Kleijn J, Koutny M, Pietkiewicz-Koutny M, Rozenberg G.

Applying Regions.

Theoretical Computer Science 2017, 658(Part A), 205-215.

DOI link to article:

http://dx.doi.org/10.1016/j.tcs.2016.01.040

Date deposited:

29/01/2016

Embargo release date:

03 February 2017

Applying Regions

Jetty Kleijn[a], Maciej Koutny[b], Marta Pietkiewicz-Koutny[b], Grzegorz Rozenberg[a]

[a]*LIACS, Leiden University,P.O.Box 9512, NL-2300 RA Leiden, The Netherlands*
[b]*School of Computing Science, Newcastle University,*
Newcastle upon Tyne, NE1 7RU, United Kingdom

Abstract

In this paper we present a brief overview of a representative fragment of the theory of regions. Regions are a powerful tool for the synthesis of concurrent systems from a behavioural specification. To demonstrate the robustness of region based synthesis we survey some of the existing results for extensions of place/transition nets. We relate in particular to the general approach founded on τ-nets and τ-regions. A new extension of region theory to the case of Petri nets with whole-place operations is presented.

Keywords: concurrency, theory of regions, transition system, synthesis problem, Petri net, step semantics, a/sync connection, whole-place operations net

---

1. Introduction

Synthesising systems from behavioural specifications is a powerful way of constructing implementations which are correct-by-design and thus requiring no costly validation efforts. In this paper, we focus on the problem of synthesising a Petri net N from a specification provided by a step transition system $\mathcal{T}$. The latter specifies the desired state space of the net N i.e., the concurrent reachability graph[1] of N should be isomorphic to $\mathcal{T}$. The approach we follow is based on the notion of a *region* of a transition system and we will show the robustness of the concept.

*Corresponding author: Maciej Koutny

[1]Synthesis from a sequential reachability graph is a special case of the concurrent case.

Regions were introduced in the seminal paper [18] for the class of EN-systems with sequential execution semantics, arguably the simplest Petri net model. The aim was to characterise the sequential transition systems (essentially, finite state automata) that are isomorphic to the reachability graph of an EN-system. Regions of a transition system $\mathcal{T}$ were defined as subsets R of its nodes enjoying the so-called *crossing property.* This property simply states that, for every label t, the arcs $q \xrightarrow{t} r$ of $\mathcal{T}$ either all enter R (i.e., $q \notin R$ and $r \in R$), or all leave R (i.e., $q \in R$ and $r \notin R$), or none of them crosses the boundary of R (i.e., $q \in R$ iff $r \in R$). A region can thus be viewed as an abstract representation of a possible place of the net N to be synthesised. Based on this idea, $\mathcal{T}$ is realisable by an EN-system if and only if there are enough regions of $\mathcal{T}$ to satisfy two fundamental regional axioms. The first axiom (*separation*) requires that for every pair of distinct states of $\mathcal{T}$, there is a *witness* region R such that one state belongs to R and the other does not. The second axiom (*forward closure*) requires that if t does not label any transition outgoing from a state q, there must be a witness region R such that t leaves R while q does not belong to R. It was shown that if both regional axioms hold for $\mathcal{T}$, an EN-system realising $\mathcal{T}$ can be constructed by taking the witness regions and turning them into places with appropriate connections to transitions derived from the crossing relationships.

Over the years, this original idea has been developed further and was extended in several different directions, including other Petri net classes (e.g., PT-nets [16, 28], pure and bounded PT-nets [6] Flip-flop nets [32], nets with inhibitor arcs [8, 31], nets with localities [26]; synthesis modules of tools (e.g., Petrify [12], ProM [34], VipTool [5], Genet [9], and Rbminer [33]); application areas (e.g., asynchronous VLSI circuits [12, 9, 33] and workflows [34]); other semantical execution models (e.g., step sequences [21, 31], (local) maximal concurrency [26], and firing policies [15]); and specification formalisms other than transition systems (e.g., languages [13] and scenarios [5]).

In this paper we present to the TCS community a brief overview of a representative fragment of the theory of regions. In the concluding section we present a number of challenging problems in the area. More details concerning the importance and long term impact of the region concept can be found in [4] and the recent monograph [3] that provides a comprehensive overview of region-based net synthesis.

Intuitively, a region captures a single net place through essential behavioural characteristics as encoded in a transition system, including its marking information and connectivity with all the transitions. 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] for dealing with region based synthesis of nets. It is founded on so-called τ-nets and corresponding τ-regions. The parameter τ is a convenient way of capturing the marking information and different connections between places and transitions of varying classes of Petri nets, removing the need to re-state and re-prove the main results every time a new kind of transitions or 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.

In this paper we demonstrate the robustness of regions as already known for several net classes. In addition, we introduce an extension of the original concept by defining a new type of regions for nets with whole-place operations (i.e., with arc weights defined in relation to all places and depending on the current marking). The nets are derived from *transfer/reset* nets [17] and *affine* nets [19], and executed under the step semantics rather than sequentially. This is yet another confirmation of the relevance and flexibility of the notion of region for the derivation of correct concurrent systems.

2. Nets with step sequence semantics

We start by presenting some basic notions concerning Petri nets; in particular, a general notion of nets defined over a transition system that captures relationships between places and transitions.

Throughout the paper, $\mathbb{Z}$ and $\mathbb{N}$ denote respectively the sets of all integers and non-negative integers. The absolute value of an integer n is denoted by $abs(n)$, e.g., $abs(2) = abs(-2) = 2$. The minimum of two integers, k and n, is denoted by $min\{k, n\}$.

2.1. Abelian monoids and multisets

An *abelian monoid* is a set $\mathbb{S}$ with a commutative and associative binary operation $+$, and a neutral 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) $\mathbb{S}_{PT} = \mathbb{N} \times \mathbb{N}$ 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 arcs between places and transitions in PT-nets, whereas $\langle T \rangle$ will represent *steps* of nets with transition set T.

The free abelian monoid $\langle T \rangle$ can be seen as the set of all the 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 $|\alpha| = \sum_{t \in T} \alpha(t)$.

2.2. 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*[2] $\delta : Q \times \mathbb{S} \to Q$ 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 *finite* if it has finitely many states, and the set of enabled elements of $\mathbb{S}$ at any of its states is finite. In the diagrams, an initial state is represented by a small square and all the remaining nodes by circles. The trivial $\mathbf{0}$-labelled transitions are omitted.

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 arbitrary abelian monoids providing ways to define various classes of nets. To ease the notations, throughout the paper we assume that:

- T is a <u>fixed</u> finite set of net transitions;
- $\mathcal{T} = \langle Q, \langle T \rangle, \delta, q_0 \rangle$ is a <u>fixed</u> step transition system; and
- $\tau = \langle \mathcal{Q}, \mathbb{S}, \Delta \rangle$ is a <u>fixed</u> net-type.

2.3. Place/transition nets

A *place/transition net* (PT-net) is a tuple $N = \langle P, T, W, M_0 \rangle$, where P and T are disjoint sets of *places* and *transitions*, $W : (P \times T) \cup (T \times P) \to \mathbb{N}$

[2] Transition functions are not related to Petri net transitions.

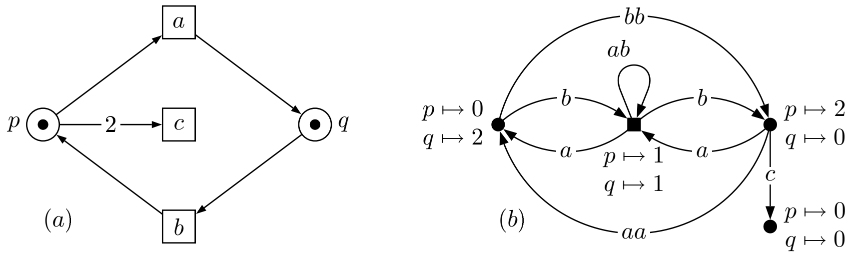

Figure 1: A PT-net (a); and its concurrent reachability graph (b).

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 $1(a)$.

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 $1(b)$ shows the concurrent reachability graph of the PT-net in Figure $1(a)$.

2.4. 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.

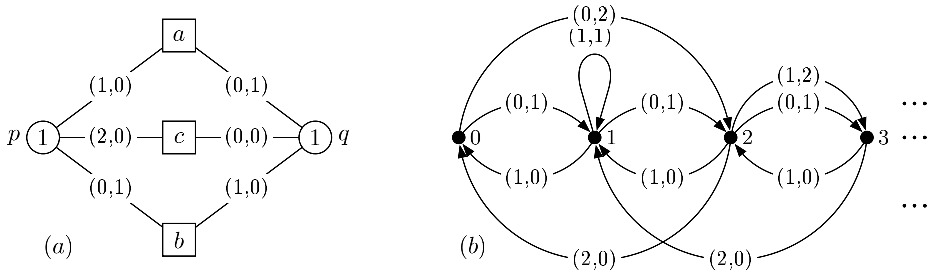

Figure 2: PT-net from Figure 1(a) as a τ_{PT}-net (a); and the net-type τ_{PT} (b).

As in [4, 15], it is possible to encode, without changing the concurrent reachability graph, a PT-net $N = \langle P, T, W, M_0 \rangle$ as τ-net, by setting $F(p,t) = (W(p,t), W(t,p))$. 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. The resulting change of notation, for the net in Figure 1(a), is illustrated in Figure 2(a). Note that $F(q,c) = (0,0)$ means that q and c in Figure 1(a) are disconnected. The markings are represented so that the lack of tokens is indicated by 0, one token by 1, etc.

After such an 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 in Section 2.1, with Δ_{PT} given by $\Delta_{PT}(n,(i,o)) = n - i + o$ and the mapping being defined provided that $n \geq i$. A fragment of $\mathbb{S}_{PT}$ is shown in Figure 2(b). To see how to read the behavioural information off the graph of τ_{PT} in Figure 2(b), suppose that p is a place which currently holds 2 tokens, and t is a transition with $W(p,t) = 1$ and $W(t,p) = 2$, i.e., $F(p,t) = (1,2)$. Then the graph of τ_{PT} tells us to: (i) look at node 2 which is the current marking of p; (ii) find out that there is an arc labelled $(1,2)$ outgoing from node 2 which means that p has 'nothing against' t being fired; and (iii) follow the arc labelled $(1,2)$ which leads to node 3, yielding the marking of p after firing t. As a result, the place p

becomes an 'accomplice' to a possible firing of transition t. However, the same p may become a 'spoiler' if its current marking is 0. For in this case we first look at node 0 and realise that there is no arc labelled $(1,2)$ outgoing from it. Thus p prevents t from being fired.

The graph of τ_{PT} provides equally accurate information about the enabling and firing of any step of transitions α. All one needs to do is calculate $(i,o) = F(p,\alpha) = (W(p,\alpha), W(\alpha,p))$ and repeat what we did for a single transition t.

τ_{PT} helps us to understand the evolution of a PT-net from the point of view of an individual place. Take again the τ_{PT}-net in Figure 2(a) and its concurrent reachability graph in Figure 1(b). For the latter, let us consider the local markings of the place p as well as the 'connections' which effected the changes of those local markings. We can do this by labelling each state with the corresponding marking of p, and each arc with the cumulative arc weights between p and the step α labelling that arc in Figure 1(b) (i.e., with $F(p,\alpha)$). The resulting graph is shown in Figure 3(a), and we may observe that it can be mapped to the graph of the net-type τ_{PT} in Figure 2(b). This can be achieved by mapping any node labelled n in the former graph to the node n in the latter, and then all the arcs in the former graph are instances of arcs in the latter. We therefore call the graph in Figure 3(a) a *τ_{PT}-labelling* of the graph in Figure 1(b). Moreover, what we just observed for the place p can be repeated for the place q, yielding another τ_{PT}-labelling, as shown in Figure 3(b). And similar conclusion can be reached for any place of any τ_{PT}-net or, more generally, any τ-net.

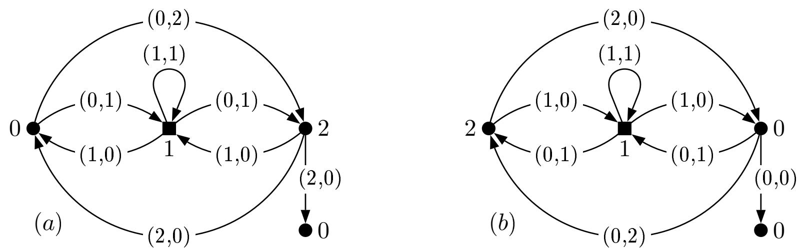

Figure 3: τ_{PT}-labellings reflecting the behaviour of places p and q in the concurrent reachability graph in Figure 1(b).

3. Synthesising τ-nets

The net synthesis problem we consider here aims to devise a procedure which constructs a τ-net with a given concurrent reachability graph. It involves both a *feasibility* problem and an *effective construction* problem. That is, one asks for an exact and effective (decidable) characterisation of the transition systems that can be realised by τ-nets. Moreover, one seeks an algorithm for deriving τ-nets from transition systems. Typically, the latter is a by-product of solving the feasibility problem. Net synthesis therefore considers the following two sub-problems, where we say that a net *realises* $\mathcal{T}$ whenever the step transition system generated by the net is isomorphic to $\mathcal{T}$ in such a way that both the initial states and arc labels are preserved.

Problem 3.1 (feasibility). *Provide necessary and sufficient conditions for $\mathcal{T}$ to be realised by some τ-net.*

Problem 3.2 (effective construction). *For a finite $\mathcal{T}$ decide whether there is a finite τ-net realising it, and in case the answer is positive construct a τ-net realising $\mathcal{T}$.*

The key aspect of any solution to the above problems is to 'read off' all the necessary net places from $\mathcal{T}$ and the library of connections given by a net-type τ. As argued at the end of the previous section, places of a τ-net induce τ-labellings of the concurrent reachability graph of this net. Conversely, if one attempts to construct a τ-net N realising $\mathcal{T}$, the τ-labellings of $\mathcal{T}$ can be used to generate all the places of N. A standard way to capture such τ-labellings is through the following fundamental notion:

> A *τ-region* of $\mathcal{T}$ is a pair of mappings $\langle \sigma : Q \to \mathcal{Q}, \eta : \langle T \rangle \to \mathbb{S} \rangle$ such that η is a morphism of monoids and, for all $q \in Q$ and $\alpha \in enb_{\mathcal{T}}(q)$, we have $\eta(\alpha) \in enb_{\tau}(\sigma(q))$ and $\Delta(\sigma(q), \eta(\alpha)) = \sigma(\delta(q, \alpha))$.

For every state q of $\mathcal{T}$, we denote by $enb_{\mathcal{T},\tau}(q)$ the set of all steps α such that $\eta(\alpha) \in enb_{\tau}(\sigma(q))$, for all τ-regions $\langle \sigma, \eta \rangle$ of $\mathcal{T}$. Hence, for every state q of $\mathcal{T}$, we have $enb_{\mathcal{T}}(q) \subseteq enb_{\mathcal{T},\tau}(q)$. Intuitively, this means that steps enabled at a state q are also *region-enabled* at q.

Remark 3.3. *The original notion of region, introduced for* EN*-systems [18, 29] and sequential transition systems, was defined as a set R of states of $\mathcal{T}$ which has a consistent 'crossing' relationship with every transition t, i.e., the arcs labelled by t either all leave R, or all enter R, or all do not cross the boundary of R. Such an R is easily seen as a τ-region; see, e.g., [4].*

In the context of the synthesis problem, a τ-region represents a place p whose local states (in τ) are consistent with the global states (in $\mathcal{T}$). Then, to deliver a realisation of $\mathcal{T}$, one needs to find enough τ-regions to construct a τ-net with the concurrent reachability graph isomorphic to $\mathcal{T}$. The need for the existence of such τ-regions is intrinsic in the following two *regional axioms*:

Axiom 3.4 (state separation). *For any pair of states $q \neq r$ of $\mathcal{T}$, there is a τ-region $\langle \sigma, \eta \rangle$ of $\mathcal{T}$ such that $\sigma(q) \neq \sigma(r)$.*

Axiom 3.5 (forward closure). *$enb_{\mathcal{T},\tau}(q) \subseteq enb_{\mathcal{T}}(q)$, for every state q of $\mathcal{T}$.*

Note that the second axiom means that, for every state q and every step α in $\langle T \rangle \setminus enb_{\mathcal{T}}(q)$, there is a τ-region $\langle \sigma, \eta \rangle$ of $\mathcal{T}$ such that $\eta(\alpha) \notin enb_{\tau}(\sigma(q))$. We then obtain the fundamental result of the theory of regions.

Theorem 3.6 (e.g., [4]). *$\mathcal{T}$ can be realised by a τ-net if and only if Axioms 3.4 and 3.5 are satisfied.*

Axioms 3.4 and 3.5 provide a generic solution to Problem 3.1 for all net-types τ. The next stage is to consider Problem 3.2 which can be challenging since, in general, a finite $\mathcal{T}$ may be realised by an infinite τ-net, but not by a finite one (cf. [15]).

For a finite net-type $\tau = \langle \mathcal{Q}, \mathbb{S}, \Delta \rangle$, i.e., one where both $\mathcal{Q}$ and $\mathbb{S}$ are finite, Problem 3.2 has always a solution since there are only finitely many potential τ-regions, and so there only finitely many non-equivalent τ-nets with a given finite transition set T. However, for an infinite net-type, Problem 3.2 needs to be considered on a case-by-case basis. E.g., for PT-nets, a solution to Problem 3.2 can be obtained by encoding the synthesis problem as a homogeneous linear system over rational numbers. Non-negative integer solutions of this system represent all the τ_{PT}-regions of $\mathcal{T}$ and, by using the techniques introduced in [10], one can derive an effective representation of all potential τ_{PT}-regions in the form of a finite basis from which all other regions can be derived as non-negative linear combinations.

Theorem 3.7 (e.g., [14]). *For $\tau = \tau_{PT}$, Problem 3.2 has a solution.*

When it comes to effective solutions of the synthesis problem, the complexity of the resulting decision procedures and synthesis algorithms depends on the properties of specific net classes. For example, it has been shown in [32] that Flip-Flop nets can be synthesised in polynomial time. On the other hand, the problem for EN-systems is NP-complete [2]. For safe PT-nets (closely related to EN-systems) an efficient synthesis algorithm was implemented in [12], and further developed in [3].

4. Nets with step firing policies

Step firing policies are a means of controlling and constraining the huge number of execution paths generated by a concurrent system.

A step firing policy is often induced by a preorder $\preceq$ on the steps in $\langle T \rangle$ (i.e., a reflexive and transitive binary relation on $\langle T \rangle$) such that an enabled step $\alpha \neq \mathbf{0}$ is *control disabled* whenever there is another enabled step β with $\alpha \prec \beta$ (i.e., $\alpha \preceq \beta$ and $\beta \not\preceq \alpha$). For example, *maximal concurrency* is captured by the preorder $\preceq_{max}$ such that $\alpha \preceq_{max} \beta$ if $\alpha = \mathbf{0} = \beta$, or $\alpha \neq \mathbf{0} \neq \beta$ and $\alpha \leq \beta$. Another example comes from control based applications where, in order to maximise a certain quantity, one can consider a *reward* map $\$: T \to \mathbb{Z}$ inducing a preorder on steps $\preceq_{\$}$ such that $\alpha \preceq_{\$} \beta$ iff $\alpha = \mathbf{0} = \beta$, or $\alpha \neq \mathbf{0} \neq \beta$ and $\sum_{t \in T} \alpha(t) \cdot \$(t) \leq \sum_{t \in T} \beta(t) \cdot \$(t)$. In general, step firing policies are defined as follows.

Let $\mathcal{X}_\tau$ be the family of all sets of steps $enb_N(M)$ which are enabled at some reachable marking M of some τ-net N with the transition set T. A *step firing policy* for τ-nets over $\langle T \rangle$ is given by a control disabled steps mapping $cds : 2^{\langle T \rangle} \to 2^{\langle T \rangle \setminus \{\mathbf{0}\}}$ such that, for all $X \subseteq \langle T \rangle$ and $Y \subseteq X$:

- $cds(Y) \subseteq cds(X) \subseteq X$; and
- $X \in \mathcal{X}_\tau$ and $X \setminus cds(X) \subseteq Y$ imply $cds(X) \cap Y \subseteq cds(Y)$.

In the above, X should be seen as the set of steps enabled at some reachable marking of some τ-net, and $X \setminus cds(X)$ as the subset of *control enabled* steps at this marking.

Applying cds to a τ-net N amounts to *control disabling* at each marking M all the enabled steps that belong to $cds(enb_N(M))$. That is, $enb_{N,cds}(M) = enb_N(M) \setminus cds(enb_N(M))$ is the set of steps enabled at a reachable marking M under cds. We then use $CRG_{cds}(N)$ to denote the induced reachable restriction of $CRG(N)$, which may be finite even for an infinite $CRG(N)$.

4.1. Synthesis with known policy

We now re-state the net synthesis problem assuming that *cds* is a given step firing policy over $\langle T \rangle$.

Problem 4.1 (feasibility with known policy). *Provide necessary and sufficient conditions for $\mathcal{T}$ to be realised by some τ-net executed under cds.*

It is a remarkable sign of the robustness of region theory that when a τ-net under *cds* realises $\mathcal{T}$, every place of the net still determines a corresponding τ-region of the transition system, without taking *cds* into account. We can therefore modify Axiom 3.5 and then obtain a counterpart of Theorem 3.6.

Axiom 4.2 (forward closure with policy). *For every state q of $\mathcal{T}$, we have $enb_{\mathcal{T}}(q) = enb_{\mathcal{T},\tau}(q) \setminus cds(enb_{\mathcal{T},\tau}(q))$.*

Theorem 4.3 ([15]). *$\mathcal{T}$ can be realised by a τ-net executed under cds* iff *Axioms 3.4 and 4.2 are satisfied.*

4.2. Synthesis with unknown policy

Instead of providing on a firing policy as part of the input to the synthesis process, one may prefer to specify a *family* $\mathcal{CDS}$ of step firing policies, aiming at solving the following problem.

Problem 4.4 (effective construction with unknown policy). *Given a family $\mathcal{CDS}$ of step firing policies, for a finite $\mathcal{T}$ decide whether there is a finite τ-net executed under some cds $\in \mathcal{CDS}$ realising it. Moreover, if the answer is positive construct a τ-net N and select a policy cds $\in \mathcal{CDS}$ such that N executed under cds realises $\mathcal{T}$.*

For a finite family of step firing policies $\mathcal{CDS}$, one can always proceed by exhaustive enumeration, considering finitely many instances of Problem 4.1. However, there are important cases when one can avoid this highly inefficient approach.

A first such case is the class of PT-nets with localities. Here step firing policies are a consequence of partitioning the set of transitions T by a co-location relation $\equiv$, and then restricting the enabledness of steps in such a way that only a step which is maximal within each active locality is control enabled. This kind of policy can be captured by a prorder on steps $\prec_{\equiv}$ such that $\alpha \prec_{\equiv} \beta$ if $\alpha \leq \beta$ and, for every $t \in \beta$ there is $t' \in \alpha$ with

$t \equiv t'$. As the set of transitions is finite, we have a finite number of possible partitions, and so also policies, to consider. If finding a suitable partition is part of the synthesis problem it can lead to a laborious process of discovery. However, if we restrict the search to nets where all the actual (dynamic) conflicts between transitions in reachable markings involve only co-located transitions, then there is only one (canonical) partition to check to solve the synthesis problem [26].

A special kind of synthesis with localities concerns Membrane Systems and Tissue Systems which are computational models inspired by biology [30]. In the Petri nets modelling them, the co-location concerns not only transitions, but also places. It was demonstrated in [24] that there is an effective solution to Tissue System synthesis under the assumption that the topology of the overall system is unknown and has to be determined.

Problem 4.4 can also be solved for an infinite set of step firing policies $\mathcal{CDS}$. For example, [15] presented a decision and synthesis algorithm for PT-nets with step firing policies aiming at maximising linear rewards of steps, where fixing the reward of individual transitions is part of the synthesis problem. PT-nets with static reward maps cover a wide range of potential uses. However, sometime one needs policies exhibiting greater flexibility. For instance, when modelling systems managing different products, the unit reward may be a function of the available quantity of a product. To allow modelling of such systems, one can use *marking dependent* preorders on steps [15]. The idea is to control disable at a marking M all resource enabled steps that are not maximal w.r.t. a preorder $\preceq_M$.

5. Nets with a/sync connections

There is a fundamental assumption in standard Petri net classes, including PT-nets and EN-systems, that in a single firing the same token cannot be both produced and consumed. This assumption does not appear to be fully justified if one allows steps of transitions to be fired. This was recognised, e.g., in [7, 22], and subsequently [23] introduced PT-nets with a/sync connections (PTASC-nets) supporting simultaneous production and consuming of tokens in a single firing. In particular, an input a/sync arc from a transition to a place and an output a/sync arc from that place to another transition can effect a synchronous transfer of tokens between these transitions in a single step. As noted in [11], such synchronous communication is not a primitive concept for the standard Petri net classes, and modelling it typically involves adding complicated sub-nets. Thus a/sync connections provide a potentially attractive succinct modelling approach.

A PT-*net with a/sync connections* (PTASC-net) is defined as a tuple $N = \langle P, T, W, AS, M_0 \rangle$, where $AS : T \times P \to \mathbb{Z}$ is an *a/sync connection* function and the remaining components are as in a PT-net. A step $\alpha \in \langle T \rangle$ is *enabled* at a marking M if, for every p, $M(p) - W(p, \alpha) + min\{0, AS(\alpha, p)\} \geq 0$, where $AS(\alpha, p) = \sum_{t \in T} \alpha(t) \cdot AS(t, p)$. That is, p must contain enough tokens for all standard connections from p to transition occurrences in α and, in addition, also enough tokens to compensate for all a/sync connections from p to transition occurrences in α that will not be supplied synchronously to p by a/sync connections to p.

A step enabled at M can be *fired* leading to the new marking M' given, for every p, by $M'(p) = M(p) + (W(\alpha, p) - W(p, \alpha) + AS(\alpha, p))$. The *concurrent reachability graph* $CRG(N)$ of N is an initialised transition system formed by firing inductively from the initial state all possible enabled steps.

PTASC-nets are a class of τ-nets. This can be demonstrated by taking the connection monoid $\mathbb{S}_{PTASC} = \langle \{-n \mid n \in \mathbb{N}\} \times \mathbb{N} \times \mathbb{Z}, \oplus, \mathbf{0} \rangle$ with $\mathbf{0} = (0, 0, 0)$ and point-wise arithmetic addition $\oplus$, and then defining a net-type $\tau_{PTASC} = \langle \mathbb{N}, \mathbb{S}_{PTASC}, \Delta_{PTASC} \rangle$ over $\mathbb{S}_{PTASC}$, where:

$$\Delta_{PTASC} = \{(\ell, (m, n, k)) \mapsto \ell + m + n + k \mid \ell + m + min\{0, k\} \geq 0\} .$$

This formalises the idea that a place containing ℓ tokens enables steps which require no more than $abs(m)$ standard tokens together with $abs(k)$ tokens moving along a/sync connections if $k < 0$. The resulting number of tokens in the place is $\ell + m + n + k$.

Region based synthesis can then be applied for the following problem.

Problem 5.1 (effective construction). *For a finite $\mathcal{T}$ decide whether there is a finite* PTASC*-net realising it, and if the answer is positive construct a* PTASC*-net realising $\mathcal{T}$.*

A solution can be obtained by encoding the problem using a homogeneous linear system over rational numbers. Non-negative integer solutions of this system represent all the τ_{PTASC}-regions of $\mathcal{T}$ and, by using the techniques introduced in [10], one can again derive an effective (finite) characterisation of all such regions. This characterisation can then be used to check the regional axioms and, if they are satisfied, to construct a PTASC-net realising $\mathcal{T}$.

Theorem 5.2 ([23]). *Problem 5.1 has a solution.*

6. Nets with whole-place operations

We now discuss a novel application of region theory to an extension of the class of nets with weight functions which linearly depend on the current marking (i.e., of all places). The nets are derived from transfer/reset nets [17] and the more general *affine* nets [19], extending PT-nets with whole-place operations [1]. Nets like these are examples of well formed transition systems [20] which are a class of tractable infinite-state models of computation.

Throughout this section markings can, for convenience (and assuming an ordering of places), be represented as vectors. The i-th component of a vector $\mathbf{x}$ is denoted by $\mathbf{x}^{(i)}$. For vectors $\mathbf{x} = (x_1, \ldots, x_n)$ and $\mathbf{y} = (y_1, \ldots, y_n)$, we denote $(\mathbf{x}, 1) = (x_1, \ldots, x_n, 1)$ and $\mathbf{x} \otimes \mathbf{y} = x_1 \cdot y_1 + \cdots + x_n \cdot y_n$.

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 implicitly 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 operation* weight function, and $\mathbf{m}_0$ is an *initial marking* belonging to the set $\mathbb{N}^n$ of *markings*. For a marking $\mathbf{m}$ and $p_i \in P$, we write $\mathbf{m}^{(i)} = \mathbf{m}(p_i)$.

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* at a marking $\mathbf{m}$ if, for every $p \in P$,

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

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 $p \in P$,

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

We denote this by $\mathbf{m}[t\rangle\mathbf{m}'$, and define the *concurrent reachability graph* $CRG(N)$ of N as the step transition system formed by firing inductively from $\mathbf{m}_0$ all possible enabled steps.

It is convenient to specify the weight function using annotations which are linear expressions involving the p_i's. For example, if $n = 3$ then $W(p, t) = (3, 0, 1, 5)$ can be written down as $3 \cdot p_1 + p_3 + 5$. A place p_j 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 $W(p_1, t) = p_1 + 2$ and $j = 1$.

Remark 6.1. *A* PT*-net can be seen as a special kind of* WPO*-net where each* $W(p, t)$ *is of the form* $W(p, t) = l \in \mathbb{N}$*, and similarly for* $W(t, p)$.

WPO-nets make the enabling of transitions and the calculation of the resulting marking dependent on the current distribution of tokens. That is, any place can, in addition to the usual token game, obtain enabling as well as token management information in the form of non-negative linear combinations of the tokens residing in net places. For example, one may stipulate that in order for a transition t to occur, the number of tokens in place p_i must be at least twice the number k of the tokens in place p_j and the firing of t adds $k+1$ tokens to p_i. This can be achieved using two arcs: one of weight $2 \cdot p_j$ from p_i to t, and the other with weight $3 \cdot p_j + 1$ from t to p_i. Two well-known examples of simple WPO-nets are transfer and reset Petri nets [17]. The affine nets [19] are equivalent to WPO-nets where each $W(p,t)$ is of the form $W(p,t) = l \in \mathbb{N}$. Moreover, in this paper we consider use steps rather than firings of single transitions.

We will now consider two synthesis problems for WPO-nets, demonstrating the robustness of the two basic notions underpinning region theory, viz. the notions of τ-net and τ-region.

6.1. k-WPO-nets

WPO-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 locations and their markings can only influence the token game in their local, relatively small neighbourhoods. One way of capturing this is to restrict the number of places which can influence arc weights.

A *k-restricted* WPO*-net* (k-WPO-net, $k \geq 1$) is a WPO-net N for which there is a partition $P_1 \uplus \cdots \uplus P_l$ of the set of places such that each P_i has at most k places and, for all $p \in P_i$ and $p' \notin P_i$, $p \not\rightsquigarrow p' \not\rightsquigarrow p$. 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.

Remark 6.2. *A* WPO*-net with k places is trivially a k-*WPO*-net. In general, a* 1-WPO*-net is not equivalent to a* PT*-net, but rather one in which inscription on all the arcs adjacent to a place p_i are of the form $a \cdot p_i + b$.*

Although k-WPO-net 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 define a suitable 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-WPO-*net-type* is a transition system:

$$\tau_{wpo}^{k} = \langle \mathbb{N}^k, (\mathbb{N}^{k+1})^k \times (\mathbb{N}^{k+1})^k, \Delta_{wpo}^{k} \rangle$$

where

$$\Delta_{wpo}^{k} : \mathbb{N}^k \times ((\mathbb{N}^{k+1})^k \times (\mathbb{N}^{k+1})^k) \to \mathbb{N}^k$$

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

$$\Delta_{wpo}^{k}(\mathbf{x}, (X, Y)) = \mathbf{x} + (\mathbf{x}, 1) \cdot (Y - X) .$$

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

Having extended net-types, a τ_{wpo}^{k}-*net* is a tuple $N = \langle \mathcal{P}, T, F, M_0 \rangle$, where $\mathcal{P} = \{P_1, \ldots, P_l\}$ 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 $\mathcal{P}$, $F : (\mathcal{P} \times T) \to (\mathbb{N}^{k+1})^k \times (\mathbb{N}^{k+1})^k$ is a *flow mapping*, and M_0 is an *initial marking* belonging to the set of *markings* defined as mappings from $\mathcal{P}$ to $\mathbb{N}^k$.

For all $P_i \in \mathcal{P}$ and $\alpha \in \langle T \rangle$, we denote $F(P_i, \alpha) = \sum_{t \in T} \alpha(t) \cdot F(P_i, t)$. Then a step $\alpha \in \langle T \rangle$ is *enabled* at a marking M if, for every $P_i \in \mathcal{P}$, $F(P_i, \alpha) \in enb_{\tau_{wpo}^{k}}(M(P_i))$. We denote this by $\alpha \in enb_N(M)$. *Firing* such a step produces the marking M', for every $P_i \in \mathcal{P}$, defined by $M'(P_i) = \Delta_{\tau_{wpo}^{k}}(M(P_i), F(P_i, \alpha))$. 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.

To see that a k-WPO-net $N = \langle P, T, W, \mathbf{m}_0 \rangle$ with $P = \{p_1, \ldots, p_n\}$ and the clusters $P_1, \ldots, P_l$ can be seen as τ_{wpo}^{k}-net, let us assume 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 $\widehat{N} = \langle \mathcal{P}, T, F, M_0 \rangle$ so that $\mathcal{P} = \{P_1, \ldots, P_l\}$ 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 $(W(t, p_1), \ldots, W(t, p_n))$ 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 in such a way that both the initial states and arc labels are preserved. Conversely, one can transform any τ^k_{wpo}-net into an equivalent, in the sense of the above result, k-WPO-net. As a result, k-WPO-net synthesis can be reduced to the following two problems of τ^k_{wpo}-net synthesis.

Problem 6.3 (feasibility with whole-places). *Given $k \geq 1$, provide necessary and sufficient conditions for $\mathcal{T}$ to be realised by some τ^k_{wpo}-net.*

Problem 6.4 (effective construction with whole-places). *For a finite $\mathcal{T}$ decide whether there is a finite τ^k_{wpo}-net realising it, and if the answer is positive construct a τ^k_{wpo}-net realising $\mathcal{T}$.*

To address Problem 6.3, we define a τ^k_{wpo}-*region* of $\mathcal{T}$ as a pair:

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

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

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

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

We can then re-define the original regional axioms and provide a full characterisation of realisable transition systems.

Axiom 6.5 (state separation with whole-places). *For any pair of states $q \neq r$ of $\mathcal{T}$, there is a τ^k_{wpo}-region $\langle \sigma, \eta \rangle$ of $\mathcal{T}$ such that $\sigma(q) \neq \sigma(r)$.*

Axiom 6.6 (forward closure with whole-places). *For every state q of $\mathcal{T}$, $enb_{\mathcal{T},\tau^k_{wpo}}(q) \subseteq enb_{\mathcal{T}}(q)$.*

Similarly as in the case of τ-regions, the second axiom means that, for every state q and every step α in $\langle T \rangle \setminus enb_{\mathcal{T}}(q)$, there is a τ^k_{wpo}-region $\langle \sigma, \eta \rangle$ of $\mathcal{T}$ such that $\eta(\alpha) \notin enb_{\tau^k_{wpo}}(\sigma(q))$. Then, by a straightforward generalisation of the proofs developed for the synthesis problem of τ-nets.

Theorem 6.7. *$\mathcal{T}$ can be realised by a τ^k_{wpo}-net* iff *Axioms 6.5 and 6.6 are satisfied.*

The next stage of the synthesis process is to seek a solution to Problem 6.4. To make use of the feasibility result provided by Theorem 6.7 one needs to find an effective representation of the τ^k_{wpo}-regions of $\mathcal{T}$. This can be achieved by building a suitable system of equalities and inequalities encoding the conditions defining τ^k_{wpo}-regions. Let $Q = \{q_0, q_1, \ldots, q_m\}$ and $T = \{t_1, \ldots, t_n\}$. The encoding we propose to use, employs the following kinds of variables:

- $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_m$ are k-vectors of nonnegative 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 nonnegative integer variables which encode the mapping η.

We then define the homogeneous system $\mathcal{S}_\mathcal{T}$:

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

It follows that the non-negative integer solutions to the system $\mathcal{S}_\mathcal{T}$ are in a straightforward one-to-one correspondence with the τ^k_{wpo}-regions of $\mathcal{T}$. Therefore, Axioms 6.5 and 6.6 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 rather than linear, and so in this paper we leave as open the problem of effective construction of τ^k_{wpo}-nets.

6.2. Synthesis of non-whole-places

The above treatment of Problem 6.4 calls for solving quadratic homogeneous Diophantine systems on non-negative integers which is a well-known hard problem. Still, the situation may not be as bad as it seems since in practice one would often want to impose bounds on the allowed range of the whole-place coefficients used in arc annotations. Then Problem 6.4 has a solution since one can replace $\mathcal{S}_\mathcal{T}$ by finitely many linear systems. There are, however, practically important cases when one can develop fully satisfactory solutions without imposing restrictions on the range of whole-place coefficients. We will now discuss one such case.

Problem 6.8 (effective construction of non-whole places). *Given a finite* WPO*-net N with transition set T, for a finite $\mathcal{T}$ decide whether there is a finite* WPO*-net extending N with non-whole-places and realising $\mathcal{T}$. Moreover, if the answer is positive, construct an extension realising $\mathcal{T}$.*

In the above, by an extension we mean that we add new places and extend the weight function and the initial marking, without changing the existing components. A new place weight function can depend on the places in N, but its marking cannot affect the weight function of any place. Thus, the whole-place part is known before the synthesis starts. Problem 6.8 has an effective solution, as we will now outline.

Let $N = \langle P, T, W, \mathbf{m}_0 \rangle$ and $|P| = k - 1$. The procedure we propose will, in fact, aim at constructing a k-WPO-net realising $\mathcal{T}$. We proceed in three stages to encode the synthesis problem at hand into a linear homogeneous system.

1. We check whether there exists a mapping $\mu : Q \to \mathbb{N}^{k-1}$ which verifies that $\mathcal{T}$ is consistent with $CRG(N)$. This means that: (i) $\mu(q_0) = \mathbf{m}_0$; and (ii) $\mu(q)[\alpha\rangle\mu(q')$ in N, for all $\delta(q_s, \alpha) = q_r$ in $\mathcal{T}$. If such a μ does not exist, Problem 6.8 is infeasible. Otherwise, μ is determined uniquely, and we proceed to the next stage.
2. We derive the system $\mathcal{S}_\mathcal{T}$ as in the previous sub-section, implicitly assuming that the first $k-1$ components correspond to the places in N, and the k-th component corresponds to a generic non-whole-place p being constructed.
3. We delete all equations and inequalities which concern the places belonging to N, i.e., those beginning with $\mathbf{x}_s^{(i)}$, for $1 \leq i < k$. We then replace by concrete values all those variables which are 'fixed' by the mapping μ, and the fact that p must be a non-whole-place. The homogeneous system $\mathcal{S}'_\mathcal{T}$ obtained in this way is <u>linear</u>.

Assume some arbitrary ordering of the variables occurring in $\mathcal{S}'_\mathcal{T}$. Using the technique 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 τ^k_{wpo}-region.

The $\mathbf{p}^l$'s are fixed and some of them turned into new places if Problem 6.8 has a solution. This is the case if we can verify Axioms 6.5 and 6.6. Clearly, if $r = 0$ then the problem is not feasible. Otherwise, we proceed as follows.

To check state separation (Axiom 6.5), let q_i and q_j be a pair of distinct states of $\mathcal{T}$. If $\mu(q_i) \neq \mu(q_j)$, then there is a place p_s in N such that $\mu_i^{(s)} \neq \mu_j^{(s)}$ and so p_s separates q_i and q_j. Suppose then that $\mu(q_i) = \mu(q_j)$, and ρ is a τ_{wpo}^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 $\psi(\mathbf{p}^l)$ to the net being constructed, together with a fresh copy of the places of N connected with p as prescribed by $\psi(\mathbf{p}^l)$.

Checking forward closure (Axiom 6.6) is carried out for each state q_i, and considers steps $\langle T \rangle$ that are not enabled at q_i in $\mathcal{T}$. Moreover, one only needs to consider steps α with $|\alpha| \leq max$ where max is the maximum size of steps labelling arcs in $\mathcal{T}$ since 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 place is clearly non-whole and disables all steps with more than max transitions.

A step α with $|\alpha| \leq max$ is not τ_{wpo}^k-region enabled at q_i iff for some integer solution $\mathbf{p} = \sum_{l=1}^r a_l \cdot \mathbf{p}^l$, we have that $\psi(\mathbf{p})$ disables α. Hence, since the coefficients a_l are non-negative, α is not 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 $\psi(\mathbf{p}^l)$ to the net being constructed, together with a fresh copy of the places of N connected with p as prescribed by $\psi(\mathbf{p}^l)$.

We therefore conclude that:

Theorem 6.9. *Problem 6.8 has a solution.*

7. Conclusions

In this paper, we surveyed some of the existing results in the area of Petri net synthesis, aiming to demonstrate the robustness of the notion of a region of a transition system. The reader is further referred to the recently published monograph [3] which provides a comprehensive picture of net synthesis and, in particular, discusses the synthesis of finite state net models, e.g., EN-systems.

We specifically focussed on various extensions of the potentially infinite state PT-net model executed under a non-sequential step semantics. We also presented a new extension of Petri net region theory for marking-dependent weight functions between places and transitions, further confirming the robustness of region theory.

Finally, among the challenging open problems related to the theory presented in this paper, we would mention the following: (i) comprehensive investigation of the problem of effective construction of k-WPO-nets; (ii) development of an approach to synthesise nets with localities from specifications other than transition systems, such that formal languages and grammars; and (iii) extension of region theory to cover modal transition systems [27].

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.: The Synthesis Problem for Elementary Net Systems is NP-complete. Theoretical Computer Science **186** (1997) 107–134

[3] Badouel, E., Bernardinello, L., Darondeau, P.: Petri Net Synthesis. Texts in Theoretical Computer Science. An EATCS Series. Springer (2015)

[4] Badouel, E., Darondeau, P.: Theory of Regions. Lecture Notes in Computer Science **1491**, Springer (1998) 529–586

[5] Bergenthum, R., Desel, J., Lorenz, R., Mauser, S.: Synthesis of Petri Nets from Scenarios with VipTool. Lecture Notes in Computer Science **5062**, Springer (2008) 388–398

[6] 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

[7] Bruni, R., Montanari, U.: Zero-Safe Nets: Comparing the Collective and Individual Token Approaches. Information and Computation **156** (2000) 46–89

[8] Busi, N., Pinna, G.M.: Synthesis of Nets with Inhibitor Arcs. Lecture Notes in Computer Science **1243**, Springer (1997) 151–165

[9] Carmona, J., Cortadella, J., Kishinevsky, M.: Genet: A Tool for the Synthesis and Mining of Petri Nets. Proc. of ACSD'09, IEEE Computer Society (2009) 181–185

[10] 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

[11] Christensen, S., Hansen, D.: Coloured Petri Nets Extended with Channels for Synchronous Communication. Lecture Notes in Computer Science **815**, Springer (1994) 159–178

[12] Cortadella, J., Kishinevsky, M., Kondratyev, A., Lavagno, L., Yakovlev, A.: Logic Synthesis of Asynchronous Controllers and Interfaces. Springer (2002)

[13] Darondeau, P.: Deriving Unbounded Petri Nets from Formal Languages. Lecture Notes in Computer Science **1466**, Springer (1998) 533–548

[14] Darondeau, P.: On the Petri Net Realization of Context-free Graphs. Theoretical Computer Science **258** (2001) 573–598

[15] Darondeau, P., Koutny, M., Pietkiewicz-Koutny, M., Yakovlev, A.: Synthesis of Nets with Step Firing Policies. Fundamenta Informaticae 94 (2009) 275–303

[16] Desel, J., Reisig, W.: The Synthesis Problem of Petri Nets. Acta Informatica 33 (1996) 297–315

[17] Dufourd, C., Finkel, A., Schnoebelen, P.: Reset Nets Between Decidability and Undecidability. Lecture Notes in Computer Science **1443**, Springer (1998) 63–115

[18] 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

[19] Finkel, A., McKenzie, P., Picaronny, C.: A well-structured Framework for Analysing Petri Net Extensions. Information and Computation **195** (2004) 1–29

[20] Finkel, A., Schnoebelen, P.: Well-structured Transition Systems Everywhere! Theoretical Computer Science **256** (2001) 63–92

[21] Hoogers, P.W., Kleijn, H.C.M., Thiagarajan, P.S.: A Trace Semantics for Petri Nets. Information and Computation **117** (1995) 98–114

[22] Kleijn, J., Koutny, M.: Causality in Structured Occurrence Nets. Lecture Notes in Computer Science **6875**, Springer (2011) 283–297

[23] Kleijn, J., Koutny, M., Pietkiewicz-Koutny, M.: Regions of Petri Nets with a/sync Connections. Theoretical Computer Science **454** (2012) 189–198

[24] Kleijn, J., Koutny, M., Pietkiewicz-Koutny, M.: Tissue Systems and Petri Net Synthesis. T. Petri Nets and Other Models of Concurrency **9** (2014) 124–146

[25] Koutny, M., Pietkiewicz-Koutny, M.: Synthesis of Elementary Net Systems with Context Arcs and Localities. Fundamenta Informaticae **88** (2008) 307–328

[26] Koutny, M., Pietkiewicz-Koutny, M.: Synthesis of Petri Nets with Localities. Sci. Ann. Comp. Sci. **19** (2009) 1–23

[27] Larsen, K.G.: Modal Specifications. Lecture Notes in Computer Science **407**, Springer (1989) 232–246

[28] Mukund, M.: Petri Nets and Step Transition Systems. International Journal of Foundations of Computer Science **3** (1992) 443–478

[29] Nielsen, M., Rozenberg, G., Thiagarajan, P.S.: Elementary Transition Systems. Theoretical Computer Science **96** (1992) 3–33

[30] Păun, G., Rozenberg, G., Salomaa, A.: The Oxford Handbook of Membrane Computing. Oxford University Press (2009)

[31] Pietkiewicz-Koutny, M.: Synthesising Elementary Net Systems with Inhibitor Arcs from Step Transition Systems. Fundamenta Informaticae **50** (2002) 175–203

[32] Schmitt, V.: Flip-Flop Nets. Lecture Notes in Computer Science **1046**, Springer (1996) 517–528

[33] Solé, M., Carmona, J.: Rbminer: A Tool for Discovering Petri Nets from Transition Systems. Lecture Notes in Computer Science **6252**, Springer (2010) 396–402

[34] van der Werf, J.M.E.M., van Dongen, B.F., Hurkens, C.A.J., Serebrenik, A.: Process Discovery Using Integer Linear Programming. Lecture Notes in Computer Science **5062**, Springer (2008) 368–387