Regions of Affine Nets

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. Regions of transition systems provide a versatile and effective tool for the synthesis of Petri nets from behavioural specifications. Intuitively, a region captures a single net place through essential behavioural characteristics as encoded in the transition system, including marking information and its 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 for dealing with region based synthesis of Petri nets. It is founded on so-called τ-nets and corresponding τ-regions.
In this paper, we discuss a region based synthesis procedure for affine nets, a class of Petri nets, in which the number of tokens produced by firing transitions depends linearly on the current marking. We then show that the notion of a τ-region can be suitably adapted to fit the semantics of affine nets.
Keywords: concurrency, theory of regions, transition system, synthesis problem, Petri net, affine net, localities, locally maximal step semantics

1 Introduction

The intended or observed behaviour of a concurrent system may be captured using a step transition system such as that depicted in Figure 1. It has six different states, including the initial state *init*, and a number of directed arcs labelled by multisets of executed actions representing possible transitions among these states.

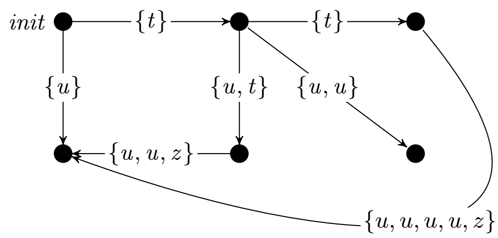


Fig. 1. A step transition system which cannot be generated by a PT-net.

Suppose that one would like to construct a Place/Transition net (PT-net) net N with its concurrent reachability graph isomorphic to the step transition system in Figure 1. Such an attempt would fail for the following reasons:

- The presence of an arc labelled by $\{u, t\}$ necessarily implies the presence of an arc outgoing from the same state labelled by $\{u\}$, and that is not true of the step transition system in Figure 1.
- The effect of executing $\{t\}\{u, t\}\{u, u, z\}$ and $\{t\}\{t\}\{u, u, u, u, z\}$ should in N be the same as both step sequences lead from the initial state to the same state of the transition system. Hence, assuming that W is the weight function of N, for every place p we would have:

$$3 \cdot (W(u,p) - W(p,u)) + 2 \cdot (W(t,p) - W(p,t)) + (W(z,p) - W(p,z)) =$$
$$4 \cdot (W(u,p) - W(p,u)) + 2 \cdot (W(t,p) - W(p,t)) + (W(z,p) - W(p,z)) \; .$$

  As a result, $W(u,p) = W(p,u)$ which means that executing u would not change the marking of N, a contradiction with the fact that executing $\{u\}$ in the initial state leads to a different state.

The latter of the above two problems is related to the fact that arc weights in PT-nets are constant and, as we demonstrated above, no net model with this property can generate the step transition system in Figure 1. We therefore need a more expressive model, and in this paper we show that a suitable formal model for behavioural descriptions like that in Figure 1 are affine nets with localities (AL-nets). *Affine* nets [13] are an example of Petri net models where arc weights depend linearly on the current marking. They are syntactically related to nets with whole-place operations [1] (WPO-nets) and transfer/reset nets [10], but they have a distinct execution semantics. In this paper, we extend the original model of [13] with step sequence semantics and transition localities. The latter feature supports the definition of the locally maximal execution semantics, allowing one to model GALS (Globally Asynchronous Locally Synchronous) systems [8, 12].

Grouping net transitions in different localities and introducing execution semantics that allows only the maximal multisets of enabled net transitions to 'fire' within a given locality will help us to address the first problem mentioned above. Allowing the weights of connections between places and transitions depend on the current marking will address the second problem.

The synthesis of an AL-net from a transition system specification will be based on the notion of a region of a transition system [11, 3, 2] suitably adapted to AL-nets, and the notion of locally maximal step semantics, a special kind of *step firing policy* (see [7, 16]).

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., [18, 17, 4, 20, 5, 19]. Later, a general approach was developed within the framework of τ-nets that take a *net-type* as a parameter [3]. In this context, [7] introduced a general approach for dealing with step firing policies, including the locally maximal execution semantics.

In this paper, we focus on the problem of synthesising AL-nets from behavioural specifications provided by step transition systems. A solution to the synthesis problem for the WPO-nets was outlined in [14], and for WPO-nets with localities in [15] and we use some of the ideas introduced there in the proposed treatment of affine nets with localities.

The paper is organised as follows. The next section recalls some basic notions concerning step transition systems and τ-nets. Section 3 introduces AL-nets, and Section 4 presents regions as an essential ingredient for a solution to the synthesis problem for AL-nets, treating them as a special kind of τ-nets.

2 Preliminaries

An *abelian monoid* is a set $\mathbb{S}$ with a commutative and associative binary operation $+$, and an identity element $\mathbf{0}$. The result of composing n copies of $s \in \mathbb{S}$ is denoted by $n \cdot s$, and so $\mathbf{0} = 0 \cdot s$. An example of an abelian monoid is the free abelian monoid $\langle T \rangle$ generated by a set T, the elements of which will represent *steps* of nets with transition set T. $\langle T \rangle$ 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 $\alpha < \beta$ whenever $\alpha \neq \beta$ and $\alpha(t) \leq \beta(t)$ for all $t \in T$.

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* $\delta : Q \times \mathbb{S} \to Q$ 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* from the initial state, 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, we denote by $enb_{TS}(q)$ the set of all s which are *enabled* at q, i.e., $\delta(q, s)$ is defined. TS is *bounded* if $enb_{TS}(q)$ is finite for every state q of TS. Moreover, such a TS is *finite* if it has finitely many states. In diagrams, $\mathbf{0}$-labelled arcs are omitted.

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 and used to define various classes of nets.

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

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

Petri nets defined by net-types. A net-type $\tau = \langle \mathcal{Q}, \mathbb{S}, \Delta \rangle$ 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*, i.e., mappings from P to $\mathcal{Q}$.

For many classes of Petri nets, including the affine nets, $\mathcal{Q}$ is the set of natural numbers $\mathbb{N} = \{0, 1, 2, \dots\}$ or its subset.

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

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

is the set of *reachable* markings, and $\delta(M, \alpha) = M'$ *iff* $M[\alpha\rangle M'$.

3 Affine Nets with Localities

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

Before introducing AL-nets, we first give the definition of affine nets and their step sequence semantics. An *affine net* (A-net), not yet considered as a τ-net, is a tuple $N = \langle P, T, W, \mathbf{m}_0\rangle$, where:

- $P = \{p_1, \dots, p_n\}$ is a finite set of implicitly ordered *places*;
- T is a finite set of *transitions* disjoint with P;
- W is a *weight* function with domain $(P \times T) \cup (T \times P)$ such that, for all $p \in P$ and $t \in T$, $W(p, t) \in \mathbb{N}$ and $W(t, p) \in \mathbb{N}^{n+1}$; and
- $\mathbf{m}_0$ is an *initial marking* belonging to the set $\mathbb{N}^n$ of *markings*.

It is convenient to specify the output weights using linear expressions involving the p_i's. For example, if $n = 3$ then $W(t, p_3) = (2, 0, 1, 4)$ can be written down as $2 \cdot p_1 + p_3 + 4$. In diagrams, arcs are annotated with their weights; arcs with weight 0 are dropped; and annotations '1' are not explicitly shown. A place p_j ($1 \le j \le n$) is a *whole-place* if $W(t, p)^{(j)} > 0$, for some $p \in P$ and $t \in T$. In such a case we also write $p_j \rightsquigarrow p$.

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 $p \in P$,

$$\mathbf{m}(p) \ge W(p, \alpha)\ . \tag{1}$$

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 $p \in P$,

$$\mathbf{m}'(p) = \mathbf{m}(p) - W(p,\alpha) + (\mathbf{m} - (W(p_1,\alpha),\ldots,W(p_n,\alpha)),1) \otimes W(\alpha,p)\ . \quad (2)$$

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. Note that in (2), the number of tokes deposited in places depends linearly on the marking of the net places *after* the tokens were removed from them by the transitions of the step being executed. In contrast, in WPO-nets [1] the number of deposited tokens is calculated on the basis of the marking before the execution of the step (in addition, the number of tokens removed from the places also depends on the current marking).

An *affine net with localities* (AL-net) is a tuple $N = \langle P, T, W, \mathbf{m}_0, \ell\rangle$ such that $\langle P, T, W, \mathbf{m}_0\rangle$ is the underlying A-net, $\ell : T \to \mathbb{N}$ is the *locality mapping* of N, and $\ell(T)$ are the *localities* of N. In diagrams, nodes representing transitions assigned the same locality are shaded in the same way, as illustrated in Figure 2 for transitions z and u.

AL-nets are executed under the *locally maximal* rule. 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 (i.e., if α is enabled in the underlying A-net). A resource enabled step α is then *control enabled* at $\mathbf{m}$ if there are no $t \in T$ and $u \in \alpha$ (not necessarily different from t) such that $\ell(t) = \ell(u)$ 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) (i.e., as in the underlying A-net). The *concurrent reachability graph* $CRG_{lmax}(N)$ of N is then formed by firing inductively from $\mathbf{m}_0$ all possible control enabled steps. The concurrent reachability graph of the AL-net in Figure 2 is isomorphic to the step transition system shown in Figure 1.

The concurrent reachability graph of an AL-net can be finite even if the concurrent reachability graph of the underlying A-net is infinite. For example, the underlying A-net of the AL-net in Figure 2 generates infinitely many step sequences $\underbrace{\{t\}\{t\}\{z\}\ldots\{t\}\{t\}\{z\}}_{k\ \text{times}}$, each of which leads to a different marking.

In general, execution semantics such as local maximal concurrency can be formulated in terms of *step firing policies* (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 is given by:

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

4 Synthesising affine nets with localities

We will now discuss how to construct an AL-net with a concurrent reachability graph that is isomorphic to a given step transition system $\mathcal{T} = \langle Q, \langle T\rangle, \delta, q_0\rangle$.

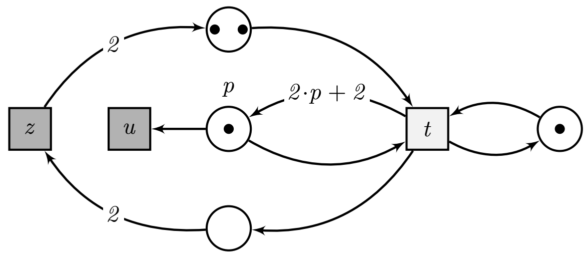


Fig. 2. An AL-net generating the step transition system of Figure 1.

For this net synthesis problem, a general approach was developed for generically defined τ-nets, each such class of nets being represented by its own net-type τ. Note that a key aspect of any solution to a net synthesis problem is to discover all the necessary net places from $\mathcal{T}$ and their connections with transitions of T from τ.

4.1 Net-type for affine nets

AL-nets employ arc weights that depend on the current marking of all places. This may be too general, e.g., in the case of systems where places are distributed among remote neighbourhoods and thus are not capable to exert direct influence on each other. This can be captured by restricting the number of places which can influence arc weights.

A *k-restricted* AL*-net* (k-AL-net, $k \geq 1$) is a AL-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 comprises at most k places and, for all $p \in P_i$ and $p' \in P_j$ $(i \neq j)$, we have $p \not\rightsquigarrow p'$. That is, there is no exchange of current marking information between different clusters of places P_i.

Although k-AL-nets are not τ-nets in the sense of the original definition, they still broadly speaking adhere to 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 clusters of places rather than the behaviour of single places. More precisely, for each $k \geq 1$, the *k-affine-net-type* is a transition system:

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

where

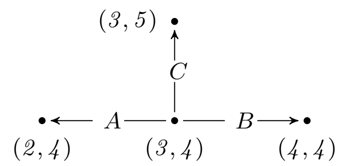
$$\Delta^k_{\mathit{aff}} : \mathbb{N}^k \times (\mathbb{N}^k \times (\mathbb{N}^{k+1})^k) \to \mathbb{N}^k$$

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

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

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

A τ^k_{aff}*-net* is a tuple $N = \langle \mathcal{P}, T, F, M_0, \ell \rangle$, where:



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

Fig. 3. A fragment of the infinite net-type τ^2_{aff}.

- $\mathcal{P} = \{P_1, \ldots, P_r\}$ 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 $\mathcal{P}$;
- $F : \mathcal{P} \times T \to \mathbb{N}^k \times (\mathbb{N}^{k+1})^k$ 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 location mapping for the transitions in T.

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 *resource enabled* at a marking M if, for every $P_i \in \mathcal{P}$, $F(P_i, \alpha) \in \mathit{enb}_{\tau^k_{\mathit{aff}}}(M(P_i))$. Such a step is then *control enabled* if

$$\alpha \in \mathit{enb}_{N,\mathit{cds}_{lmax}}(M) = \mathit{enb}_N(M) \setminus \mathit{cds}_{lmax}(\mathit{enb}_N(M)) \ . \tag{3}$$

Firing a control enabled step produces the marking M', for every $P_i \in \mathcal{P}$, defined by $M'(P_i) = \Delta^k_{\mathit{aff}}(M(P_i), F(P_i, \alpha))$. We denote this by $M[\alpha\rangle M'$, and then define the *concurrent reachability graph* $\mathit{CRG}_{lmax}(N)$ of N as the step transition system formed by firing inductively from M_0 all possible control enabled steps.

4.2 From transition systems to nets

First we need to express a k-AL-net $N = \langle P, T, W, \mathbf{m}_0, \ell \rangle$, with a set of places $P = \{p_1, \ldots, p_n\}$ and clusters $P_1, \ldots, P_r$, as a τ^k_{aff}-net with localities. 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 $N' = \langle \mathcal{P}, T, F, M_0, \ell \rangle$ so that $\mathcal{P} = \{P_1, \ldots, P_r\}$ and, for all $P_i \in \mathcal{P}$ and $t \in T$:

- $F(P_i, t) = (X, Y)$, where $X = (W(p_1, t), \ldots, W(p_n, t))$ is a vector, and Y is the array $[W(t, p_1), \ldots, W(t, p_n)]$ (the $W(t, p_i)$'s are column vectors), both obtained by deleting the rows and/or columns corresponding to the places in $P \setminus P_i$;

– $M_0(P_i)$ 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 N' are isomorphic (when we apply the cds_{lmax} policy, or ignore it, in both nets). Conversely, one can transform any τ^k_{aff}-net with localities into an equivalent k-AL-net and, trivially, each AL-net is a $|P|$-AL-net. Hence k-AL-net synthesis can be reduced to the following two synthesis problems for τ^k_{aff}-net with localities.

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_{aff}-net with the locality mapping ℓ, i.e., $\mathcal{T}$ is isomorphic with the concurrent reachability graph of the net executed under the cds_{lmax} policy defined by ℓ.

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_{aff}-net with the locality mapping ℓ realising $\mathcal{T}$. Moreover, if the answer is positive construct such a net.

To address Problem 1, we define a τ^k_{aff}*-region* of $\mathcal{T}$ as a pair:

$$\langle \sigma : Q \to \mathbb{N}^k, \eta : T \to \mathbb{N}^k \times (\mathbb{N}^{k+1})^k \rangle \tag{4}$$

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

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

where $\eta(\alpha) = \sum_{t \in T} \alpha(t) \cdot \eta(t)$. Moreover, for every state q of Q, we denote by $enb_{\mathcal{T},\tau^k_{\mathit{aff}}}(q)$ the set of all steps α such that $\eta(\alpha) \in enb_{\tau^k_{\mathit{aff}}}(\sigma(q))$, for all τ^k_{aff}-regions $\langle \sigma, \eta \rangle$ of $\mathcal{T}$ (intuitively, in this case α is *region* enabled).

In the context of the synthesis problem, a τ^k_{aff}-region represents a cluster of places whose local states (in τ^k_{aff}) are consistent with the global states (in $\mathcal{T}$). Then, to deliver a realisation of $\mathcal{T}$, one needs to find *enough*[3] τ^k_{aff}-regions to construct a τ^k_{aff}-net with localities realising $\mathcal{T}$ (under the cds_{lmax} policy). The need for the existence of such τ^k_{aff}-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_{aff}-region $\langle \sigma, \eta \rangle$ of $\mathcal{T}$ such that $\sigma(q) \neq \sigma(r)$.*

Axiom 2 (forward closure) *For every state q of $\mathcal{T}$, $enb_{\mathcal{T}}(q) = enb_{\mathcal{T},\tau^k_{\mathit{aff}}}(q) \setminus cds_{lmax}(enb_{\mathcal{T},\tau^k_{\mathit{aff}}}(q))$.*

[3] We need here only a subset of all possible regions, called admissible regions in [9], that act as 'witnesses' for the satisfaction of every instance of the regional axioms.

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 that:

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

Note that when a τ^k_{aff}-net with localities realises $\mathcal{T}$, every cluster of places of the net still determines a corresponding τ^k_{aff}-region of the transition system, without taking cds_{lmax} into account.

For Problem 1, by suitably adapting the proofs developed in [15] for the WPO-nets with localities, one can show that $\mathcal{T}$ can be realised by a τ^k_{aff}-net ($k \geq 1$) executed under cds_{lmax} *iff* Axioms 1 and 2 are satisfied.

To address Problem 2 using the feasibility result provided by the above statement we need to find an effective representation of the τ^k_{aff}-regions of $\mathcal{T}$. Similarly as in [14], one can define a system $\mathcal{S}_{\mathcal{T}}$ of equations and inequalities encoding the conditions defining τ^k_{aff}-regions. Then, all the non-negative integer solutions of $\mathcal{S}_{\mathcal{T}}$ are in one-to-one correspondence with the τ^k_{aff}-regions of $\mathcal{T}$. Therefore, Axioms 1 and 2 can be checked using the solutions of $\mathcal{S}_{\mathcal{T}}$.

In general, the (homogenous) system $\mathcal{S}_{\mathcal{T}}$ is quadratic. In practice, one might often want to impose bounds on the allowed range of the whole-place coefficients used in arc annotations. In such a case, Problem 2 has a solution since one can replace $\mathcal{S}_{\mathcal{T}}$ by finitely many linear systems that can be dealt with using the techniques developed for PT-nets that employ the results of [6]. One can also consider modified versions of Problem 2, where there is no need to resort to bounding the whole-place coefficients, and still obtain a solution, see, e.g., [14, 15].

5 Conclusions

In this paper, we extended the notions of τ-nets and τ-regions to the class of affine nets. We also discussed how these two notions can be used to develop a synthesis procedure for affine nets with locally maximal step semantics.

Among possible directions for future work, we single out two challenges. The first is to investigate the relationship between the locality mapping and the grouping of the places into clusters. The second is effective construction without the locality mapping being given as input.

Acknowledgement

We would like to thank Alex Yakovlev for a long standing collaboration on topics related to the modelling, analysis, and synthesis of concurrent systems.

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. Dasgupta, S., Yakovlev, A.: Comparative Analysis of GALS Clocking Schemes. IET Computers and Digital Techniques **1** (2007) 59–69
9. Desel, J., Reisig, W.: The Synthesis Problem of Petri Nets. Acta Informatica 33 (1996) 297–315
10. Dufourd, C., Finkel, A., Schnoebelen, P.: Reset Nets Between Decidability and Undecidability. Lecture Notes in Computer Science **1443**, Springer (1998) 63–115
11. 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
12. Fernandes, J., Koutny, M., Mikulski, L., Pietkiewicz-Koutny, M., Sokolov, S., Yakovlev, A.: Persistent and Non-violent Steps and the Design of GALS Systems. Fundamenta Informaticae **137** (2015) 143–170
13. Finkel, A., McKenzie, P., Picaronny, C.: A Well-structured Framework for Analysing Petri Net Extensions. Information and Computation **195** (2004) 1–29
14. Kleijn, J., Koutny, M., Pietkiewicz-Koutny, M., Rozenberg, G.: Applying Regions. Theoretical Computer Science (2016)
15. Kleijn, J., Koutny, M., Pietkiewicz-Koutny, M.: Synthesis of Petri Nets with Whole-places and Localities. (submitted)
16. Koutny, M., Pietkiewicz-Koutny, M.: Synthesis of Petri Nets with Localities. Sci. Ann. Comp. Sci. **19** (2009) 1–23
17. Mukund, M.: Petri Nets and Step Transition Systems. International Journal of Foundations of Computer Science **3** (1992) 443–478
18. Nielsen, M., Rozenberg, G., Thiagarajan, P.S.: Elementary Transition Systems. Theoretical Computer Science **96** (1992) 3–33
19. Pietkiewicz-Koutny, M.: Transition Systems of Elementary Net Systems with Inhibitor Arcs. Lecture Notes in Computer Science **1248**, Springer (1997) 310–327
20. Schmitt, V.: Flip-Flop Nets. Lecture Notes in Computer Science **1046**, Springer (1996) 517–528