Scientific Annals of Computer Science vol. 28 (2), 2018, pp. 199–236

doi: 10.7561/SACS.2018.2.199

Simultaneous Petri Net Synthesis

Eike Best[1]
Raymond Devillers[2]
Uli Schlachter[1]
Harro Wimmel[1]

Abstract

Petri net synthesis deals with the problem whether, given a labelled transition system TS, one can find a Petri net N with an initial marking M_0 such that the reachability graph of (N, M_0) is isomorphic to TS. This may be preceded by a pre-synthesis phase that will quickly reject ill-formed transition systems (and give structural reasons for the failure) and otherwise build data structures needed by the proper synthesis. The last phase proceeds by solving systems of linear inequalities, and may still fail but for less transparent reasons. In this paper, we consider an extended problem. A finite set of transition systems $\{TS_1, \ldots, TS_m\}$ shall be called *simultaneously Petri net solvable* if there is a single Petri net N with several initial markings $\{M_{01}, \ldots, M_{0m}\}$, such that for every $i = 1, \ldots, m$, the reachability graph of (N, M_{0i}) is isomorphic to TS_i. The focus will be on *choice-free* nets, that is, nets without structural choices, and we explore how previously published efficient algorithms for the pre-synthesis and proper synthesis of bounded and choice-free Petri nets can be generalised for the simultaneous pre-synthesis and synthesis of such multi-marked nets. At the same time, the choice-free pre-synthesis of a single transition system shall be strengthened by introducing new structural checks.

Keywords: Choice-Freeness, Labelled Transition Systems, Petri Nets, Synthesis.

[1] Department of Computing Science, Carl von Ossietzky Universität Oldenburg, Germany, Email: firstname.lastname@informatik.uni-oldenburg.de

[2] Département d'Informatique, Université Libre de Bruxelles, Belgium, Email: rdevil@ulb.ac.be

1 Introduction

Being able to execute a single piece of hardware (or software) from two or more different initial states is not an uncommon phenomenon. For example, the world's favourite operating systems are usually startable either in normal mode or in various shades of safe mode, and the resulting behaviours may be rather different. For instance, a device driver may malfunction in normal mode, while a replacement driver may still be operational in downgraded mode.

In this paper, a related question will be considered in the context of Petri net synthesis [1] where the task is to construct a system realising a single given behavioural specification. In simultaneous system synthesis, the more general task is to realise several possible given behaviours through different initial configurations of a system. We shall assume m behaviours to be specified by m finite labelled transition systems. The task is to find a single unmarked choice-free net N, together with m initial markings $M_{01}, \ldots M_{0m}$, such that the i'th behaviour is realised by N, together with M_{0i}. Choice-free nets are a non-trivial class of weighted Petri nets allowing fully distributed implementations and presenting interesting behavioural characteristics. The existence of choice-free realisations is of importance in several fields of application [12, 16].

The structure of this paper is as follows. The next section contains basic definitions about transition systems and Petri nets. After introducing central concepts about choice-free synthesis (Section 3), the pre-synthesis previously developed for individual solvability [3] is extended and adapted to the simultaneous case (Section 4). Two new checks are added with respect to the ones detailed in previous papers, and the resulting algorithms are specified in full, and analysed, in Section 5. In Section 6, techniques that have been developed for individual choice-free synthesis [4, 17] are extended to simultaneous choice-free synthesis. In Section 7, some concluding remarks and future plans are evoked.

2 Labelled Transition Systems and Petri Nets

A labelled transition system with initial state, lts for short, is a quadruple $TS = (S, T, \rightarrow, \imath)$ describing an edge-labelled directed graph. The nodes of this graph represent states $s \in S$; the arcs are triples $\rightarrow \subseteq (S \times T \times S)$, where $(s, t, s') \in \rightarrow$ denotes an edge with label $t \in T$ leading from state s to state s'; and $\imath$ is the initial state. TS is finite if S and T (hence also $\rightarrow$) are finite sets.

A label t is enabled at a state s if a t-labelled edge leads out of s. A state s' is reachable from s by an occurrence of t, denoted by $s[t\rangle s'$, if $(s, t, s') \in \rightarrow$. More generally, a state s' is reachable from s by a sequence $\sigma \in T^*$, denoted by $s[\sigma\rangle s'$, if there is a directed path with label sequence σ leading from s to s'. The set of states reachable from s is denoted by $[s\rangle$. A sequence $s[\sigma\rangle s'$ is called a cycle, or more precisely a cycle at (or around) state s, if $s = s'$.

For instance, the state set of TS_1 in Figure 1 is $\{\imath_1, r, s\}$, the label set is $\{a, b, c, d\}$, the edge set is $\{(\imath_1, a, r), (r, b, r), (r, d, s), (s, b, s), (s, c, s)\}$, and the initial state is $\imath_1$. State s is reachable from $\imath_1$ by the sequence ad (that is, $\imath_1[ad\rangle s$), but also by $\imath_1[abbdbcb\rangle s$. Also, $[r\rangle = \{r, s\}$.

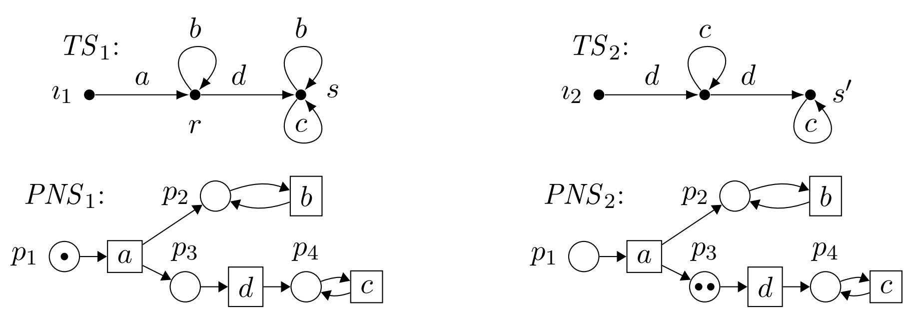

Figure 1: Two finite transition systems TS_1 and TS_2. PNS_1 and PNS_2 simultaneously solve TS_1 and TS_2 choice-freely, since the reachability graph of PNS_i is isomorphic to TS_i ($i = 1, 2$) and the underlying Petri net is choice-free (and the same in both systems).

A T-vector is a function $\Phi\colon T \to \mathbb{N}$, and its support is $supp(\Phi) = \{t \in T \mid \Phi(t) > 0\}$. It is prime if the greatest common divisor of its function values is 1. Two T-vectors $\Phi_1, \Phi_2\colon T \to \mathbb{N}$ are label-disjoint (or simply disjoint when there is no ambiguity) if their supports are disjoint.

These notions can be transferred to sequences $\sigma \in T^*$ through their Parikh vectors: the Parikh vector of a sequence $\sigma \in T^*$, denoted $\mathcal{P}(\sigma)$, is the T-vector counting, for each label $t \in T$, the number of occurrences of t in σ. We extend the notation $s[\cdots\rangle s'$ to T-vectors and write $s[\Phi\rangle s'$ if $s[\tau\rangle s'$

for some $\tau \in T^*$ with $\mathcal{P}(\tau) = \Phi$. The support of a sequence σ is the support of $\mathcal{P}(\sigma)$, and two sequences $\sigma_1, \sigma_2 \in T^*$ are label-disjoint if $\mathcal{P}(\sigma_1)$ and $\mathcal{P}(\sigma_2)$ are label-disjoint, and Parikh-equivalent if they have the same Parikh vector, i.e. if $\mathcal{P}(\sigma_1) = \mathcal{P}(\sigma_2)$.

We use alphabetical ordering for T-vectors, hence also for Parikh vectors. For example, in TS_1, the sequence $\imath_1[abbdbcb\rangle s$ has the Parikh vector $\mathcal{P}(abbdbcb) = (1,4,1,1)$. The cycles $r[bb\rangle r$ and $s[bb\rangle s$ are Parikh-equivalent, while $r[bb\rangle r$ and $s[bcb\rangle s$ are not.

T- vectors can be compared componentwise. Thus, for instance, with $\lneqq$ meaning "less than or equal in each component, but not entirely equal", we have $(0,0,0,0) \lneqq (0,1,0,0) \lneqq (0,2,0,0)$. And with $\nleq$ meaning "not less than or equal", $(1,2,3,4) \nleq (4,4,4,3)$. We define $\gneqq$ and $\ngeq$ analogously.

A cycle $s[\sigma\rangle s$ is called empty if $\mathcal{P}(\sigma)$ is the null vector, i.e. $\sigma = \varepsilon$. A non-empty cycle is called small if there is no non-empty cycle $s'[\sigma'\rangle s'$ with $\mathcal{P}(\sigma') \lneqq \mathcal{P}(\sigma)$. For example, in TS_1, there are exactly three small cycles, $r[b\rangle r$, $s[b\rangle s$, and $s[c\rangle s$ (but infinitely many cycles). Note that they happen to be either Parikh-equivalent or label-disjoint.

Two lts with the same label set $TS = (S, T, \to, \imath)$ and $TS' = (S', T, \to', \imath')$ are isomorphic if there is a bijection $\zeta\colon S \to S'$ with $\zeta(\imath) = \imath'$ and $(s,t,s') \in \to \Leftrightarrow (\zeta(s), t, \zeta(s')) \in \to'$, for all $s, s' \in S$ and $t \in T$.

A finite, unmarked Petri net will be denoted as $N = (P, T, F)$ where P is a finite set of places, T is a finite set of transitions, with $P \cap T = \emptyset$, and F is a flow function $F\colon ((P \times T) \cup (T \times P)) \to \mathbb{N}$. A net N is choice-free [5, 16] (not to be confused with free-choice [6]) if $\forall p \in P\colon |p^\bullet| \leq 1$, where $p^\bullet = \{t \in T \mid F(p,t) > 0\}$ is the set of output transitions of p.

A Petri net system $PNS = (N, M_0)$ is a Petri net N provided with an initial marking M_0, where a marking is a mapping $M\colon P \to \mathbb{N}$, indicating the number of tokens in each place. A transition $t \in T$ is enabled by a marking M, denoted by $M[t\rangle$, if for all places $p \in P$, $M(p) \geq F(p,t)$. If t is enabled at M, then t can occur (or fire) in M, leading to the marking M' defined by $M'(p) = M(p) - F(p,t) + F(t,p)$ (denoted by $M[t\rangle M'$). From this, one can define reachability (of a marking from some marking) etc., analogously as above. In particular, the set $[M_0\rangle$ of markings reachable from M_0 is well-defined, and if it is finite, then the system is called bounded. The reachability graph of a Petri net system $PNS = (N, M_0)$ is the labelled transition system with the set of vertices $[M_0\rangle$, initial state M_0 and arcs $\{(M,t,M') \mid M, M' \in [M_0\rangle \wedge M[t\rangle M'\}$.

The incidence matrix C of a net $N = (P,T,F)$ is the integer place-transition matrix with components $C(p,t) = F(t,p) - F(p,t)$, where p is a place and t is a transition. An elementary property of Petri nets is the state equation [14] which expresses that, if $M[\sigma\rangle M'$, then $M' = M + C \cdot \mathcal{P}(\sigma)$. A semiflow of a net N is a T-vector Φ such that $\Phi \gneqq \mathbf{0}$ and $C \cdot \Phi = \mathbf{0}$. A semiflow is minimal if there is no smaller one. From the state equation, it follows immediately that if $M[\sigma\rangle M'$ and $\mathcal{P}(\sigma)$ is a semiflow, then $M = M'$.

For example, the net N shown in the middle of Figure 2 contains two different semiflows, $(1,1,0)$ and $(1,0,1)$, with supports $\{a,b\}$ and $\{a,c\}$, respectively (again considering transitions in alphabetical order).

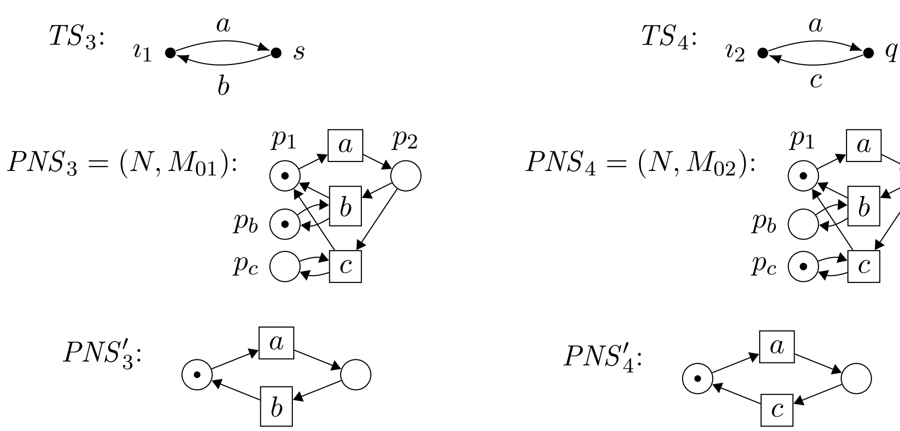

Figure 2: Two finite transition systems TS_3 and TS_4 with initial states $\imath_1, \imath_2$. The reachability graph of PNS_i is isomorphic to TS_i, thus PNS_i solves the corresponding TS_i ($i = 3,4$). As PNS_3 and PNS_4 differ only in their initial markings and have the same underlying Petri net N, this is also a simultaneous solution. Observe that the net N is not choice-free, since place p_2 has two different output transitions. TS_3 and TS_4 also have individual choice-free solutions, depicted as PNS'_3 and PNS'_4. But they have no simultaneous choice-free solution. We will prove this later.

Two Petri net systems $N_1 = (P_1, T, F_1, M_0^1)$ and $N_2 = (P_2, T, F_2, M_0^2)$ with the same transition set T are isomorphic if there is a bijection $\zeta \colon P_1 \to P_2$ such that, $\forall p_1 \in P_1, t \in T$: $M_0^1(p_1) = M_0^2(\zeta(p_1))$, $F_1(p_1,t) = F_2(\zeta(p_1),t)$

and $F_1(t, p_1) = F_2(t, \zeta(p_1))$. It is obvious that isomorphic Petri net systems have isomorphic reachability graphs, so that if we have a Petri net solution for some labelled transition system, any isomorphic Petri net system is also a solution.

3 Necessary Conditions for Choice-Free Synthesis

Individual Petri net synthesis has first been considered in [10] and is described in [1]. Its extension to simultaneous synthesis may be formalised as follows.

Definition 1 (Simultaneous) solvability of *lts* by *PNS*
Let $TS_1, \ldots, TS_m$ be m labelled transition systems with disjoint state sets (but possibly overlapping label sets). They are called simultaneously Petri net synthesisable (or solvable, for short) if there are a Petri net N and m markings $M_{01}, \ldots, M_{0m}$ such that, for $1 \leq i \leq m$, the reachability graph of the Petri net system (N, M_{0i}) is isomorphic to TS_i. If the same is true with a choice-free net N, then $TS_1, \ldots, TS_m$ are called simultaneously choice-freely solvable (cf-solvable, for short). $\square$

Thus, given $TS_1, \ldots, TS_m$ as an input, the choice-free synthesis problem asks whether there are a choice-free Petri net N and m initial markings $M_{01}, \ldots, M_{0m}$ such that (N, M_{0i}) cf-solves TS_i for all $1 \leq i \leq m$, and if so, to construct them. If this is not possible, an interesting addition would be to produce reasons for that failure, as simple and intuitive as possible.

For example, consider the two cyclic behaviours specified by means of labelled transition systems TS_3 and TS_4 on the top row of Figure 2. They can individually (and choice-freely) be solved by the Petri net systems PNS'_3 and PNS'_4 shown on the bottom of the figure, and simultaneously (but not choice-freely) by the net systems PNS_3 and PNS_4 shown in the middle.

In solving a simultaneous choice-free synthesis problem, a two-phase strategy will be adopted. An initial pre-synthesis check eliminates inputs that cannot be synthesised for some structural reasons. If such an input is detected, a reason (or a small set of reasons) for the failure can usually be delineated, and the transition system designer may focus on mending those deficiencies. After this step (pre-synthesis), a second step (synthesis) is executed, but on a reduced set of inputs. This step may still detect some non-synthesisable transition systems. However, the reasons for the unsuitability may now be more obscure. If no failure is detected in either stage, the input is synthesisable, and one of the possible solutions – a net N together with m initial markings – is returned by the synthesis step.

In the remainder of this paper, we shall describe, and analyse, an algorithm solving the simultaneous choice-free synthesis problem. The following theorem, whose proof can be found in [3, 4, 17], specifies a number of properties a single transition system must *necessarily* enjoy in order to have a choice-free solution. It forms the basis of pre-synthesis.

Theorem 1 Necessary conditions for individual cf-solvability
The reachability graph of a bounded, choice-free Petri net system is finite, totally reachable, forward and backward deterministic, and persistent. It also satisfies the prime cycle property and the disjoint small cycles property. Distances between states are well-defined, and the distance path property is satisfied. □

We shall now explain these properties (more details, examples and counter-examples may be found in [3, 4]).

A labelled transition system $TS = (S, T, \rightarrow, \imath)$ is *totally reachable* if $[\imath\rangle = S$ (i.e. every state is reachable from $\imath$); *(forward) deterministic* if, for all states $s, s', s'' \in S$ and for all transitions $t \in T$, $(s[t\rangle s' \wedge s[t\rangle s'')$ entails $s' = s''$ (i.e. immediate successor states are uniquely determined by the labels); *backward deterministic* if, for all states $s, s', s'' \in S$ and transitions $t \in T$, $(s'[t\rangle s \wedge s''[t\rangle s)$ entails $s' = s''$; and *persistent* [13] if for all states $s, s', s'' \in S$, and labels $t \neq u$, if $s[t\rangle s'$ and $s[u\rangle s''$, then there is some state $r \in S$ such that both $s'[u\rangle r$ and $s''[t\rangle r$ (i.e. once two different labels are both enabled, none can disable the other).

Total reachability and both versions of determinism are always satisfied by Petri net reachability graphs. Persistence is always satisfied by choice-free nets.

TS has *the prime cycle property* if every small cycle has a prime Parikh vector, and *the disjoint small cycles property* if there exist an integer $n \leq |T|$ and a finite set of mutually label-disjoint T-vectors $\Upsilon_1, \ldots, \Upsilon_n : T \rightarrow \mathbb{N}$ such that

$$\{\, \mathcal{P}(\beta) \mid TS \text{ contains a state } s \text{ and a small cycle } s[\beta\rangle s \,\} \;=\; \{\, \Upsilon_1, \ldots, \Upsilon_n \,\}$$

The latter implies that the Parikh vectors of small cycles are either equal or label-disjoint.

Assume that the disjoint small cycles property is satisfied with T-vectors $\Upsilon_1, \ldots, \Upsilon_n$. For a T-vector $\Phi \colon T \to \mathbb{N}$, its modulo vector $\Phi \bmod \{\Upsilon_1, \ldots, \Upsilon_n\}$ is the smallest, non-negative vector $\Phi - \sum_{i \in \{1,\ldots,n\}} k_i \cdot \Upsilon_i$ for $k_1, \ldots, k_n \in \mathbb{N}$.[3] For Parikh vectors of paths between two given states, the modulo vector is an invariant in any transition system that is finite, totally reachable, deterministic, persistent, and has the disjoint small cycles property [3]. This justifies the following definition. Let $r, s \in S$, and let $r[\alpha\rangle s$ be a path of TS. Then $\Delta_{r,s} = \mathcal{P}(\alpha) \bmod \{\Upsilon_1, \ldots, \Upsilon_n\}$ is called *the distance* from r to s. For the sake of brevity, let Δ_s denote $\Delta_{\imath,s}$. TS satisfies *the distance path property* if, between any pair of states $r, s \in S$ with $s \in [r\rangle$, there is a *distance path*, i.e. a path $r[\sigma\rangle s$ satisfying $\mathcal{P}(\sigma) = \Delta_{r,s}$. For instance, in TS_1 (Figure 1), we have $\Delta_s = \Delta_{\imath,s} = (1,0,0,1)$, with a distance path $\imath[ad\rangle s$.

Figures 3 and 4 illustrate the necessity of some of the constraints in Theorem 1.

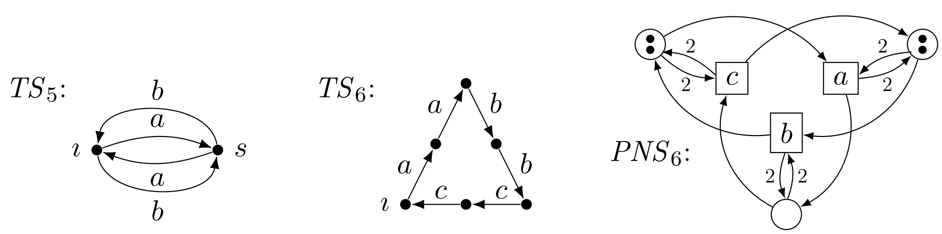

Figure 3: The transition system shown on the left-hand side is deterministic and persistent but does not satisfy the prime cycle property, nor the disjoint small cycles property. It has no Petri net realisation. The transition system shown in the middle satisfies the disjoint small cycles property but not the prime cycle property. It has a Petri net realisation, PNS_6, but not a choice-free one.

The pre-synthesis algorithms described in [4, 17] check the properties listed in Theorem 1 in some clever and efficient way – more precisely, in time $O(|S| \cdot |T|^2)$ – and create, individually for every TS_i, data structures containing all distances from the initial state as well as the Parikh vectors of all small cycles. Hence we can assume from now on that all transition

[3] The idea is to subtract from Φ the vectors Υ_i as often as possible, without getting negative entries.

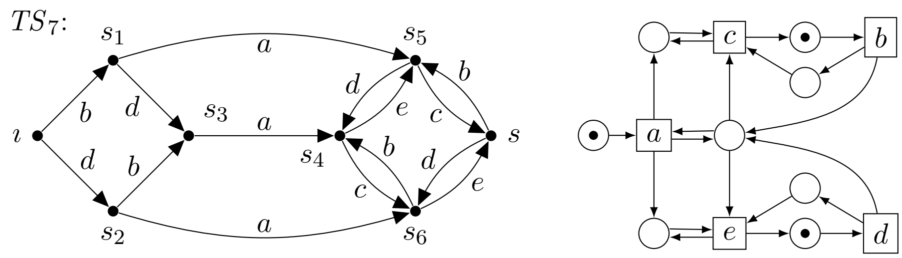

Figure 4: TS_7 is not choice-freely Petri net solvable while it is deterministic, persistent, and satisfies both the prime cycle and the disjoint small cycles properties. There are two Parikh vectors of small cycles, $\mathcal{P}(bc) = (0,1,1,0,0)$ and $\mathcal{P}(de) = (0,0,0,1,1)$. For instance, $\Delta_{\iota,s} = (1,0,0,0,0)$ (as can be evaluated either by subtracting the cyclic T-vector $\mathcal{P}(bc)$ from the Parikh vector $\mathcal{P}(bac) = (1,1,1,0,0)$ of the path $\iota[bac\rangle s$, or by subtracting the cyclic T-vector $\mathcal{P}(de)$ from the Parikh vector $\mathcal{P}(dae) = (1,0,0,1,1)$ of the path $\iota[dae\rangle s$). The distance path property is not satisfied, since there is no path from ι to s having Parikh vector $(1,0,0,0,0)$. Right-hand side: A non-choice-free solution of TS_7.

systems satisfy these basic properties, and that such data structures are available.

In the analysis of deterministic and persistent transition systems, a precious tool is the notion of a residue and the associated theorem by Keller that we shall detail now. Let $\tau, \sigma \in T^*$ be two sequences over T. The (left) residue of τ with respect to σ, denoted by $\tau \stackrel{\bullet}{-} \sigma$, arises from cancelling successively in τ the leftmost occurrences of all symbols from σ, read from left to right. Inductively: $\tau \stackrel{\bullet}{-} \varepsilon = \tau$; $\tau \stackrel{\bullet}{-} t = \tau$ if $t \notin \mathit{supp}(\tau)$; $\tau \stackrel{\bullet}{-} t$ is the sequence obtained by erasing the leftmost t in τ if $t \in \mathit{supp}(\tau)$; and $\tau \stackrel{\bullet}{-} (t\sigma) = (\tau \stackrel{\bullet}{-} t) \stackrel{\bullet}{-} \sigma$. Said differently, $\tau \stackrel{\bullet}{-} \sigma$ is τ with, for each $t \in T$, the first $\min(\mathcal{P}(\tau)(t), \mathcal{P}(\sigma)(t))$ occurrences of t dropped. This may be extended to T-vectors as follows. Let $\sigma \in T^*$ and $\Phi \in \mathbb{N}^T$: $\sigma \stackrel{\bullet}{-} \Phi$ is the sequence obtained from σ by cancelling the $\min(\mathcal{P}(\tau)(t), \Phi(t))$ leftmost occurrences of t for each $t \in T$. Let $\Phi, \Psi \in \mathbb{N}^T$: $\Psi \stackrel{\bullet}{-} \Phi$ is the T-vector such that, for each $t \in T$, $(\Psi \stackrel{\bullet}{-} \Phi)(t) = \max(\Psi(t) - \Phi(t), 0) = \Psi(t) - \min(\Psi(t), \Phi(t))$. The consistency between these various forms of residues arises from the observations that $\tau \stackrel{\bullet}{-} \sigma = \tau \stackrel{\bullet}{-} \mathcal{P}(\sigma)$ and $\mathcal{P}(\tau \stackrel{\bullet}{-} \mathcal{P}(\sigma)) = \mathcal{P}(\tau) \stackrel{\bullet}{-} \mathcal{P}(\sigma)$. Other interesting properties about residues are that $(\sigma \stackrel{\bullet}{-} \sigma_1) \stackrel{\bullet}{-} \sigma_2 = \sigma \stackrel{\bullet}{-} (\mathcal{P}(\sigma_1) + \mathcal{P}(\sigma_2)) = (\sigma \stackrel{\bullet}{-} \sigma_2) \stackrel{\bullet}{-} \sigma_1$ and $\sigma\sigma' \stackrel{\bullet}{-} \sigma = \sigma'$.

Theorem 2 Keller [11]
Let $(S, T, \rightarrow, \imath)$ be a deterministic and persistent lts. Let τ and σ be two label sequences enabled at some state s. Then $\tau(\sigma \stackrel{\bullet}{-} \tau)$ and $\sigma(\tau \stackrel{\bullet}{-} \sigma)$ are also enabled at s. Furthermore, the state reached after $\tau(\sigma \stackrel{\bullet}{-} \tau)$ and the state reached after $\sigma(\tau \stackrel{\bullet}{-} \sigma)$ are the same. In terms of T-vectors, $s[\Theta\rangle \wedge s[\Phi\rangle$ implies $s[\Theta + (\Phi \stackrel{\bullet}{-} \Theta)\rangle s'$ as well as $s[\Phi + (\Theta \stackrel{\bullet}{-} \Phi)\rangle s'$, for some $s' \in S$.

Consider the sequences $\imath[bad\rangle s_4$ and $\imath[dae\rangle s$ of Fig. 4 as an example. We get $bad \stackrel{\bullet}{-} dae = b$ and $dae \stackrel{\bullet}{-} bad = e$, and the theorem guarantees that $\imath[bad(dae \stackrel{\bullet}{-} bad)\rangle s_5$ and $\imath[dae(bad \stackrel{\bullet}{-} dae)\rangle s_5$ lead to the same state s_5.

4 Individual and Simultaneous Pre-Synthesis

In this section, we augment the set of necessary individual conditions given in Theorem 1 by ones that apply to individual cf-synthesis, but also to simultaneous cf-synthesis. To start with, the next theorem establishes a new, rather general property of choice-free net systems. This property generalises Lemma 16 in [16], but we give an independent proof.

Theorem 3 Reduced sequences
In a choice-free net N, assume that $M[\sigma\rangle$ and that Φ is a semiflow. Then $M[\sigma \stackrel{\bullet}{-} \Phi\rangle$.

Proof: Let us proceed by induction on the length of σ.

If $\mathcal{P}(\sigma) \leq \Phi$, $\sigma \stackrel{\bullet}{-} \Phi = \varepsilon$ and the property is trivial.

Otherwise, let $\sigma = \sigma_1 t \sigma_2$ with $\mathcal{P}(\sigma_1)(t) = \Phi(t)$ and $\mathcal{P}(\sigma_1)(t') < \Phi(t')$ for each $t' \in T \setminus \{t\}$, i.e. t is the first "excessive" transition with respect to Φ, hence also the first transition in $\sigma \stackrel{\bullet}{-} \Phi$.

If $M[\sigma_1\rangle M'$ and C is the incidence matrix of the net N, from the fundamental state equation we have $M' = M + C \cdot \mathcal{P}(\sigma_1)$, hence also $M' = M + C \cdot (\mathcal{P}(\sigma_1) - \Phi)$ since Φ is a semiflow. Since $\mathcal{P}(\sigma_1)(t) = \Phi(t)$, for any place p with $t \in p^\bullet$ we thus have $M(p) = M'(p) + \sum_{u \neq t} C(p, u) \cdot (\Phi(u) - \mathcal{P}(\sigma_1)(u)) \geq M'(p)$, since $\mathcal{P}(\sigma_1) \leq \Phi$ and, by choice-freeness, t is the only transition able to decrease the marking of p. As a consequence, since $M'[t\rangle$, we also have $M[t\rangle$, i.e. the first "excessive" t may be pushed to the left. By persistence (Keller's theorem), we then have $M[t(\sigma_1 t\sigma_2 \stackrel{\bullet}{-} t)\rangle$, which can be rewritten into $M[t((\sigma_1 t) \stackrel{\bullet}{-} t)\sigma_2\rangle$.

Let $\sigma'_1 = (\sigma_1 t) \stackrel{\bullet}{-} t$. We thus have $M[t\sigma'_1\sigma_2\rangle$. Since $\sigma'_1\sigma_2$ is shorter (by one transition) than σ, we may use the induction hypothesis (from the marking reached after $M[t\rangle$): $M[t((\sigma'_1\sigma_2) \stackrel{\bullet}{-} \Phi)\rangle$. Now, $(\sigma'_1\sigma_2) \stackrel{\bullet}{-} \Phi = \sigma_2 \stackrel{\bullet}{-} (\Phi - \mathcal{P}(\sigma'_1)) = \sigma_2 \stackrel{\bullet}{-} (\Phi - \mathcal{P}(\sigma_1))$ since $\mathcal{P}(\sigma'_1) = \mathcal{P}(\sigma_1) \leq \Phi$, and $\sigma \stackrel{\bullet}{-} \Phi = (\sigma_1 t\sigma_2) \stackrel{\bullet}{-} \Phi = t(\sigma_2 \stackrel{\bullet}{-} (\Phi - \mathcal{P}(\sigma_1))$ since $\Phi(t) = \mathcal{P}(\sigma_1)(t)$. The claim results. □

In the following we shall derive several interesting special cases of this general property.

Φ may be derived from the examination of cycles in some reachability graph(s) of N. Hence, in the context of an individual or simultaneous choice-free synthesis, assuming there is a solution, it may be obtained from the analysis of cycles in the given labelled transition system(s).

We shall first exploit Theorem 3 in order to establish a close correspondence between small cycles and minimal semiflows. This is important, since the former are behavioural objects, but the latter are of a structural nature (i.e. independent of any initial markings and applying only to the underlying net).

Corollary 1 Minimal semiflows
In the reachability graph of a choice-free net system, the Parikh vector of a small cycle is a minimal semiflow.

Proof: The Parikh vector of any non-empty cycle $M[\sigma\rangle M$ is a semiflow. Hence, if it is a minimal semiflow, there is no small cycle with a lower Parikh vector and we have a small cycle. Conversely, if it is not a minimal semiflow, we have $\Phi \lneqq \mathcal{P}(\sigma)$ for some semiflow Φ and, from Theorem 3 and Keller's theorem, $M[\sigma \stackrel{\bullet}{-} \Phi\rangle M'[\sigma \stackrel{\bullet}{-} (\sigma \stackrel{\bullet}{-} \Phi)\rangle M$ with $\mathcal{P}(\sigma \stackrel{\bullet}{-} (\sigma \stackrel{\bullet}{-} \Phi)) = \Phi$, so that $M' = M$ and σ cannot be a small cycle. □

The property that the Parikh vectors of small cycles are also minimal semiflows depends intimately on choice-freeness. This is illustrated by Figures 3 and 5. In Figure 3, TS_6 shows that Corollary 1 is wrong if the premise of choice-freeness is omitted, since $(2, 2, 2)$ is not a minimal semiflow in PNS_6

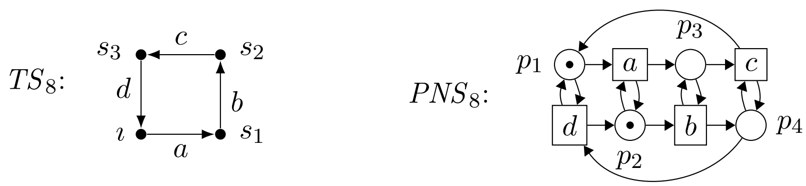

Figure 5: A net system, PNS_8, with a small cycle *abcd* in its reachability graph (l.h.s.) that does not represent a minimal semiflow. Minimal semiflows are $(1, 0, 1, 0)$ and $(0, 1, 0, 1)$. The prime cycle property is satisfied.

(instead, $(1, 1, 1)$ is). Figure 5 demonstrates that even if the prime cycle property holds true, Corollary 1 is wrong unless choice-freeness is assumed.

If $\Phi = \Phi_1 + \Phi_2$, where Φ_1 and Φ_2 are semiflows of N, we may observe that $\sigma \stackrel{\bullet}{-} \Phi = (\sigma \stackrel{\bullet}{-} \Phi_1) \stackrel{\bullet}{-} \Phi_2$. Hence, in the applications of Theorem 3, we may restrict our attention to minimal semiflows, hence to small cycles from Corollary 1.

The property expressed in Theorem 3 yields a necessary condition of cf-solvability: if we may find a configuration $s[\sigma\rangle$ while $\neg s[\sigma \stackrel{\bullet}{-} \Phi\rangle$ for some state s, sequence σ and wanted semiflow Φ in a given transition system, we may deduce that the (simultaneous) choice-free synthesis is impossible. It may thus be useful to check those properties as a part of the pre-synthesis phase. However, unless we consider an acyclic transition system, this involves checking arbitrarily long sequences from all states s, which is highly impractical. We now develop a range of more practical and feasible checks which, taken together, will eventually imply it (see Theorem 4; note also that the remark on minimal semiflows and Corollary 1 already reduce the set of semiflows to consider).

First, we show that it is not necessary to consider every state s; the initial one suffices.

Proposition 1 Reduced sequences from the initial state
In a finite, deterministic and persistent transition system $TS = (S, T, \to, \imath)$, *if, for some sequence* $\tau \in T^*$, $\forall \sigma \in T^* : \imath[\sigma\rangle \Rightarrow \imath[\sigma \stackrel{\bullet}{-} \tau\rangle$, *then also for each reachable state* s *we have* $\forall \sigma \in T^* : s[\sigma\rangle \Rightarrow s[\sigma \stackrel{\bullet}{-} \tau\rangle$.

Proof: Let us assume that $\sigma = \sigma_1\sigma_2$, with $\imath[\sigma_1\rangle s[\sigma_2\rangle$. From the hypothesis, $\imath[\sigma_1\sigma_2 \stackrel{\bullet}{-} \tau\rangle$. Then, from Keller's theorem, $\imath[\sigma_1\rangle s[(\sigma_1\sigma_2 \stackrel{\bullet}{-} \tau) \stackrel{\bullet}{-} \sigma_1\rangle$. But from the properties mentioned for residues in Section 2, $(\sigma_1\sigma_2 \stackrel{\bullet}{-} \tau) \stackrel{\bullet}{-} \sigma_1 = (\sigma_1\sigma_2) \stackrel{\bullet}{-} (\mathcal{P}(\tau) + \mathcal{P}(\sigma_1)) = (\sigma_1\sigma_2 \stackrel{\bullet}{-} \sigma_1) \stackrel{\bullet}{-} \tau = \sigma_2 \stackrel{\bullet}{-} \tau$. Hence $s[\sigma_2 \stackrel{\bullet}{-} \tau\rangle$ for any sequence σ_2 enabled at s, as claimed. $\square$

Since we intend to check semiflows, we shall thus introduce the following definition:

Definition 2 REDUCED SEQUENCES WITH RESPECT TO A T-VECTOR
Let $TS = (S, T, \rightarrow, \imath)$ be any lts and Φ be any T-vector. We shall say that TS satisfies the *property of reduced sequences with respect to* Φ if $\forall \sigma \in T^*\colon \imath[\sigma\rangle \Rightarrow \imath[\sigma \stackrel{\bullet}{-} \Phi\rangle$. $\square$

Thus, if an lts is cf-solvable, it must satisfy the reduced sequences property with respect to any semiflow of the underlying net, and in particular, with respect to any semiflow derived from any small cycle anywhere in the set of given transition systems if the considered lts is part of a simultaneous synthesis problem. However, all enabled sequences from $\imath$ need to be considered, and this is still impractical. We turn to distance paths.

From the definition of distances, we know that, in the reachability graph of a bounded choice-free system, for any reachable marking M and any Parikh vector Υ of a small cycle we have $\Delta_M \not\geq \Upsilon$; moreover, from the distance path property, there is a path $M_0[\sigma\rangle M$ such that $\mathcal{P}(\sigma) = \Delta_M$. The following result allows to simplify the corresponding checks, by avoiding the explicit computation of the modulo operations and by only considering a single short path to each state (i.e., a path from the initial state to another state such that there is no shorter one to that state) of the kind produced by the computation of a breadth-first spanning tree.

Lemma 1 SHORT PATHS AND DISTANCE PATHS
Let $TS = (S, T, \rightarrow, \imath)$ be a finite, totally reachable, deterministic and persistent lts satisfying the disjoint small cycle property for the set $\{\Upsilon_1, \ldots, \Upsilon_n\}$. If, for some state $s \in S$, there is a path $\imath[\sigma\rangle s$ such that, $\forall i \in \{1, \ldots, n\}$, $\mathcal{P}(\sigma) \not\geq \Upsilon_i$, then $\imath[\sigma\rangle s$ is a short path, all the short paths to s have the same Parikh vector and are distance paths. It also results that distance paths are short paths.

Proof: We know that $\Delta_s = \mathcal{P}(\sigma) \bmod \{\Upsilon_1, \ldots, \Upsilon_n\}$, so that if $\mathcal{P}(\sigma) \not\geq \Upsilon_i$ for all i's, then $\mathcal{P}(\sigma) = \Delta_s$ and $\imath[\sigma\rangle s$ is a distance path. It is also a short path since for any path $\imath[\sigma'\rangle s$, we know that $\mathcal{P}(\sigma) = \Delta_s = \mathcal{P}(\sigma') \bmod \{\Upsilon_1, \ldots, \Upsilon_n\}$ $\leq \mathcal{P}(\sigma')$. This also implies that if $\imath[\sigma'\rangle s$ is a short path, $\mathcal{P}(\sigma) = \mathcal{P}(\sigma')$. □

Distance paths are thus characterised by the property that their Parikh vectors are not greater or equal to the Parikh vectors of the small cycles, but their shortness also respects any other (minimal) semiflow:

Corollary 2 CONSTRAINT ON DISTANCES
If Φ is a semiflow of a bounded choice-free net system, in its reachability graph we have that $\forall M \in [M_0\rangle$: $\Delta_M \not\geq \Phi$.

Proof: Let $M_0[\sigma\rangle M$ be a distance path. Theorem 3 shows that $M_0[\sigma \mathbin{\dot{-}} \Phi\rangle M'$ for some M', and from Keller's theorem, if $\Delta_M = \mathcal{P}(\sigma) \geq \Phi$, we have $M_0[\sigma \mathbin{\dot{-}} \Phi\rangle M'[\sigma'\rangle M$ with $\mathcal{P}(\sigma') = \Phi$. Since Φ is a semiflow, $M' = M$, and since $\sigma \mathbin{\dot{-}} \Phi$ is smaller than σ, $M_0[\sigma \mathbin{\dot{-}} \Phi\rangle M$ is also a distance path to M, smaller than σ, a contradiction. □

We shall introduce a corresponding property to be checked during pre-synthesis:

Definition 3 SHORT DISTANCES WITH RESPECT TO A T-VECTOR
Let $TS = (S, T, \to, \imath)$ be an lts with a distance notion Δ and let Φ be a T-vector. TS will be said to satisfy the *short distance property with respect to* Φ if $\forall s \in S : \Delta_s \not\geq \Phi$. □

Corollary 3 REDUCED DISTANCES
For each reachable marking M and each semiflow Φ of a choice-free net system, there is a reachable marking M' such that $\Delta_{M'} = \Delta_M \mathbin{\dot{-}} \Phi$.

Proof: This arises from the fact that, from the distance path property, for each reachable marking M there is a path σ such that $M_0[\sigma\rangle M$ and $\mathcal{P}(\sigma) = \Delta_M$. Then, from Theorem 3, there is a marking M' such that $M_0[\sigma \mathbin{\dot{-}} \Phi\rangle M'$. Moreover, distance paths are characterised by the fact that for any small cycle $\widetilde{M}[\tau\rangle\widetilde{M}$ the Parikh vector of the path is not greater or equal to $\mathcal{P}(\tau)$. Hence, since $\mathcal{P}(\sigma \mathbin{\dot{-}} \Phi) \leq \mathcal{P}(\sigma)$, $\sigma \mathbin{\dot{-}} \Phi$ is also a distance path and $\Delta_M \mathbin{\dot{-}} \Phi = \mathcal{P}(\sigma \mathbin{\dot{-}} \Phi) = \Delta_{M'}$. □

Corollary 3 is illustrated by Figure 6. In TS_9, we have $\imath[abb\rangle$ and a small cycle $s[abc\rangle s$ (with $\mathcal{P}(abc)$ being a semiflow if this lts is solvable), but not $\imath[b\rangle$ (which is what $\imath[abb \stackrel{\bullet}{-} abc\rangle$ evaluates to). Hence Corollary 3 (or, *a fortiori*, Theorem 3) prohibits any choice-free synthesis of TS_9. By contrast, TS_{10} allows the choice-free solution PNS_{10}.

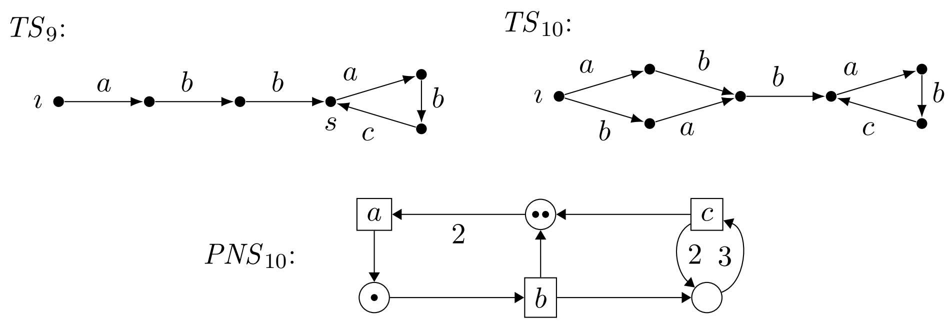

Figure 6: Illustration of the reduced distance property. TS_9 does not satisfy the reduced distance property with respect to $\mathcal{P}(abc)$, but TS_{10} does.

Definition 4 REDUCED DISTANCES WITH RESPECT TO A T-VECTOR
Let $TS = (S, T, \rightarrow, \imath)$ be any finite, deterministic, persistent lts with disjoint small cycles (so that distances are well defined) and Φ be any T-vector. We shall say that TS satisfies the *property of reduced distances with respect to* Φ if $\forall s \in S, \exists s' \in S : \Delta_{s'} = \Delta_s \stackrel{\bullet}{-} \Phi$. □

Thus, from Corollary 3, if a finite lts is cf-solvable, it must satisfy the reduced distances property with respect to any semiflow. By only considering Parikh vectors of distance paths, we get a finite number of checks to be performed on the given finite lts during a cf-pre-synthesis. This does not deliver the full power of Theorem 3 however, but the next property will fill the gap.

In a deterministic and persistent lts, cycles may be pushed forward Parikh-equivalently [2]: if $s[\tau\rangle s[\sigma\rangle s'$, then $s'[\tau'\rangle s'$ for some τ' with $\mathcal{P}(\tau') = \mathcal{P}(\tau)$. We shall now see that, if it is solvable by a cf-system, they may also be pushed backward, up to some extent.

Proposition 2 Backward pushing of cycles
In a choice-free net system, if $M_0[\sigma\rangle M[\tau\rangle M$ and Φ is a semiflow disjoint from τ, then there is some M' such that $M_0[\sigma \overset{\bullet}{-} \Phi\rangle M'[\tau\rangle M'$.

Proof: By Theorem 3, $M_0[(\sigma\tau) \overset{\bullet}{-} \Phi\rangle M'$, but since τ is disjoint from Φ, $(\sigma\tau) \overset{\bullet}{-} \Phi = (\sigma \overset{\bullet}{-} \Phi)\tau$. As a consequence, since $\mathcal{P}(\tau)$ is a semiflow, $M_0[\sigma \overset{\bullet}{-} \Phi\rangle M'[\tau\rangle M'$. □

This leads to another property to be checked during pre-synthesis:

Definition 5 Earliest cycles with respect to a T-vector
Let $TS = (S, T, \rightarrow, \imath)$ be any finite, deterministic, persistent lts with disjoint small cycles (so that distances are well defined) and Φ be any T-vector. We shall say that TS satisfies the *property of earliest cycles with respect to* Φ if $\forall s \in S$ with $s[\tau\rangle s$ for some τ disjoint from Φ, $\exists s' \in S$ with $\Delta_{s'} = \Delta_s \overset{\bullet}{-} \Phi$ and $s'[\tau\rangle s'$. □

In particular, Φ may be the Parikh vector of a small cycle. This is illustrated by Figure 7.

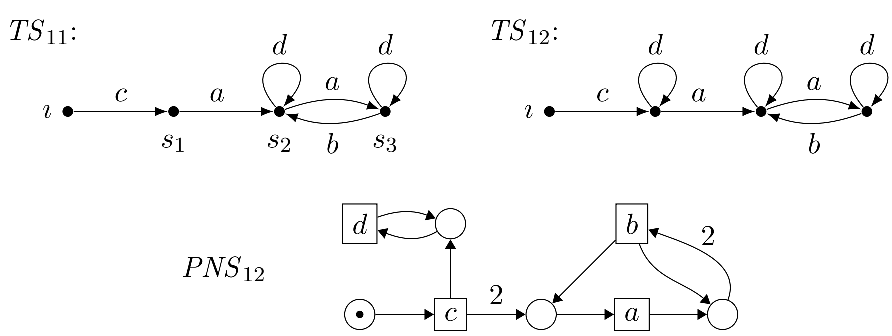

Figure 7: Illustration of the earliest cycle property. TS_{11} satisfies the distance path and the reduced distance properties, but not the earliest cycles property since there is a cycle ba at s_3, a cycle d at s_2 and $\Delta_{s_2} = (1, 0, 1, 0)$, so that there should be a cycle d at distance $\Delta_{s_2} \overset{\bullet}{-} \mathcal{P}(ba) = (0, 0, 1, 0)$, i.e. at s_1. Hence TS_{11} has no choice-free solution. This is corrected in TS_{12}, which has the choice-free solution PNS_{12}.

We also consider a slightly weaker version of the early cycle property:

Definition 6 EARLIEST PARIKH CYCLES WITH RESPECT TO A T-VECTOR
Let $TS = (S, T, \to, \imath)$ be any finite, deterministic, persistent lts with disjoint small cycles (so that distances are well defined) and Φ be any T-vector. We shall say that TS satisfies the *property of earliest Parikh cycles with respect to* Φ if $\forall s \in S$ with $s[\tau\rangle s$ for some τ disjoint from Φ, $\exists s' \in S$ with $\Delta_{s'} = \Delta_s \stackrel{\bullet}{-} \Phi$ and $s'[\tau'\rangle s'$ with $\mathcal{P}(\tau') = \mathcal{P}(\tau)$. $\square$

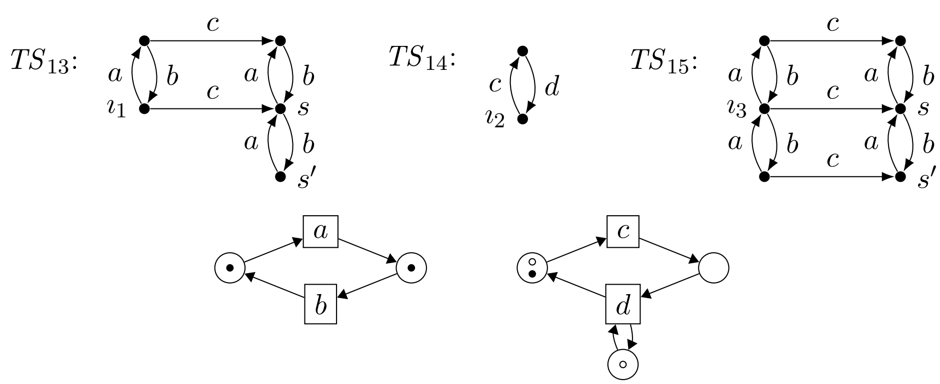

Figure 8: Illustration of the difference between the earliest cycles property and the earliest Parikh cycles property.

The difference between the earliest cycles and earliest Parikh cycles constraints is illustrated in Figure 8. The earliest cycles property is not satisfied by the pair TS_{13} and TS_{14}. Indeed, the cycle ba is missing at $\imath_1$ since there is the cycle $s[ba\rangle s$ at s and subtracting the cyclic Parikh vector $\mathcal{P}(cd)$ arising at $\imath_2$ from the path $\imath_1[c\rangle s$ we should have the cycle $\imath_1[ba\rangle\imath_1$. Hence the simultaneous cf-synthesis of TS_{13} and TS_{14} necessarily fails. By contrast, the earliest Parikh cycles property is satisfied by that pair since we have cycles with Parikh vector $\mathcal{P}(ab)$ at each state of TS_{13}. However, we may also derive the non-simultaneous cf-synthesisability of TS_{13} and TS_{14} from the reduced distance property for the Parikh vector $\mathcal{P}(cd)$, since, in TS_{13}, $\Delta_{s'} = \mathcal{P}(cb)$ implies the existence of a state s'' such that $\Delta_{s''} = \mathcal{P}(cb) \stackrel{\bullet}{-} \mathcal{P}(cd) = \mathcal{P}(b)$, but this is not the case. This is not an accident, as will be showed by Theorem 4 next: when combined with other properties, the earliest Parikh cycles property implies the reduced sequence property, hence in turn the earliest cycles property, so that the earliest Parikh cycles property is equivalent to the earliest cycles property in presence of these other properties. The missing cycle is added in TS_{15} whose simultaneous cf-synthesis with TS_{14} yields the solution depicted on the bottom of the figure: the black tokens correspond to TS_{15}, the hollow ones to TS_{14}.

All checks derived so far can be joined together in the next theorem,[4] forming the blueprint for a practical algorithm as described in Section 5.

Theorem 4 EVERYTHING CAN BE CHECKED WITH FINITELY MANY CHECKS
Let $TS = (S, T, \to, \imath)$ be a finite, deterministic and persistent labelled transition system with prime disjoint small cycles (hence with a distance notion; we shall denote by $\mathcal{G}$ the set of their Parikh vectors) and satisfying the distance path property. Also, let Φ be a prime T-vector satisfying

1. *$\forall \Upsilon \in \mathcal{G}$: either $\Phi = \Upsilon$ or Φ is disjoint from Υ (disjointness property);*
2. *for any state s, $\Delta_s \not\geq \Phi$ (short distance property);*
3. *for any state s, we have $\imath[\Delta_s \mathbin{\dot{-}} \Phi\rangle$ (reduced distance property);*
4. *for any state s and Parikh cycle $\Upsilon \in \mathcal{G}$ disjoint from Φ, we have $s[\Upsilon\rangle \Rightarrow \imath[\Delta_s \mathbin{\dot{-}} \Phi\rangle\tilde{s}[\Upsilon\rangle\tilde{s}$ for some state $\tilde{s} \in S$ (earliest Parikh cycles property).*

Then, for any firing sequence $\imath[\sigma\rangle$, also $\imath[\sigma \mathbin{\dot{-}} \Phi\rangle$ (reduced sequence property; more precisely, Proposition 1).

Proof: Let us proceed by induction on the length of the path $\imath[\sigma\rangle s$ (the base case $\sigma = \varepsilon$ is trivial since then $\sigma \mathbin{\dot{-}} \Phi = \varepsilon$ and the empty sequence can be realised everywhere). Hence we assume $\sigma = \sigma' a$, $\imath[\sigma'\rangle s'$ and $\imath[\sigma' \mathbin{\dot{-}} \Phi\rangle$: we want to show that $\imath[\sigma \mathbin{\dot{-}} \Phi\rangle$.

If $\sigma \mathbin{\dot{-}} \Phi = \sigma' \mathbin{\dot{-}} \Phi$, i.e. if $\mathcal{P}(\sigma')(a) < \Phi(a)$, we trivially have $\imath[\sigma \mathbin{\dot{-}} \Phi\rangle$ as requested.

Otherwise, $\mathcal{P}(\sigma')(a) \geq \Phi(a)$ and $\sigma \mathbin{\dot{-}} \Phi = (\sigma' \mathbin{\dot{-}} \Phi)a$. If we can show that $\imath[\mathcal{P}(\sigma \mathbin{\dot{-}} \Phi)\rangle$, by Keller's theorem we also have $\imath[\sigma' \mathbin{\dot{-}} \Phi\rangle[(\sigma \mathbin{\dot{-}} \Phi) \mathbin{\dot{-}} (\sigma' \mathbin{\dot{-}} \Phi)\rangle = \imath[\sigma' \mathbin{\dot{-}} \Phi\rangle[\mathcal{P}(a)\rangle = \imath[\sigma' \mathbin{\dot{-}} \Phi\rangle[a\rangle = \imath[\sigma \mathbin{\dot{-}} \Phi\rangle$ and we are done.

In general, since the small cycles are disjoint and we have the distance path property by assumption, we have $\mathcal{P}(\sigma) = \Delta_s + \sum_{\Upsilon \in \mathcal{G}} k_\Upsilon \cdot \Upsilon$, for some integer coefficients $k_\Upsilon \in \mathbb{N}$, $\imath[\Delta_s\rangle s$ and, with $\mathcal{G}' = \{\Upsilon \in \mathcal{G} \mid k_\Upsilon > 0\}$, for each

[4] Whose premises can be assumed after successful pre-synthesis according to Theorem 1.

$\Upsilon \in \mathcal{G}'$, $s[\Upsilon\rangle s$ (see Theorem 22 in [3]). We shall now consider two subcases based on the fact that, from the disjointness property hypothesis (point 1 above), either $\Phi = \Upsilon$ for some $\Upsilon \in \mathcal{G}'$ or Φ is disjoint from each $\Upsilon \in \mathcal{G}'$.

If $\Phi = \Upsilon$ for some $\Upsilon \in \mathcal{G}'$, $\mathcal{P}(\sigma \dot{-} \Phi) = \Delta_s + (k_\Phi - 1)\cdot\Phi + \sum_{\Upsilon\in\mathcal{G}'\setminus\{\Phi\}} k_\Upsilon \cdot \Upsilon$. Since $\imath[\Delta_s\rangle s[\Phi^{(k_\Phi-1)}\rangle s \ldots s[\Upsilon^{k_\Upsilon}\rangle s \ldots$, we may apply the argument above to deduce $\imath[\sigma \dot{-} \Phi\rangle$.

If Φ is disjoint from each $\Upsilon \in \mathcal{G}'$, $\mathcal{P}(\sigma \dot{-} \Phi) = (\Delta_s \dot{-} \Phi) + \sum_{\Upsilon\in\mathcal{G}'} k_\Upsilon \cdot \Upsilon$. From point 3 above we have $\imath[\Delta_s \dot{-} \Phi\rangle\tilde{s}$ for some $\tilde{s} \in S$, and from point 4 we have $\tilde{s}[\Upsilon\rangle\tilde{s}$ for each $\Upsilon \in \mathcal{G}'$. Hence $\imath[\sigma \dot{-} \Phi\rangle$, as requested. □

In Theorem 4, we assume that Φ is either one of the members of $\mathcal{G}$ or disjoint from them. Hence it may be any member of $\mathcal{G}$. Our next aim is to show that this extends to any small cycle in a set of transition systems, provided they are simultaneously cf-solvable.

Lemma 2 Increasing residues
In a choice-free net with incidence matrix C, if Φ_1 and Φ_2 are two different minimal semiflows with a non-empty intersection, then $C \cdot (\Phi_1 \dot{-} \Phi_2) \gneqq \mathbf{0}$ (and, similarly, $C \cdot (\Phi_2 \dot{-} \Phi_1) \gneqq \mathbf{0}$).

Proof: Let us choose an initial marking M_0 high enough in order to have $M_0[\sigma_1\rangle M_0$ and $M_0[\sigma_2\rangle M_0$ with $\mathcal{P}(\sigma_1) = \Phi_1$ and $\mathcal{P}(\sigma_2) = \Phi_2$. They are small cycles around M_0 (see Corollary 1) and $\sigma_1 \neq \sigma_1 \dot{-} \sigma_2 \neq \varepsilon$ as well as $\sigma_2 \neq \sigma_2 \dot{-} \sigma_1 \neq \varepsilon$ ($\sigma_1 \neq \sigma_1 \dot{-} \sigma_2$ since they have a common transition in their supports and $\sigma_1 \dot{-} \sigma_2 \neq \varepsilon$ otherwise σ_1 is smaller than σ_2 while their Parikh vectors are different minimal semiflows of N; and similarly for $\sigma_2 \dot{-} \sigma_1$). Since the reachability graph is deterministic and persistent, by Keller's theorem there is a marking M_1 such that $M_0[\sigma_2\rangle M_0[\sigma_1 \dot{-} \sigma_2\rangle M_1$ and $M_0[\sigma_1\rangle M_0[\sigma_2 \dot{-} \sigma_1\rangle M_1$. We may apply Keller again to get $M_1[(\sigma_1 \dot{-} \sigma_2) \dot{-} (\sigma_2 \dot{-} \sigma_1)\rangle M_2$ and $M_1[(\sigma_2 \dot{-} \sigma_1) \dot{-} (\sigma_1 \dot{-} \sigma_2)\rangle M_2$ for some marking M_2. Since $\sigma_1 \dot{-} \sigma_2$ and $\sigma_2 \dot{-} \sigma_1$ have disjoint supports, the residue does not change the sequence, e.g. $(\sigma_1 \dot{-} \sigma_2) \dot{-} (\sigma_2 \dot{-} \sigma_1) = \sigma_1 \dot{-} \sigma_2$. Thus, we may apply Keller again and again, constructing a series of markings $M_2, M_3, \ldots, M_n, \ldots$ with $M_i[\sigma_1 \dot{-} \sigma_2\rangle M_{i+1}$ and $M_i[\sigma_2 \dot{-} \sigma_1\rangle M_{i+1}$ for each $i \in \mathbb{N}$. Since the difference between two consecutive markings $M_{i+1} - M_i = C \cdot (\Phi_1 \dot{-} \Phi_2) = C \cdot (\Phi_2 \dot{-} \Phi_1)$ stays constant, either $M_0 = M_1$, or $i \neq j \Rightarrow M_i \neq M_j$. In the first case, $\sigma_1 \dot{-} \sigma_2$ (as well as $\sigma_2 \dot{-} \sigma_1$) yields through its Parikh vector a smaller cycle than σ_1, a contradiction. In the second case, since $M_{i+1} - M_i = C \cdot (\Phi_1 \dot{-} \Phi_2)$

and $M_i \geq \mathbf{0}$ for each i, $C \cdot (\Phi_1 \mathbin{\dot{-}} \Phi_2)$ may not have a negative component, and must have at least one strictly positive component. Hence the first claim. The other one follows by symmetry. □

Theorem 5 Small cycles in a simultaneous synthesis
Let (N, M_0^1) and (N, M_0^2) be two bounded systems arising from the same choice-free net N. Let $M_1[\sigma_1\rangle M_1$ be a small cycle in the reachability graph of the former and $M_2[\sigma_2\rangle M_2$ be a small cycle in the reachability graph of the latter. Then σ_1 and σ_2 are either Parikh-equivalent or disjoint.

Proof: Let us assume this is not the case so that, from Corollary 1, $\mathcal{P}(\sigma_1)$ and $\mathcal{P}(\sigma_2)$ are different minimal semiflows with a non-empty intersection. From Theorem 3, we have $M_1[\sigma_1 \mathbin{\dot{-}} \mathcal{P}(\sigma_2)\rangle M_1'$ for some marking M_1' and, since $\mathcal{P}(\sigma_1 \mathbin{\dot{-}} \mathcal{P}(\sigma_2)) = \mathcal{P}(\sigma_1) \mathbin{\dot{-}} \mathcal{P}(\sigma_2)$, from Lemma 2 we have $M_1' = M_1 + C \cdot (\mathcal{P}(\sigma_1) \mathbin{\dot{-}} \mathcal{P}(\sigma_2)) \gneqq M_1$, which implies the non-boundedness of (N, M_0^1), a contradiction. □

Theorem 5 implies that there is no simultaneous choice-free solution of the two transition systems TS_3 and TS_4 shown in Figure 2, since the two small cycles (one in TS_3, the other one in TS_4) are neither Parikh-equivalent nor label-disjoint. This is true despite the fact that TS_3 and TS_4 have individual choice-free solutions.

Let us now assume that for a specific input $\{TS_1, \ldots, TS_m\}$, the properties mentioned in Theorem 1 have been checked on all TS_i. In case any of these tests fail, there can be no successful individual (and by implication, no successful simultaneous) cf-synthesis. If all of them succeed, we shall also get, for each $i \in \{1, \ldots, m\}$, the set $\mathcal{G}_i = \{\Upsilon_1^i, \ldots, \Upsilon_{n_i}^i\}$ of the different Parikh vectors of the small cycles in TS_i, as well as the distances Δ_s between the i'th initial state $\imath_i$ and state $s \in S_i$ (recall that we assumed the various state spaces S_i disjoint).

Let $\mathcal{G} = \bigcup_{i=1}^m \mathcal{G}_i$. From Corollary 1, each member of $\mathcal{G}$ must be a minimal semiflow of any simultaneous cf-solution. Theorem 5 implies that all the different members of $\mathcal{G}$ must have disjoint supports; if this is not the case, this is a new reason for the simultaneous synthesis failure (as exemplified in Figure 2).

The pre-synthesis in [4, 17] already checks that, in each TS_i, for each state $s \in S_i$ and each $\Upsilon \in \mathcal{G}_i$, we have $\Delta_s \not\geq \Upsilon$, where Δ_s denotes both the distance in TS_i and the unique Parikh vector of all the short paths from $\imath_i$

to s. Following Corollary 2, we may now check that the constraint $\Delta_s \not\gtrsim \Upsilon$ remains true for each $\Upsilon \in \mathcal{G} \setminus \mathcal{G}_i$, satisfying Definition 3. Again, if this is not true, we have a new reason of failure. This is exemplified in Figure 9.

From Corollary 3, we may improve the individual pre-synthesis procedure in [4] and [17] by checking that, in each TS_i, for each state $s \in S_i$ and each $\Upsilon \in \mathcal{G}_i$, for some $s' \in S_i$, we have $\Delta_{s'} = \Delta_s \mathbin{\dot{-}} \Upsilon$. For the simultaneous pre-synthesis, we may do the same with each $\Upsilon \in \mathcal{G}$. Again, if this is not true, we have another reason of failure (of an individual or simultaneous synthesis, respectively). This is also exemplified in Figure 9.

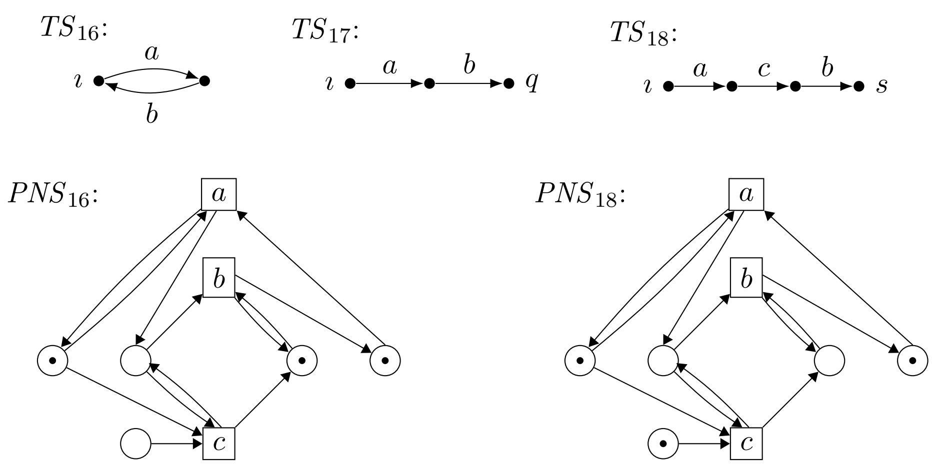

Figure 9: Two transition systems TS_{16} and TS_{17} which are individually, but not simultaneously, solvable by a choice-free net. In TS_{16}, there is a small cycle $\imath[ab\rangle\imath$, while in TS_{17}, $\Delta_q = \mathcal{P}(ab)$ for a state $q \neq \imath$, so that $\neg(\Delta_q \not\gtrsim \mathcal{P}(ab))$. This contravenes Corollary 2 (in fact, TS_{16} and TS_{17} are not simultaneously solvable by any Petri net, since $\mathcal{P}(ab)$ should be a semiflow of the solution, so that q should coincide with the initial state of TS_{17}). TS_{16} and TS_{18} are individually and simultaneously solvable (see PNS_{16} and PNS_{18}), but not choice-freely. In TS_{18}, there is no state s' with $\Delta_{s'} = \mathcal{P}(c) = \Delta_s \mathbin{\dot{-}} \mathcal{P}(ab)$, contravening Corollary 3. The net underlying the simultaneous non-choice-free solution given here contains three choice places, and the developments above prove that at least one such place must be present in any simultaneous solution.

Similarly, it is possible to use Proposition 2, i.e. the earliest cycles property in Definition 4, to strengthen the individual as well as the simultaneous pre-synthesis procedure. Here, however, it is necessary to complete the structures built during the first part of the individual pre-syntheses. Besides constructing the distances Δ_s for each $s \in S_i$ and the Parikh vectors of small cycles, we need to determine for each $s \in S_i$ the *set* $\mathcal{C}_s$ *of small cycles around* s. Then, for the individual pre-synthesis, we may check that, for each $s \in S_i$, $\tau \in \mathcal{C}_s$ and $\Upsilon \in (\mathcal{G}_i \setminus \{\mathcal{P}(\tau)\})$, there is a state $s' \in S_i$ such that $\tau \in \mathcal{C}_{s'}$ while $\Delta_{s'} = \Delta_s \stackrel{\bullet}{-} \Upsilon$. For the simultaneous pre-synthesis, for each $i \in \{1, \dots, m\}$, we add that, for each $\Upsilon' \in \mathcal{G} \setminus \mathcal{G}_i$, there is a state $s' \in S_i$ such that $\tau \in \mathcal{C}_{s'}$ while $\Delta_{s'} = \Delta_s \stackrel{\bullet}{-} \Upsilon'$. This is illustrated by Figure 10.

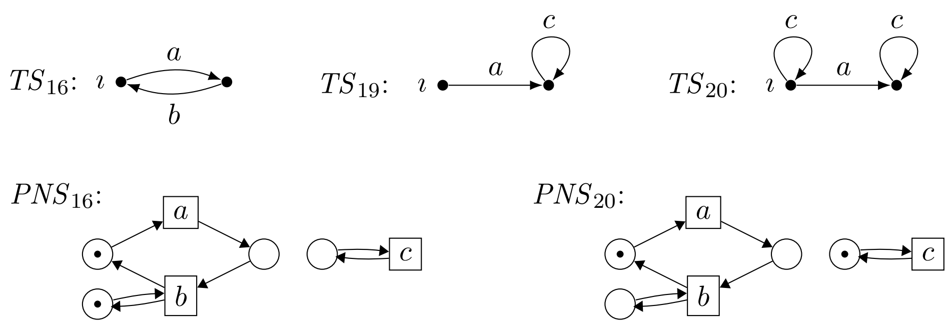

Figure 10: The two transition systems TS_{19} and TS_{20} are individually cf-solvable. TS_{19} is not simultaneously cf-solvable with TS_{16}, since there is a loop c after a, but not initially while $a \stackrel{\bullet}{-} ab = \varepsilon$, contravening Proposition 2. By contrast, TS_{20} is simultaneously cf-solvable with TS_{16}, as shown by PNS_{16} and PNS_{20}.

It is possible to alleviate the burden of computing and recording all the small cycles by only recording the Parikh vectors of small cycles around states, i.e. to use the earliest Parikh cycles property in Definition 6 instead of the earliest cycles property. One advantage is that one records less information, since it may happen that many cycles around s have the same Parikh vector. Moreover, it is not even necessary to record the information for *every* state. Indeed, since we know that cycles may be pushed forward Parikh-equivalently (see [2]), one only has to record an information for s when this information is not recorded for a state s' with $s \in [s'\rangle$.

Usually, during pre-synthesis, the state space is explored by means of a breadth-first spanning tree (see Section 5). This tree can conveniently be used, since one only has to record a minimal semiflow corresponding to a small cycle around s if its direct predecessor in the spanning tree does not present it (let us recall that, since cycles may be pushed forward Parikh equivalently, if an indirect predecessor presents the information, so will do all the intermediate states in the spanning tree). We shall thus define $\mathcal{D}_s$ *as the set of Parikh vectors of small cycles around* s *which do not occur in* $\mathcal{D}_{s'}$ *where* s' *is the direct predecessor of* s *in the considered spanning tree.* For the individual pre-synthesis of the various TS_i's, we then simply have to check that $\forall s \in S_i, \Phi \in \mathcal{D}_s : \Delta_s$ is disjoint from each $\Upsilon \in \mathcal{G}_i \setminus \{\Phi\}$. For the simultaneous pre-synthesis, we have to add that $\forall s \in S_i, \Phi \in \mathcal{D}_s : \Delta_s$ is disjoint from each $\Upsilon \in \mathcal{G} \setminus \mathcal{G}_i$. When taken individually, those tests based on $\mathcal{D}$ are weaker than the tests based on $\mathcal{C}$ (see Figure 8), since they only check Parikh vectors instead of small cycles but, as already noticed above, when considered in conjunction with the other checks above this is no longer the case: point 4 of Theorem 4 only uses Parikh vectors of small cycles, not the small cycles themselves, and this is enough to imply the reduced sequence property of Theorem 3 through Proposition 1, hence also Proposition 2.

5 Algorithmic Complexity of the Pre-Synthesis

In this section we take a closer look at the algorithmic complexity of the approach suggested by Theorem 4. In [4, 17] we have seen that finding the Parikh vectors of all small cycles, testing their disjointness, and checking the short distance and distance path properties can be done in $O(|S| \cdot |T|^2)$ for a finite, deterministic, and persistent lts $(S, T, \to, \imath)$.

Algorithm 1 shows how to test the reduced distance property with respect to the set of the Parikh vectors of the small cycles of a given lts. Since these Parikh vectors will be semiflows in the synthesised net, Theorem 3 must be applicable. Consider now a breadth-first spanning tree $E \subseteq \to$ of our lts, with $|E| = |S| - 1$, i.e. each state except $\imath$ has exactly one incoming edge in E. Since E is constructed breadth-first, each distance Δ_s (for $s \in S$) is represented by a unique path $\imath[\sigma\rangle s$ in E. By Theorem 3, $\imath[\sigma \stackrel{\bullet}{-} \Phi\rangle$ must be true for any small cycle with Parikh vector Φ. We conclude $\imath[\Delta_s \stackrel{\bullet}{-} \Phi\rangle$, i.e. to check the reduced distances it is sufficient to check $\imath[\sigma\rangle \Rightarrow \imath[\sigma \stackrel{\bullet}{-} \Phi\rangle$ for paths $\imath[\sigma\rangle$ *in the spanning tree.*

Algorithm 1 Test for the reduced distances with respect to small cycles

```
Input: a finite, deterministic, persistent lts (S, T, →, ı)
       a breadth-first spanning tree E ⊆ → (with |E| = |S| − 1)
       the set 𝒢 of Parikh vectors of small cycles
  check disjointness of 𝒢 and the short distance property
  for each Φ ∈ 𝒢:
    if ¬reducedDistance(ı, ı, Φ) then return false endif
  return true

  procedure reducedDistance(s ∈ S, s′ ∈ S, Θ: T → ℕ): boolean
    for each (s, t, r) ∈ E:
      if Θ(t) > 0 then
        let Θ(t) := Θ(t) − 1
        if ¬reducedDistance(r, s′, Θ) then return false endif
        let Θ(t) := Θ(t) + 1
      else if ∃(s′, t, r′) ∈ → then
        if ¬reducedDistance(r, r′, Θ) then return false endif
      else return false endif
    return true
```

The algorithm uses two state variables s and s' to represent the states reached by $\imath[\sigma\rangle s$ and $\imath[\sigma \stackrel{\bullet}{-} \Phi\rangle s'$ for paths σ in the spanning tree. The parameter Θ realises the part "$\stackrel{\bullet}{-}\Phi$" by preventing advancement of s' via a t-edge as long as $\Theta(t) > 0$, instead, $\Theta(t)$ is reduced by one. Overall, the algorithm makes a depth-first run through the spanning tree E for each Parikh vector $\Phi \in \mathcal{G}$ of a small cycle, where $|\mathcal{G}| \leq |T|$ by disjointness and $|E| < |S|$. When we recursively follow an edge $s[t\rangle r$ there are three cases: not to advance s' if $\Theta(t) > 0$, otherwise to follow the edge $s'[t\rangle r'$ if it exists, or to report a failure if such an edge is not found. This is realised by the if-statement in the recursive procedure *reducedDistance*. A failure is immediately propagated back by the "**return** *false*" statements, stopping the whole recursion. If arguments are passed on by reference to *reducedDistance* (or we implement the algorithm iteratively), no copying of Θ is required. The checks for edges $(s, t, r) \in E$ and $(s', t, r') \in \rightarrow$ can be done in constant time with adequate data structures. As the algorithm makes a depth-first run through the spanning tree (via the variable s) for each element of $\mathcal{G}$, we obtain a run time of $O(|S| \cdot |T|)$ for the outer loop. The initial checks for disjointness and short distances take $O(|S| \cdot |T|^2)$.

Checking the earliest Parikh cycle property is a bit more involved. As discussed before Theorem 4, after ensuring the reduced distance property by

Algorithm 1, we need to find the earliest states at which a small cycle with Parikh vector $\Phi \in \mathcal{G}$ occurs, in any permutation.

Using our spanning tree E, we can define the Parikh vector of an edge $e = (s, t, s') \in \rightarrow$ by $\mathcal{P}_E(e) = \Delta_s + \mathcal{P}(t) - \Delta_{s'}$. Clearly, for any edge $e \in E$, $\mathcal{P}_E(e) = \mathbf{0}$ since the distances $\Delta_{s'}$ and $\Delta_s + \mathcal{P}(t)$ are realised by the same path in the spanning tree. Also, for any cycle $s_0[t_1\rangle s_1 \ldots s_{n-1}[t_n\rangle s_0$, $\mathcal{P}_E(s_0[t_1 \ldots t_n\rangle s_0) = \sum_{1\le i\le n} \mathcal{P}_E(s_{i-1}[t_i\rangle s_{i \bmod n}) = \mathcal{P}(t_1 \ldots t_n)$ since each occurring $-\Delta_{s_i}$ is annihilated by a $+\Delta_{s_i}$ for the following edge, leaving only the labels of the cycle to be added up.

In [17] it was shown that in the reachability graph of a choice-free net, for every edge e either $\mathcal{P}_E(e) = \mathbf{0}$ or $\mathcal{P}_E(e) \in \mathcal{G}$, no matter how we construct the breadth-first spanning tree E. Since $\mathcal{G}$ is a set of disjoint and thus linearly independent vectors, the Parikh vector $\mathcal{P}(\sigma) = \Phi \in \mathcal{G}$ of a small cycle $s[\sigma\rangle s$ must be composed from one edge $e \in \rightarrow \setminus E$ in σ with $\mathcal{P}_E(e) = \Phi$ while all other edges used for σ contribute the Parikh vector $\mathbf{0}$. We thus find an edge (in $\rightarrow \setminus E$) with Parikh vector Φ such that s is reachable from its target state.

To see that the reverse is also true, assume an edge $e = (r, t, r') \in \rightarrow \setminus E$ with $\mathcal{P}_E(e) = \Phi$. By construction of the spanning tree E, there are distance paths $\imath[\sigma\rangle r$ and $\imath[\sigma'\rangle r'$ in E. With Keller's theorem we find $\imath[\sigma t\rangle r'[\sigma' \mathbin{\dot{-}} (\sigma t)\rangle s$ and $\imath[\sigma'\rangle r'[(\sigma t) \mathbin{\dot{-}} \sigma'\rangle s$ for some state s. As $\mathbf{0} \le \mathcal{P}_E(e) = \Delta_r + \mathcal{P}(t) - \Delta_{r'} = \mathcal{P}(\sigma t) - \mathcal{P}(\sigma')$, we conclude $\sigma' \mathbin{\dot{-}} (\sigma t) = \varepsilon$ and $r' = s$. Thus, $(\sigma t) \mathbin{\dot{-}} \sigma'$ is a (small) cycle with Parikh vector $\mathcal{P}_E(e) = \Phi$ at the target state $r' = s$ of the edge e. Any successor state of s also allows a small cycle with Parikh vector Φ. If $s[t\rangle s'$, Keller's theorem yields $s[t\rangle s'[\Phi \mathbin{\dot{-}} t\rangle s''$ and $s[\Phi\rangle s[t \mathbin{\dot{-}} \Phi\rangle s''$ with either $t \mathbin{\dot{-}} \Phi = t$, $\Phi \mathbin{\dot{-}} t = \Phi$, and $s'' = s'$, or $t \mathbin{\dot{-}} \Phi = \varepsilon$, $s'' = s$, and $\mathcal{P}(t) + \Phi \mathbin{\dot{-}} t = \Phi$.

In summary, s allowing a small cycle with Parikh vector Φ is equivalent to s being reachable from the target state of an edge in $\rightarrow \setminus E$ with Parikh vector Φ. So, we can find the earliest states allowing a small cycle with Parikh vector Φ by running a depth-first search through our spanning tree, checking if we encounter the target state s of such an edge $e \in \rightarrow \setminus E$ with $\mathcal{P}_E(e) = \Phi$, and backtracking if we find one. Before we backtrack, we check if the state fulfills the simplified version $\mathcal{D}$ (see p. 221 top) of the earliest Parikh cycle property, i.e. if the distance Δ_s is disjoint with all other Parikh vectors of small cycles, $\mathcal{G} \setminus \{\Phi\}$. If not, the lts can be dismissed immediately. This is done in Algorithm 2.

Algorithm 2 Test for the earliest Parikh cycles with respect to $\mathcal{G}$

```
Input: a finite, totally reachable, deterministic, persistent lts (S, T, →, ı)
  compute a breadth-first spanning tree E ⊆ → (with |E| = |S| − 1)
  compute the set G of Parikh vectors of small cycles
  collect the sets target(Φ) of target states of edges e ∈ → \ E
          with P_E(e) = Φ for each Φ ∈ G
  check the reduced distance property via Algorithm 1
  for each Φ ∈ G:
    if ¬checkCycles(ı, true, Φ) then return false endif
  return true

  procedure checkCycles(s ∈ S, psionly ∈ {true, false}, Φ: T → ℕ): boolean
    if s ∈ target(Φ) then return psionly
    for each (s, t, r) ∈ E:
      if psionly and ∃Θ ∈ G \ {Φ}: Θ(t) > 0 then
        if ¬checkCycles(r, false, Φ) then return false endif
      else if ¬checkCycles(r, psionly, Φ) then return false endif
      endif
    return true
```

Here, the state s is used for the depth-first search through our spanning tree, Φ is the Parikh vector of one small cycle (and remains unchanged during the whole recursion), and *psionly* (representing the current truth value of property $\mathcal{D}$) is true on some initial part of the spanning tree, until we encounter a label from another small cycle. It will revert to false then and remain so in the whole subtree we travel from there. If we collect the target states of edges $e \in \rightarrow \setminus E$ with Parikh vector Φ during the computation of $\mathcal{G}$, the check $s \in target(\Phi)$ can be done in constant time. The test $\exists \Theta \in \mathcal{G} \setminus \{\Phi\} : \Theta(t) > 0$ can also be done in constant time with an appropriate pre-computation: At the time we compute $\mathcal{G}$ we provide a boolean array for each $\Phi \in \mathcal{G}$ where each label t with $\Phi(t) > 0$ is marked true. One further array is constructed, where all labels not occurring in small cycles are marked true. Two $O(1)$ checks, one on the latter array, one for Φ will then tell us if a label occurs anywhere in $\mathcal{G} \setminus \{\Phi\}$. As in the previous algorithm, we have two loops, one over $\mathcal{G}$ and one over E, which lead to a run time of $O(|S| \cdot |T|)$ (starting at the outer for-each-loop, the initial computation of $\mathcal{G}$ remains in $O(|S| \cdot |T|^2)$).

Outsourcing the pre-computations to earlier checks of other (necessary) properties, we can argue that the additional computations for the reduced distance and earliest Parikh cycle properties can be done in $O(|S| \cdot |T|)$ ($O(|S| \cdot |T|^2)$ including pre-computations). Since the pre-synthesis already needs $O(|S| \cdot |T|^2)$ anyway, this is easily affordable. In terms of solving our initial task (Definition 1), if m transition systems $(S_i, T_i, \to_i, \imath_i)$ $(1 \leq i \leq m)$ are given, this adds up to time $O((\sum_{i=1}^{m} |S_i|) \cdot |\bigcup_{i=1}^{m} T_i|^2)$ for the entire pre-synthesis, which compares favourably with synthesis where the size of the state space typically enters in higher powers.

6 Simultaneous Synthesis Algorithm

We now describe a simultaneous choice-free synthesis algorithm. To start with, we emphasise that this additional step is necessary, since simultaneous cf-synthesis may still fail. For example, consider the systems TS_{21} and TS_{22} in Figure 11. They successfully pass all checks based on the disjoint minimal semiflows, reduced distances and early Parikh cycles, and they have individual cf-solutions as illustrated on the bottom row of Figure 11. However, the pair TS_{21}, TS_{22} is not simultaneously cf-solvable. To see why this is true, we give an *ad-hoc* argument. There must be a place, say p, which prevents the firing of a at state s. If μ denotes the marking of p at $\imath_1$ and a and b (temporarily) denote the effect of the firings of a and b on place p, respectively, this means that $\mu + 2a + 2b < 0$, or $-\mu - 2a - 2b > 0$. The semiflow $(3, 2)$ demanded by the second transition system enforces $3a + 2b = 0$ on every place including p. Inserting this in the inequality $-\mu - 2a - 2b > 0$ yields $-\mu + a > 0$. Because p enables the firing sequences a and abb, $\mu + a \geq 0$ as well as $\mu + a + 2b \geq 0$, hence $\mu - 2a \geq 0$ (using again the semiflow $(3, 2)$). Adding the first with $-\mu + a > 0$ yields $2a > 0$, and adding the second with $-\mu + a > 0$ yields $-a > 0$, a contradiction.

In the remainder of this section, suppose that we have, as an input, m labelled transition systems $TS_1, \ldots, TS_m$ which have passed the entire pre-synthesis described in Sections 3 and 4. Also suppose that we have an individual choice-free solution PNS_i for every TS_i. As before, let $\mathcal{G}_i$ be the set of (Parikh vectors of) small cycles in TS_i; let $\mathcal{G} = \bigcup_{1 \leq i \leq m} \mathcal{G}_i$; and let $\mathcal{MSF}_i$ be the set of minimal semiflows in PNS_i.

6.1 Transition Synchronisation

We start with the simplest case, which is when all the Petri net systems $PNS_i = (P_i, T_i, F_i, M_0^i)$ have the same set T of transitions and include the

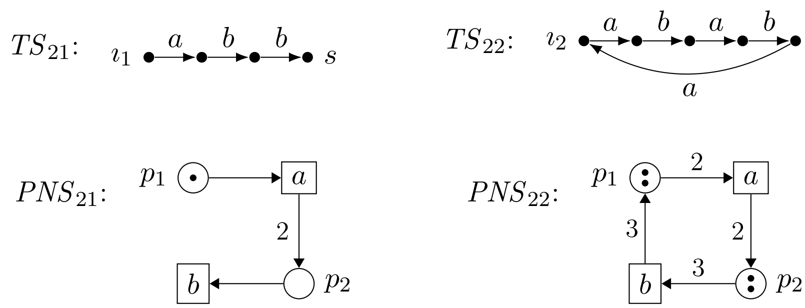

Figure 11: Two transition systems TS_{21}, TS_{22} with individual choice-free solutions PNS_{21}, PNS_{22}.

set $\mathcal{G}$ (the set of Parikh vectors of small cycles in the various TS_j's) in their sets $\mathcal{MSF}_i$ of minimal semiflows, i.e. when

$$\forall i \in \{1, \ldots, m\}\colon\ T_i = T \ \ \wedge\ \ \mathcal{G} \subseteq \mathcal{MSF}_i \tag{1}$$

In this case, the individual Petri nets can be synchronised at their equally-labelled transitions: assuming the place sets P_i are disjoint (which is always possible since isomorphic Petri net systems have isomorphic reachability graphs), this amounts to build the net (P, T, F) such that $P = \bigcup_i P_i$ and $\forall t \in T, p \in P_i : F(p,t) = F_i(p,t) \wedge F(t,p) = F_i(t,p)$. It is easy to see that this preserves choice-freeness and that the set of minimal semiflows is the intersection of the $\mathcal{MSF}_i$'s, hence include $\mathcal{G}$. We now show that the resulting Petri net has m initial markings which solve the m given transition systems. For each $i \in \{1, \ldots, m\}$, we may use for the places p from P_i the initial marking of PNS_i: $M_{0i}(p) = M_0^i(p)$. This allows to exclude the arcs not enabled by TS_i (hence to keep the system bounded), and to keep the arcs enabled by TS_i provided they are not excluded by the other places. It thus remains to show that it is possible to put initially sufficiently many tokens on the places *not* belonging to P_i such that the result does not prevent any of the desired firings. The next proposition guarantees that this can be done.

Proposition 3 Choice of marking
Let $PNS_1, \ldots, PNS_m$ be solutions of $TS_1, \ldots, TS_m$ (respectively), all with the same transition set T, while respecting all the semiflows in $\mathcal{G}$; then in their

synchronisation it is possible to choose a marking in order to generate TS_i, for all $i \in \{1, \ldots, m\}$.

Proof: In the remainder of the proof, we fix some index $i \in \{1, \ldots, m\}$ and let j never be the same index as i, i.e. $j \neq i$.

If we forget the places from all PNS_j, by definition, the places from PNS_i with their initial marking M_{0i} generate a solution to TS_i. If we now add the places of the PNS_j with their connections to / from transitions in T, we have to determine their initial marking so that this does not restrict the firing sequences enabled by TS_i.

By induction on the length of the enabled firing sequences, we must show that for each place $p_j \in P_j$ and for some initial marking $M_{0i}(p_j)$ of it, for each $\imath_i[\sigma\rangle s$ enabled by TS_i and $s[t\rangle$, the marking $M_s(p_j)$ of p_j after executing σ only relies on s and is at least $F_j(p_j, t)$. The property results from the observation that $\mathcal{P}(\sigma) = \Delta_s + \sum_{\Upsilon \in \mathcal{G}_i} k_\Upsilon \cdot \Upsilon$ for some natural integers k_Υ. Hence, since $\mathcal{G}_i \subseteq \mathcal{G} \subseteq \mathcal{MSF}_j$, PNS_j conforms to the semiflows in $\mathcal{G}_i$ and $M_s(p_j) = M_{0i}(p_j) + C_j \cdot (\Delta_s + \sum_{\Upsilon \in \mathcal{G}} k_\Upsilon \cdot \Upsilon) = M_{0i}(p_j) + C_j \cdot \Delta_s$, which leads to finitely many constraints: $\forall s \in S_i, s[t\rangle$: $M_{0i}(p_j) + C_j \cdot \Delta_s \geq F_j(p_j, t)$, and it is always possible to find an adequate $M_{0i}(p_j)$. With the convention that $\max_\emptyset = 0$, we thus simply have the constraint $M_{0i}(p_j) \geq \max_{s \in S_i}(\max_{s[t\rangle} F_j(p_j, t) - C_j \cdot \Delta_s)$ (note that, for $s = \imath_i$, we get $\max_{s[t\rangle} F_j(p_j, t) \geq 0$, so that this formula always yields a non-negative value, as requested for a marking). □

6.2 Homogenisation of Transition Sets and Semiflows

In general, we may not have Property (1) in our individual solutions. The transition sets may be different, and even if they coincide, it may happen that some T-vectors in $\mathcal{G}$ are not semiflows of the considered solution of some TS_i.

As an example, consider the two transition systems TS_{23} and TS_{24} shown in Figure 12 which have individual solutions (thick portions of PNS_{23} and PNS_{24}, respectively) not satisfying (1). We may enforce the equality of the transition sets by adding non-executable transitions where they are missing, thus establishing the first conjunct of (1). This happened in Figure 12 where a b transition with a token-empty input place p_b was added to PNS_{23}. The part of PNS_{23} drawn with solid lines is again a choice-free solution of TS_{23}. However, PNS_{24} has a minimal semiflow, viz. $(1, 1)$, which is missing in PNS_{23} (with only the solid lines). The synthesis algorithm specified next will add such a semiflow if at all possible.

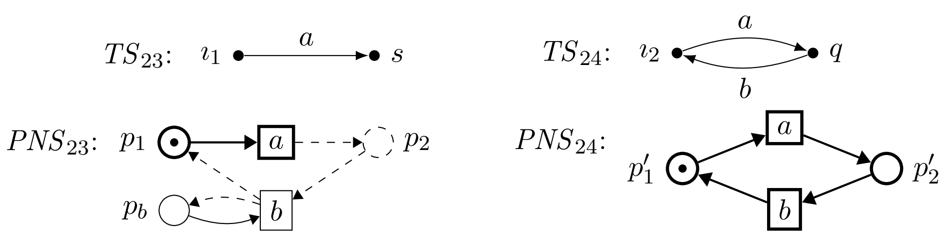

Figure 12: Two transition systems TS_{23} and TS_{24}. Two individual solutions of them are given by the thick parts of PNS_{23} and PNS_{24}. In TS_{23}, p_b and b are added by the homogenisation of transition sets, and the dashed arrows by an invocation of one of the existing synthesis procedures which creates a semiflow (in TS_{23}) that corresponds to the cycle ab (in TS_{24}).

To this end, we may exploit one of the synthesis procedures described in [4, 17] for bounded choice-free net systems. With a view to Theorem 5, all small cycles of TS_i should correspond not just to minimal semiflows of PNS_i, but also to minimal semiflows of PNS_j with $j \neq i$. The algorithms in [4, 17] have as a parameter a set of semiflows that can be enforced on the solution. This means that we may use them in conjunction with the *entire* set of small cycle Parikh vectors $\mathcal{G}$, instead of just with the individual small cycles $\mathcal{G}_i$. If successful, these procedures then impose a semiflow on *every* small cycle in $\mathcal{G}$ in the synthesised solution. This will be detailed in the next section.

For instance, in Figure 12, this leads to the addition of the dashed elements in PNS_{23}, which otherwise would not be necessary, so that the T-vector $(1, 1)$, which is a (minimal) semiflow generated by TS_{24}, is also a semiflow of the considered solution of TS_{23}. In general, the correctness of the underlying choice-free synthesis algorithm ensures the eventual truth of the second conjunct of (1), whenever possible. In Figure 11, this will fail because it is impossible to augment PNS_{21} by a semiflow $(3, 2)$.

Note, however, that (1) must be established in its entirety for the necessary marking distribution according to Proposition 3 to be possible. For example, if we drop the dashed elements in PNS_{23}, then we need an infinite number of initial tokens in p_1 and in p_b in order to allow an unbounded number of repetitions of ab as requested by TS_{24}, which is not allowed. However, synchronising PNS_{23} and PNS_{24} *with* the dashed elements, we get the net and markings exhibited in Figure 13: a single token in p_b is enough to get a solution of TS_{24}, and dropping this token yields a solution to TS_{23}. Note that the obtained solution is not optimal (p_1 and p_1' play the same role, as well as p_2 and p_2'), but this is irrelevant here since we do not aim at minimality, simply at the correctness of the solution.

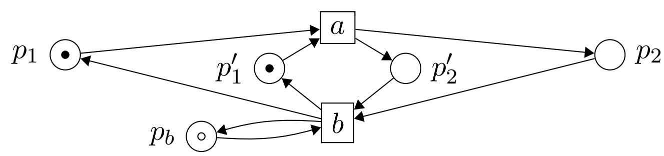

Figure 13: A simultaneous solution of TS_{23} and TS_{24} obtained by synchronising PNS_{23} and PNS_{24} on the transitions. Depending on whether a token is absent or not on p_b, TS_{23} or TS_{24} is solved.

6.3 Extended Individual Synthesis

In this section, we develop the semiflow homogenisation just explained in the previous section, extending (a slightly simplified version of) the algorithm described in [4] (but using [17] works equally well). The main addition is to take care of the minimal semiflows arising from the analysis of all the given transition systems when synthesising any one of them.

Let us thus have a closer look at the way to perform the choice-free synthesis of a finite labelled transition system TS_i ($i = 1, ..., m$) in presence of an extended set of (minimal) semiflow $\mathcal{G}$ (including the Parikh vectors $\mathcal{G}_i$ of its small cycles, but also the ones arising from other TS_j's). In general synthesis, two kinds of separation problems need — and suffice — to be solved, *state separation problems* and *event/state separation problems* [1], but it is known [4] that the state separation problems play no role in bounded choice-free synthesis.

Places of a (choice-free) solution have the general form illustrated in Figure 14, and they must satisfy three conditions:

- the weights must be compatible with the semiflows in $\mathcal{G}$,

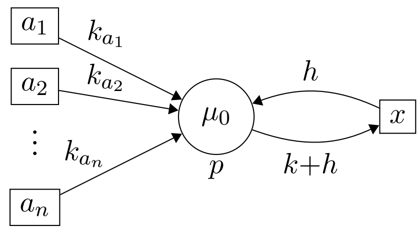

Figure 14: A general pure ($h = 0$) or non-pure ($h > 0$) choice-free place p with initial marking μ_0. Place p has at most one outgoing transition named x. The set $\{a_1, \ldots, a_n\}$ comprises all other transitions, i.e. $T = \{x, a_1, \ldots, a_n\}$, and k_{a_j} denotes the weight of the arc from a_j to p (which could be zero).

- the initial markings must be high enough to allow to reach all the states of the TS_i (each such place, together with its markings corresponding to the various reachable states, then yields a region in the sense of [1]),
- the initial markings should be low enough to allow excluding forbidden transitions in TS_i (this materialises the event/state separation problems).

In the following, we shall fix x, but in the procedure we shall of course consider all the labels in T.

First, the weights must satisfy the following cyclic constraints:

$$\forall \Upsilon \in \mathcal{G} : \sum_{l \in \{1,\ldots,n\}} k_{a_l} \cdot \Upsilon(a_l) = k \cdot \Upsilon(x).$$

As a consequence, if x does not belong to the support of Υ, we may deduce that $k_{a_l} = 0$ for all values of l such that $\Upsilon(a_l) > 0$. This may thus reduce considerably the number of pre-transitions of a place p as in Figure 14, and in the following we shall denote by $A(x)$ the set of those possible pre-transitions (it may happen that more k_{a_l}'s will finally be null, but we shall not capture them now).

Let us denote by $T_\Upsilon = supp(\Upsilon)$ the support of $\Upsilon \in \mathcal{G}$, and by $T_0 = T \setminus \bigcup_{\Upsilon \in \mathcal{G}} T_\Upsilon$ the transitions that do not belong to the support of some semiflow in $\mathcal{G}$. There are then two different cases:

1. if $x \in T_\Upsilon$ belongs to the support of some (unique[5]) Υ in $\mathcal{G}$, $A(x) = (T_\Upsilon \setminus \{x\}) \cup T_0$, and

$$k = \sum_{a \in T_\Upsilon \setminus \{x\}} k_a \cdot \Upsilon(a)/\Upsilon(x)$$

2. if $x \in T_0$, $A(x) = T_0 \setminus \{x\}$ and there is no special relation fixing k with respect to the other weights.

As a consequence, if $s[y\rangle r$ with $s, r \in S_i$ and $y \neq x$, if we represent by M_s the marking reached after following any sequence leading to s from the initial state (the result does not rely on the specific sequence which is followed), we have $M_s(p) \leq M_r(p)$, and if $y \notin A(x)$ we have $M_s(p) = M_r(p)$. We shall thus define on S_i the partial order $s \leq_x r$ iff $s[\alpha\rangle r$ with $\alpha \in (T \setminus \{x\})^*$, $s \sim_x r$ iff $s[y\rangle r$ or $r[y\rangle s$ with $y \notin A(x)$, $\equiv_x = (\sim_x)^*$ is the equivalence relation $s \equiv_x r$ generated by $\sim_x$, and $<_x = \leq_x \setminus \equiv_x$. We then have $s <_x r \Rightarrow M_s(p) \leq M_r(p)$ and $s \equiv_x r \Rightarrow M_s(p) = M_r(p)$.

As to the various reachable markings of such a place p, we have that, for each state $s \in S_i$, the marking of that place corresponding to s is

$$M_s(p) = \mu_0 + \sum_{a \in A(x)} k_a \cdot \Delta_s(a) - k \cdot \Delta_s(x) \geq 0.$$

In order to allow executing x whenever needed, this leads to the constraint that

$$\forall s \in S_i : s[x\rangle \Rightarrow M_s(p) \geq k + h. \tag{2}$$

However, as mentioned in [4], it is not necessary to check all the constraints (2) to get an initial marking μ_0 high enough. Let us consider the following cases:

- if x does not occur in TS_i, then the constraint is empty;
- if x occurs in TS_i and $\mathcal{P}(x) \in \mathcal{G}$, we also have $\mathcal{P}(x) \in \mathcal{G}_i$ and $s[x\rangle \Rightarrow s[x\rangle s$. Then also $k = 0$ and $s[y\rangle r \Rightarrow M_s(p) \leq M_r(p) \Rightarrow r[x\rangle r$ (which is compatible with the well known property that one may push forward Parikh-equivalently any cycle in a finite, deterministic and persistent lts [2]). We thus only have to check the *initial x-loops*. If $\imath_i[x\rangle\imath_i$, then

[5] since the various Υ's are disjoint.

we simply have to make x isolated, and there is no constraint to check. Otherwise, let $IL_i(x) = \{s \in S_i \mid s[x\rangle \wedge \forall r \in S_i : r <_x s \Rightarrow \neg r[x\rangle\}$, and $mX_i(x)$ be the subset of $IL_i(x)$ where one only keeps a single representative of the equivalence classes of $\equiv_x$. The constraints (2) then reduce to

$$\forall s \in mX_i(x) : M_s(p) \geq h; \tag{3}$$

- otherwise, since $\mathcal{P}(x) \notin \mathcal{G}_i$, if $s[x^l\rangle r$ with $l > 0$, then we must have $s \neq r$. Moreover, $M_s(p) \geq l \cdot k + h$, or equivalently, $M_r(p) \geq h$. For this reason, when considering the marking of p at a state s with $s[x\rangle$, the idea is to follow x-chains in forward direction as long as possible: from determinism and $\mathcal{P}(x) \notin \mathcal{G}_i$, there is a unique last state. Thus, we are interested in the following subsets of states of S_i:

  $$\begin{aligned} XNX_i(x) &= \{r \in S_i \mid [x\rangle r \wedge \neg r[x\rangle\} \\ mXNX_i(x) &= \{r \in XNX_i(x) \mid \nexists s \in XNX_i(x) \wedge s <_x r\} \end{aligned}$$

  which are produced by x but do not enable x, and in the subset $mX_i(x)$ of $mXNX_i(x)$ where one only keeps a single representative of the equivalence classes of $\equiv_x$. The above considerations amount to a proof of the fact that we may again replace the constraints (2) by the constraints (3).

Similarly, in order to exclude forbidden transitions, it is not necessary to construct a place p such that $M_s(p) < k + h$ for each state s such that $\neg s[x\rangle$. Indeed, if $M_s(p) < k+h$, the same place p will exclude $r[x\rangle$ whenever $r \equiv_x s$ since then $M_s(p) = M_r(p)$, and more generally whenever $r \leq_x s$ since then $M_r(p) \leq M_s(p)$. This leads to consider the set of states:

$$NXX_i(x) = \{s \in S_i \mid \neg s[x\rangle \text{ and } \forall r \in S_i : s <_x r \Rightarrow r[x\rangle\}$$

i.e. the set of states not enabling x such that it is not possible to find a potentially "better" one from it, and $mNXX_i(x)$ as the subset of $NXX_i(x)$ where one only keeps a single representative of the equivalence classes of $\equiv_x$.

Hence, extending the analysis in [4], in order to check if there is a choice-free solution of TS_i compatible with the set of semiflows $\mathcal{G}$, and to build it, for each $x \in T$ and $s \in mNXX_i(x)$, we need to solve the system

$$\boxed{\begin{array}{l} \forall r{\in}mX_i(x){:}\ 0 < k \cdot \left[\ 1 + \Delta_s(x) - \Delta_r(x)\ \right] + \sum\limits_{a\in A(x)} k_a \cdot \left[\ \Delta_r(a) - \Delta_s(a)\ \right] \\ \text{if } x \in T_\Upsilon : \sum_{a\in A(x)} k_a \cdot \Upsilon(a) = k \cdot \Upsilon(x) \end{array}}$$

If this system is solvable in the domain of natural numbers (with $k_a = 0$ if $a \notin A(x)$), let us define $\mu = \max\{k \cdot \Delta_r(x) - \sum_{a \in A(x)} k_a \cdot \Delta_r(a) \mid r \in mXNX_i(x)\}$. If $\mu \geq 0$, by choosing $h = 0$ and $M_0^i = \mu$ we shall get a place satisfying all the needed conditions. If $\mu < 0$, it is not possible to create a suitable pure place from this solution, but we may choose $h = -\mu$ and $M_0^i = 0$ and we shall again get an adequate place.

If x occurs in some $\Upsilon \in \mathcal{G}$, it is possible to eliminate k from the unknowns of the above system, and save an equation, but we shall not do it here.

For instance, for systems TS_1 and TS_2 of Figure 1, for which $\mathcal{G} = \{\mathcal{P}(b), \mathcal{P}(c)\}$ and $T_0 = \{a, d\}$ is the set of non-live transitions, when $x = d$, we get three systems of one inequation each. With $k_a = F(a, p)$ and $k = F(p, d)$, these systems are

$$\begin{array}{ll} 0 < k \cdot [1 + 0 - 1] + k_a \cdot [1 - 0] & (\text{for } \imath_1 \in mNXX_1(d),\ s \in mX_1(d)) \\ 0 < k \cdot [1 + 1 - 1] + k_a \cdot [1 - 1] & (\text{for } s \in mNXX_1(d),\ s \in mX_1(d)) \\ 0 < k \cdot [1 + 2 - 2] + k_a \cdot [0 - 0] & (\text{for } s' \in mNXX_2(d),\ s' \in mX_2(d)) \end{array}$$

There is a way to solve all of them with a single place p_3 (as in Figure 1): $F(a, p_3) = F(p_3, d) = 1$. With these values, the minimal (and adequate) initial marking of this place has no token for the solution of TS_1 and 2 tokens for the solution of TS_2.

7 Concluding Remarks

We have described (the theory behind) an algorithm solving the simultaneous synthesis problem for choice-free Petri nets – in fact: we specified several such algorithms, since pre-synthesis can be curtailed if desired, albeit at the peril of extending proper synthesis, which may give rise to more costly procedures and less informative results. We also enhanced the pre-synthesis phase of an individual choice-free synthesis problem with new checks (namely, reduced distances and earliest Parikh cycles) that were not used in previous work. Hence, in this paper, about everything from section 4 is new, also when we only consider a single lts, but section 6.3 is mildly new.

Many directions are open for future research: to consider special cases (e.g., acyclic transition systems, marked graphs); or to generalise the setup (e.g., to several labelled transition systems with a single initial state each, but with possibly some common states – see also [8]); or to switch the focus to other classes of nets, or systems. We also plan to combine our synthesis procedure with a factorisation technique [7, 8, 9] in order to reduce the complexity by a divide and conquer strategy.

It also seems possible to capture a range of examples similar to the one depicted in Figure 11 during pre-synthesis, by generalising Theorem 3. The authors are engaged in this extension and have obtained partial results. We are also interested in incorporating the novel algorithms described in this paper in our tool APT [15].

Acknowledgements. The authors would like to thank the reviewers for helpful comments.

References

[1] É. Badouel, L. Bernardinello, P. Darondeau: *Petri Net Synthesis*. Texts in Theoretical Computer Science, Springer-Verlag, ISBN 978-3-662-47967-4, 339 pages (2015). `doi:10.1007/978-3-662-47967-4`.

[2] E. Best, P. Darondeau: A Decomposition Theorem for Finite Persistent Transition Systems. Acta Informatica 46: 237–254 (2009). `doi:10.1007/s00236-009-0095-6`.

[3] E. Best, R. Devillers: Petri Net Pre-Synthesis Based on Prime Cycles and Distance Paths. Science of Computer Programming 157 (2018), 41–55. `doi:10.1016/j.scico.2017.07.005`.

[4] E. Best, R. Devillers, U. Schlachter: Bounded Choice-Free Petri Net Synthesis: Algorithmic Issues. Acta Informatica (2017). `doi:10.1007/s00236-017-0310-9`.

[5] S. Crespi-Reghizzi, D. Mandrioli. A Decidability Theorem for a Class of Vector-Addition Systems. Inf. Process. Lett., 3(3):78–80, 1975. `doi:10.1016/0020-0190(75)90020-4`.

[6] J. Desel, J. Esparza: *Free Choice Petri Nets*. Cambridge Tracts in Theoretical Computer Science 40, 242 pages (1995). `doi:10.1017/CBO9780511526558`.

[7] R. Devillers: Products of Transition Systems and Additions of Petri Nets. Proceedings of the16th International Conference on Application

of Concurrency to System Design, ACSD, 65–73 (2016). doi:10.1109/acsd.2016.10.

[8] R. Devillers: Factorisation of Transition Systems. Acta Informatica (2017). doi:10.1007/s00236-017-0300-y.

[9] R. Devillers, U. Schlachter: Factorisation of Petri Net Solvable Transition Systems. Proceedings of the 39th International Conference on Applications and Theory of Petri Nets and Concurrency (ICATPN 18), LNCS 10877, 82–98 (2018). doi:10.1007/978-3-319-91268-4_5.

[10] A. Ehrenfeucht, G. Rozenberg: Partial 2-Structures, Part I: Basic Notions and the Representation Problem, and Part II: State Spaces of Concurrent Systems. Acta Informatica, Vol. 27(4), 315–368 (1990). doi:10.1007/BF00264611 and doi:10.1007/BF00264612.

[11] R.M. Keller: A Fundamental Theorem of Asynchronous Parallel Computation. Parallel Processing, LNCS Vol. 24, Springer-Verlag, 102–112 (1975). doi:10.1007/3-540-07135-0_113.

[12] A. Kondratyev, J. Cortadella, M. Kishinevsky, E. Pastor, O. Roig, A. Yakovlev: Checking Signal Transition Graph Implementability by Symbolic BDD Traversal. Proc. European Design and Test Conference, 325–332, Paris, France (1995). doi:10.1109/EDTC.1995.470376.

[13] L.H. Landweber, E.L. Robertson: Properties of Conflict-Free and Persistent Petri Nets. JACM 25(3), 352–364 (1978). doi:10.1145/322077.322079.

[14] W. Reisig: *Petri Nets*. EATCS Monographs on Theoretical Computer Science 4, Springer-Verlag (1985). doi:10.1007/978-3-642-69968-9.

[15] U. Schlachter et al.: https://github.com/CvO-Theory/apt (2013–2018).

[16] E. Teruel, J.M. Colom, M. 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 (1997), 73-83. doi:10.1109/3468.553226.

[17] H. Wimmel: Presynthesis of Bounded Choice-Free or Fork-Attribution Nets. Technical Report No. 01/18 in: Berichte aus dem Department

für Informatik, University of Oldenburg, Germany, ISSN 1867-9218, 29 pages (2018).