$ and "system (6)" if $x \in T_\ell$ for $1 \le \ell \le n$.

▶ **Theorem 4.3** (Validity of the construction). *If, for some $x \in T$ and $s \in mNX(x)$, the corresponding system (5/6) is not solvable, then TS has no CF solution. Otherwise, the constructed net is a CF solution of TS.*

Proof. If the system (5/6) associated with some $x \in T$ and $s \in mNX(x)$ is not solvable, then from the analysis above there is no place of a CF net both allowing all valid evolutions and excluding the invalid transition x from s. Let us thus assume all those systems are solvable and let us consider the net constructed as above.

We have seen that for any $s \in S$, if $s[x\rangle$, there is a state $r \in mXNX(x)$ and a state $s' \in XNX(x)$ such that $r[\alpha\rangle s'$ and $s[x^\ell\rangle s'$, with $\ell > 0$ and $\alpha \in (T \setminus \{x\})^*$. By the construction of each place p, $M_r(p) \ge h$, $M_{s'}(p) \ge h$, and $M_s(p) \ge k + h$. We have also seen that, for any $s \in S$, there is an x-free sequence α such that either $s_0[\alpha\rangle s$ and $M_s(p) \ge M_0(p) \ge 0$, or there is some $r \in XNX(x)$ with $r[\alpha\rangle s$ so that $M_s(p) \ge M_r(p) \ge h$. As a consequence, place p allows all valid evolutions specified by the lts. If $\neg s'[x\rangle$, we know there is some $s \in mNX(x)$ such that $M_{s'}(p) \le M_s(p)$. From the choice of $M_0(p)$ for the corresponding place p, and from (5), we have (4) whatever h, which excludes to perform x from s as well as from s'.

Thus, the solvability of all systems (5/6) implies that all event/state separation problems can be solved, and we get a CF net N with the same language as TS. The only way to

have non-isomorphism is that some state separation problem cannot be solved, i.e., that two states s_1 and s_2 correspond to the same marking. In that case, let $s_1[\beta\rangle q_1$ be a path to a home state q_1 of TS. Since s_1 and s_2 correspond to the same marking and $L(N) = L(TS)$, $s_2[\beta\rangle q_2$ for some state q_2 corresponding to the same marking as q_1. Since q_1 is a home state, there is a path $q_2[\alpha\rangle q_1$. Since the language is the same and q_1, q_2 correspond to the same marking, we have $q_2[\alpha\rangle q_1[\alpha\rangle q_3[\alpha\rangle q_4 \ldots$, and from finiteness and cycle consistency, we must have $q_1 = q_2$. But then, by strong determinism, we also have $s_1 = s_2$. ◀

Two examples

- Consider TS_3 (Figure 3). For $x = a$, we get $x \in T_0 = \{a, d\}$, $I_\ell = I_0 = \{1\}$ if $a_1 = d$, $mXNX(a) = \{s_1\}$, $mNX(a) = \{s_2\}$ and equation (5) reduces to $0 < k \cdot [1+1-1] + k_1 \cdot [0-1]$, which leads to the solution $k = 1$, $k_1 = 0$, $\mu_0 = 1$ and $h = 0$, corresponding to place p_1 in N_3. For $x = b$, we get $x \in T_1 = \{b\}$, $T_0 = \{a, d\}$, $I_0 = \{1, 2\}$ if $a_1 = a$ and $a_2 = d$, $I_1 = \emptyset$, $mXNX(b) = \{s_1\}$, $mNX(b) = \{s_0\}$ and equation (6) reduces to $0 < k_1 \cdot [0 - 1 \cdot (0-1)] + k_2 \cdot [0 - 1 \cdot (0-0)]$, which leads to the solution $k_1 = 1$, $k = k_2 = 0$, $\mu_0 = 0$ and $h = 1$, corresponding to place p_2 in N_3. The treatments of c and d are similar.
- Consider TS_4 (Figure 4). For $x = a$, we get $x \in T_0 = \{a\}$, $I_\ell = I_0 = \emptyset$, $MXNX(a) = \{s_4, s_5, s_6\}$ (all states are $\equiv_0$-equivalent), $mXNX(a) = \{s_4\}$ (for example), $MNX(a) = \{s_0, s_4, s_5, s_6, s\}$, $mNX(a) = \{s_0, s\}$ (for example), and for $x = s_0$, equation (5) reduces to $0 < k \cdot [1 + 0 - 1] + 0$, which is unsolvable.

Some special cases

- All transitions are live. Then $T_0 = \emptyset$; $\equiv_0$ is identity; and any $a_j \in {}^\bullet p$ corresponds to the same small cycle as $x \in p^\bullet$. If synthesis succeeds, the resulting CF net is a disjoint composition of n individual CF nets, each one connected by itself and corresponding to one of the Parikh vectors Υ_i (for $1 \leq i \leq n$). In essence, therefore, this reduces to the next case.
- All transitions are live and $n = 1$ (cf. [9]). Then both $\equiv_0$ and $\equiv_1$ reduce to identity. All sets $mXNX(x)$ and $mNX(x)$ are unique and can be simplified as follows:

$$\begin{array}{lcl} XNX(x) & = & \{s \in S \mid [x\rangle s \wedge \neg s[x\rangle\} \\ mXNX(x) & = & \{s \in XNX(x) \mid \nexists s' \in XNX(x) : s'[\alpha\rangle s \text{ with } \alpha \in (T \setminus \{x\})^+\} \\ mNX(x) & = & \{s \in S \mid \neg s[x\rangle \wedge \forall s' \in S, a \in T : s[a\rangle s' \Rightarrow s'[x\rangle\} \end{array}$$

  Such transition systems are "almost reversible" (i.e., they consist of an initial acyclic part, followed by a single strongly connected component, such as TS_2 in Figure 2).
- TS is reversible, i.e., s_0 is a home state (cf. [6, 8]). All previous simplifications hold and, additionally, if there is a CF solution, then there is also a pure solution. Nevertheless (despite the simpler setting), it is possible to construct reversible persistent transition systems which can be solved by a plain and pure Petri net, but not by a CF net.
- TS belongs to a marked graph (a plain net with $\forall p \in P : |p^\bullet| = 1 \wedge |{}^\bullet p| = 1$) or to a T-system (a plain net with $\forall p \in P : |p^\bullet| \leq 1 \wedge |{}^\bullet p| \leq 1$) (cf. [5, 7]). Such transition systems have full characterisations for bounded as well as for unbounded Petri nets.

5 Concluding remarks

The first part of this paper describes a partial characterisation – more precisely, an upper approximation – of the state spaces of bounded choice-free Petri nets. The reachability graphs of such Petri nets are finite, totally reachable, strongly deterministic, strongly cycle-consistent,

persistent, and enjoy the small cycle, the prime cycle, and the distance properties. A full structural characterisation seems to be very difficult to obtain. Even for finite words, such as TS_0 in Figure 2, it is very difficult to obtain exact structural (i.e., so to speak, free of linear algebra) conditions characterising the PN-solvable ones amongst them [3].

In its second part (starting with Section 4), the paper describes how choice-freely solvable transition systems can be detected and their Petri net solutions be constructed if possible. The aim here was to use structural knowledge in order to limit the set of states for which event/state separation problems need to be solved, and also to reduce the number of linear inequalities needed for each one of these problems.

The algorithmic gains are threefold: (1) It is possible to check, before starting synthesis, some of the properties given by the upper state space characterisation and to discard any given lts failing to satisfy one of them as not being solvable choice-freely. Such an algorithm has already been included for marked graphs in APT [10] and performs very satisfactorily. In general, though, these properties may not be easy to check. (2) State separation problems all but disappear (though this came as no surprise, given the result described in [13]). (3) The number of unknowns and the number of systems is reduced for the event/state separation problems that remain. This allows more efficient synthesis, and our first experiments confirm that our algorithm exhibits interesting performances. However, even for the special cases discussed at the end of Section 4, it seems difficult to estimate exactly how much can be gained, and in particular how the size of the essential sets $mXNX(x)$ and $mNX(x)$ evolves with the size of the lts, and possibly with some of its specific characteristics (like the out- and in-degrees of its nodes, symmetries of the structure, etc.).

There are many extensions and potential applications of choice-free synthesis. One particularly promising generalisation is to allow partially non-injective transition labellings for Petri nets (for instance, by forbidding equally labelled transitions in parallel components).

Acknowledgements. We are indebted to the reviewers for valuable comments.

References

1 Eric Badouel, Benoît Caillaud, and Philippe Darondeau. Distributing finite automata through Petri net synthesis. *Formal Asp. Comput.*, 13(6):447–470, 2002.

2 Eric Badouel and Philippe Darondeau. Theory of regions. In *Lectures on Petri Nets I: Basic Models, LNCS Vol. 1491*, pages 529–586, 1998.

3 Kamila Barylska, Eike Best, Evgeny Erofeev, Łukasz Mikulski, and Marcin Piątkowski. On binary words being Petri net solvable. In *Algorithms and Theories for the Analysis of Event Data (ATAED 2015)*, pages 1–15, 2015.

4 Eike Best and Philippe Darondeau. A decomposition theorem for finite persistent transition systems. *Acta Inf.*, 46(3):237–254, 2009.

5 Eike Best and Raymond R. Devillers. Characterisation of the state spaces of live and bounded marked graph Petri nets. In *8th International Conference on Language and Automata Theory and Applications (LATA 2014)*, pages 161–172, 2014.

6 Eike Best and Raymond R. Devillers. Synthesis of persistent systems. In *35th International Conference on Application and Theory of Petri Nets and Concurrency (ICATPN 2014)*, pages 111–129, 2014.

7 Eike Best and Raymond R. Devillers. State space axioms for T-systems. *Acta Inf.*, 52(2-3):133–152, 2015.

8 Eike Best and Raymond R. Devillers. Synthesis and reengineering of persistent systems. *Acta Inf.*, 52(1):35–60, 2015.

9 Eike Best and Raymond R. Devillers. Synthesis of live and bounded persistent systems. *Fundamenta Informaticae*, 140:39–59, 2015.

10 Eike Best and Uli Schlachter. Analysis of Petri nets and transition systems. In *Proc. 8th Interaction and Concurrency Experience (ICE), Grenoble*, 2015.

11 Jordi Cortadella, Michael Kishinevsky, Alex Kondratyev, Luciano Lavagno, and Alex Yakovlev. *Logic Synthesis for Asynchronous Controllers and Interfaces*, volume 8 of *Advanced Microelectronics*. Springer, 2002.

12 Stefano Crespi-Reghizzi and Dino Mandrioli. A decidability theorem for a class of vector-addition systems. *Inf. Process. Lett.*, 3(3):78–80, 1975.

13 Philippe Darondeau. Equality of languages coincides with isomorphism of reachable state graphs for bounded and persistent Petri nets. *Inf. Process. Lett.*, 94(6):241–245, 2005.

14 Paul Gastin and Nathalie Sznajder. Fair synthesis for asynchronous distributed systems. *ACM Trans. Comput. Log.*, 14(2):9, 2013.

15 Robert M. Keller. A fundamental theorem of asynchronous parallel computation. In *Sagamore Computer Conference, August 20-23 1974, LNCS Vol. 24*, pages 102–112, 1975.

16 Leslie Lamport. Arbitration-free synchronization. *Distributed Computing*, 16(2-3):219–237, 2003.

17 Lawrence H. Landweber and Edward L. Robertson. Properties of conflict-free and persistent Petri nets. *J. ACM*, 25(3):352–364, 1978.

18 Tadao Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, 1989.

19 Peter J. Ramadge and W. M. Wonham. Supervisory control of a class of discrete event processes. *SIAM Journal on Control and Optimization*, 25(1):206–230, 1987.

20 Enrique Teruel, José Manuel Colom, and Manuel Silva. Choice-free Petri nets: a model for deterministic concurrent systems with bulk services and arrivals. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 27(1):73–83, 1997.

21 Harro Wimmel and Karsten Wolf. Applying CEGAR to the Petri net state equation. In *17th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2011)*, pages 224–238, 2011.