Scientific Annals of Computer Science vol. 27 (2), 2017, pp. 177–212
doi: 10.7561/SACS.2017.2.177

Adding A/Sync Places to the Synthesis Procedure for Whole-Place Operations Nets with Localities

Jetty KLEIJN[1], Maciej KOUTNY[2],
Marta PIETKIEWICZ-KOUTNY[2]

Abstract

Assuming that the behavioural specification of a concurrent system is given in the form of a step transition system, where the arcs between states are labelled by steps (multisets of executed actions), we focus on the problem of synthesising a Petri net generating a reachability graph isomorphic to a given step transition system. To deal with step transition systems more complicated than those generated by standard Place/Transition nets, we consider in this paper Petri nets with whole-place operations, localities, and a/sync places. We adapt and extend the general approach developed within the framework of τ-nets and the theory of regions of step transition systems. Building on the results presented in [23], emphasis here is on the role of a/sync places with their potential for an instantaneous transfer of tokens within a step. In a series of results we demonstrate the robustness of the notion of region for Petri net synthesis.

Keywords: concurrency, theory of regions, transition system, synthesis problem, Petri net, step semantics, locality, whole-place operations, synchronous and asynchronous communication, a/sync places

[1] Leiden Institute of Advanced Computer Science, Leiden University, PO Box 9512, 2300 RA, The Netherlands, E-mail: h.c.m.kleijn@liacs.leidenuniv.nl

[2] School of Computing, Newcastle University, Newcastle upon Tyne, NE4 5TG United Kingdom, E-mail: {maciej.koutny,marta.koutny}@ncl.ac.uk

1 Introduction

Synthesising systems from behavioural specifications is an attractive way of constructing implementations which are correct-by-design and thus requiring no costly validation efforts. In this paper, we assume that the behaviour of concurrent systems is given in the form of a (step) transition system where the arcs are labelled by multisets of executed actions. Systems are modelled by Petri nets. We are, however, interested in transition systems that may reflect behaviour beyond that which could be expressed by standard Place/Transition nets (PT-nets).

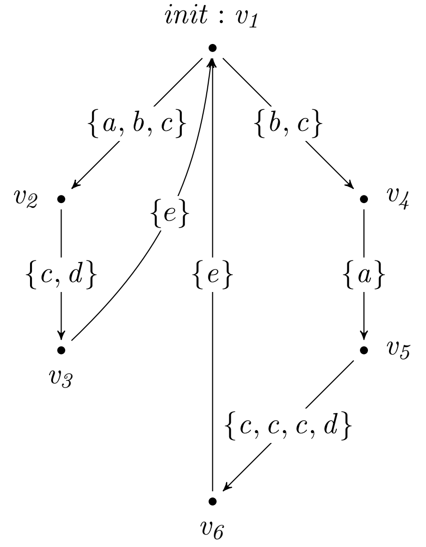


Figure 1: A step transition system.

The transition systems generated by PT-nets satisfy two crucial properties that might be expected to hold in many modelled systems:

1. *Backward determinism*, demanding that two arcs labelled by the same step and incoming to a given state start at the same state.

2. *Subset closure* for steps enabled at a given state, demanding that for a step enabled at a state all its subsets are enabled at this state.

Considering the above properties, we conclude that the step transition system depicted in Figure 1 cannot be generated by any PT-net. Firstly, there are two incoming arcs labelled by $\{e\}$ meeting at the state v_1, but coming from two different states, v_3 and v_6, violating property (1). Secondly, at the state v_1 we can see step $\{a, b, c\}$ being enabled without many of its subsets being enabled at v_1, violating property (2).

A Petri net model that could deliver a suitable formal model for behavioural descriptions like that in Figure 1, would incorporate two features to deal with the constraints imposed by PT-nets mentioned above. The first added feature that is not present in PT-nets are arcs labelled by linear expressions involving places making their weights to depend on the current marking of places, and in consequence introducing the so called *whole-place operations* (see also [16, 18, 1, 23]). This new feature will address the problem concerning the inability to model backward non-deterministic behaviour within PT-nets. The second added feature are *localities* for actions (net-transitions). All the net-transitions will be assigned their localities and by that will be divided into groups of transitions that share a locality. A new semantics will govern the executions of steps in the nets with localities: only the maximal enabled multisets of net-transitions will be allowed to be executed within a given locality. This feature will address the problem concerning the lack of subset closure for steps enabled at a given state. However, we need yet another net feature to be able to generate the behaviour depicted in Figure 1. Even having localities at our disposal, we cannot envisage the right locality mapping that would allow the situation, where two steps like $\{a, b, c\}$ and $\{b, c\}$ are enabled at state v_1, as shown in Figure 1, without also step $\{a\}$ being enabled at v_1. This problem can be combated by introducing *a/sync places* that facilitate the instantaneous transfer of tokens between net-transitions involved in the same step. This last feature, the introduction of a/sync places, will add extra expressive power to the class of Petri nets that we considered in [23] in the context of synthesis problem, and is the major contribution of this paper.

Originally, a/sync places were introduced in [20], in a more restricted setting as channel places between component nets. Allowing tokens to be simultaneously produced and consumed (synchronously) provides additional modelling power and behavioural expressiveness. As, e.g., noted in [11], synchronous communication is not a primitive concept for standard Petri nets. The concept of a/sync places provides a succinct abstraction. Such places can be seen as playing a role similar to zero places [7], supporting modular translations of concurrent languages. In particular, a/sync places can model transactions involving several individual transitions through single step executions. Potential applications of a/sync places also include channel-based models for exogenous coordination of (software) components (see, e.g., [2]). The idea also proved to be useful in the context of the biologically inspired tissue systems where the a/sync transfer of tokens motivated a mechanism for the modelling of instantaneous or fast passing of signals within the system. In [22], the idea was further generalised by introducing a/sync connections between places and transitions rather than dedicated a/sync places.

In [21], a/sync places as we consider them in this paper, were used for the synchronous transfer of tokens from an input transition to an output transition. These places moreover allow asynchronous communication, because tokens that are not consumed instantaneously, remain available as ordinary tokens.

A class of nets with whole-place operations, a/sync places, and localities that we will focus on in this paper will be called WPOASL-nets.

The synthesis of WPOASL-nets from step transition system specifications will build on our previous work [24, 23]. It will be based on a suitably adapted notion of *region* of a (step) transition system [17, 4, 3], as well as their locally maximal execution semantics, a special kind of *step firing policy* (see [25, 14]). Regions were introduced in the seminal paper [17] for the class of Elementary Net Systems (EN-systems) with sequential execution semantics. After that, the original idea has been developed (see, for example, [27]) and extended in several different directions, including: other Petri net classes (e.g., bounded PT-nets without loops [6], PT-nets [26], Flip-flop nets [29], nets with inhibitor arcs [8, 28], and nets with localities [25]); synthesis modules of implemented tool frameworks (e.g., Petrify [12], ProM [31], VipTool [5], Genet [9], and Rbminer [30]); application areas (e.g., asynchronous VLSI circuits [12, 9, 30] and workflows [31]); other semantical execution models (e.g., step sequences [19, 28], (local) maximal concurrency [25], and firing policies [14]); and specification formalisms other than transition systems (e.g., languages [13] and scenarios [5]). More details concerning the importance and long term impact of the region concept can be found in the monograph [3].

One of the key advances in the design of region based solutions for a variety of synthesis problems has been the development of a general approach [4]. It is founded on so-called τ-nets and corresponding τ-regions. The parameter τ (called *net-type*) is a convenient way of capturing the marking information and different connections between places and transitions of different classes of Petri nets, removing the need to re-state and re-prove the main results every time a new kind of arcs is introduced. This approach can be applied once a class of Petri nets has been shown to be a class of τ-nets, i.e., to correspond to a class of τ-nets for some suitable τ. It should be kept in mind however, that although the theory provides necessary and sufficient conditions for the existence of a τ-net whose reachability graph is isomorphic to a given transition system, it does not provide ready answers for decidability and algorithmic concerns.

The paper is organised as follows. The next section recalls some basic notions concerning transition systems, PT-nets, and τ-nets. Section 3 introduces WPOASL-nets and discusses their expressive power. Section 4 presents a solution to the synthesis problems for WPOASL-nets, treating them as a special kind of τ-nets. In Section 5, we discuss a procedure for the synthesis of a WPOASL-net when the whole-places (the places that influence arc weights) are known. The paper ends with a conclusion that includes some directions for future work.

This paper is an extended version of the conference paper [23]. The main difference is that the latter did not consider a/sync places. By generalising the concept of region to cater for the instantaneous transfer of tokens and providing explicit proofs for the thus generalised setting, we demonstrate the robustness of the notion of region for Petri net synthesis.

2 Preliminaries

An *abelian monoid* is a set $\mathbb{S}$ with a commutative and associative binary operation $+$, and an identity element $\mathbf{0}$. The result of composing n copies of $s \in \mathbb{S}$ is denoted by $n \cdot s$, and so $\mathbf{0} = 0 \cdot s$. In this paper, the abelian monoid $\mathbb{S}_{PT} = \mathbb{N} \times \mathbb{N}$, will represent the (weighted) arcs between places and transitions in PT-nets; here $\mathbb{N}$ denotes all non-negative integers, and $\mathbb{S}_{PT}$ has pointwise arithmetic addition as its operation with $(0, 0)$ as identity element. Furthermore, the free abelian monoid $\langle T \rangle$ generated by a set T can be seen as the set of all finite multisets over T. Thus, e.g., $abb = bab = bba$ represents the multiset $\{a, b, b\}$. In particular, $\langle T \rangle$ represents the *steps* (multisets of transitions) of nets with transition set T. We use $\alpha, \beta, \gamma, \ldots$ to range over the elements of $\langle T \rangle$. For $t \in T$ and $\alpha \in \langle T \rangle$, $\alpha(t)$ denotes the multiplicity of t in α, and so we can represent α as $\alpha = \sum_{t \in T} \alpha(t) \cdot t$. Then $t \in \alpha$ whenever $\alpha(t) > 0$, and $\alpha \leq \beta$ whenever $\alpha(t) \leq \beta(t)$ for all $t \in T$. Moreover, $\alpha < \beta$ if $\alpha \leq \beta$ and $\alpha \neq \beta$. The size of α is given by $|\alpha| = \sum_{t \in T} \alpha(t)$.
In our example $\alpha = \{a, b, b\}$, we could write $\alpha(a) = 1$, $\alpha(b) = 2$, and $\alpha(c) = 0$

for all other $c \in T$; thus $\alpha = a + 2 \cdot b$ and $\alpha < 2 \cdot a + 2 \cdot b = \{a, a, b, b\}$.

Transition systems. A *(deterministic) transition system* $\langle Q, \mathbb{S}, \delta \rangle$ over an abelian monoid $\mathbb{S}$ consists of a set of *states* Q and a partial function $\delta : Q \times \mathbb{S} \to Q$ such that $\delta(q, \mathbf{0}) = q$ for all $q \in Q$. An *initialised* transition system $\langle Q, \mathbb{S}, \delta, q_0 \rangle$ is a transition system with an *initial* state $q_0 \in Q$ such that each state $q \in Q$ is *reachable*, i.e., there are $s_1, \ldots, s_n$ and $q_1, \ldots, q_n = q$ $(n \geq 0)$ with $\delta(q_{i-1}, s_i) = q_i$, for $1 \leq i \leq n$. For every state q of a transition system TS, we denote by $enb_{TS}(q)$ the set of all s which are *enabled* at q, i.e., $\delta(q, s)$ is defined. TS is *bounded* if $enb_{TS}(q)$ is finite for every state q of TS. Moreover, such a TS is *finite* if it has finitely many states.
In diagrams, $\mathbf{0}$-labelled arcs are omitted and singleton steps are written without brackets.

Initialised transition systems $\mathcal{T}$ over free abelian monoids — called *step transition systems* or *concurrent reachability graphs* — represent behaviours of Petri nets. *Net-types* are non-initialised transition systems τ over abelian monoids used to define various classes of nets.

Let $\mathcal{T} = \langle Q, \langle T \rangle, \delta, q_0 \rangle$ and $\mathcal{T}' = \langle Q', \langle T \rangle, \delta', q_0' \rangle$ be step transition systems. $\mathcal{T}$ and $\mathcal{T}'$ are *isomorphic*, $\mathcal{T} \cong \mathcal{T}'$, if there is a bijection $f : Q \to Q'$ with $f(q_0) = q_0'$ and

$$\delta(q, \alpha) = q' \Leftrightarrow \delta'(f(q), \alpha) = f(q'), \textit{ for all } q, q' \in Q \textit{ and } \alpha \in \langle T \rangle\ .$$

Place/Transition nets. A *Place/Transition net* (PT-net, for short) is a tuple $N = \langle P, T, W, M_0 \rangle$, where P and T are disjoint sets of *places* and *transitions*, $W : (P \times T) \cup (T \times P) \to \mathbb{N}$ is a *weight function*, and M_0 is an *initial marking* belonging to the set of *markings* defined as mappings from P to $\mathbb{N}$, thus assigning a number of *tokens* to each place. N is *finite* if both P and T are finite. We use the standard conventions concerning the graphical representation of PT-nets, as illustrated in Figure 2(a).

The weight function is extended to steps as follows. For all $p \in P$ and $\alpha \in \langle T \rangle$, we let

$$W(p, \alpha) = \sum_{t \in T} \alpha(t) \cdot W(p, t) \quad \textit{and} \quad W(\alpha, p) = \sum_{t \in T} \alpha(t) \cdot W(t, p)\ .$$

Then a *step* $\alpha \in \langle T \rangle$ is *enabled* and may be *fired* at a marking M if, for every $p \in P$, $M(p) \geq W(p, \alpha)$. We denote this by $\alpha \in enb_N(M)$. *Firing* such a step leads to the marking M', for every $p \in P$ defined by

$$M'(p) = M(p) - W(p, \alpha) + W(\alpha, p)\ .$$

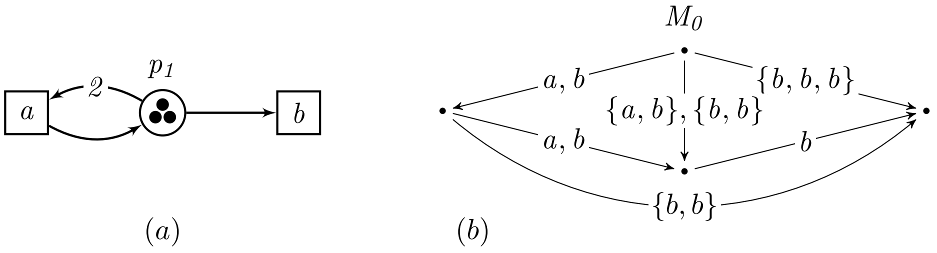


Figure 2: A PT-net (a); and its concurrent reachability graph (b).

We denote this by $M[\alpha\rangle M'$. The *concurrent reachability graph* $CRG(N)$ of N is the step transition system formed by firing inductively from M_0 all possible enabled steps, i.e., $CRG(N) = \langle [M_0\rangle, \langle T\rangle, \delta, M_0\rangle$ where

$$[M_0\rangle = \{M_n \mid \exists\alpha_1, \dots, \alpha_n \ \exists M_1, \dots M_{n-1} \ \forall 1 \leq i \leq n \ : \ M_{i-1}[\alpha_i\rangle M_i\}$$

is the set of *reachable* markings and $\delta(M, \alpha) = M'$ *iff* $M[\alpha\rangle M'$. Figure 2(b) shows the concurrent reachability graph of the PT-net in Figure 2(a).

Maximal concurrency. One of the issues dealt with in this paper is maximal concurrency in the executions of Petri nets. To capture this idea formally, one needs to decide, in particular, which of the steps enabled at a marking M of a PT-net N are maximal. There is no problem to do so if the set $enb_N(M)$ is finite as one can declare all $\leq$-maximal steps in $enb_N(M)$ (i.e., no further occurrences of transitions can be added without destroying the enabledness of the step) as being maximally concurrent. The situation is more complicated if $enb_N(M)$ is infinite. For example, suppose that $enb_N(M) = \{\alpha_1, \alpha_2, \dots\}$ and $\alpha_1 < \alpha_2 < \dots$ holds. In this case, the intuitive sense of being a maximally concurrent step is lost as none of the α_i's is $\leq$-maximal in $enb_N(M)$.

The situation just described can arise in PT-nets, for example when there exist transition(s) without any non-zero input arcs. This is easily excluded by assuming that N is *restricted* (i.e., for every transition t, there is place p such that $W(p, t) > 0$). However, this situation can also arise in a net with an infinite set of transitions T. This again can be excluded by assuming that T is finite. There can, however, also be other reasons for this unboundedness in the classes of nets considered in this paper. Therefore, in our treatment of maximal concurrency, we follow a slightly different approach by assuming that *all* steps in $enb_N(M)$ are maximally concurrent if $enb_N(M)$ is infinite. This approach does not diminish the generality of our results since the synthesis problem to be considered takes as inputs bounded transition systems, and the nets being synthesised will never enable infinitely many steps at a single marking.

Petri nets defined by net-types. A net-type $\tau = \langle \mathcal{Q}, \mathbb{S}, \Delta \rangle$ is a non-initialised transition system over an abelian monoid $\mathbb{S}$. It specifies the values that can be stored in places ($\mathcal{Q}$); the operations and tests that a net-transition may perform on these values ($\mathbb{S}$); and the enabling condition and the newly generated values for steps of transitions (Δ). It is a parameter in the definition of τ-nets: A *τ-net* is a tuple $N = \langle P, T, F, M_0 \rangle$, where P and T are disjoint sets of places and transitions respectively; $F : (P \times T) \to \mathbb{S}$ is a *flow mapping*; and M_0 is an *initial marking* belonging to the set of *markings*, i.e., mappings from P to $\mathcal{Q}$. N is *finite* if both P and T are finite. For all $p \in P$ and $\alpha \in \langle T \rangle$, we set

$$F(p, \alpha) = \sum_{t \in T} \alpha(t) \cdot F(p, t) \ .$$

Thus $F(p, \alpha)$ specifies the combined flow relation of the transition occurrences forming step α with place p. Step $\alpha \in \langle T \rangle$ is *enabled* at a marking M if, for every $p \in P$, $F(p, \alpha) \in enb_\tau(M(p))$. We denote this by $\alpha \in enb_N(M)$. *Firing* such a step produces the marking M', for every $p \in P$ defined by

$$M'(p) = \Delta(M(p), F(p, \alpha)) \ .$$

We denote this by $M[\alpha\rangle M'$, and then define the *concurrent reachability graph* $CRG(N)$ of N as the step transition system formed by firing inductively from M_0 all possible enabled steps.

As demonstrated in [4, 14], it is possible to encode any PT-net $N = \langle P, T, W, M_0 \rangle$ as a τ-net without affecting its concurrent reachability graph. It is enough to take $F(p, t) = (W(p, t), W(t, p))$. Thus $F(p, t) = (i, o)$ means that i is the weight of the arc from p to t, and o the weight of the arc in the opposite direction. With this encoding, N becomes a τ_{PT}-net where $\tau_{PT} = \langle \mathbb{N}, \mathbb{S}_{PT}, \Delta_{PT} \rangle$ is the infinite net-type over $\mathbb{S}_{PT} = \mathbb{N} \times \mathbb{N}$ as introduced at the beginning of Section 2, with Δ_{PT} given by

$$\Delta_{PT}(n, (i, o)) = n - i + o$$

provided that $n \geq i$ (see Figure 6(a)).

3 Whole-place Operations and a/sync Places

Assuming an ordering of places, markings can be represented as vectors. The i-th component of a vector $\mathbf{x}$ is denoted by $\mathbf{x}^{(i)}$. For $\mathbf{x} = (x_1, \ldots, x_n)$ and $\mathbf{y} = (y_1, \ldots, y_n)$, $(\mathbf{x}, 1) = (x_1, \ldots, x_n, 1)$ and $\mathbf{x} \otimes \mathbf{y} = x_1 \cdot y_1 + \cdots + x_n \cdot y_n$. Moreover, $\otimes$ will also denote the multiplication of two-dimensional arrays.

We now introduce a class of Petri nets that extends the class of PT-nets by allowing the arcs in nets to be annotated with linear expressions involving places. Thus the weights of arcs will be represented by vectors with $n+1$ entries, where n is the (total) number of places. The entries of such vectors are coefficients used in the annotating expressions. Moreover, we will allow for a/sync places in addition to the standard places.

A *net with whole-place operations and a/sync places* (WPOAS-net) is a tuple

$$N = \langle P', P'', T, W, \mathbf{m}_0 \rangle \ , \tag{1}$$

where:

- $P' = \{p_1, \ldots, p_{n'}\}$ $(n' \geq 0)$ is a finite set of ordered *a/sync places* and $P'' = \{p_{n'+1}, \ldots, p_n\}$ $(n \geq 1)$ is a finite set of ordered *standard places*;
- T is a finite set of *transitions* disjoint with the ordered set of *places* $P = P' \cup P'' = \{p_1, \ldots, p_n\}$;
- $W : (P \times T) \cup (T \times P) \to \mathbb{N}^{n+1}$ is a *whole-place* weight function; and
- $\mathbf{m}_0$ is an *initial marking* belonging to the set $\mathbb{N}^n$ of *markings*.

Similarly as for PT-nets, we can extend the whole-place weight function to steps. Thus we denote, for all places $p \in P$ and steps of transitions $\alpha \in \langle T \rangle$,

$$W(p, \alpha) = \sum_{t \in T} \alpha(t) \cdot W(p, t) \quad \textit{and} \quad W(\alpha, p) = \sum_{t \in T} \alpha(t) \cdot W(t, p) \ .$$

As for PT-nets, a step can occur at a marking if every input place to the step (its transitions), has enough tokens assigned to it. How many tokens are needed is determined by the arc weights. The extended whole-place weight function relating places and steps defines the dependency of the arc weights on the current number of tokens in each of the n places (weighted according to the function's values first n entries). In case, the first n entries are 0, the weight is invariant and fully specified by the $(n+1)$-st entry.

For the a/sync places, the meaning of token flow is different from that of the token flow through standard places. Whereas in the latter case, all tokens to be consumed by a step from a place must be already present in the place, in the former case, some or all of those tokens can be produced by the firing of the transitions in the step.
Formalising the above, step α is *enabled* at marking $\mathbf{m}$ if, for all $p' \in P'$ and $p'' \in P''$,

$$\begin{array}{rcl} \mathbf{m}(p') & \geq & (\mathbf{m},1) \otimes (W(p',\alpha) - W(\alpha,p')) \\ \mathbf{m}(p'') & \geq & (\mathbf{m},1) \otimes W(p'',\alpha) \ . \end{array} \tag{2}$$

We denote this by $\alpha \in enb_N(\mathbf{m})$. An enabled α can be *fired* leading to a new marking such that, for every $p \in P$,

$$\mathbf{m}'(p) = \mathbf{m}(p) + (\mathbf{m},1) \otimes (W(\alpha,p) - W(p,\alpha)) \ . \tag{3}$$

We denote this by $\mathbf{m}[\alpha\rangle\mathbf{m}'$, and define the *concurrent reachability graph* $\mathit{CRG}(N)$ of N as one built by firing inductively from $\mathbf{m}_0$ all possible enabled steps.

In net diagrams, an a/sync place is depicted as a circle with double border.

As an example consider the WPOAS-net in Figure 3. Since the weights of the arcs are all invariant vectors $(0,0,0,0,1)$, the whole-place weight function is trivial and its values are not indicated. Moreover, the arcs with a whole-place weight vector with only 0 as entries are not drawn. The places p_1 and p_2 are the a/sync places of this net. Step $\alpha = \{a,b,c\}$ is enabled at the given marking $\mathbf{m}_0$ because, with $(\mathbf{m}_0,1) = (1,1,1,1,1)$, $W(p_1,\alpha) = W(p_2,\alpha) = (0,0,0,0,2)$ and $W(\alpha,p_1) = W(\alpha,p_2) = (0,0,0,0,1)$, we have:

$$\begin{array}{rcl} \mathbf{m}_0(p_1) = 1 & \geq & 1 = (\mathbf{m}_0,1) \otimes (W(p_1,\alpha) - W(\alpha,p_1)) \\ \mathbf{m}_0(p_2) = 1 & \geq & 1 = (\mathbf{m}_0,1) \otimes (W(p_2,\alpha) - W(\alpha,p_2)) \ . \end{array}$$

It is convenient to specify the values of a whole-place weight function of a WPOAS-net using linear expressions involving the p_i's. For example, if $n = 3$ and $W(p_2,t) = (2,0,1,4)$, then we may write $W(p_2,t) = 2 \cdot p_1 + p_3 + 4$. A place p_j $(1 \leq j \leq n)$ is a *whole-place* for place $p \in P$ if $W(p,t)^{(j)} \neq 0$ or $W(t,p)^{(j)} \neq 0$, for some $t \in T$. In such a case we also write $p_j \rightsquigarrow p$. Note that it may happen that $p = p_j$; see, for example, Figure 4(a), where $W(p_5,e) = p_5$. In general, in net diagrams, we will be using for arc annotations linear expressions involving places rather than vectors with their coefficients.

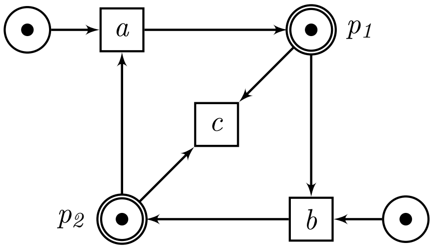


Figure 3: A WPOAS-net.

3.1 WPOAS-nets with Localities

We are now in a position to introduce nets with whole-place operations, a/sync places, and localities, the central concept of this paper.

A WPOAS-*net with localities* (or WPOASL-net) is a tuple

$$N = \langle P', P'', T, W, \mathbf{m}_0, \ell \rangle \tag{4}$$

such that $N' = \langle P', P'', T, W, \mathbf{m}_0 \rangle$ is a WPOAS-net as in (1), and $\ell : T \to \{1, 2, \ldots, l\}$, where $l \geq 1$, is the *locality mapping* of N with $\{1, 2, \ldots, l\}$ the *localities* of N. For a multiset of transitions $\alpha \in \langle T \rangle$, we write $\ell(\alpha)$ to denote the set $\{\ell(t) : \alpha(t) > 0\}$ consisting of the localities of the transitions in α. In diagrams, nodes representing transitions with the same locality are shaded in the same way (see Figure 4, where a and e share one locality and b, c and d share another locality).

The locality mapping assigns to each transition of the net a locality. Grouping the net-transitions in different localities makes it possible to implement an execution semantics based on *locally maximal* steps. To define such semantics of N, we use a *step firing policy* (see [14]). Step firing policies are a means of controlling and constraining the potentially huge number of execution paths generated by a concurrent system. Our step firing policy here is to forbid at each marking $\mathbf{m}$ the occurrence of those enabled steps that are not locally maximal. So, our step firing policy is given by a *control disabled steps* mapping [3] $cds_{lmax} : 2^{\langle T \rangle} \to 2^{\langle T \rangle \setminus \{\mathbf{0}\}}$ such that $cds_{lmax}(X) = \varnothing$ if $X \subseteq \langle T \rangle$ is infinite — see our remarks on maximal concurrency in Section 2

[3] Control disabled steps mappings are defined in [14] in the context of τ-nets, and this is how cds_{lmax} will be used in Section 4.

— and otherwise:

$$cds_{lmax}(X) = \{\alpha \in X \setminus \{\mathbf{0}\} \mid \exists \beta \in X : \ell(\beta) \subseteq \ell(\alpha) \wedge \alpha < \beta\} \ .$$

Thus if X is finite, $cds_{lmax}(X)$ consists of all steps α in X for which there is a multiset $\beta \neq \alpha$ in X which has no other localities than those associated with the elements of α and which strictly contains α (meaning that $\alpha(t) < \beta(t)$ for all $t \in T$). We then apply this idea to the steps enabled at a marking in the underlying WPOAS-net N'. So, the cds_{lmax} mapping takes as an argument a set of steps enabled at some marking of N' and returns its subset with the steps rejected by the locally maximal step firing policy of N.

Let $\mathbf{m}$ be a marking of N' (and thus also a marking of N). The steps in $enb_{N'}(\mathbf{m})$ are called *resource enabled* at $\mathbf{m}$, and the steps in

$$enb_N(\mathbf{m}) = enb_{N'}(\mathbf{m}) \setminus cds_{lmax}(enb_{N'}(\mathbf{m}))$$

are said to be *control enabled* at $\mathbf{m}$. A control enabled step α when fired leads to the marking $\mathbf{m}'$ given, for every $p \in P$, by the formula (3).

We then define the *concurrent reachability graph* $CRG(N)$ of N as the step transition system formed by firing inductively from $\mathbf{m}_0$ all possible control enabled steps. This is illustrated in Figure 4 which depicts a WPOASL-net, N, which generates a concurrent reachability graph isomorphic to that shown in Figure 1. Note that the concurrent reachability graph of the underlying net N' of the net N in Figure 4(a) would show much richer behaviour than that depicted in Figure 4(b), as it would contain all resource enabled steps of N, including those rejected by its locally maximal step firing policy. For example, resource enabled steps of N at $\mathbf{m}_1$ are: $\mathbf{0}$, $\{b\}$, $\{c\}$, $\{b,c\}$, $\{a,b\}$, and $\{a,b,c\}$. However, steps $\{b\}$, $\{c\}$, and $\{a,b\}$ are rejected by the locally maximal step firing policy of N, as they can be extended to other resource enabled steps at $\mathbf{m}_1$, $\{b,c\}$ and $\{a,b,c\}$, within their existing localities: $\ell(\{b\}) = \ell(\{c\}) = \ell(\{b,c\})$ and $\ell(\{a,b\}) = \ell(\{a,b,c\})$.

3.1.1 Expressiveness of WPOASL-nets

In a WPOASL-net the control enabledness of the step $\{a,b\}$ at a marking $\mathbf{m}$, without $\{a\}$ and $\{b\}$ being control enabled at $\mathbf{m}$, can be the consequence *either* of a and b 'cooperating' with each other through a/sync place(s) (and in this way being dependent on each other), *or* of being co-located (and being forced to synchronise according to the locally maximal step firing policy). However, the first possibility represents a 'weaker' coupling than the

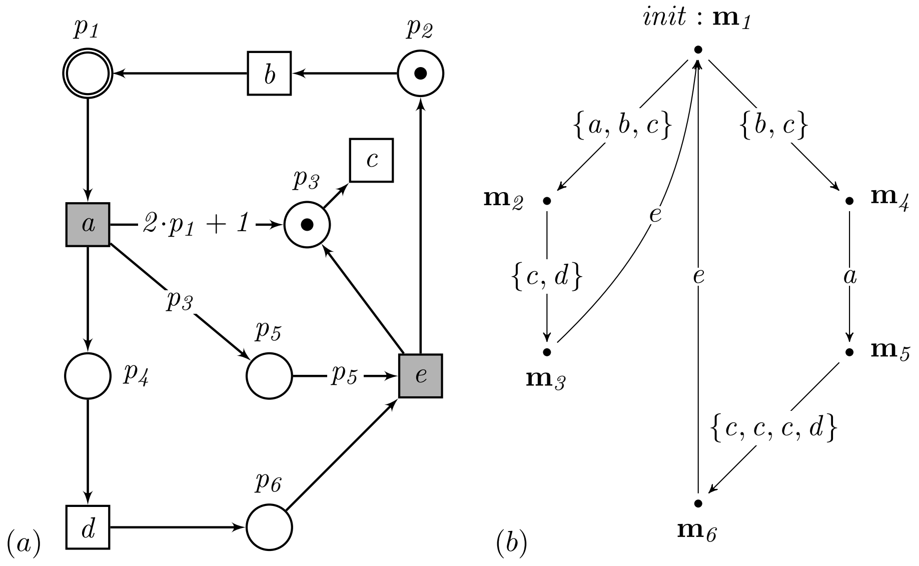


Figure 4: A WPOASL-net N (a); and its concurrent reachability graph (b), where $\mathbf{m}_1 = (0, 1, 1, 0, 0, 0)$, $\mathbf{m}_2 = (0, 0, 1, 1, 1, 0)$, $\mathbf{m}_3 = (0, 0, 0, 0, 1, 1)$, $\mathbf{m}_4 = (1, 0, 0, 0, 0, 0)$, $\mathbf{m}_5 = (0, 0, 3, 1, 0, 0)$, and $\mathbf{m}_6 = (0, 0, 0, 0, 0, 1)$.

second one as transitions would cooperate with each other as a 'last resort', and this is marking dependent, but being co-located is a global property for transitions that applies at every marking. We also observe that whole-place operations and a/sync places support, in their unique ways, dynamic semantics of nets, allowing the enabling conditions for steps to depend on a current marking, while the localities of transitions help to express a global behavioural property of steps.

Having said that, each of the three non-standard features of WPOASL-nets adds to their modelling power. We will now show that dropping either of them restricts the expressiveness of WPOASL-nets in terms of the generated concurrent reachability graphs.

Whole-place operations. Consider the WPOASL-net N, with initial marking $\mathbf{m}_0$, depicted in Figure 5(a). Suppose that N' is a WPOASL-net without whole-places with initial marking $\mathbf{m}'_0$ and having concurrent reachability graph isomorphic to that of N.

Let $\mathbf{m}_1$ be the marking reached in N by firing from $\mathbf{m}_0$ $\{a\}$ followed by

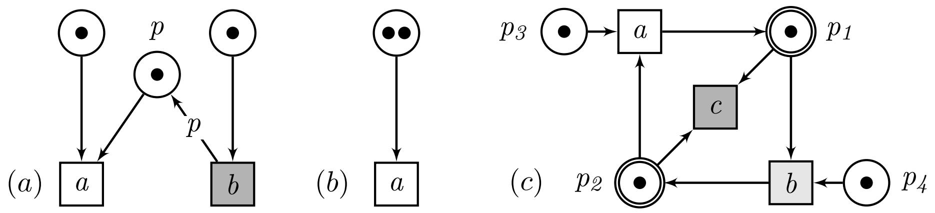


Figure 5: A wpoasl-net whose concurrent reachability graph cannot be generated by: any wpoasl-net without whole-places (a); any wpoas-net (b); and any wpoasl-net without a/sync places (c).

$\{b\}$, and $\mathbf{m}_2$ be the marking reached in N by firing from $\mathbf{m}_0$ $\{b\}$ followed by $\{a\}$. Clearly these two markings are different. Hence the corresponding markings of N', $\mathbf{m}'_1$ and $\mathbf{m}'_2$, must also be different. However, as the arc weights of N' are constant, this is impossible as both $\{a\}$ and $\{b\}$ remove and deposit the same number of tokens in each place of N' no matter in which order they are executed. We thus obtained a contradiction.

Locally maximal semantics. Consider the wpoasl-net N, with initial marking $\mathbf{m}_0$, depicted in Figure 5(b). Observe that in that net a can be considered to be co-located with itself. Suppose that N' is an wpoas-net with initial marking $\mathbf{m}'_0$ and having concurrent reachability graph isomorphic to that of N.

We have $enb_{N'}(\mathbf{m}'_0) = enb_N(\mathbf{m}_0) = \{\mathbf{0}, \{a, a\}\}$. Hence there must be a place p in N' which blocks $\{a\}$ at $\mathbf{m}'_0$, but does not block $\{a, a\}$. Clearly, p has to be an a/sync place. Suppose that, in the initial marking, the weight of arc from p to a evaluates to m, and the weight of arc from a to p evaluates to n, and there are k tokens in p. Therefore, we must have $2m \leq k + 2n$ and $m > k + n$. Hence $2k + 2n < 2m \leq k + 2n$, and so $k < 0$, a contradiction.

a/sync places. Consider the wpoasl-net N, with initial marking $\mathbf{m}_0$, depicted in Figure 5(c). Suppose that N' is a wpoasl-net without a/sync places with initial marking $\mathbf{m}'_0$ and having concurrent reachability graph isomorphic to that of N.

Let $\mathbf{m}_1$ be the marking of N with $\mathbf{m}_1(p_1) = \mathbf{m}_1(p_2) = 0$ and $\mathbf{m}_1(p_3) = \mathbf{m}_1(p_4) = 1$, obtained by executing step $\{c\}$ at $\mathbf{m}_0$. We observe that:

$$enb_N(\mathbf{m}_1) = \{\mathbf{0}, \{a, b\}\} \qquad \textit{and} \qquad \{a\}, \{a, b\} \in enb_N(\mathbf{m}_0)\ .$$

Then, there are two markings of N', $\mathbf{m}'_0$ and $\mathbf{m}'_1$, such that:

$$enb_{N'}(\mathbf{m}'_1) = \{\mathbf{0}, \{a, b\}\} \;\; (*) \qquad \textit{and} \qquad \{a\}, \{a, b\} \in enb_{N'}(\mathbf{m}'_0) \; . \;\; (**)$$

From the definition of enabledness of steps in WPOASL-nets, because N' has no a/sync places, it follows that for every marking $\mathbf{m}'$ in N': if a step is resource enabled at $\mathbf{m}'$ then each of its subsets is resource enabled at $\mathbf{m}'$. Therefore, if N' is to have the same behaviour as N (and satisfy $(*)$), it can only be achieved by its locality mapping. Let ℓ' be the locality mapping of N'. Then, since N' does not contain a/sync places and (*) holds, it must be the case that $\ell'(a) = \ell'(b)$ (otherwise we would have $\{a\}, \{b\} \in enb_{N'}(\mathbf{m}'_1)$). But this contradicts (**) as in such a case $\ell'(\{a, b\}) = \{\ell'(a)\}$ and $\{a\} < \{a, b\}$. We therefore obtained a contradiction.

4 Synthesis of WPOASL-nets

The net synthesis problem we consider here aims to devise a procedure which constructs a WPOASL-net with a concurrent reachability graph (based on the locally maximal step firing policy) that is isomorphic to a given step transition system $\mathcal{T} = \langle Q, \langle T \rangle, \delta, q_0 \rangle$.

The synthesis problem was first investigated in the literature for individual classes of Petri nets, and later a general approach was developed for τ-nets, where each class of nets is represented by its own net-type τ. The key aspect of any solution to the synthesis problems is to discover all the necessary net places and their connections with transitions of T using $\mathcal{T}$ and τ. All information needed to construct a place in a net that realises $\mathcal{T}$, is encapsulated in the notion of region, which depends on the parameter τ. Before we give the definition of a region relevant to our problem, we need to realise that for nets with whole-place operations like WPOASL-nets, discovering places for the net to be constructed is complicated by the fact that the (new) markings of places dynamically depend on the markings of other places. Therefore, instead of discovering individual places of the net to be constructed, one needs a procedure to discover clusters of related places, each cluster containing places that depend only on one another. We will therefore re-define WPOASL-nets as nets containing clusters of related places, each containing at most k' a/sync and at most k'' standard places (k'/k''-WPOASL-nets) and express them as τ-nets, so that we can synthesise them as τ-nets, using the general approach for τ-nets.[4]

[4] Note that by including an extra parameter k' for a/sync places, we extend the results

The re-definition with clusters of places, as described above, requires extra information about the net to be synthesised: the values k' and k'' must be given upfront. This simplifies the synthesis problem. The consequence of this simplification is that if we solve the synthesis problem for a given step transition system and obtain a k'/k''-WPOASL-net, we can then express it as an equivalent WPOASL-net by combining clusters of places. However, if the solution is not found in the class of k'/k''-WPOASL-nets, then it is still possible that there is a solution in the class of arbitrary WPOASL-nets.

4.1 k'/k"-WPOASL-nets and their Net-type

A k'/k''*-restricted* WPOASL-*net* (or k'/k''-WPOASL-net), where $k'+k'' \geq 1$, is a WPOASL-net N for which there is a partition $P_1 \uplus \cdots \uplus P_r$ of the set of places such that each P_i has at most k' a/sync places, at most k'' standard places and, for all $p \in P_i$ and $p' \notin P_i$, $p \not\rightsquigarrow p' \not\rightsquigarrow p$. In other words, the places can be partitioned into clusters of bounded size (in terms of both a/sync and standard places) so that there is no exchange of whole-place marking information between different clusters.

Strictly speaking, k'/k''-WPOASL-nets (nor WPOASL-nets) are not τ-nets as the newly generated marking of a place does not only depend on its current marking and its connections to transitions. Thanks to the clustering of the places, however, they still fit the ideas behind the definition of τ-nets and we can define a suitably extended net-type capturing the behaviour of sets of several places rather than the behaviour of single places. More precisely, for all $k', k'' \geq 0$ with $k'+k'' \geq 1$, the k'/k''-WPOASL-*net-type* is a transition system $\tau^{k'/k''}$ — generalising the net-type τ^k_{wpo} introduced and used in [24, 23] — defined in the following way.

Let $k = k' + k''$. Then :

$$\tau^{k'/k''} = \langle \mathbb{N}^k, (\mathbb{N}^{k+1})^k \times (\mathbb{N}^{k+1})^k, \Delta^{k'/k''} \rangle$$

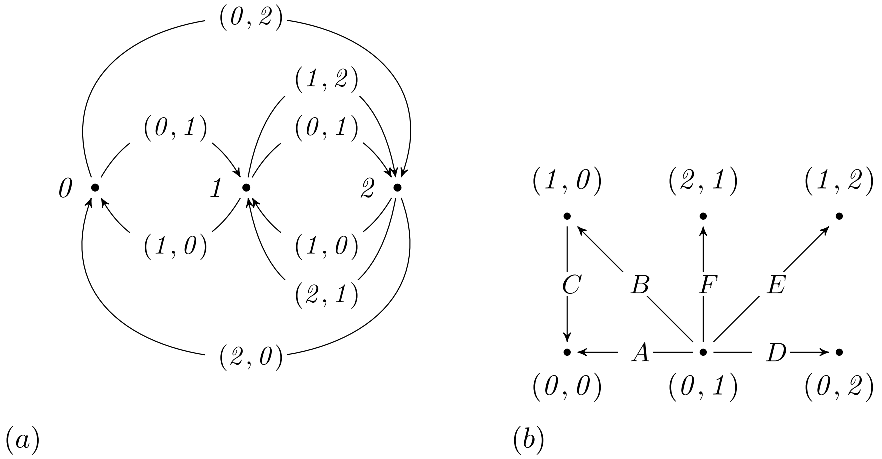
where

$$\Delta^{k'/k''} : \mathbb{N}^k \times ((\mathbb{N}^{k+1})^k \times (\mathbb{N}^{k+1})^k) \to \mathbb{N}^k$$

is a partial function such that $\Delta^{k'/k''}(\mathbf{x}, (X, Y))$ is defined if, for all $j' \in \{1, \ldots, k'\}$ (indices for a/sync places) and $j'' \in \{k'+1, \ldots, k\}$ (indices for standard places):

$$\begin{array}{rcl} \mathbf{x}^{(j')} & \geq & ((\mathbf{x}, 1) \otimes (X - Y))^{(j')} \\ \mathbf{x}^{(j'')} & \geq & ((\mathbf{x}, 1) \otimes X)^{(j'')} \end{array} \qquad (5)$$

from [23].



$$
\begin{array}{rclrcl}
A & = & \left(\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \end{bmatrix}\right) & B & = & \left(\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \end{bmatrix}\right) \\
C & = & \left(\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}\right) & D & = & \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} 2 & 1 \\ 1 & 2 \\ 0 & 0 \end{bmatrix}\right) \\
E & = & \left(\begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 2 \\ 1 & 1 \\ 0 & 0 \end{bmatrix}\right) & F & = & \left(\begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 2 \\ 1 & 0 \\ 1 & 0 \end{bmatrix}\right)
\end{array}
$$

Figure 6: Fragments of two infinite net-types: τ_{PT} (a); and $\tau^{1/1}$ (b).

and, if that is the case,

$$\Delta^{k'/k''}(\mathbf{x}, (X, Y)) = \mathbf{x} + (\mathbf{x}, 1) \otimes (Y - X) \ .$$

Note that here we treat tuples of vectors in $(\mathbb{N}^{k+1})^k$ as $(k+1) \times k$ arrays.

Having defined the net-type $\tau^{k'/k''}$, we can now define a $\tau^{k'/k''}$-*net* as a tuple $N = \langle \mathcal{P}, T, F, M_0, \ell \rangle$, where $\mathcal{P} = \{P_1, \ldots, P_r\}$ is a set of disjoint sets of implicitly ordered places comprising exactly k' a/sync places and k'' standard places each, T is a set of transitions being different from the places in the sets of $\mathcal{P}$,

$$F : (\mathcal{P} \times T) \to (\mathbb{N}^{k+1})^k \times (\mathbb{N}^{k+1})^k$$

is a *flow mapping*, M_0 is an *initial marking* belonging to the set of *markings* defined as mappings from $\mathcal{P}$ to $\mathbb{N}^k$, and ℓ is a locality mapping for the transitions in T.

For all $P_i \in \mathcal{P}$ and $\alpha \in \langle T \rangle$, we set

$$F(P_i, \alpha) = \sum_{t \in T} \alpha(t) \cdot F(P_i, t) \ .$$

Then a step $\alpha \in \langle T \rangle$ is *resource enabled* at a marking M if, for every $P_i \in \mathcal{P}$, $F(P_i, \alpha) \in enb_{\tau^{k'/k''}}(M(P_i))$. We denote this by $\alpha \in enb_N(M)$.

Firing such a step (for now we ignore the firing policy) produces the marking M', for every $P_i \in \mathcal{P}$, defined by

$$M'(P_i) = \Delta^{k'/k''}(M(P_i), F(P_i, \alpha)) \ .$$

We denote this by $M[\alpha\rangle M'$, and then define the *concurrent reachability graph* $CRG(N)$ of N as the step transition system formed by firing inductively from M_0 all possible resource enabled steps.

However, we want to execute N under the locally maximal step firing policy. The related control disabled steps mapping cds_{lmax}, when applied to N, would control disable at each marking M all the resource enabled steps that belong to $cds_{lmax}(enb_N(M))$. That is,

$$enb_{N,cds_{lmax}}(M) = enb_N(M) \setminus cds_{lmax}(enb_N(M)) \tag{6}$$

is the set of *control enabled* steps at a reachable marking M under cds_{lmax}. We then use $CRG_{cds_{lmax}}(N)$ to denote the induced reachable restriction of $CRG(N)$, which may be finite even for an infinite $CRG(N)$.

As an example of a $\tau^{k'/k''}$-net we can take the net N in Figure 7(a). It can be treated as $\tau^{1/1}$-net, $N = \langle \mathcal{P}, T, F, M_0, \ell \rangle$, where we have only one cluster of places $P_1 = \{p_1, p_2\}$ (so $\mathcal{P} = \{P_1\}$), $T = \{a, b\}$, $M_0(P_1) = (0, 1)$, ℓ can be defined in two different ways (leading to two different concurrent reachability graphs of N, as depicted in Figure 7(b, c)), and the flow mapping is defined as follows:

$$F(P_1, a) = \left(\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \right) \quad F(P_1, b) = \left(\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \end{bmatrix} \right)$$

A fragment of the net-type $\tau^{1/1}$, capturing the behaviour of clusters of places in $\tau^{1/1}$-nets, is depicted in Figure 6(b).

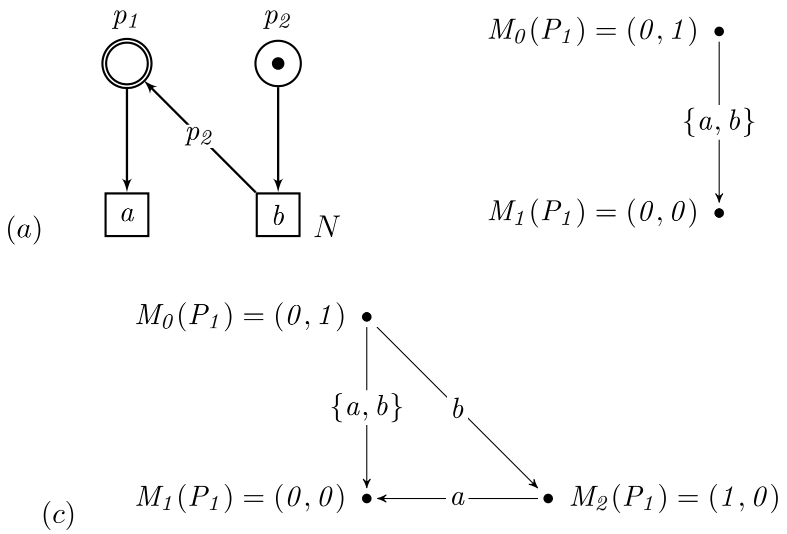


Figure 7: A $\tau^{1/1}$-net N (a); its $CRG_{cds_{lmax}}(N)$ with $\ell(a) = \ell(b)$ (b); and its $CRG_{cds_{lmax}}(N)$ with $\ell(a) \neq \ell(b)$ (c).

4.2 Synthesising k'/k"-WPOASL-nets as $\tau^{k'/k''}$-nets

First we need to express a k'/k''-WPOASL-net $N = \langle P', P'', T, W, \mathbf{m}_0, \ell\rangle$, with set of places $P = P' \cup P'' = \{p_1, \ldots, p_n\}$ and clusters $P_1, \ldots, P_r$, as a $\tau^{k'/k''}$-net.

Suppose that each set P_i in the partition has exactly k' a/sync places and k'' standard places. (We can always add dummy empty places disconnected from the original transitions and places, if needed.) We then define $\widehat{N} = \langle \mathcal{P}, T, F, M_0, \ell\rangle$ so that $\mathcal{P} = \{P_1, \ldots, P_r\}$ and, for all $P_i \in \mathcal{P}$ and $t \in T$:

- $F(P_i, t) = (X, Y)$ where X and Y are arrays respectively obtained from the arrays $[W(p_1, t), \ldots, W(p_n, t)]$ and $[W(t, p_1), \ldots, W(t, p_n)]$, where the $W(\cdot, \cdot)$ are column vectors, by deleting the rows and columns corresponding to the places in $P \setminus P_i$; and

- $M_0(P_i)$ is obtained from $\mathbf{m}_0$ by deleting the entries corresponding to the places in $P \setminus P_i$.

It is straightforward to check that the concurrent reachability graphs of N and $\widehat{N}$ are isomorphic (when we execute both nets under the cds_{lmax} policy or ignore the policy in both nets). Conversely, one can transform any $\tau^{k'/k''}$-net into an equivalent k'/k''-WPOASL-net, and trivially the WPOASL-net as in (4) is an $n'/(n-n')$-WPOASL-net. All this implies that — just as in case of any synthesis problem that asks for a procedure to construct a τ-net with a given concurrent reachability graph, see, e.g., [24] — k'/k''-WPOASL-net synthesis involves the following two problems of $\tau^{k'/k''}$-net synthesis.

In the rest of this section we assume that k' and k'' are non-negative integers such that $k'+k'' \geq 1$, and $k = k'+k''$.

Problem 1 (feasibility) *Let $\mathcal{T} = \langle Q, \langle T\rangle, \delta, q_0\rangle$ be a bounded step transition system, and ℓ be a locality mapping for T.*
Provide necessary and sufficient conditions for $\mathcal{T}$ to be realised by some $\tau^{k'/k''}$-net, $\widehat{N}$, executed under the cds_{lmax} policy defined by ℓ. (That is, it is required that $\mathcal{T} \cong CRG_{cds_{lmax}}(\widehat{N})$.)

Problem 2 (effective construction) *Let $\mathcal{T} = \langle Q, \langle T\rangle, \delta, q_0\rangle$ be a finite step transition system, and ℓ be a locality mapping for T.*
Decide whether there is a finite $\tau^{k'/k''}$-net realising $\mathcal{T}$ when executed under the cds_{lmax} policy defined by ℓ. Moreover, if the answer is positive construct such a $\tau^{k'/k''}$-net.

To address Problem 1, we define a $\tau^{k'/k''}$*-region* of $\mathcal{T} = \langle Q, \langle T\rangle, \delta, q_0\rangle$ as a pair:

$$\langle \sigma : Q \to \mathbb{N}^k, \eta : T \to (\mathbb{N}^{k+1})^k \times (\mathbb{N}^{k+1})^k\rangle$$

such that, for all $q \in Q$ and $\alpha \in enb_{\mathcal{T}}(q)$,

$$\eta(\alpha) \in enb_{\tau^{k'/k''}}(\sigma(q)) \quad \text{and} \quad \Delta^{k'/k''}(\sigma(q), \eta(\alpha)) = \sigma(\delta(q,\alpha)) ,$$

where $\eta(\alpha) = \sum_{t\in T} \alpha(t)\cdot\eta(t)$. Moreover, for every state q of Q, we denote by $enb_{\mathcal{T},\tau^{k'/k''}}(q)$ the set of all steps α such that

$$\eta(\alpha) \in enb_{\tau^{k'/k''}}(\sigma(q)) ,$$

for all $\tau^{k'/k''}$-regions $\langle\sigma,\eta\rangle$ of $\mathcal{T}$. Hence for every state q of $\mathcal{T}$, we have

$$enb_{\mathcal{T}}(q) \subseteq enb_{\mathcal{T},\tau^{k'/k''}}(q). \tag{7}$$

In the context of the synthesis problem, a $\tau^{k'/k''}$-region represents a cluster of places whose local states (in $\tau^{k'/k''}$) are consistent with the global states (in $\mathcal{T}$). Then, to deliver a realisation of $\mathcal{T}$, one needs to find *enough* $\tau^{k'/k''}$-regions[5] to construct a $\tau^{k'/k''}$-net $\widehat{N}$ satisfying $\mathcal{T} \cong CRG_{cds_{lmax}}(\widehat{N})$. The following two *regional axioms* describe the conditions that should be satisfied by a desirable set of regions. Together they provide a full characterisation of realisable transition systems, i.e., those that can be realised by a $\tau^{k'/k''}$-net executed under some cds_{lmax} policy, as we later show in Theorem 1.

$\mathcal{T}$ can be realised by a WPOASL-net if and only if Axioms 1 and 2 are satisfied (see, e.g., [24]).

Axiom 1 (state separation) *For any pair of states $q \neq r$ of $\mathcal{T}$, there is a $\tau^{k'/k''}$-region $\langle\sigma,\eta\rangle$ of $\mathcal{T}$ such that $\sigma(q) \neq \sigma(r)$.*

Axiom 2 (forward closure) *For every state q of $\mathcal{T}$,*

$$enb_{\mathcal{T}}(q) = enb_{\mathcal{T},\tau^{k'/k''}}(q) \setminus cds_{lmax}(enb_{\mathcal{T},\tau^{k'/k''}}(q)) \ .$$

The first axiom links the states of $\mathcal{T}$ with markings of the net to be constructed, making sure that a difference between two states of $\mathcal{T}$ is reflected in a different number of tokens held in the two markings of the net representing the said states. The second axiom means that, for every state q and every step α in $\langle T\rangle \setminus enb_{\mathcal{T}}(q)$, we have either of the following:

- there is a $\tau^{k'/k''}$-region $\langle\sigma,\eta\rangle$ of $\mathcal{T}$ such that $\eta(\alpha) \notin enb_{\tau^{k'/k''}}(\sigma(q))$ (the step α is not *region enabled*), or
- $\alpha \in cds_{lmax}(enb_{\mathcal{T},\tau^{k'/k''}}(q))$ (the step α is not *control enabled*, meaning that it is rejected by the cds_{lmax} policy).

Note that when a $\tau^{k'/k''}$-net under cds_{lmax} realises $\mathcal{T}$, every cluster of places of the net still determines a corresponding $\tau^{k'/k''}$-region of the transition system, without taking cds_{lmax} into account. This is why the same kind of regions would be used if we are asked to synthesise a WPOAS-net (rather than a WPOASL-net).

Before we prove the main result of the paper that gives the solution to Problem 1, we need two auxiliary results. The first one presents an important property enjoyed by control disabled steps mappings, and in particular by cds_{lmax}.

[5] By enough we mean a minimal number of regions to witness the satisfaction of every instance of Axioms 1 and 2 given next (see also [15]).

Proposition 1 *Let X be a finite set of resource enabled steps at some reachable marking of some $\tau^{k'/k''}$-net and Y be its subset ($Y \subseteq X$). Then:*

$$X \setminus cds_{lmax}(X) \subseteq Y \Longrightarrow cds_{lmax}(X) \cap Y \subseteq cds_{lmax}(Y).$$

Proof: Let $\alpha \in cds_{lmax}(X) \cap Y$. We need to show that $\alpha \in cds_{lmax}(Y)$. From $\alpha \in cds_{lmax}(X)$ it follows that there is $\beta \in X$ such that $\ell(\beta) \subseteq \ell(\alpha)$ and $\alpha < \beta$. We now consider two cases:

Case 1: $\beta \in Y$. Then $\alpha \in cds_{lmax}(Y)$.

Case 2: $\beta \in X \setminus Y$. Then, by $X \setminus cds_{lmax}(X) \subseteq Y$, we have that $\beta \in cds_{lmax}(X)$. Hence, there is $\gamma \in X$ such that $\ell(\gamma) \subseteq \ell(\beta)$ and $\beta < \gamma$. If $\gamma \in Y$ we can continue as in case 1, with γ replacing β and obtain $\alpha \in cds_{lmax}(Y)$ due to the transitivity of $\subseteq$ and $<$. Otherwise, we continue as in Case 2 with γ replacing β and so $\gamma \in cds_{lmax}(X)$. Then we can repeat the same argument. Now, because X is a finite set, one must find sooner or later in this iteration some step $\phi \in Y$ such that case 1 holds with ϕ replacing β, and so $\alpha \in cds_{lmax}(Y)$. □

The second auxiliary result associates a region of a step transition system $\mathcal{T}$ with a particular cluster of places of the net to be synthesised from $\mathcal{T}$. The mappings σ and η hold all the information about the associated cluster of places, their connections to transitions in the net and their markings for every state of the net. In fact, for the mapping σ, if we know η, it is enough to know its value for the initial state q_0 to uniquely compute the values for the remaining states of $\mathcal{T}$.

Proposition 2 *Let $\widehat{N} = \langle \mathcal{P}, T, F, M_0, \ell \rangle$ be a $\tau^{k'/k''}$-net such that $\mathcal{T} \cong CRG_{cds_{lmax}}(\widehat{N})$. Then, for each cluster $P_i \in \mathcal{P}$ ($i = 1, \ldots, r$), there is exactly one $\tau^{k'/k''}$-region $\langle \sigma, \eta \rangle$ of $\mathcal{T}$ such that $\sigma(q_0) = M_0(P_i)$ and $\eta(\alpha) = F(P_i, \alpha)$ for all steps $\alpha \in \langle T \rangle$.*

Proof: All step transition systems we consider are deterministic. Observe that both δ and $\Delta^{k'/k''}$ are functions rather than relations. Also observe that $\mathcal{T}$ is reachable (i.e., each of its states is reachable from the initial one). Hence,

$$\sigma(q_0) \quad and \quad \eta : \langle T \rangle \to (\mathbb{N}^{k+1})^k \times (\mathbb{N}^{k+1})^k$$

determine at most one mapping $\sigma : Q \to \mathbb{N}^k$ such that

$$\Delta^{k'/k''}(\sigma(q), \eta(\alpha)) = \sigma(\delta(q, \alpha))$$

whenever $\alpha \in enb_{\mathcal{T}}(q)$, and therefore they determine at most one $\tau^{k'/k''}$-region of $\mathcal{T}$.

We now define the mapping σ. Let $P_i \in \mathcal{P}$ $(i = 1, \ldots, r)$. By assumption, $\mathcal{T} \cong CRG_{cds_{lmax}}(\widehat{N})$ and $CRG_{cds_{lmax}}(\widehat{N})$ is a sub-graph of $CRG(\widehat{N})$. Let $\sigma : Q \to \mathbb{N}^k$ be defined as follows: $\sigma(q) = f(q)(P_i)$, where $f(q)$ is the image of q through the isomorphism $\cong$ ($f(q)$ is a marking of $\widehat{N}$). Then, for every $\alpha \in enb_{\mathcal{T}}(q)$, we have, from $\mathcal{T} \cong CRG_{cds_{lmax}}(\widehat{N})$, that α is resource enabled at $f(q)$ in $\widehat{N}$, and hence

$$F(P_i, \alpha) \in enb_{\tau^{k'/k''}}(f(q)(P_i))$$

and the marking of P_i after α is fired is

$$f(\delta(q, \alpha))(P_i) = \Delta^{k'/k''}(f(q)(P_i), F(P_i, \alpha)) \ .$$

Therefore, we have, for σ defined as above and $\eta(\alpha) = F(P_i, \alpha)$ (as stated in the assumptions), that

$$\eta(\alpha) \in enb_{\tau^{k'/k''}}(\sigma(q)) \quad \textit{and} \quad \sigma(\delta(q, \alpha)) = \Delta^{k'/k''}(\sigma(q), \eta(\alpha)) \ .$$

Hence $\langle \sigma, \eta \rangle$, with σ defined as above, is a $\tau^{k'/k''}$-region of $\mathcal{T}$ associated with P_i. Also,

$$\sigma(q_0) = f(q_0)(P_i) = M_0(P_i)$$

as $\cong$ is an isomorphism preserving the initial states. Therefore, the result holds. □

Theorem 1 *Let $\mathcal{T} = \langle Q, \langle T \rangle, \delta, q_0 \rangle$ be a bounded step transition system and cds_{lmax} be the locally maximal step firing policy associated with a locality mapping ℓ defined for T.*

Then $\mathcal{T}$ can be realised by a $\tau^{k'/k''}$-net under cds_{lmax} iff *Axioms 1 and 2 are satisfied.*

Proof: ($\Longrightarrow$) Let $\mathcal{T}$ be realised by the $\tau^{k'/k''}$-net $\widehat{N}$ under cds_{lmax}.

We have $\mathcal{T} \cong CRG_{cds_{lmax}}(\widehat{N})$. Let $f : Q \to (\mathcal{P} \to \mathbb{N}^k)$ be a bijection linking the states of $\mathcal{T}$ with the reachable markings of $\widehat{N}$. First, we show that:

$$enb_{\mathcal{T},\tau^{k'/k''}}(q) \subseteq enb_{\widehat{N}}(f(q)). \tag{8}$$

Let $\alpha \notin enb_{\widehat{N}}(f(q))$. Then there is a cluster $P_i \in \mathcal{P}$ $(1 \le i \le r)$ in $\widehat{N}$ such that

$$F(P_i, \alpha) \notin enb_{\tau^{k'/k''}}(f(q)(P_i)) \ .$$

Let $\langle\sigma,\eta\rangle$ be the $\tau^{k'/k''}$-region of $\mathcal{T}$ induced by P_i according to Proposition 2. Then $\sigma(q) = f(q)(P_i)$ and $\eta(\alpha) = F(P_i,\alpha)$. Hence,

$$\eta(\alpha) \notin \mathit{enb}_{\tau^{k'/k''}}(\sigma(q))$$

and so $\alpha \notin \mathit{enb}_{\mathcal{T},\tau^{k'/k''}}(q)$.

To show Axiom 1 let $q \neq r$ in Q. As $\mathcal{T} \cong \mathit{CRG}_{\mathit{cds}_{lmax}}(\widehat{N})$, we have $f(q) \neq f(r)$, and therefore $f(q)(P_i) \neq f(r)(P_i)$, for some $1 \leq i \leq r$. Let $\langle\sigma,\eta\rangle$ be the $\tau^{k'/k''}$-region of $\mathcal{T}$ induced by P_i according to Proposition 2. Then $\sigma(q) = f(q)(P_i) \neq f(r)(P_i) = \sigma(r)$. Hence, $\sigma(q) \neq \sigma(r)$.

To show Axiom 2, we first show that, for all $\alpha \in \langle T\rangle$ and $q \in Q$:

$$\alpha \notin \mathit{enb}_{\mathcal{T}}(q) \Longrightarrow \alpha \notin \mathit{enb}_{\mathcal{T},\tau^{k'/k''}}(q) \setminus \mathit{cds}_{lmax}(\mathit{enb}_{\mathcal{T},\tau^{k'/k''}}(q)). \qquad (9)$$

Let $q \in Q$ and $\alpha \notin \mathit{enb}_{\mathcal{T}}(q)$. From (6) and $\mathcal{T} \cong \mathit{CRG}_{\mathit{cds}_{lmax}}(\widehat{N})$, either:

(i) $\alpha \notin \mathit{enb}_{\widehat{N}}(f(q))$ or

(ii) $\alpha \in \mathit{enb}_{\widehat{N}}(f(q)) \cap \mathit{cds}_{lmax}(\mathit{enb}_{\widehat{N}}(f(q)))$.

If (i) holds then, by (8), we have $\alpha \notin \mathit{enb}_{\mathcal{T},\tau^{k'/k''}}(q)$ and so (9) holds. In (ii) two cases are possible. If $\alpha \notin \mathit{enb}_{\mathcal{T},\tau^{k'/k''}}(q)$ we have (9); otherwise $\alpha \in \mathit{enb}_{\mathcal{T},\tau^{k'/k''}}(q)$ and we set the following:

$$X = \mathit{enb}_{\widehat{N}}(f(q)) \quad \mathit{and} \quad Y = \mathit{enb}_{\mathcal{T},\tau^{k'/k''}}(q) \ .$$

By (8), we have $Y \subseteq X$. Moreover, by (6,7) and $\mathcal{T} \cong \mathit{CRG}_{\mathit{cds}_{lmax}}(\widehat{N})$, we have $X \setminus \mathit{cds}_{lmax}(X) \subseteq Y$. Hence, by Proposition 1 and the fact that $\mathcal{T}$ is bounded,

$$\alpha \in \mathit{cds}_{lmax}(X) \cap Y \subseteq \mathit{cds}_{lmax}(\mathit{enb}_{\mathcal{T},\tau^{k'/k''}}(q)) \ ,$$

and so (9) holds.

To finish the proof of Axiom 2, we show that, for all $q \in Q$:

$$\mathit{enb}_{\mathcal{T}}(q) \subseteq \mathit{enb}_{\mathcal{T},\tau^{k'/k''}}(q) \setminus \mathit{cds}_{lmax}(\mathit{enb}_{\mathcal{T},\tau^{k'/k''}}(q)). \qquad (10)$$

By isomorphism $\mathcal{T} \cong \mathit{CRG}_{\mathit{cds}_{lmax}}(\widehat{N})$ and (6), we have

$$\mathit{enb}_{\mathcal{T}}(q) = \mathit{enb}_{\widehat{N}}(f(q)) \setminus \mathit{cds}_{lmax}(\mathit{enb}_{\widehat{N}}(f(q))) \ .$$

Hence

$$\mathit{enb}_{\mathcal{T}}(q) \cap \mathit{cds}_{lmax}(\mathit{enb}_{\widehat{N}}(f(q))) = \varnothing \ .$$

Thus, by (8) and $cds_{lmax}(Y) \subseteq cds_{lmax}(X)$ (for $Y \subseteq X$),

$$enb_{\mathcal{T}}(q) \cap cds_{lmax}(enb_{\mathcal{T},\tau^{k'/k''}}(q)) = \varnothing \ .$$

Moreover, by (7), which always holds, we can conclude that (10) holds.

($\Longleftarrow$) Let $\mathcal{R}$ be the set of all $\tau^{k'/k''}$-regions of $\mathcal{T}$.

Let $\widehat{N} = \langle \mathcal{P}, T, F, M_0, \ell \rangle$ be a $\tau^{k'/k''}$-net defined as follows: $\mathcal{P} = \mathcal{R}$, $M_0(P_i) = \sigma(q_0)$ and $F(P_i, t) = \eta(t)$ for any cluster $P_i = \langle \sigma, \eta \rangle \in \mathcal{P}$ and $t \in T$. We will show that if $\mathcal{T}$ satisfies Axioms 1 and 2 then $\mathcal{T} \cong CRG_{cds_{lmax}}(\widehat{N})$.

We denote by $\mathcal{RM}_{cds_{lmax}}$ the set of all markings in $CRG_{cds_{lmax}}(\widehat{N})$ and by $M \xrightarrow{\alpha} M'$ the directed arcs in this graph. We now define a relation $\sim \subseteq Q \times \mathcal{RM}_{cds_{lmax}}$ as the smallest relation that includes $q_0 \sim M_0$ and such that

$$q \sim M,\ \delta(q, \alpha) = q' \text{ and } M \xrightarrow{\alpha} M' \text{ implies } q' \sim M'.$$

We prove first that $\sim$ is a partial bijection between Q and $\mathcal{RM}_{cds_{lmax}}$. By construction of $\widehat{N}$, $M_0(P_i) = \sigma(q_0)$ for every $P_i = \langle \sigma, \eta \rangle$ of $\widehat{N}$. Now let $q \sim M$ with $\delta(q, \alpha) = q'$ and $M \xrightarrow{\alpha} M'$, and assume for the sake of induction that $M(P_i) = \sigma(q)$ for every $P_i = \langle \sigma, \eta \rangle$ of $\widehat{N}$. As $\langle \sigma, \eta \rangle$ is a $\tau^{k'/k''}$-region of $\mathcal{T}$, $\sigma(\delta(q, \alpha)) = \Delta^{k'/k''}(\sigma(q), \eta(\alpha))$. As $P_i = \langle \sigma, \eta \rangle$ is a cluster of places in $\widehat{N}$ and $F(P_i, t) = \eta(t)$ for all $t \in T$ by construction of $\widehat{N}$, we have

$$\sigma(\delta(q, \alpha)) = \Delta^{k'/k''}(M(P_i), F(P_i, \alpha)) \ .$$

From $M \xrightarrow{\alpha} M'$, we have $M'(P_i) = \Delta^{k'/k''}(M(P_i), F(P_i, \alpha))$. As a result,

$$M'(P_i) = \sigma(\delta(q, \alpha)) = \sigma(q')$$

and we have $q' \sim M'$. So, $q \sim M$ implies $M(P_i) = \sigma(q)$ for all $P_i = \langle \sigma, \eta \rangle$ of $\widehat{N}$. Furthermore, from Axiom 1, $q \neq r$ implies $\sigma(q) \neq \sigma(r)$ for some $\tau^{k'/k''}$-region $\langle \sigma, \eta \rangle$ of $\mathcal{T}$. Therefore, the relation $\sim$ is a partial bijection between Q and $\mathcal{RM}_{cds_{lmax}}$.

Next, we show that the following implication is satisfied:

$$q \sim M \Longrightarrow enb_{\mathcal{T},\tau^{k'/k''}}(q) = enb_{\widehat{N}}(M). \tag{11}$$

Let $\alpha \in enb_{\mathcal{T},\tau^{k'/k''}}(q)$. This means that $\eta(\alpha) \in enb_{\tau^{k'/k''}}(\sigma(q))$, for all $\tau^{k'/k''}$-regions $\langle \sigma, \eta \rangle$ of $\mathcal{T}$. It was shown above that, for every cluster of places $P_i = \langle \sigma, \eta \rangle$ of $\widehat{N}$, $M(P_i) = \sigma(q)$, where $q \sim M$. Furthermore, by construction of $\widehat{N}$, $F(P_i, t) = \eta(t)$, for all $t \in T$, and $P_i = \langle \sigma, \eta \rangle$. Hence, $\eta(\alpha) = F(P_i, \alpha)$. Therefore, $F(P_i, \alpha) \in enb_{\tau^{k'/k''}}(M(P_i))$, for every cluster of places P_i of $\widehat{N}$. This in turn means that α is resource enabled at M in $\widehat{N}$: $\alpha \in enb_{\widehat{N}}(M)$.

To show the reverse inclusion, let $\alpha \in enb_{\widehat{N}}(M)$. Then, by the fact that α is resource enabled at M, in $\widehat{N}$, we have

$$F(P_i, \alpha) \in enb_{\tau^{k'/k''}}(M(P_i)) \ ,$$

for every cluster P_i of $\widehat{N}$. From the construction of $\widehat{N}$, it follows that $F(P_i, t) = \eta(t)$ for all $t \in T$ and $P_i = \langle \sigma, \eta \rangle$, hence

$$\eta(\alpha) \in enb_{\tau^{k'/k''}}(M(P_i)) \ .$$

For every cluster $P_i = \langle \sigma, \eta \rangle$ of $\widehat{N}$, $M(P_i) = \sigma(q)$ when $q \sim M$. So,

$$\eta(\alpha) \in enb_{\tau^{k'/k''}}(\sigma(q)) \ ,$$

for every $\tau^{k'/k''}$-region of $\mathcal{T}$. Hence,

$$\alpha \in enb_{\mathcal{T},\tau^{k'/k''}}(q) \ .$$

We now observe that $q \sim M$ implies

$$enb_{\mathcal{T}}(q) = enb_{\widehat{N},cds_{lmax}}(M) \ ,$$

which follows from (11), Axiom 2, and (6). Hence $\sim$ is a bijection between Q and $\mathcal{RM}_{cds_{lmax}}$, and so $\mathcal{T} \cong CRG_{cds_{lmax}}(\widehat{N})$. □

To solve Problem 2 using the feasibility result provided by Theorem 1 one needs to find an effective representation of the $\tau^{k'/k''}$-regions of $\mathcal{T}$. Similarly as in [24, 23], one can define a system of equations and inequalities encoding the conditions that must be satisfied by $\tau^{k'/k''}$-regions. Let $Q = \{q_0, q_1, \ldots, q_m\}$ and $T = \{t_1, \ldots, t_n\}$ (recall that $\mathcal{T}$ is assumed finite in Problem 2). The encoding employs the following variables:

- $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_m$ are k-vectors of non-negative integer variables which encode the mapping σ; and

- $\mathbf{X}_1, \ldots, \mathbf{X}_n$ and $\mathbf{Y}_1, \ldots, \mathbf{Y}_n$ are $(k + 1) \times k$ arrays of non-negative integer variables, which encode the mapping η.

We then define the homogeneous system $\mathcal{S}_{\mathcal{T}}$ such that, for all $\delta(q_s, \alpha) = q_r$ in $\mathcal{T}$, for all $j' \in \{1, \ldots, k'\}$, and for all $j'' \in \{k'+1, \ldots, k\}$:

$$\begin{aligned}
&(\mathbf{x}_s - (\mathbf{x}_s, 1) \otimes \textstyle\sum_{i=1}^{n} \alpha(t_i) \cdot (\mathbf{X}_i - \mathbf{Y}_i))^{(j')} \geq 0 \\
&(\mathbf{x}_s - (\mathbf{x}_s, 1) \otimes \textstyle\sum_{i=1}^{n} \alpha(t_i) \cdot \mathbf{X}_i)^{(j'')} \geq 0 \\
&\mathbf{x}_r - \mathbf{x}_s - (\mathbf{x}_s, 1) \otimes \textstyle\sum_{i=1}^{n} \alpha(t_i) \cdot (\mathbf{Y}_i - \mathbf{X}_i) = \mathbf{0}\ .
\end{aligned} \tag{12}$$

Then the (non-negative) integer solutions of $\mathcal{S}_{\mathcal{T}}$ are in a one-to-one correspondence with the $\tau^{k'/k''}$-regions of $\mathcal{T}$. Therefore, Axioms 1 and 2 can be checked using the solutions of $\mathcal{S}_{\mathcal{T}}$.

In the case of PT-net synthesis, a similar procedure has been shown to be effective since the homogeneous system considered there was linear and one could always find a sufficiently representative finite basis for all the solutions. Here the problem is clearly harder as the system $\mathcal{S}_{\mathcal{T}}$ is quadratic. In practice, one would often want to impose bounds on the allowed range of the whole-place coefficients used in arc annotations. Then Problem 2 has a solution since one could replace $\mathcal{S}_{\mathcal{T}}$ by finitely many linear systems that can be dealt with using the techniques developed for PT-nets. However, one can consider a modified version of Problem 2 without bounding the whole-place coefficients and still obtain a solution, as described in the next section.

5 Synthesis with Known Whole-places

We now assume that all the whole-places of the net to be synthesised are known, that means their markings are known at every state of the initial step transition system $\mathcal{T}$. However, we still do not know their connections to the transitions. The discovery of these is a part of the synthesis problem discussed in this section. Also, we assume that all the whole-places are standard places. The remaining places to be discovered in the synthesis procedure can be a/sync or standard places.

Problem 3 (construction with known whole-places)
Let $\mathcal{T} = \langle Q, \langle T \rangle, \delta, q_0 \rangle$ be a finite step transition system, m be a positive integer, and κ be a mapping assigning tuples in $\mathbb{N}^m$ to the elements of Q. Decide whether there is a WPOASL*-net N with implicitly ordered places $p_1, \ldots, p_m, \ldots, p_n$ realising $\mathcal{T}$ such that:*

- *$p_1, \ldots, p_m$ are standard places;*

- *each whole-place p_i of N satisfies $i \leq m$; and*
- *for every state $q \in Q$, it is the case that $\kappa(q) = (\mu(q)^{(1)}, \ldots, \mu(q)^{(m)})$, where μ is a bijection from Q to the reachable markings of N establishing the isomorphism between $\mathcal{T}$ and the concurrent reachability graph of N.*

Moreover, if the answer is positive, construct such a WPOASL*-net N.*

We will now describe how the above problem can be solved using results from the last section.

Since T is finite, there are only finitely many semantically distinct ways in which one can assign localities to the transitions in T. We can explore them all one-by-one, and below we assume that ℓ is a *fixed* locality mapping for T.

We next discuss the coefficients on the arcs adjacent to $p_1, \ldots, p_m$. Suppose first that $1 \leq i, j \leq m$ and $W(p_i, t) = v_1 \cdot p_1 + \cdots + v_m \cdot p_m + v_0$ in a net solving Problem 3, and μ is a corresponding bijection. We consider two cases:

- $\kappa(q)^{(j)} > 0$, for some $\delta(q, \alpha) = q'$ with $t \in \alpha$. Then, since α is enabled at $\mu(q)$, it must be the case that $\kappa(q)^{(i)} \geq v_j \cdot \kappa(q)^{(j)} \cdot \alpha(t)$, and so

  $$v_j \leq \min \left\{ \frac{\kappa(q)^{(i)}}{\kappa(q)^{(j)} \cdot \alpha(t)} \;\middle|\; \delta(q, \alpha) = q' \textit{ and } t \in \alpha \right\} .$$

  Hence, the range of possible values for v_j is finite.

- $\kappa(q)^{(j)} = 0$, for each $\delta(q, \alpha) = q'$ with $t \in \alpha$. Then we can assume $v_j = 1 + \max\{\kappa(q)^{(i)} \mid q \in Q\}$. This does not 'contradict' any of the arcs in $\mathcal{T}$ and, at the same time, ensures a maximal disabling power of coefficient v_j.

Suppose next that $i, j \leq m$ and $W(t, p_i) = v_1 \cdot p_1 + \cdots + v_m \cdot p_m + v_0$. We again consider two cases:

- $\kappa(q)^{(j)} > 0$, for some $\delta(q, \alpha) = q'$ with $t \in \alpha$. Then, since executing α at $\mu(q)$ leads to $\mu(q')$, it must be the case that $\kappa(q')^{(i)} \geq v_j \cdot \kappa(q)^{(j)} \cdot \alpha(t)$, and so

  $$v_j \leq \min \left\{ \frac{\kappa(q')^{(i)}}{\kappa(q)^{(j)} \cdot \alpha(t)} \;\middle|\; \delta(q, \alpha) = q' \textit{ and } t \in \alpha \right\} .$$

  Hence, the range of possible values for v_j is again finite.

- $\kappa(q)^{(j)} = 0$, for each $\delta(q,\alpha) = q'$ with $t \in \alpha$. Then we set $v_j = 0$.

As a result, we need to take into account only finitely many assignments of values to the whole-place coefficients of arcs between the transitions in T and $p_1, \ldots, p_m$. We can consider them one-by-one and, after filtering out those inconsistent with κ, carry out independent searches for a solution. Therefore, below we assume that such whole-place coefficients are *fixed*, and proceed further unless the net constructed so far is a solution (the initial marking is $\kappa(q_0)$).

Having fixed transition localities and whole-place coefficients involving the potential whole-places, we can proceed with the main part of the decision procedure, i.e., the construction of additional non-whole-places that can use $p_1, \ldots, p_m$ in their arc annotations. Moreover, we need to consider two cases as a new place may be either a standard or an a/sync place.

First, we derive the system $\mathcal{S}_\mathcal{T}$ as in (12) with $k = m+1$, implicitly assuming that the last m components correspond to the standard places $p_1, \ldots, p_m$, and the first component corresponds to a generic standard non-whole-place p being constructed (i.e., we assume that $k' = 0$ and $k'' = m+1$). We then replace by concrete values all those variables which are 'fixed' by the mapping κ, and the fact that p must be a non-whole-place. We finally delete all equations and inequalities which concern $p_1, \ldots, p_m$, i.e., those beginning with $\mathbf{x}_s^{(i)}$, for $2 \leq i \leq m+1$.

The homogeneous system $\mathcal{S}'_\mathcal{T}$ obtained in this way is *linear*.

Assume some arbitrary ordering of the variables of $\mathcal{S}'_\mathcal{T}$, so that its solutions can be treated as vectors. Using the results from [10], one can find a finite set $\mathbf{p}^1, \ldots, \mathbf{p}^r$ of non-negative integer solutions of $\mathcal{S}'_\mathcal{T}$ such that each non-negative integer solution $\mathbf{p}$ of $\mathcal{S}'_\mathcal{T}$ is a linear combination $\mathbf{p} = \sum_{l=1}^{r} a_l \cdot \mathbf{p}^l$ with non-negative rational coefficients a_l. For every non-negative integer solution $\mathbf{p}$ of $\mathcal{S}'_\mathcal{T}$, let $\psi(\mathbf{p})$ be a corresponding $\tau^{0/m+1}$-region.

We then repeat the above construction with one change, namely the first component is assumed to correspond to a generic a/sync non-whole-place v, leading to a linear system $\mathcal{S}''_\mathcal{T}$. In this case, we also can find a finite set $\mathbf{v}^1, \ldots, \mathbf{v}^u$ of non-negative integer solutions of $\mathcal{S}''_\mathcal{T}$ such that each non-negative integer solution $\mathbf{v}$ of $\mathcal{S}''_\mathcal{T}$ is a linear combination $\mathbf{v} = \sum_{j=1}^{u} b_j \cdot \mathbf{v}^j$ with non-negative rational coefficients b_j. For every non-negative integer solution $\mathbf{v}$ of $\mathcal{S}''_\mathcal{T}$, let $\psi(\mathbf{v})$ be a corresponding $\tau^{1/m}$-region.

The $\mathbf{p}^l$'s and $\mathbf{v}^j$'s are *fixed* and some of them will be turned into additional places if Problem 3 has a solution under the fixed localities and coefficients. This, in turn, is the case if we can verify Axioms 1 and 2.

Clearly, if $r = u = 0$ then the problem is not feasible for the current fixed parameters. Otherwise, we proceed as follows.

To check state separation (Axiom 1), let q_i and q_j be a pair of distinct states of $\mathcal{T}$. If $\kappa(q_i) \neq \kappa(q_j)$, then we are done. Suppose then that $\kappa(q_i) = \kappa(q_j)$, and ρ is a $\tau^{0/m+1}$-region separating q_i and q_j. Then there is a solution of $\mathcal{S}'_{\mathcal{T}}$, $\mathbf{p} = \sum_{l=1}^{r} a_l \cdot \mathbf{p}^l$, such that $\rho = \psi(\mathbf{p})$. This means that $\mathbf{p}$ assigns different values to q_i and q_j. Hence, there must be $\mathbf{p}^l$ which also assigns different values to q_i and q_j. Therefore, $\psi(\mathbf{p}^l)$ separates q_i and q_j. Similarly, if $\rho = \psi(\mathbf{v})$ is a $\tau^{1/m}$-region separating q_i and q_j, there must be $\mathbf{v}^j$ which also assigns different values to q_i and q_j. We therefore only need to check the $\mathbf{p}^l$'s and $\mathbf{v}^j$'s in order to establish the separation of q_i and q_j. If a suitable $\mathbf{p}^l$ or $\mathbf{v}^j$ is found, we add a non-whole standard place p corresponding to the first place of $\psi(\mathbf{p}^l)$, or we add a non-whole a/sync place v corresponding to the first place of $\psi(\mathbf{v}^j)$, to the net being constructed.

Checking forward closure (Axiom 2) is carried out for each state q_i, and considers steps $\alpha \in \langle T \rangle$ that are not enabled at q_i in $\mathcal{T}$. Moreover, one does not need to consider $\alpha \neq \mathbf{0}$ in the following cases:

- α is already disabled by the whole-places, or $|\alpha| > max$, where max is the maximum size of steps labelling arcs in $\mathcal{T}$. The latter exclusion is justified by the fact that one can always add to a net a standard non-whole-place which is connected with each transition by an incoming and outgoing arc of weight 1, and is initially marked with max tokens. Such a non-whole-place disables all steps with more than max transitions, and does not disable any other steps.

- There is a step β enabled at q_i such that $\ell(\beta) \subseteq \ell(\alpha)$ and $\alpha < \beta$.

In all other cases, α is not region enabled at q_i iff $\psi(\mathbf{p})$ disables α, for some solution $\mathbf{p} = \sum_{l=1}^{r} a_l \cdot \mathbf{p}^l$ of $\mathcal{S}'_{\mathcal{T}}$, or $\psi(\mathbf{v})$ disables α, for some solution $\mathbf{v} = \sum_{j=1}^{u} b_j \cdot \mathbf{v}^j$ of $\mathcal{S}''_{\mathcal{T}}$. Suppose that the former holds. Then, since the coefficients a_l are non-negative, α is not $\tau^{0/m+1}$-region enabled at q_i iff there is $\mathbf{p}^l$ such that $\psi(\mathbf{p}^l)$ disables α. We therefore only need to check the $\mathbf{p}^l$'s in order to establish the disabling of α. If a suitable $\mathbf{p}^l$ is found, we add a non-whole standard place p corresponding to the first place of $\psi(\mathbf{p}^l)$ to the net being constructed. In the latter case, we proceed similarly, and if a suitable $\mathbf{v}^j$ is found, we add a non-whole a/sync place v corresponding to the first place of $\psi(\mathbf{v}^j)$ to the net being constructed.

Finally, if one can validate all cases of state separation and forward closure, the resulting net is a solution to Problem 3, and otherwise there is no solution.

6 Conclusions

In this paper, we extended the results of [23] by adding a/sync places to the synthesis procedure for the nets with whole-place operations and localities (WPOL-nets). That new added feature could not have been simulated by any combination of existing net features of WPOL-nets and therefore considered redundant. In the paper, we showed that indeed it extends the expressive power of WPOL-nets.

Similarly, as with WPOL-nets, when designing the synthesis algorithm for WPOASL-nets, we had to resort to some additional assumptions. Firstly, the size of the clusters of related places needs to be known in advance. Secondly, the realisation that the obtained synthesis algorithm involves solving a quadratic system of equations and inequalities, led us to yet another simplification: the assumption that the information about the whole-places is also known upfront, which helped to reduce the quadratic system of equations and inequalities of the synthesis algorithm to a linear system of equations and inequalities.

As, in general, the algorithmic solutions to the synthesis problems for both WPOL-nets and WPOASL-nets involve solving a quadratic system of equations and inequalities, we feel that the following open questions are worth investigating:

- How to find interesting subclasses of these classes for which the synthesis procedure involves solving a linear system of equations and inequalities?

- How to find classes of Petri nets of the equivalent expressive power to these classes (using a/sync connection rather than a/sync places, perhaps) for which the synthesis procedure involves solving a linear system of equations and inequalities?

Among other possible directions for future work, we single out two challenges. The first one is the development of a synthesis approach for WPOAS-nets executed under more general step firing policies, e.g., those based on linear rewards of steps, where the reward for firing a single transition is either fixed or it depends on the current net marking [14]. The second task, more specific to k'/k''-WPOASL-nets, is to investigate the relationship between the locality mapping and the grouping of the places into clusters.

Acknowledgements

We are grateful to the anonymous reviewers for constructive comments and suggestions.

References

[1] P.A. Abdulla, G. Delzanno, and L. Van Begin. A language-based comparison of extensions of Petri nets with and without whole-place operations. In A.-H. Dediu, A.-M. Ionescu, and C. Martín-Vide, editors, *Language and Automata Theory and Applications, Third International Conference, LATA 2009, Tarragona, Spain, April 2-8, 2009. Proceedings*, volume 5457 of *Lecture Notes in Computer Science*, pages 71–82. Springer, 2009. doi:10.1007/978-3-642-00982-2_6.

[2] F. Arbab. Reo: a channel-based coordination model for component composition. *Mathematical Structures in Computer Science*, 14(3):329–366, 2004. doi:10.1017/S0960129504004153.

[3] E. Badouel, L. Bernardinello, and P. Darondeau. *Petri Net Synthesis*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2015. doi:10.1007/978-3-662-47967-4.

[4] E. Badouel and P. Darondeau. Theory of regions. In Wolfgang Reisig and Grzegorz Rozenberg, editors, *Lectures on Petri Nets I: Basic Models, Advances in Petri Nets, the volumes are based on the Advanced Course on Petri Nets, held in Dagstuhl, September 1996*, volume 1491 of *Lecture Notes in Computer Science*, pages 529–586. Springer, 1996. doi:10.1007/3-540-65306-6_22.

[5] R. Bergenthum, J. Desel, R. Lorenz, and S. Mauser. Synthesis of Petri nets from scenarios with VipTool. In K.M. van Hee and R. Valk, editors, *Applications and Theory of Petri Nets, 29th International Conference, PETRI NETS 2008, Xi'an, China, June 23-27, 2008. Proceedings*, volume 5062 of *Lecture Notes in Computer Science*, pages 388–398. Springer, 2008. doi:10.1007/978-3-540-68746-7_25.

[6] L. Bernardinello, G. De Michelis, K. Petruni, and S. Vigna. On the synchronic structure of transition systems. In Jörg Desel, editor, *Structures in Concurrency Theory*, Workshops in Computing, pages 69–84. Springer, 1995. doi:10.1007/978-1-4471-3078-9_5.

[7] R. Bruni and U. Montanari. Zero-safe nets: Comparing the collective and individual token approaches. *Information and Computation*, 156(1-2):46–89, 2000. doi:10.1006/inco.1999.2819.

[8] N. Busi and G. Michele Pinna. Synthesis of nets with inhibitor arcs. In A.W. Mazurkiewicz and J. Winkowski, editors, *CONCUR '97: Concurrency Theory, 8th International Conference, Warsaw, Poland, July 1-4, 1997, Proceedings*, volume 1243 of *Lecture Notes in Computer Science*, pages 151–165. Springer, 1997. doi:10.1007/3-540-63141-0_11.

[9] J. Carmona, J. Cortadella, and M. Kishinevsky. Genet: A tool for the synthesis and mining of Petri nets. In *Ninth International Conference on Application of Concurrency to System Design, ACSD 2009, Augsburg, Germany, 1-3 July 2009*, pages 181–185. IEEE Computer Society, 2009. doi:10.1109/ACSD.2009.6.

[10] N. Chernikova. Algorithm for finding a general formula for the non-negative solutions of a system of linear inequalities. *USSR Computational Mathematics and Mathematical Physics*, 5:228–233, 1965. doi:10.1016/0041-5553(65)90045-5.

[11] S. Christensen and N.D. Hansen. Coloured Petri nets extended with channels for synchronous communication. In R. Valette, editor, *Application and Theory of Petri Nets 1994, 15th International Conference, Zaragoza, Spain, June 20-24, 1994, Proceedings*, volume 815 of *Lecture Notes in Computer Science*, pages 159–178. Springer, 1994. doi:10.1007/3-540-58152-9_10.

[12] J. Cortadella, M. Kishinevsky, A. Kondratyev, L. Lavagno, and A. Yakovlev. *Logic Synthesis of Asynchronous Controllers and Interfaces*, volume 8 of *Springer Series in Advanced Microelectronics*. Springer, 2002. doi:10.1007/978-3-642-55989-1.

[13] P. Darondeau. Deriving unbounded Petri nets from formal languages. In D. Sangiorgi and R. de Simone, editors, *CONCUR '98: Concurrency Theory, 9th International Conference, Nice, France, September 8-11, 1998, Proceedings*, volume 1466 of *Lecture Notes in Computer Science*, pages 533–548. Springer, 1998. doi:10.1007/BFb0055646.

[14] P. Darondeau, M. Koutny, M. Pietkiewicz-Koutny, and A. Yakovlev. Synthesis of nets with step firing policies. *Fundamenta Informaticae*, 94(3-4):275–303, 2009. doi:10.3233/FI-2009-132.

[15] J. Desel and W. Reisig. The synthesis problem of Petri nets. *Acta Informatica*, 33(4):297–315, 1996. doi:10.1007/s002360050046.

[16] C. Dufourd, A. Finkel, and P. Schnoebelen. Reset nets between decidability and undecidability. In K.G. Larsen, S. Skyum, and G. Winskel, editors, *Automata, Languages and Programming, 25th International Colloquium, ICALP'98, Aalborg, Denmark, July 13-17, 1998, Proceedings*, volume 1443 of *Lecture Notes in Computer Science*, pages 103–115. Springer, 1998. doi:10.1007/BFb0055044.

[17] A. Ehrenfeucht and G. Rozenberg. Partial (set) 2-structures. part I: basic notions and the representation problem. part II: state spaces of concurrent system. *Acta Informatica*, 27(4):315–368, 1990. doi:10.1007/BF00264611.

[18] A. Finkel, P. McKenzie, and C. Picaronny. A well-structured framework for analysing Petri net extensions. *Information and Computation*, 195(1-2):1–29, 2004. doi:10.1016/j.ic.2004.01.005.

[19] P. W. Hoogers, H. C. M. Kleijn, and P. S. Thiagarajan. A trace semantics for Petri nets. *Information and Computation*, 117(1):98–114, 1995. doi:10.1006/inco.1995.1032.

[20] J. Kleijn and M. Koutny. Causality in structured occurrence nets. In C.B. Jones and J.L. Lloyd, editors, *Dependable and Historic Computing - Essays Dedicated to Brian Randell on the Occasion of His 75th Birthday*, volume 6875 of *Lecture Notes in Computer Science*, pages 283–297. Springer, 2011. doi:10.1007/978-3-642-24541-1_22.

[21] J. Kleijn and M. Koutny. Localities in systems with a/sync communication. *Theoretical Computer Science*, 429:185–192, 2012. doi:10.1016/j.tcs.2011.12.038.

[22] J. Kleijn, M. Koutny, and M. Pietkiewicz-Koutny. Regions of Petri nets with a/sync connections. *Theoretical Computer Science*, 454:189–198, 2012. doi:10.1016/j.tcs.2012.04.016.

[23] J. Kleijn, M. Koutny, and M. Pietkiewicz-Koutny. Synthesis of Petri nets with whole-place operations and localities. In A. Sampaio and F. Wang, editors, *Theoretical Aspects of Computing - ICTAC 2016 - 13th International Colloquium, Taipei, Taiwan, ROC, October 24-31,*

2016, Proceedings, volume 9965 of *Lecture Notes in Computer Science*, pages 103–120, 2016. doi:10.1007/978-3-319-46750-4_7.

[24] J. Kleijn, M. Koutny, M. Pietkiewicz-Koutny, and G. Rozenberg. Applying regions. *Theoretical Computer Science*, 658:205–215, 2017. doi:10.1016/j.tcs.2016.01.040.

[25] M. Koutny and M. Pietkiewicz-Koutny. Synthesis of Petri nets with localities. *Scientific Annals of Computer Science*, 19:1–23, 2009. URL: http://www.info.uaic.ro/bin/Annals/Article?v=XIX&a=0.

[26] M. Mukund. Petri nets and step transition systems. *International Journal of Foundations of Computer Science*, 3(4):443–478, 1992. doi:10.1142/S0129054192000231.

[27] M. Nielsen, G. Rozenberg, and P. S. Thiagarajan. Elementary transition systems. *Theoretical Computer Science*, 96(1):3–33, 1992. doi:10.1016/0304-3975(92)90180-N.

[28] M. Pietkiewicz-Koutny. Transition systems of elementary net systems with inhibitor arcs. In P. Azéma and G. Balbo, editors, *Application and Theory of Petri Nets 1997, 18th International Conference, ICATPN '97, Toulouse, France, June 23-27, 1997, Proceedings*, volume 1248 of *Lecture Notes in Computer Science*, pages 310–327. Springer, 1997. doi:10.1007/3-540-63139-9_43.

[29] V. Schmitt. Flip-flop nets. In C. Puech and R. Reischuk, editors, *STACS 96, 13th Annual Symposium on Theoretical Aspects of Computer Science, Grenoble, France, February 22-24, 1996, Proceedings*, volume 1046 of *Lecture Notes in Computer Science*, pages 517–528. Springer, 1996. doi:10.1007/3-540-60922-9_42.

[30] M. Solé and J. Carmona. Rbminer: A tool for discovering Petri nets from transition systems. In A. Bouajjani and W.-N. Chin, editors, *Automated Technology for Verification and Analysis - 8th International Symposium, ATVA 2010, Singapore, September 21-24, 2010. Proceedings*, volume 6252 of *Lecture Notes in Computer Science*, pages 396–402. Springer, 2010. doi:10.1007/978-3-642-15643-4_33.

[31] J. M. E. M van der Werf, B.F. van Dongen, C.A.J. Hurkens, and A. Serebrenik. Process discovery using integer linear programming.

In K.M. van Hee and R. Valk, editors, *Applications and Theory of Petri Nets, 29th International Conference, PETRI NETS 2008, Xi'an, China, June 23-27, 2008. Proceedings*, volume 5062 of *Lecture Notes in Computer Science*, pages 368–387. Springer, 2008. doi:10.1007/978-3-540-68746-7_24.