Synthesis of Petri Nets with Whole-place Operations and Localities

Jetty Kleijn[1], Maciej Koutny[2], and Marta Pietkiewicz-Koutny[2]

[1] LIACS, Leiden University, PO Box 9512, 2300 RA, The Netherlands
[2] School of Computing Science, Newcastle University, NE1 7RU, UK

Abstract. 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, systems are modelled by Petri nets and the behavioural specifications are provided in the form of step transition systems, where arcs are labelled by multisets of executed actions. We focus on the problem of synthesising Petri nets with whole-place operations and localities (WPOL-nets), which are a class of Petri nets powerful enough to express a wide range of system behaviours, including inhibition of actions, resetting of local states, and locally maximal executions.
The synthesis problem was solved for several specific net classes and later a general approach was developed within the framework of τ-nets. In this paper, we follow the synthesis techniques introduced for τ-nets that are based on the notion of a region of a transition system, which we suitably adapt to work for WPOL-nets.
Keywords: concurrency, theory of regions, transition system, synthesis problem, Petri net, step semantics, locality, whole-place operations net

1 Introduction

The starting point of a scientific investigation that aims at describing and analysing a dynamic system or an experiment is very often a record of a series of observations as depicted, for example, by a graph like that in Figure 1(a). The observation graph captures important information about the system, e.g., the fact that it can be in three different states in which the quantity of some crucial resource ξ has been measured to be equal to 2, 1, or 0 units. Other relevant information is that the moves between these three states result from executions of three distinct actions: A, B, and C. Moreover, these actions can sometimes be performed simultaneously (for example, B and C), as well as individually (for example, A).

Suppose now that one would like to construct a formal system model matching the observations depicted by the graph in Figure 1(a). Such a model could then be used for further analyses of the real-life system using suitable techniques and tools. Since the observation graph conveys a mix of state and action information, a natural way of proceeding might be to develop a Petri net model, as Petri nets deal explicitly with both state and action based issues and are

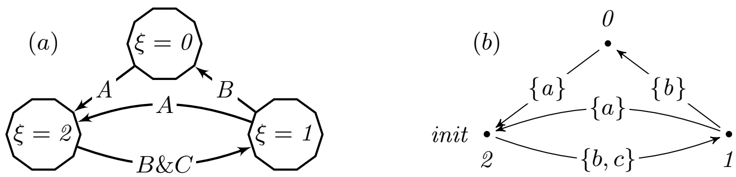


Fig. 1. A record of real-life observations of a system (a); and its step transition system representation (b).

able to express different relationships between actions and/or states: causality, simultaneity, and competing for resources.

To construct a Petri net model for the observation graph in Figure 1(a), we first convert it into a slightly more formal representation in terms of a transition system as shown in Figure 1(b) where the actions A, B, and C are respectively represented by net-transitions a, b, and c, the arcs are labelled by sets of executed net-transitions, and the nodes are labelled by integers representing the volume of the crucial resource ξ. Moreover, one node is designated as the initial state. A key idea is that the quantity of the crucial resource can be represented by a specific place (local state) p_1 in a Petri net model to be constructed, and the overall aim of the synthesis process is to build a Petri net whose reachability graph is isomorphic to the graph in Figure 1(b), and the tokens assigned to place p_1 in different markings (global states) are as specified by the integers labelling the nodes.

It is natural to aim at a model as simple as possible, and so one might attempt to synthesise from Figure 1(b) a Place/Transition net (PT-net). since these are the simplest Petri net model allowing one to represent integer-valued quantities. However, such an attempt would fail, as the transition system in Figure 1(b) does not represent the behaviour of any PT-net. The first reason is that to be so it should have contained two more arcs, labelled by $\{b\}$ and $\{c\}$, outgoing from the initial state. Another problem is that it contains two $\{a\}$-labelled arcs coming to the same (initial) state from two distinct states. Since PT-nets are backward-deterministic, they would never produce this kind of behaviour.

The Petri net model we will use to construct a suitable formal model for behavioural descriptions like that in Figure 1(b), will be nets with whole-place operations (i.e., the weight of an arc may depend on the current total number of tokens in a subset of places) and localities (WPOL-nets). Grouping net-transitions in different localities and introducing an execution semantics that allows only maximal multisets of enabled net-transitions to 'fire' within a given locality helps to overcome the first problem mentioned above. Allowing the weights of connections between places and transitions to depend on the current marking and, in consequence, introducing *whole-place operations* addresses the second problem concerning the backward non-deterministic behaviour.

The synthesis of a WPOL-net from a transition system specification will be based on the notion of a *region* of a transition system [10, 3, 2] suitably adapted to WPOL-nets and their locally maximal execution semantics, a special kind of *step firing policy* (see [13, 7]). This paper shows for the first time how to synthesise a net, whose execution depends dynamically on the current marking (distribution of 'resources'), under an additional constraint in the form of a step firing policy.

Synthesising systems from behavioural specifications is an attractive way of constructing implementations which are correct-by-design and thus requiring no costly validation efforts. The synthesis problem was solved for many specific classes of nets, e.g., [15, 14, 4, 17, 8, 5, 16]. Later, a general approach was developed within the framework of τ-nets that takes a *net-type* as a parameter [3]. In this paper, we focus on the problem of synthesising WPOL-nets from behavioural specifications provided by step transition systems. WPOL-nets are nets with whole-place operations (WPO-nets) extended with transition localities. WPO-nets in turn are derived from transfer/reset nets [9] and *affine* nets [11], extending PT-nets with whole-place operations [1]. A solution to the synthesis problem for WPO-nets was outlined in [12], and we use some of the ideas introduced there in this paper, at the same time dealing with the additional constraint of the locally maximal execution semantics.

The paper is organised as follows. The next section recalls some basic notions concerning transition systems, PT-nets, and τ-nets. Section 3 introduces WPO-nets and WPOL-nets, and Sections 4 and 5 present a solution to the synthesis problem for WPOL-nets, treating them as a special kind of τ-nets. The paper ends with a brief conclusion that outlines some directions for future work.

2 Preliminaries

An *abelian monoid* is a set $\mathbb{S}$ with a commutative and associative binary operation $+$, and 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 $\mathbf{0} = 0 \cdot s$. Two examples of abelian monoids are: (i) $\mathbb{S}_{PT} = \mathbb{N} \times \mathbb{N}$, where $\mathbb{N}$ are all non-negative integers, with the pointwise arithmetic addition operation and $\mathbf{0} = (0, 0)$ and (ii) the free abelian monoid $\langle T \rangle$ generated by a set T. $\mathbb{S}_{PT}$ will represent (weighted) arcs between places and transitions in PT-nets, whereas $\langle T \rangle$ will represent *steps* (multisets of transitions) of nets with transition set T. The free abelian monoid $\langle T \rangle$ can be seen as the set of all finite multisets over T, e.g., $aab = aba = baa = \{a, a, b\}$. 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 $\alpha = \sum_{t \in T} \alpha(t) \cdot t$. Then $t \in \alpha$ whenever $\alpha(t) > 0$, and $\alpha \leq \beta$ whenever $\alpha(t) \leq \beta(t)$ for all $t \in T$. The size of α is $|\alpha| = \sum_{t \in T} \alpha(t)$.

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 *transition function*[3] $\delta : Q \times \mathbb{S} \to Q$ 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

[3] Transition functions are not related to (Petri) net-transitions.

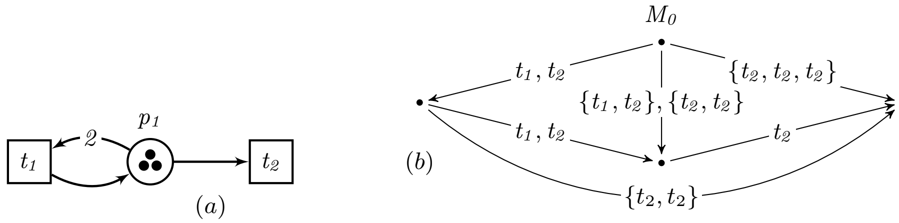


Fig. 2. A PT-net (a); and its concurrent reachability graph (b).

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, **0**-labelled arcs are omitted and singleton steps 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 with $f(q_0) = q_0'$ and $\delta(q, \alpha) = q' \Leftrightarrow \delta'(f(q), \alpha) = f(q')$, for all $q, q' \in Q$ 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 $\mathbb{N}$. We use the standard conventions concerning the graphical representation of PT-nets, as illustrated in Figure 2(a).

For all $p \in P$ and $\alpha \in \langle T \rangle$, we denote $W(p, \alpha) = \sum_{t \in T} \alpha(t) \cdot W(p, t)$ and $W(\alpha, p) = \sum_{t \in T} \alpha(t) \cdot W(t, p)$. 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 $M'(p) = M(p) - W(p, \alpha) + W(\alpha, p)$. 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

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

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).

Petri nets defined by net-types. A net-type $\tau = \langle \mathcal{Q}, \mathbb{S}, \Delta \rangle$ is a parameter in the definition of τ*-nets*. It specifies the values (markings) that can be stored in places ($\mathcal{Q}$), the operations and tests (inscriptions on the arcs) 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).

A τ*-net* is a tuple $N = \langle P, T, F, M_0 \rangle$, where P and T are respectively disjoint sets of places and transitions, $F : (P \times T) \to \mathbb{S}$ is a *flow mapping*, and M_0 is an *initial marking* belonging to the set of *markings* defined as 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 denote $F(p, \alpha) = \sum_{t \in T} \alpha(t) \cdot F(p, t)$. Then a 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 in [3, 7], it is possible to encode a 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 an infinite net-type over $\mathbb{S}_{PT}$ defined earlier, with Δ_{PT} given by $\Delta_{PT}(n, (i, o)) = n - i + o$ provided that $n \geq i$ (see Figure 5(a)).

3 Nets with whole-place operations

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.

A *net with whole-place operations* (WPO-net) is a tuple $N = \langle P, T, W, \mathbf{m}_0 \rangle$, where $P = \{p_1, \ldots, p_n\}$ is a finite set of ordered *places*, T is a finite set of *transitions* disjoint with P, $W : (P \times T) \cup (T \times P) \to \mathbb{N}^{n+1}$ is a *whole-place* weight function, and $\mathbf{m}_0$ is an *initial marking* belonging to the set $\mathbb{N}^n$ of *markings*.

For $p \in P$ and $\alpha \in \langle T \rangle$, $W(p, \alpha) = \sum_{t \in T} \alpha(t) \cdot W(p, t)$ and $W(\alpha, p) = \sum_{t \in T} \alpha(t) \cdot W(t, p)$. Then α is *enabled* at a marking $\mathbf{m}$ if, for every $p \in P$,

$$\mathbf{m}(p) \geq (\mathbf{m}, 1) \otimes W(p, \alpha) . \quad (1)$$

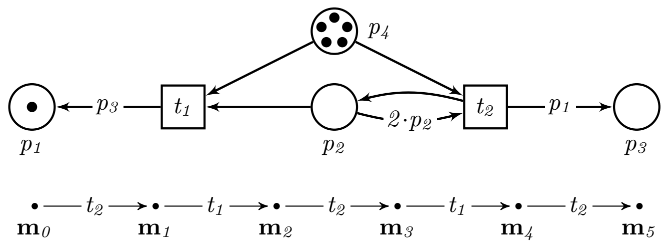
We denote this by $\alpha \in enb_N(\mathbf{m})$. An enabled α can be *fired* leading to a new marking such that, for every $p \in P$,

$$\mathbf{m}'(p) = \mathbf{m}(p) + (\mathbf{m}, 1) \otimes (W(\alpha, p) - W(p, \alpha)) . \quad (2)$$

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

(a)

(b)



(c)

$$\begin{bmatrix} W(p_1,t_1) \\ W(p_2,t_1) \\ W(p_3,t_1) \\ W(p_4,t_1) \end{bmatrix} = \begin{bmatrix} 0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,1 \\ 0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,1 \end{bmatrix} \qquad \begin{bmatrix} W(t_1,p_1) \\ W(t_1,p_2) \\ W(t_1,p_3) \\ W(t_1,p_4) \end{bmatrix} = \begin{bmatrix} 0\,0\,1\,0\,0 \\ 0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0 \end{bmatrix}$$

$$\begin{bmatrix} W(p_1,t_2) \\ W(p_2,t_2) \\ W(p_3,t_2) \\ W(p_4,t_2) \end{bmatrix} = \begin{bmatrix} 0\,0\,0\,0\,0 \\ 0\,2\,0\,0\,0 \\ 0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,1 \end{bmatrix} \qquad \begin{bmatrix} W(t_2,p_1) \\ W(t_2,p_2) \\ W(t_2,p_3) \\ W(t_2,p_4) \end{bmatrix} = \begin{bmatrix} 0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,1 \\ 1\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0 \end{bmatrix}$$

Fig. 3. A WPO-net generating the first six Fibonacci numbers (a); its concurrent reachability graph (b); and the weight function (c).

It is convenient to specify the weight function using arc annotations which are linear expressions involving the p_i's. For example, if $n = 3$ then $W(p_3, t) = (3, 0, 1, 5)$ can be written down as $3 \cdot p_1 + p_3 + 5$. A place p_j $(1 \leq j \leq n)$ is a *whole-place* if $W(p, t)^{(j)} > 0$ or $W(t, p)^{(j)} > 0$, for some $p \in P$ and $t \in T$. In such a case we also write $p_j \rightsquigarrow p$. Note that it may happen that $p = p_j$; for example, if $W(p_1, t) = p_1$. This is useful, e.g., for simulating inhibitor arcs (see $W(p_2, t_2) = 2 \cdot p_2$ in Figure 3(a)). In diagrams, arcs with '0' annotation are dropped, and '1' annotations are not shown.

Figure 3 shows a modified example, taken from [9], of a WPO-net for the generation of the first six Fibonacci numbers. Its markings are as follows: $\mathbf{m}_0 = (\mathbf{1}, 0, 0, 5)$, $\mathbf{m}_1 = (1, 1, \mathbf{1}, 4)$, $\mathbf{m}_2 = (\mathbf{2}, 0, 1, 3)$, $\mathbf{m}_3 = (2, 1, \mathbf{3}, 2)$, $\mathbf{m}_4 = (\mathbf{5}, 0, 3, 1)$, and $\mathbf{m}_5 = (5, 1, \mathbf{8}, 0)$. Hence, the markings of places p_1 and p_3, in alternation, represent the first six Fibonacci numbers (written above in bold). As $W(t_1, p_1)^{(3)} = 1 > 0$, $W(t_2, p_3)^{(1)} = 1 > 0$ and $W(p_2, t_2)^{(2)} = 2 > 0$, the net has three whole-places, p_1, p_2 and p_3 with $p_3 \rightsquigarrow p_1$, $p_1 \rightsquigarrow p_3$ and $p_2 \rightsquigarrow p_2$. Moreover, p_4, acting as a simple counting place, is a *non-whole-place*.

A WPO-*net with localities* (or WPOL-net) is a tuple $N = \langle P, T, W, \mathbf{m}_0, \ell \rangle$ such that $N' = \langle P, T, W, \mathbf{m}_0 \rangle$ is a WPO-net, and $\ell : T \to \{1, 2, \ldots, l\}$, where $l \geq 1$, is the *locality mapping* of N and $\{1, 2, \ldots, l\}$ are the *localities* of N. In diagrams, nodes representing transitions assigned the same locality are shaded in the same way, as illustratated in Figure 4(a) for transitions b and c. Finally N inherits the notations introduced for N'.

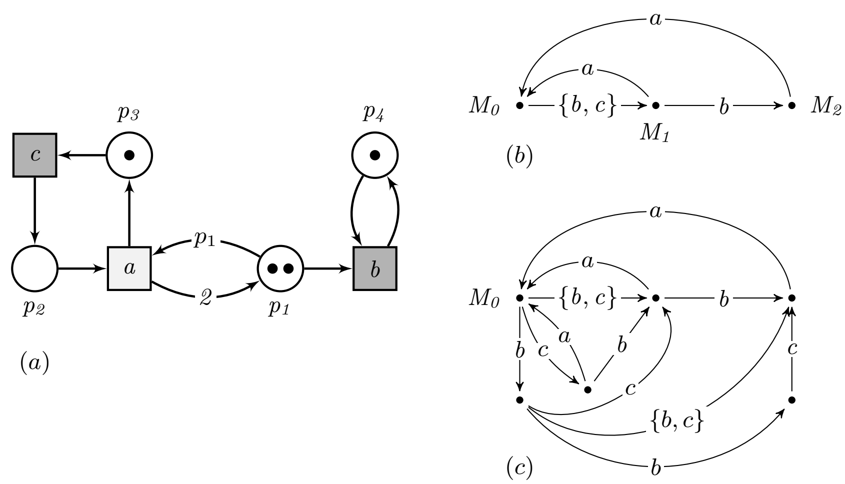


Fig. 4. A WPOL-net (a); its concurrent reachability graph (b); and the concurrent reachability graph of the underlying WPO-net (c).

WPOL-nets are executed under the *locally maximal* step firing policy. A step $\alpha \in \langle T \rangle$ is *resource enabled* at a marking $\mathbf{m}$ if, for every $p \in P$, the inequality (1) is satisfied. Such a step is then *control enabled* if there is no $t \in T$ such that there exists a transition $t' \in \alpha$ with $\ell(t) = \ell(t')$ and the step $t + \alpha$ is resource enabled at $\mathbf{m}$. A control enabled step α can be then fired leading to the marking $\mathbf{m}'$, for every $p \in P$ given by the formula (2).

In general (see [7]), a step firing policy is given by a *control disabled steps* mapping $cds : 2^{\langle T \rangle} \to 2^{\langle T \rangle \setminus \{\mathbf{0}\}}$ that, for a set of resource enabled steps at some reachable marking, returns the set of steps disabled by this policy at that marking. For the locally maximal step firing policy this mapping will be denoted by cds_{lmax} and we will identify this policy with its cds_{lmax} mapping:[4]

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

Step firing policies are a means of controlling and constraining the potentially huge number of execution paths generated by a concurrent system. The concurrent reachability graph of a net executed under a step firing policy contains only the control enabled steps (see Figure 4(b, c)).

4 Synthesis of WPOL-nets

The net synthesis problem we consider here aims to devise a procedure which constructs a WPOL-net with a concurrent reachability graph (reflecting the use

[4] Control disabled steps mappings are defined in [7] in the context of τ-nets, and this is how cds_{lmax} will be used in Section 4.

of 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 from $\mathcal{T}$ and their connections with transitions of T from τ. All necessary information needed to construct a place in a net that realises $\mathcal{T}$ is encapsulated in the notion of a region, which depends on 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, discovering places for the net to be constructed is more complicated than in previously considered synthesis problems (except for [12]), as the 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 WPOL-nets as nets containing clusters of — at most k — related places (k-WPOL-nets) and express them as τ-nets, so that we can synthesise them as τ-nets, using the general framework of net synthesis theory.

4.1 k-WPOL-nets and their net-type

WPOL-nets allow arc weights to depend on the current marking of all places. This may be too generous, e.g., in the case of systems where places are distributed among different neighbourhoods, forming the scopes where their markings can influence the token game. One way of capturing this is to restrict the number of places which can influence arc weights.

A *k-restricted* WPOL-*net* (k-WPOL-net, $k \geq 1$) is a WPOL-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 places and, for all $p \in P_i$ and $p' \notin P_i$, $p \not\rightharpoonup p' \not\rightharpoonup p$. In other words, the places can be partitioned into clusters of bounded size so that there is no exchange of whole-place marking information between different clusters.

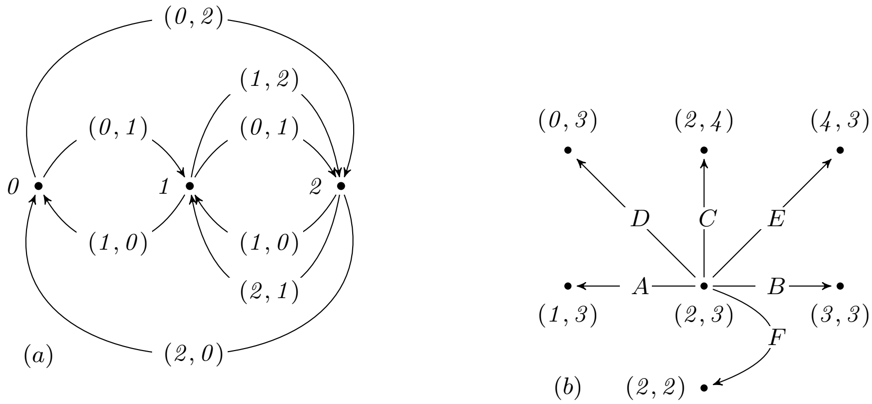
Although k-WPOL-nets (as well as WPOL-nets) are not τ-nets in the sense of the original definition as the change of a marking of a place does not only depend on its marking and the connections to the transitions, they still fit the ideas behind the definition of τ-nets. All we need to do is to define a suitably extended net-type capturing the behaviour of sets of k places rather than the behaviour of single places. More precisely, for each $k \geq 1$, the k-WPOL-*net-type* is a transition system [5]:

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

where

$$\Delta^k : \mathbb{N}^k \times ((\mathbb{N}^{k+1})^k \times (\mathbb{N}^{k+1})^k) \rightarrow \mathbb{N}^k$$

[5] As will be explained later, the same net-type can be defined for a given kind of nets to be executed without any specific policy or with some policy. Therefore, we can re-use here the τ^k_{wpo} net-type introduced in [12], which coincides with τ^k.



$$A = \left(\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \right) \quad B = \left(\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \right) \quad C = \left(\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \right)$$

$$D = \left(\begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \right) \quad E = \left(\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \right) \quad F = \left(\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \right)$$

Fig. 5. Fragments of two infinite net-types: τ_{PT} (a); and τ^2 (b).

is a partial function such that $\Delta^k(\mathbf{x}, (X, Y))$ is defined if $\mathbf{x} \geq (\mathbf{x}, 1) \otimes X$ and, if that is the case,

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

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

Having defined a net-type τ^k, a τ^k*-net* is 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 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 $P_i \in \mathcal{P}$, $F(P_i, \alpha) \in enb_{\tau^k}(M(P_i))$. 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 $M'(P_i) = \Delta^k(M(P_i), F(P_i, \alpha))$. 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 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,

$$enb_{N,cds_{lmax}}(M) = enb_N(M) \setminus cds_{lmax}(enb_N(M)) \tag{3}$$

is the set of steps enabled 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 $CRG(N)$.

4.2 Synthesising k-WPOL-nets as τ^k-nets

First we need to express a k-WPOL-net $N = \langle P, T, W, \mathbf{m}_0, \ell \rangle$, with set of places $P = \{p_1, \ldots, p_n\}$ and clusters $P_1, \ldots, P_r$, as a τ^k-net. Suppose that each set P_i in the partition has exactly k places. (If any of the sets P_i has $m < k$ places, we can always add to it $k - m$ fresh dummy empty places disconnected from the original transitions and places.) 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 $t \in T$: (i) $F(P_i, t) = (X, Y)$ 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 $W(\cdot, \cdot)$ are column vectors, by deleting the rows and columns corresponding to the places in $P \setminus P_i$; and (ii) $M_0(P_i)$ 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 τ^k-net into an equivalent k-WPOL-net, and trivially each WPOL-net is a $|P|$-WPOL-net.

We can turn the WPO-net of Figure 3(a) into a WPOL-net with locality mapping ℓ such that $\ell(t_1) = 1$ and $\ell(t_2) = 2$. The result can be represented as a τ^2-net $\widehat{N} = \langle \{P_1, P_2\}, \{t_1, t_2\}, F, M_0, \ell \rangle$, $P_1 = \{p_1, p_3\}$ and $P_2 = \{p_2, p_4\}$, $M_0(P_1) = (1, 0)$, $M_0(P_2) = (0, 5)$ and:

$$F(P_1, t_1) = \left(\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \end{bmatrix} \right) \quad F(P_1, t_2) = \left(\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \right)$$

$$F(P_2, t_1) = \left(\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \right) \quad F(P_2, t_2) = \left(\begin{bmatrix} 2 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \right)$$

The above discussion implies that k-WPOL-net synthesis can be reduced to the following two problems of τ^k-net synthesis.

Problem 1 (feasibility) *Let $\mathcal{T} = \langle Q, \langle T \rangle, \delta, q_0 \rangle$ be a bounded step transition system, k be a positive integer, and ℓ be a locality mapping for T.*
Provide necessary and sufficient conditions for $\mathcal{T}$ to be realised by some τ^k-net, $\widehat{N}$, executed under the cds_{lmax} policy defined by ℓ ($\mathcal{T} \cong CRG_{cds_{lmax}}(\widehat{N})$).

Problem 2 (effective construction) *Let $\mathcal{T} = \langle Q, \langle T\rangle, \delta, q_0\rangle$ be a finite step transition system, k be a positive integer, and ℓ be a locality mapping for T. Decide whether there is a finite τ^k-net realising $\mathcal{T}$ when executed under the cds_{lmax} policy defined by ℓ. Moreover, if the answer is positive construct such a τ^k-net.*

To address Problem 1, we define a τ^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 $\alpha \in enb_{\mathcal{T}}(q)$,

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

where $\eta(\alpha) = \sum_{t\in T} \alpha(t) \cdot \eta(t)$. Moreover, for every state q of Q, we denote by $enb_{\mathcal{T},\tau^k}(q)$ the set of all steps α such that $\eta(\alpha) \in enb_{\tau^k}(\sigma(q))$, for all τ^k-regions $\langle\sigma, \eta\rangle$ of $\mathcal{T}$. Hence for every state q of $\mathcal{T}$, we have

$$enb_{\mathcal{T}}(q) \subseteq enb_{\mathcal{T},\tau^k}(q). \tag{4}$$

In the context of the synthesis problem, a τ^k-region represents a cluster of places whose local states (in τ^k) are consistent with the global states (in $\mathcal{T}$). Then, to deliver a realisation of $\mathcal{T}$, one needs to find *enough* τ^k-regions to construct a τ^k-net $\widehat{N}$ satisfying $\mathcal{T} \cong CRG_{cds_{lmax}}(\widehat{N})$. The need for the existence of such τ^k-regions is dictated by the following two *regional axioms*:

Axiom 1 (state separation) *For any pair of states $q \neq r$ of $\mathcal{T}$, there is a τ^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 $\mathcal{T}$, $enb_{\mathcal{T}}(q) = enb_{\mathcal{T},\tau^k}(q) \setminus cds_{lmax}(enb_{\mathcal{T},\tau^k}(q))$.*

The above axioms provide a full characterisation of realisable transition systems. 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:

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

Note that when a τ^k-net under cds_{lmax} realises $\mathcal{T}$, every cluster of places of the net still determines a corresponding τ^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 WPO-net (rather than a WPOL-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}.

Proposition 1. *Let X be a finite set of resource enabled steps at some reachable marking of some τ^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 $\mathcal{T} \cong CRG_{cds_{lmax}}(\widehat{N})$ for a τ^k-net $\widehat{N} = \langle \mathcal{P}, T, F, M_0, \ell \rangle$. Then, for each cluster $P_i \in \mathcal{P}$ ($i = 1, \ldots, r$), there is exactly one τ^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 in this paper are deterministic. Observe that both δ and Δ^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, $\sigma(q_0)$ and $\eta : \langle T \rangle \to (\mathbb{N}^{k+1})^k \times (\mathbb{N}^{k+1})^k$ determine at most one map $\sigma : Q \to \mathbb{N}^k$ such that $\Delta^k(\sigma(q), \eta(\alpha)) = \sigma(\delta(q, \alpha))$ whenever $\alpha \in enb_{\mathcal{T}}(q)$, and therefore they determine at most one τ^k-region of $\mathcal{T}$.

We now define the map 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 $F(P_i, \alpha) \in enb_{\tau^k}(f(q)(P_i))$ and the marking of P_i after α is fired is $f(\delta(q, \alpha))(P_i) = \Delta^k(f(q)(P_i), F(P_i, \alpha))$. Therefore, we have, for σ defined as above and $\eta(\alpha) = F(P_i, \alpha)$ (as stated in the assumptions), that $\eta(\alpha) \in enb_{\tau^k}(\sigma(q))$ and $\sigma(\delta(q, \alpha)) = \Delta^k(\sigma(q), \eta(\alpha))$. Hence $\langle \sigma, \eta \rangle$, with σ defined as above, is a τ^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 τ^k-net ($k \geq 1$) under cds_{lmax} iff *Axioms 1 and 2 are satisfied.*

Proof. ($\Longrightarrow$) $\mathcal{T}$ can be realised by the τ^k-net $\widehat{N}$ under cds_{lmax}. That means that $\mathcal{T} \cong CRG_{cds_{lmax}}(\widehat{N})$. 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 prove that:

$$enb_{\mathcal{T},\tau^k}(q) \subseteq enb_{\widehat{N}}(f(q)). \tag{5}$$

Let $\alpha \notin enb_{\widehat{N}}(f(q))$. Then there is a cluster $P_i \in \mathcal{P}$ ($1 \leq i \leq r$) in $\widehat{N}$ such that $F(P_i, \alpha) \notin enb_{\tau^k}(f(q)(P_i))$. Let $\langle \sigma, \eta \rangle$ be the τ^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, $\eta(\alpha) \notin enb_{\tau^k}(\sigma(q))$ and so $\alpha \notin enb_{\mathcal{T},\tau^k}(q)$.

To show Axiom 1 let $q \neq r$ in Q. As $\mathcal{T} \cong CRG_{cds_{lmax}}(\widehat{N})$, 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 τ^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 $q \in Q$:

$$\alpha \notin enb_{\mathcal{T}}(q) \Longrightarrow \alpha \notin enb_{\mathcal{T},\tau^k}(q) \setminus cds_{lmax}(enb_{\mathcal{T},\tau^k}(q)). \tag{6}$$

Let $q \in Q$ and $\alpha \notin enb_{\mathcal{T}}(q)$. From (3) and $\mathcal{T} \cong CRG_{cds_{lmax}}(\widehat{N})$, either:

(i) $\alpha \notin enb_{\widehat{N}}(f(q))$ or

(ii) $\alpha \in enb_{\widehat{N}}(f(q)) \cap cds_{lmax}(enb_{\widehat{N}}(f(q)))$.

If (i) holds then, by (5), we have $\alpha \notin enb_{\mathcal{T},\tau^k}(q)$ and so (6) holds. In (ii) two cases are possible. If $\alpha \notin enb_{\mathcal{T},\tau^k}(q)$ we have (6); otherwise $\alpha \in enb_{\mathcal{T},\tau^k}(q)$ and we set the following: $X = enb_{\widehat{N}}(f(q))$ and $Y = enb_{\mathcal{T},\tau^k}(q)$. By (5), we have $Y \subseteq X$. Moreover, by (3,4) and $\mathcal{T} \cong CRG_{cds_{lmax}}(\widehat{N})$, we have $X \setminus cds_{lmax}(X) \subseteq Y$. Hence, by Proposition 1 and the fact that $\mathcal{T}$ is bounded, $\alpha \in cds_{lmax}(X) \cap Y \subseteq cds_{lmax}(enb_{\mathcal{T},\tau^k}(q))$, and so (6) holds.

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

$$enb_{\mathcal{T}}(q) \subseteq enb_{\mathcal{T},\tau^k}(q) \setminus cds_{lmax}(enb_{\mathcal{T},\tau^k}(q)). \tag{7}$$

By isomorphism $\mathcal{T} \cong CRG_{cds_{lmax}}(\widehat{N})$ and (3), we have $enb_{\mathcal{T}}(q) = enb_{\widehat{N}}(f(q)) \setminus cds_{lmax}(enb_{\widehat{N}}(f(q)))$. Hence $enb_{\mathcal{T}}(q) \cap cds_{lmax}(enb_{\widehat{N}}(f(q))) = \varnothing$. Thus, by (5) and $cds_{lmax}(Y) \subseteq cds_{lmax}(X)$ (for $Y \subseteq X$), $enb_{\mathcal{T}}(q) \cap cds_{lmax}(enb_{\mathcal{T},\tau^k}(q)) = \varnothing$. Moreover, by (4), which always holds, we can conclude that (7) holds.

($\Longleftarrow$) Let $\mathcal{R}$ be the set of all τ^k-regions of $\mathcal{T}$. Let $\widehat{N} = \langle \mathcal{P}, T, F, M_0, \ell \rangle$ be a τ^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 $P_i = \langle \sigma, \eta \rangle \in \mathcal{P}$ 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 reachable 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 an 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 τ^k-region of $\mathcal{T}$, $\sigma(\delta(q,\alpha)) = \Delta^k(\sigma(q),\eta(\alpha))$. 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 $\sigma(\delta(q,\alpha)) = \Delta^k(M(P_i),F(P_i,\alpha))$. From $M \xrightarrow{\alpha} M'$, we have $M'(P_i) = \Delta^k(M(P_i),F(P_i,\alpha))$. 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 τ^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:

$$q \sim M \Longrightarrow enb_{\mathcal{T},\tau^k}(q) = enb_{\widehat{N}}(M). \tag{8}$$

Let $\alpha \in enb_{\mathcal{T},\tau^k}(q)$. This means that $\eta(\alpha) \in enb_{\tau^k}(\sigma(q))$, for all τ^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, $F(P_i,\alpha) \in enb_{\tau^k}(M(P_i))$, 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 $F(P_i,\alpha) \in enb_{\tau^k}(M(P_i))$, 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 $\eta(\alpha) \in enb_{\tau^k}(M(P_i))$. For every cluster $P_i = \langle\sigma,\eta\rangle$ of $\widehat{N}$, $M(P_i) = \sigma(q)$ when $q \sim M$. So, $\eta(\alpha) \in enb_{\tau^k}(\sigma(q))$ for every τ^k-region of $\mathcal{T}$. Hence, $\alpha \in enb_{\mathcal{T},\tau^k}(q)$.

We now observe that $q \sim M$ implies $enb_{\mathcal{T}}(q) = enb_{\widehat{N},cds_{lmax}}(M)$, which follows from (8), Axiom 2, and (3). 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 τ^k-regions of $\mathcal{T}$. Similarly as in [12], one can define a system of equations and inequalities encoding the conditions that must be satisfied by τ^k-regions. Let $Q = \{q_0,q_1,\ldots,q_m\}$ and $T = \{t_1,\ldots,t_n\}$. 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 $\mathcal{S}_\mathcal{T}$:

$$\begin{cases} \mathbf{x}_s - (\mathbf{x}_s, 1) \otimes \sum_{i=1}^{n} \alpha(t_i) \cdot \mathbf{X}_i \geq \mathbf{0} & \text{for all } \delta(q_s, \alpha) = q_r \\ \mathbf{x}_r - \mathbf{x}_s - (\mathbf{x}_s, 1) \otimes \sum_{i=1}^{n} \alpha(t_i) \cdot (\mathbf{Y}_i - \mathbf{X}_i) = \mathbf{0} & \text{in } \mathcal{T} . \end{cases} \tag{9}$$

Then the non-negative integer solutions of $\mathcal{S}_\mathcal{T}$ are in a one-to-one correspondence with the τ^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, however, the situation is much 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 will now outline how one can develop a fully satisfactory procedure for synthesis problems like that discussed in the introduction.

Problem 3 (effective 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 Q. Decide whether there is a* WPOL*-net N with implicitly ordered places $p_1, \ldots, p_m, \ldots, p_n$ realising $\mathcal{T}$ such that:*

1. *each whole-place p_i of N satisfies $i \leq m$, and*
2. *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 WPOL*-net N.*

Figure $1(b)$ defines an instance of the above problem with $m = 1$. 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. Note that for the the example in Figure $1(b)$, we must have $\ell(b) = \ell(c)$ since otherwise the initial state would have to enable also a step α with $b \in \alpha$ and $c \notin \alpha$. Hence here one only needs to consider two locality assignments.

We next discuss the coefficients on the arcs adjacent to $p_1, \ldots, p_m$. Suppose first that $i, j \leq m$ 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 $v_j = 0$.

Note that for the example in Figure 1(b), all coefficients v_j satisfy $0 \leq v_j \leq 1$. Moreover, as $\{b, c\}$ is an enabled step, it is not possible to have both $W(p_1, b) = p_1 + v$ and $W(p_1, c) = p_1 + w$.

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.

First, we derive the system $\mathcal{S}_{\mathcal{T}}$ as in (9) with $k = m+1$, implicitly assuming that the first m components correspond to $p_1, \ldots, p_m$, and the k-th component corresponds to a generic non-whole-place p being constructed. We then delete all equations and inequalities which concern $p_1, \ldots, p_m$, i.e., those beginning with $\mathbf{x}_s^{(i)}$, for $1 \leq i \leq m$. We finally 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. The homogeneous system $\mathcal{S}'_{\mathcal{T}}$ obtained in this way is *linear*.

Assume some arbitrary ordering of the variables of $\mathcal{S}'_\mathcal{T}$. Using the results from [6], 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 τ^k-region.

The $\mathbf{p}^l$'s are *fixed* and some of them turned into new 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 $r = 0$ 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 τ^k-region separating q_i and q_j. Then there is a solution $\mathbf{p} = \sum_{l=1}^{r} a_l \cdot \mathbf{p}^l$ such that $\rho = \psi(\mathbf{p})$. This means that $\mathbf{p}$ assigns different values to q_i and q_j. Hence, since the coefficients a_l are non-negative, 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. We therefore only need to check the $\mathbf{p}^l$'s in order to establish the separation of q_i and q_j. If a suitable $\mathbf{p}^l$ is found, we add a non-whole-place p corresponding to the last place of $\psi(\mathbf{p}^l)$ 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 $\mathcal{T}$. In the latter case, one can always add a standard PT-net 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 $\beta \neq \alpha$ enabled at q_i such that $\ell(\beta) \subseteq \ell(\alpha)$ and $\alpha \leq \beta$.

In all other cases, α is not τ^k-region enabled at q_i iff $\psi(\mathbf{p})$ disables α, for some solution $\mathbf{p} = \sum_{l=1}^{r} a_l \cdot \mathbf{p}^l$. Hence, since the coefficients a_l are non-negative, α is not τ^k-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-place p corresponding to the last place of $\psi(\mathbf{p}^l)$ 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

Among the possible directions for future work, we single out two challenges. The first one is the development of a synthesis approach for WPO-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 [7]. The second task, more specific to k-WPOL-nets, is to investigate the relationship between the locality mapping and the grouping of the places into clusters.

Acknowledgements

We would like to thank the anonymous reviewers for useful comments and suggestions.

References

1. Abdulla, P.A., Delzanno, G., Van Begin, L.: A Language-Based Comparison of Extensions of Petri Nets with and without Whole-Place Operations. Lecture Notes in Computer Science **5457**, Springer (2009) 71–82
2. Badouel, E., Bernardinello, L., Darondeau, P.: Petri Net Synthesis. Texts in Theoretical Computer Science. An EATCS Series. Springer (2015)
3. Badouel, E., Darondeau, P.: Theory of Regions. Lecture Notes in Computer Science **1491**, Springer (1998) 529–586
4. Bernardinello, L., De Michelis, G., Petruni, K., Vigna, S.: On the Synchronic Structure of Transition Systems. In: Structures in Concurrency Theory, J.Desel (ed.) (1995) 69–84
5. Busi, N., Pinna, G.M.: Synthesis of Nets with Inhibitor Arcs. Lecture Notes in Computer Science **1243**, Springer (1997) 151–165
6. Chernikova, N.: Algorithm for Finding a General Formula for the Non-negative Solutions of a System of Linear Inequalities. USSR Computational Mathematics and Mathematical Physics **5** (1965) 228–233
7. Darondeau, P., Koutny, M., Pietkiewicz-Koutny, M., Yakovlev, A.: Synthesis of Nets with Step Firing Policies. Fundamenta Informaticae 94 (2009) 275–303
8. Desel, J., Reisig, W.: The Synthesis Problem of Petri Nets. Acta Informatica 33 (1996) 297–315
9. Dufourd, C., Finkel, A., Schnoebelen, P.: Reset Nets Between Decidability and Undecidability. Lecture Notes in Computer Science **1443**, Springer (1998) 63–115
10. Ehrenfeucht, A., Rozenberg, G.: Partial 2-structures; Part I: Basic Notions and the Representation Problem, and Part II: State Spaces of Concurrent Systems. Acta Informatica **27** (1990) 315–368
11. Finkel, A., McKenzie, P., Picaronny, C.: A Well-structured Framework for Analysing Petri Net Extensions. Information and Computation **195** (2004) 1–29
12. Kleijn, J., Koutny, M., Pietkiewicz-Koutny, M., Rozenberg, G.: Applying Regions. Theoretical Computer Science (2016)
13. Koutny, M., Pietkiewicz-Koutny, M.: Synthesis of Petri Nets with Localities. Sci. Ann. Comp. Sci. **19** (2009) 1–23
14. Mukund, M.: Petri Nets and Step Transition Systems. International Journal of Foundations of Computer Science **3** (1992) 443–478
15. Nielsen, M., Rozenberg, G., Thiagarajan, P.S.: Elementary Transition Systems. Theoretical Computer Science **96** (1992) 3–33
16. Pietkiewicz-Koutny, M.: Transition Systems of Elementary Net Systems with Inhibitor Arcs. Lecture Notes in Computer Science **1248**, Springer (1997) 310–327
17. Schmitt, V.: Flip-Flop Nets. Lecture Notes in Computer Science **1046**, Springer (1996) 517–528