Synthesis Problem for Petri Nets with Localities

Maciej Koutny and Marta Pietkiewicz-Koutny

School of Computing Science
Newcastle University
Newcastle upon Tyne, NE1 7RU
United Kingdom
{maciej.koutny,marta.koutny}@newcastle.ac.uk

Abstract. There is a growing need to introduce and develop computational models capable of faithfully modelling systems whose behaviour combines synchrony with asynchrony in a variety of complicated ways. Examples of such real-life systems can be found from VLSI hardware to systems of cells within which biochemical reactions happen in synchronised pulses. One way of capturing the resulting intricate behaviours is to use Petri nets with localities partitioning transitions into disjoint groups within which execution is synchronous and maximally concurrent. In this paper, we generalise this type of nets by allowing each transition to belong to several localities. Moreover, we define this extension in a generic way for all classes of nets defined by net-types.
The semantics of nets with overlapping localities can be defined in different ways, and we here discuss four fundamental interpretations, each of which turns out to be an instance of the general model of nets with policies. Thanks to this fact, it is possible to automatically synthesise nets with localities from behavioural specifications given in terms of finite transition systems. We end the paper outlining some initial ideas concerning net synthesis when the association of transitions to localities is not given and has to be determined by the synthesis algorithm.
Keywords: theory of concurrency, Petri net, locality, analysis and synthesis, step sequence semantics, conflict, theory of regions, transition system, step firing policy, net-type.

1 Introduction

In the formal modelling of computational systems there is a growing need to faithfully capture real-life systems exhibiting behaviour which can be described as 'globally asynchronous locally (maximally) synchronous' (GALS). Examples can be found in hardware design, where a VLSI chip may contain multiple clocks responsible for synchronising different subsets of gates [5], and in biologically inspired membrane systems representing cells within which biochemical reactions happen in synchronised pulses [15]. To capture such systems in a formal manner, [8] introduced *Place/Transition-nets with localities* (PTL-nets), where each locality identifies a distinct set of transitions which must be executed synchronously, i.e., in a maximally concurrent manner (akin to *local maximal concurrency*).

The modelling power of PTL-nets (even after enhancing them with inhibitor and activator arcs in [7]) was constrained by the fact that each transition belonged to a unique locality, and therefore localities were all *non-overlapping*. In this paper, we drop this restriction aiming at a net model which we feel could provide a greater scope for faithful (or direct) modelling features implied by the complex nature of, for example, modern VLSI systems or biological systems. The paper deals with theoretical underpinnings of such an approach.

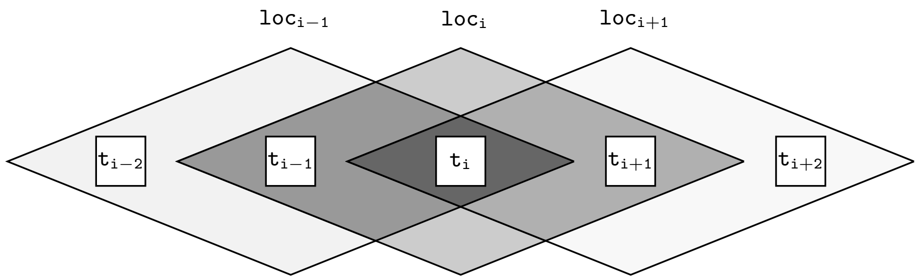


Fig. 1. Transitions with multiple overlapping localities.

To explain the basic idea behind nets with *overlapping localities*, let us consider transitions $\mathtt{t}_0, \mathtt{t}_1, \ldots, \mathtt{t}_{\mathtt{n}-1}$ arranged in a circular manner, i.e., $\mathtt{t}_\mathtt{i}$ is adjacent to $\mathtt{t}_{(\mathtt{i}+\mathtt{n}-1)\,\mathsf{mod}\,\mathtt{n}}$ and $\mathtt{t}_{(\mathtt{i}+1)\,\mathsf{mod}\,\mathtt{n}}$ which are transitions forming its 'neighbourhood'. Figure 1 shows the overlapping of the localities

$$\mathtt{loc}_{\mathtt{i}-1} = \{\mathtt{t}_{\mathtt{i}-2}, \mathtt{t}_{\mathtt{i}-1}, \mathtt{t}_\mathtt{i}\} \quad \mathtt{loc}_\mathtt{i} = \{\mathtt{t}_{\mathtt{i}-1}, \mathtt{t}_\mathtt{i}, \mathtt{t}_{\mathtt{i}+1}\} \quad \mathtt{loc}_{\mathtt{i}+1} = \{\mathtt{t}_\mathtt{i}, \mathtt{t}_{\mathtt{i}+1}, \mathtt{t}_{\mathtt{i}+2}\}$$

to which a transition $\mathtt{t}_\mathtt{i}$ belongs (note that in the diagrams localities are depicted as shaded diamonds encompassing the transitions they contain). Each of the transitions belongs to some subsystem which is left unspecified apart from the fact that, in the initial marking, all the $\mathtt{t}_\mathtt{i}$'s are concurrently enabled.

One can consider at least two different interpretations of the meaning of the localities as in Figure 1 from the point of view of transitions' executability.

> 1st Interpretation: *The execution is triggered by the stimulation of localities, and at each stimulated locality one executes as many (enabled) transitions as possible.*

For example, the following would be examples of legal steps:

$\{\mathtt{t}_2, \mathtt{t}_3, \mathtt{t}_4\}$	$\mathtt{loc}_3$ stimulated
$\{\mathtt{t}_2, \mathtt{t}_3, \mathtt{t}_4, \mathtt{t}_5, \mathtt{t}_8, \mathtt{t}_9, \mathtt{t}_{10}\}$	$\mathtt{loc}_3$, $\mathtt{loc}_4$ and $\mathtt{loc}_9$ stimulated

and $\{\mathtt{t}_3\}$ would be example of an illegal step. According to the first interpretation, a transition can be forced to fire if at least one of its localities has been stimulated. This changes in the second interpretation when this happens only if all of its localities have been stimulated.

> 2nd Interpretation: *To be executed, transition* $\mathtt{t_i}$ *needs all the localities it belongs to be stimulated.*

For example, the following would be examples of legal steps:

$\{\mathtt{t_3}\}$	$\mathtt{loc_2}$, $\mathtt{loc_3}$ and $\mathtt{loc_4}$ stimulated
$\{\mathtt{t_0}, \mathtt{t_1}, \ldots, \mathtt{t_{n-1}}\}$	$\mathtt{loc_0}, \mathtt{loc_1}, \ldots, \mathtt{loc_{n-1}}$ stimulated

and $\{\mathtt{t_2}, \mathtt{t_4}\}$ would be example of an illegal step. The above will be two out of the four fundamental interpretations of nets with overlapping localities which we will introduce and investigate in this paper. It is not our intention in this paper to make a judgement as to which of these four interpretations is more useful or reasonable. Instead, our aim is to investigate and compare their key properties, in particular, those relating to the net synthesis problem.

In this paper, rather than introducing overlapping localities for PT-nets or their standard extensions, we will move straight to the general case of τ-nets [2] which encapsulate a majority of Petri net classes for which the synthesis problem has been investigated and solved. In fact, the task of defining τ-nets with (potentially) overlapping localities is straightforward, as the resulting model of *τ-nets with localities* turns out to be an instance of the general framework of τ-nets with *policies* introduced in [4].

After introducing the new model of nets, we turn our attention to their automatic synthesis from behavioural specifications given in terms of step transition systems. Since τ-nets with localities are an instance of a more general scheme treated in [4], we directly import synthesis results presented there which are based on the regions of a transition system studied in other contexts, in particular, in [1–3, 6, 9, 10, 13, 14, 16].

The results in [4] assume that policies are given which, in our case, means that we know exactly the localities associated with all the net transitions. This may be difficult to guarantee in practice, and we end the paper outlining some initial ideas concerning net synthesis when this is not the case.

The paper is organised in the following way. In the next section, in order to make the paper self-contained, we recall the notions and results relating to the general theory of the synthesis of nets with policies. After that, we define four semantic interpretations of nets with overlapping localities, and prove that in each case the resulting model defines nets with policies. We also discuss and compare some basic properties of the new policies, in particular, we formulate a main result concerning the synthesis of nets with overlapping localities. In the last section, we outline some initial ideas concerning the synthesis problem when not only the net, but also the localities need to be constructed. The appendix presents proofs of results omitted from the main body of the paper.

2 Preliminaries

In this section, we recall some basic notions concerning general Petri nets, policies and the synthesis problem as presented in [4].

Abelian monoids and multisets. An *abelian monoid* is a set S with a commutative and associative binary (composition) operation $+$ on S, and a neutral element $\mathbf{0}$. The monoid element resulting from composing n copies of $s \in S$ will be denoted by $n \cdot s$, and so $\mathbf{0} = 0 \cdot s$ and $s = 1 \cdot s$. As we will see, weighted arcs between places and transitions in PT-nets can be expressed using the abelian monoid $\mathbb{S}_{PT}$ which is the product $\mathbb{N} \times \mathbb{N}$ with the pointwise arithmetic addition operation and $\mathbf{0} = (0, 0)$.

Steps of transitions. Potential *steps* of a Petri net with transition set T can be captured by the free abelian monoid $\langle T \rangle$ generated by T. Note that $\langle T \rangle$ can be seen as the set of all the multisets over T; for example, $aaab = \{a, a, a, b\}$. We will use $\alpha, \beta, \gamma, \dots$ to range over the elements of $\langle T \rangle$. Moreover, for all $t \in T$ and $\alpha \in \langle T \rangle$, we will use $\alpha(t)$ to denote the multiplicity of t in α.

The sum of two multisets, α and β, will be denoted by $\alpha + \beta$, and a singleton multiset $\{t\}$ simply by t. We will then write $t \in \alpha$ whenever $\alpha(t) > 0$, and use $supp(\alpha)$ to denote the set of all $t \in \alpha$. We denote $\alpha \leq \beta$ whenever $\alpha(t) \leq \beta(t)$ for all $t \in T$ (and $\alpha < \beta$ if $\alpha \leq \beta$ and $\alpha \neq \beta$). Whenever $\alpha = \beta + \gamma$, we denote the multiset γ by $\alpha - \beta$. For $X \subseteq \langle T \rangle$, we denote by $\max_{\leq}(X)$ the set of all $\leq$-maximal elements of X.

Transition systems. A *transition system* over an abelian monoid S is a triple (Q, S, δ) such that Q is a set of *states*, and $\delta : Q \times S \to Q$ a partial *transition function*[1] satisfying $\delta(q, \mathbf{0}) = q$ for all $q \in Q$. An *initialised* transition system $\mathcal{T} \stackrel{\mathrm{df}}{=} (Q, S, \delta, q_0)$ has in addition an *initial* state $q_0 \in Q$ from which every other state is *reachable*. For every state q of a (non-initialised or initialised) transition system TS, $enbld_{TS}(q) \stackrel{\mathrm{df}}{=} \{s \in S \mid \delta(q, s) \text{ is defined}\}$.

Initialised transition systems $\mathcal{T}$ over free abelian monoids — called *step transition systems* — will represent concurrent behaviours of Petri nets. Non-initialised transition systems τ over arbitrary abelian monoids — called *net-types* — will provide ways to define various classes of nets. Throughout the paper, we will assume that:

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

Assumption 1 *In this paper, we will assume that τ is* sub-step closed *which means that, for every state $q \in \mathbb{Q}$, if $\alpha + \beta \in enbld_\tau(q)$ then also $\alpha \in enbld_\tau(q)$.*

[1] Transition functions and net transitions are unrelated notions.

The above assumption will imply that sub-steps of *resource enabled steps* (i.e., steps enabled by the standard token game) are also resource enabled which is a condition usually satisfied in practice.

Petri nets defined by net-types. The net-type $\tau = (\mathbb{Q}, \mathbb{S}, \Delta)$ may be conveniently used as a parameter in the definition of a class of nets, called *τ-nets*. The net-type specifies the values (markings) that can be stored within net places ($\mathbb{Q}$), 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).

Definition 1 (τ-net). *A* τ-net *is a bi-partite graph (P, T, F), where P and T are respectively disjoint sets of places and transitions, and $F : (P \times T) \to \mathbb{S}$ is a (generalised) flow mapping. A* marking *of the τ-net is a map $M : P \to \mathbb{Q}$. A* τ-net system $\mathcal{N}$ *is a τ-net with an initial marking M_0.*

In what follows, for each place $p \in P$ and step $\alpha \in \langle T \rangle$ we will denote the cumulative flow between α and p by $F(p, \alpha) = \sum_{t \in T} \alpha(t) \cdot F(p, t)$.

Definition 2 (step semantics). *Given a τ-net system $\mathcal{N} = (P, T, F, M_0)$, a step $\alpha \in \langle T \rangle$ is* (resource) enabled *at a marking M if, for every place $p \in P$:*

$$F(p, \alpha) \in \mathit{enbld}_\tau(M(p)) \; .$$

We denote this by $\alpha \in \mathit{enbld}_\mathcal{N}(M)$. The firing *of such a step produces the marking M' such that $M'(p) = \Delta(M(p), F(p, \alpha))$, for every place $p \in P$. We denote the fact that α is enabled at M and its firing leads to M' by $M[\alpha\rangle M'$, and then define the* concurrent reachability graph $\mathit{CRG}(\mathcal{N})$ *of $\mathcal{N}$ as the step transition system formed by firing inductively from M_0 all possible (resource) enabled steps of $\mathcal{N}$.*

Note that a step α is resource enabled at a marking M in a τ-net system if for every place p there is an $F(p, \alpha)$-labelled arc outgoing from the node $M(p)$ in τ, and the firing of such a step leads to the new marking M', where $M'(p)$ is simply the target node of such an arc in τ.

PT-nets are $\boldsymbol{\tau}$-nets. A *PT-net* is a triple $N = (P, T, W)$, where P and T are disjoint sets of places and transitions, and $W : (P \times T) \cup (T \times P) \to \mathbb{N}$ specifies directed edges with integer weights. Its markings are mappings $M : P \to \mathbb{N}$, and a *PT-net system* is N together with an initial marking M_0, as illustrated in Figure 2(a). Figure 2(b) shows the concurrent reachability graph of the PT-net system in Figure 2(a).

As we will shortly see, it is possible to render PT-nets as τ-nets. Crucially, one can encode the PT-net system's arc weights, $W(p, t)$ and $W(t, p)$, by setting $F(p, t) = (W(p, t), W(t, p)) \in \mathbb{S}_{PT}$. The resulting change of notation, for the net from Figure 2(a), is represented graphically in Figure 2(c). Notice that, in

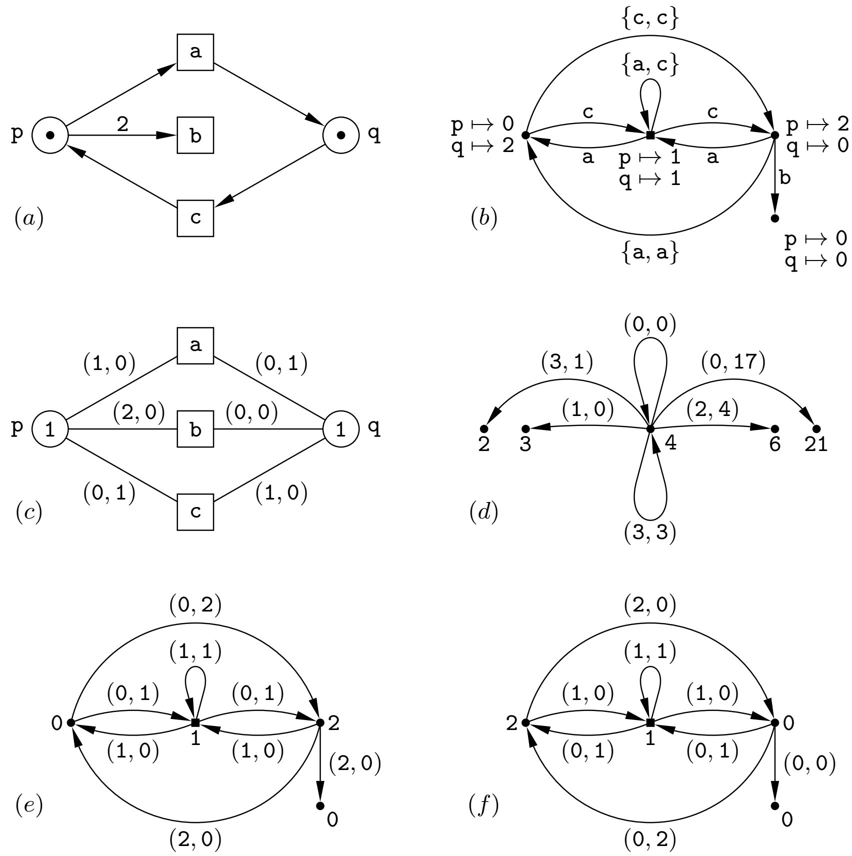


Fig. 2. A PT-net system (a); its concurrent reachability graph (b) with the initial state represented by a small square and the trivial **0**-labelled arcs being omitted; and its rendering as a τ_{PT}-net system (c). A fragment of the net-type τ_{PT} is shown in (d). In (e) and (f) we re-trace in (b) the behaviour of places p and q, respectively, in terms of the net-type τ_{PT}.

particular, $F(\mathtt{q},\mathtt{b}) = (0,0)$ means that q and b in Figure 2(a) are disconnected. The markings are represented so that the lack of tokens is indicated by *0*, one token by *1*, two tokens by *2*, etc.

To show that a PT-net can indeed be seen as a τ-net, we define a suitable (infinite) net-type, $\tau_{PT} = (\mathbb{N}, \mathbb{S}_{PT}, \Delta_{PT})$ over $\mathbb{S}_{PT}$, a fragment of which is shown in Figure 2(d). In general, for every $n \in \mathbb{N}$ and $(in, out) \in \mathbb{S}_{PT}$, $(in, out) \in enbld_{\tau_{PT}}(n) \Leftrightarrow in \leq n$. Moreover, in such a case $\Delta_{PT}(n,(in,out)) = n - in + out$. Then, in order to transform a PT-net into an equivalent τ_{PT}-net, all one needs to do is to insert integers, representing the number of tokens, in each place and set $F(p,t) = (W(p,t), W(t,p))$, for all places p and transitions t, as already mentioned. In other words, $F(p,t) = (in, out)$ means that in is the weight of the arc from p to t, and *out* the weight of the arc in the opposite direction, see Figure 2(a,c).

Although we talked about a single transition t, the graph of τ_{PT} provides equally accurate information about the enabling and firing of a step of transitions α. Indeed, all one needs to do is calculate

$$(in, out) = F(p,\alpha) = (W(p,\alpha), W(\alpha,p)) \ .$$

For the net in Figure 2(a), we obtain $F(\mathtt{p},\{\mathtt{a},\mathtt{c}\}) = (\mathtt{1},\mathtt{0}) + (\mathtt{0},\mathtt{1}) = (\mathtt{1},\mathtt{1})$ and $F(\mathtt{q},\{\mathtt{a},\mathtt{c}\}) = (\mathtt{0},\mathtt{1}) + (\mathtt{1},\mathtt{0}) = (\mathtt{1},\mathtt{1})$ which, together with $\Delta_{PT}(\mathtt{1},(\mathtt{1},\mathtt{1})) = \mathtt{1}$, means that: (i) the net in Figure 2(a) enables the step $\{\mathtt{a},\mathtt{c}\}$ at the initial marking; and (ii) its firing results in the same marking.

Any evolution of a PT-net system can be 're-traced' from the point of view of an individual place. Consider again the PT-net system in Figure 2(a) and its concurrent reachability graph in Figure 2(b). For the latter, let us consider the local markings of the place $\mathtt{p}$ 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 $\mathtt{p}$, and each arc with the cumulative arc weights between $\mathtt{p}$ and the step α labelling that arc, i.e., $F(\mathtt{p},\alpha)$. The result is shown in Figure 2(e).

The graph in Figure 2(e) can be 're-discovered' in the graph of the net-type τ_{PT}. This can be achieved by mapping any node labelled $\mathtt{n}$ in the former graph to the node $\mathtt{n}$ in the latter, and then all the arcs in the former graph are instances of arcs in the latter. We call the graph in Figure 2(e) a τ_{PT}*-labelling* of the graph in Figure 2(b). Clearly, we may repeat the same procedure for the place $\mathtt{q}$, obtaining another τ_{PT}-labelling depicted in Figure 2(f).

$\boldsymbol{\tau}$-nets with policies. Step firing policies are means of controlling and constraining the huge number of execution paths resulting from the highly concurrent nature of many computing systems.

Let $\mathcal{X}_\tau$ be the family of all sets of steps enabled at some reachable marking M of some τ-net $\mathcal{N}$ with the set of transitions T.

Definition 3 (bounded step firing policy). *A* bounded 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 $X \subseteq \langle T\rangle$, the following hold:*

1. *If X is infinite then $cds(X) = \varnothing$.*
2. *If X is finite then, for every $Y \subseteq X$:*
   (a) $cds(X) \subseteq X$;
   (b) $cds(Y) \subseteq cds(X)$; and
   (c) $X \in \mathcal{X}_\tau$ and $X \setminus cds(X) \subseteq Y$ imply $cds(X) \cap Y \subseteq cds(Y)$.

Intuitively, Definition 3(2.c) captures a kind of monotonicity in control disabling resource enabled steps. If control disabling a step in X is due to the (resource) enabling of some steps included in X, then if these disabling steps are also present in Y, any $\alpha \in Y$ which is control disabled in X will also be control disabled in Y.

Step firing policies constrain the behaviour of nets by blocking some of the resource enabled steps.

Definition 4 (τ-net system with policy). *A* τ-net system with policy *is a tuple* $\mathcal{NP} \stackrel{\mathrm{df}}{=} (P, T, F, M_0, cds)$ *such that* $\mathcal{N} = (P, T, F, M_0)$ *is a* τ*-net system and* cds *is a bounded step firing policy for* τ*-nets over* $\langle T \rangle$*.*

The notions of marking and execution of enabled steps in $\mathcal{NP}$ *are inherited from* $\mathcal{N}$*. Moreover, the* resource enabled *and* control enabled *steps of* $\mathcal{NP}$ *at a marking* M *are given, respectively, by:*

$$\begin{aligned} enbld_{\mathcal{NP}}(M) &\stackrel{\mathrm{df}}{=} enbld_{\mathcal{N}}(M) \\ Enbld_{\mathcal{NP}}(M) &\stackrel{\mathrm{df}}{=} enbld_{\mathcal{N}}(M) \setminus cds(enbld_{\mathcal{N}}(M)). \end{aligned}$$

We will denote by $CRG(\mathcal{NP})$ *the step transition system with the initial state* M_0 *formed by firing inductively from* M_0 *all possible control enabled steps of* $\mathcal{NP}$*, and call it the* concurrent reachability graph *of* $\mathcal{NP}$*.*

Step firing policies can often be defined by pre-orders on step sequences. More precisely a bounded step firing policy given by $cds : 2^{\langle T \rangle} \to 2^{\langle T \rangle \setminus \{\mathbf{0}\}}$ is *pre-order based* if there is a pre-order $\preceq$ on $\langle T \rangle$ such that, for all finite $X \subseteq \langle T \rangle$,

$$cds(X) = \{\alpha \in X \mid \alpha \neq \varnothing \wedge \exists \beta \in X : \alpha \prec \beta\} .$$

In such a case we denote cds by $cds_{\preceq}$. For example, the maximally concurrent execution semantics of a PT-net can be captured by the bounded step firing policy $cds_{\max}$ such that, for every non-empty set of steps X, $cds_{\max}(X) \stackrel{\mathrm{df}}{=} \{\alpha \in X \mid \alpha \neq \varnothing \wedge \alpha \notin \max_{\leq}(X)\}$. Such a policy is in fact pre-order based (it suffices to take $\preceq$ to be sub-multiset order $\leq$).

Synthesis of τ-net systems with policies. In this paper, by solving a synthesis problem we mean finding a procedure for building a net of a certain class with a given concurrent reachability graph, as follows.

SYNTHESIS PROBLEM

> Let $\mathcal{T}$ be a given finite step transition system, τ a net-type, and cds a control disabled steps mapping for τ-nets over $\langle T \rangle$. Provide necessary and sufficient conditions for $\mathcal{T}$ to be *realised* by some τ-net system with policy $\mathcal{NP} = (P, T, F, M_0, cds)$ (i.e., $\mathcal{T} \cong CRG(\mathcal{NP})$ where $\cong$ is transition system isomorphism preserving the initial states and transition labels).

The solution of the synthesis problem we seek is based on the idea of a region of a transition system.

Definition 5 (τ-region). *A* τ-region *of* $\mathcal{T}$ *is a pair of mappings*

$$(\sigma : Q \to \mathbb{Q} \ , \ \eta : \langle T \rangle \to \mathbb{S})$$

such that η is a morphism of monoids and, for all $q \in Q$ and $\alpha \in enbld_{\mathcal{T}}(q)$:

$$\eta(\alpha) \in enbld_\tau(\sigma(q)) \quad and \quad \Delta(\sigma(q), \eta(\alpha)) = \sigma(\delta(q, \alpha)) \ .$$

For every state q of Q, we denote by $enbld_{\mathcal{T},\tau}(q)$ the set of all steps α such that $\eta(\alpha) \in enbld_\tau(\sigma(q))$, for all τ-regions (σ, η) of $\mathcal{T}$.

Intuitively, for PT-net systems, τ_{PT}-regions correspond to the τ_{PT}-labellings of the concurrent reachability graph like those depicted in Figure 2(e, f). We then obtain a general net synthesis result [4].

Theorem 1. *$\mathcal{T}$ can be realised by a τ-net system with a (bounded step firing) policy cds* iff *the following two regional axioms are satisfied:*

AXIOM I: STATE SEPARATION

For any pair of states $q \neq r$ of $\mathcal{T}$, there is a τ-region (σ, η) of $\mathcal{T}$ such that $\sigma(q) \neq \sigma(r)$.

AXIOM II: FORWARD CLOSURE WITH POLICIES

For every state q of $\mathcal{T}$, $enbld_{\mathcal{T}}(q) = enbld_{\mathcal{T},\tau}(q) \setminus cds(enbld_{\mathcal{T},\tau}(q))$. $\square$

A net solution to the synthesis problem is obtained if one can compute a finite set $\mathcal{WR}$ of τ-regions of $\mathcal{T}$ *witnessing* the satisfaction of all instances of AXIOMS I and II [6]. A suitable τ-net system with policy *cds*, $\mathcal{NP}_{\mathcal{WR}} = (P, T, F, M_0, cds)$, can be then constructed with $P = \mathcal{WR}$ and, for any place $p = (\sigma, \eta)$ in P and every $t \in T$, $F(p, t) = \eta(t)$ and $M_0(p) = \sigma(q_0)$ (recall that q_0 is the initial state of $\mathcal{T}$, and $T \subseteq \langle T \rangle$).

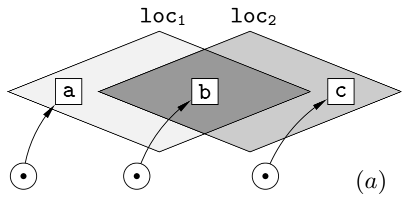
3 Nets with general localities

We will now introduce a general class of Petri nets with localities, and then introduce four fundamental ways of interpreting the semantics of such nets based on specific kinds of *cds* mappings.

A *locality set* for the transition set T is any finite family $\mathcal{L}$ of non-empty sets of transitions — called *localities* — covering T, i.e., $\bigcup \mathcal{L} = T$. Below we will denote by $\mathcal{L}_t$ the set of all localities to which a given transition t belongs. Moreover, $\mathcal{L}_\alpha \overset{\text{df}}{=} \bigcup_{t \in \alpha} \mathcal{L}_t$ is the set of localities involved in a step α. Note that if we additionally assume that the sets in $\mathcal{L}$ are disjoint, then we obtain the model of transition localities considered in [9–11].

Policies based on localities. We consider four policy mappings based on localities, $cds_{\mathcal{L}}^{\maltese}$, where $\maltese \in \{\exists, \forall, \exists \subseteq, \forall \subseteq\}$. Each of these four is a mapping $cds_{\mathcal{L}}^{\maltese} : 2^{\langle T \rangle} \to 2^{\langle T \rangle \setminus \{\mathbf{0}\}}$ such that, for every infinite set of steps $X \subseteq \langle T \rangle$, we have $cds_{\mathcal{L}}^{\maltese}(X) = \varnothing$, and for every finite set of steps $X \subseteq \langle T \rangle$:

$$\begin{aligned}
cds_{\mathcal{L}}^{\exists}(X) &\overset{\text{df}}{=} \{\alpha \in X \mid \exists v \in \alpha\ \exists \ell oc \in \mathcal{L}_v\ \exists \alpha + t \in X :\ t \in \ell oc\} \\
cds_{\mathcal{L}}^{\exists\subseteq}(X) &\overset{\text{df}}{=} \{\alpha \in X \mid \exists v \in \alpha\ \exists \ell oc \in \mathcal{L}_v\ \exists \alpha + t \in X :\ t \in \ell oc \wedge \mathcal{L}_t \subseteq \mathcal{L}_\alpha\} \\
cds_{\mathcal{L}}^{\forall}(X) &\overset{\text{df}}{=} \{\alpha \in X \mid \exists v \in \alpha\ \forall \ell oc \in \mathcal{L}_v\ \exists \alpha + t \in X :\ t \in \ell oc\} \\
cds_{\mathcal{L}}^{\forall\subseteq}(X) &\overset{\text{df}}{=} \{\alpha \in X \mid \exists v \in \alpha\ \forall \ell oc \in \mathcal{L}_v\ \exists \alpha + t \in X :\ t \in \ell oc \wedge \mathcal{L}_t \subseteq \mathcal{L}_\alpha\}
\end{aligned} \tag{1}$$



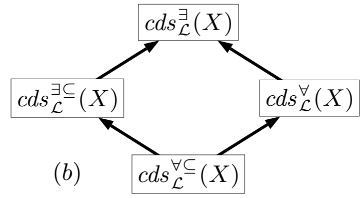


Fig. 3. Comparing different policies based on localities (a), and relationships between different policy mappings (b).

Consider, for example, a PT-net with two localities, $\mathtt{loc}_1 = \{\mathtt{a}, \mathtt{b}\}$ and $\mathtt{loc}_2 = \{\mathtt{b}, \mathtt{c}\}$, depicted in Figure 3$(a)$. In its initial marking, the set $\mathtt{X}$ of resource enabled steps is:

$$\mathtt{X} = \{\varnothing, \{\mathtt{a}\}, \{\mathtt{b}\}, \{\mathtt{c}\}, \{\mathtt{a}, \mathtt{b}\}, \{\mathtt{a}, \mathtt{c}\}, \{\mathtt{b}, \mathtt{c}\}, \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}\} \ .$$

Thus, according to the definitions of the four policy mappings, we have:

$$\begin{aligned} cds_{\mathcal{L}}^{\exists}(\mathtt{X}) &= \{\{\mathtt{a}\}, \{\mathtt{b}\}, \{\mathtt{c}\}, \{\mathtt{a}, \mathtt{b}\}, \{\mathtt{a}, \mathtt{c}\}, \{\mathtt{b}, \mathtt{c}\}\} \\ cds_{\mathcal{L}}^{\exists\subseteq}(\mathtt{X}) &= \{\{\mathtt{b}\}, \{\mathtt{a}, \mathtt{b}\}, \{\mathtt{a}, \mathtt{c}\}, \{\mathtt{b}, \mathtt{c}\}\} \\ cds_{\mathcal{L}}^{\forall}(\mathtt{X}) &= \{\{\mathtt{a}\}, \{\mathtt{b}\}, \{\mathtt{c}\}, \{\mathtt{a}, \mathtt{c}\}\} \\ cds_{\mathcal{L}}^{\forall\subseteq}(\mathtt{X}) &= \{\{\mathtt{b}\}, \{\mathtt{a}, \mathtt{c}\}\} \ . \end{aligned} \tag{2}$$

Our main result is that the *cds* mappings we have just introduced give rise to bounded step firing policies.

Theorem 2. $cds_{\mathcal{L}}^{\maltese}$ *is a bounded step firing policy, for each* $\maltese \in \{\exists, \forall, \exists\subseteq, \forall\subseteq\}$.

As to the direct relationships between the policy mappings based on localities, Figure 3(b) shows an inclusion diagram with arrows indicating set inclusions which hold in all cases. No other arrows (i.e., set inclusions) can be added as can be seen by inspecting the PT-net with localities depicted in Figure 3(a) and the sets returned by each of the four policy mappings shown in (2). Hence, in general, the four *cds* mappings induce *different* control policies for nets with localities.

In the papers [9–11], localities $\mathcal{L}$ formed a partition of T. In such a case, the four policy mappings collapse to:

$$cds_{\mathcal{L}}(X) \stackrel{\mathrm{df}}{=} \{\alpha \in X \mid \exists v \in \alpha \ \exists \alpha + t \in X : \ \ell oc_t = \ell oc_v\}$$

where, for every transition u, ℓoc_u denotes the unique locality belonging to $\mathcal{L}_u$. Thus all four policies introduced in this paper are conservative extensions of that investigated previously.

It is interesting to observe that in the (previously considered) case of non-overlapping localities, $cds_{\mathcal{L}}$ can be defined through a pre-order on steps. This is

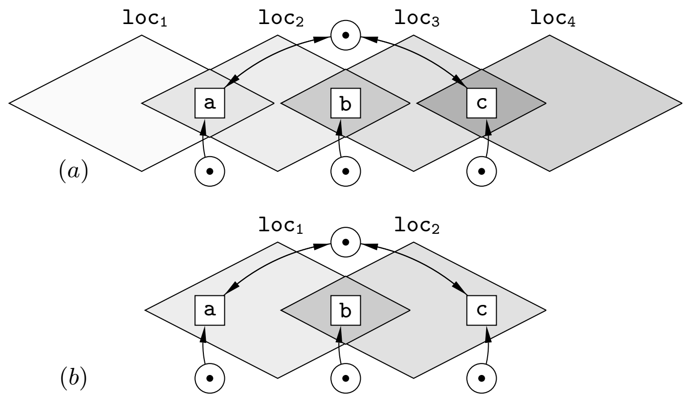


Fig. 4. $cds^{\forall}_{\mathcal{L}}$ and $cds^{\forall\subseteq}_{\mathcal{L}}$ are not pre-order based policies.

no longer the case for the general locality mappings. In the proof of Theorem 2 we established that both $cds^{\exists}_{\mathcal{L}}$ and $cds^{\exists\subseteq}_{\mathcal{L}}$ are pre-order based policies. This, however, does not extend to the remaining two mappings, as we show next.

Proposition 1. *$cds^{\forall}_{\mathcal{L}}$ and $cds^{\forall\subseteq}_{\mathcal{L}}$ are not pre-order based policies.*

Proof. To show the result in the first case, let us assume that $cds^{\forall}_{\mathcal{L}}$ for the PT-net system with localities in Figure 4(a) can be captured by a suitable pre-order $\preceq$ on steps. In the initial marking M_0, the resource enabled steps are: $\{\mathtt{a}\}$, $\{\mathtt{b}\}$, $\{\mathtt{c}\}$, $\{\mathtt{a},\mathtt{b}\}$ and $\{\mathtt{b},\mathtt{c}\}$. Since one of them, $\{\mathtt{b}\}$, is not control enabled there must be a resource enabled step α such that $\{\mathtt{b}\} \preceq \alpha$. As the net is symmetric w.r.t. transitions $\mathtt{a}$ and $\mathtt{c}$, we can suppose w.l.o.g. that $\{\mathtt{b}\} \preceq \{\mathtt{a},\mathtt{b}\}$ or $\{\mathtt{b}\} \preceq \{\mathtt{a}\}$ holds. We then consider the marking M obtained by firing the control enabled step $\{\mathtt{c}\}$, i.e., $M_0[\{\mathtt{c}\}\rangle M$. At such a marking, the steps $\{\mathtt{a}\}$, $\{\mathtt{b}\}$ and $\{\mathtt{a},\mathtt{b}\}$ are both resource enabled and control enabled. But this contradicts the assumption that $\{\mathtt{b}\} \preceq \{\mathtt{a},\mathtt{b}\}$ or $\{\mathtt{b}\} \preceq \{\mathtt{a}\}$ holds.

The result can also be shown for $cds^{\forall\subseteq}_{\mathcal{L}}$ by taking the PT-net system with localities in Figure 4(b) and applying exactly the same reasoning as above. $\square$

Net systems with general localities. For each $\maltese \in \{\exists, \forall, \exists\subseteq, \forall\subseteq\}$, we will call a τ-net system with the bounded step firing policy $cds^{\maltese}_{\mathcal{L}}$ a *$\tau^{\maltese}_{\mathcal{L}}$-net system.* Moreover, we will call $\mathcal{T}$ a *$\tau^{\maltese}_{\mathcal{L}}$-transition system* if AXIOM I and AXIOM II are satisfied for $\mathcal{T}$ with the policy $cds = cds^{\maltese}_{\mathcal{L}}$. Below, for $\maltese \in \{\exists, \forall, \exists\subseteq, \forall\subseteq\}$, we denote:

$$Enbld^{\maltese}_{\mathcal{NP}}(M) \stackrel{\mathrm{df}}{=} enbld_{\mathcal{NP}}(M) \setminus cds^{\maltese}_{\mathcal{L}}(enbld_{\mathcal{NP}}(M)) \; . \tag{3}$$

The above equation defines the set of control enabled steps at a given marking M of a $\tau^{\maltese}_{\mathcal{L}}$-net system $\mathcal{NP}$.

For $\tau_{\mathcal{L}}^{\exists}$-net system, the control enabled steps at a given marking are those which cannot be extended within any of their localities any further as resource enabled steps. Looking at the example from Figure 3(a), we can see that this policy is very restrictive, leaving the step $\{\mathtt{a},\mathtt{b},\mathtt{c}\}$ as the only non-empty step control enabled at the initial marking. The second policy, defined by the policy mapping $cds_{\mathcal{L}}^{\exists\subseteq}$, is less restrictive. It takes into consideration, when extending a resource enabled step, not only localities of this step, but as well the new ones that might be introduced when the step is extended. This time control enabled steps are not only the steps that are 'maximal' within their existing localities, but as well those which can be extended to other resource enabled steps only at the cost of introducing some new localities. This allows, in the example of Figure 3(a), steps $\{\mathtt{a}\}$ and $\{\mathtt{c}\}$ to join the set of control enabled steps at the initial marking. Although they both can be extended to resource enabled steps, $\{\mathtt{a},\mathtt{b}\}$ and $\{\mathtt{b},\mathtt{c}\}$, respectively, the extension has a new locality ($\mathtt{loc}_2$ in the first case, and $\mathtt{loc}_1$ in the second). So, this policy treats steps $\{\mathtt{a}\}$ and $\{\mathtt{c}\}$ as 'maximal' within the sets of their existing localities.

The third policy, defined by the policy mapping $cds_{\mathcal{L}}^{\forall}$, is looking only at the possibility of extending resource enabled steps within their existing localities (no matter whether the extension brings new localities or not), and this time the requirement for being a control enabled step is less demanding. Any resource enabled step that is already 'maximal' within at least one locality per step's transition is considered control enabled. So, $\{\mathtt{a}\}$ and $\{\mathtt{c}\}$ are not control enabled under this policy. They can be extended to bigger resource enabled steps, $\{\mathtt{a},\mathtt{b}\}$ and $\{\mathtt{b},\mathtt{c}\}$, respectively. However, $\{\mathtt{a},\mathtt{b}\}$ and $\{\mathtt{b},\mathtt{c}\}$, that were excluded by the previous policy, are control enabled under this policy, as the first one is 'maximal' within locality $\mathtt{loc}_1$ (for both $\mathtt{a}$ and $\mathtt{b}$) and the second one within locality $\mathtt{loc}_2$ (for both $\mathtt{b}$ and $\mathtt{c}$).

The last policy, defined by the policy mapping $cds_{\mathcal{L}}^{\forall\subseteq}$, is the least restrictive and considers a step to be control enabled if it is 'maximal' within at least one locality per step's transition, or if any extension would introduce some new localities. These permissive conditions mean that only $\{\mathtt{b}\}$ and $\{\mathtt{a},\mathtt{c}\}$ fail to satisfy them as both can be extended to resource enabled steps $\{\mathtt{a},\mathtt{b}\}$ or $\{\mathtt{b},\mathtt{c}\}$ (in the case of $\{\mathtt{b}\}$), and $\{\mathtt{a},\mathtt{b},\mathtt{c}\}$ (in the case of $\{\mathtt{a},\mathtt{c}\}$) within their existing localities.

Synthesis of nets with localities

We obtain an immediate solution of the synthesis problem for all proposed policies based on possibly overlapping localities.

Theorem 3. *For each* $\maltese \in \{\exists, \forall, \exists\subseteq, \forall\subseteq\}$*, a finite step transition system* $\mathcal{T}$ *can be realised by a* $\tau_{\mathcal{L}}^{\maltese}$*-net system* iff $\mathcal{T}$ *is a* $\tau_{\mathcal{L}}^{\maltese}$*-transition system.*

Proof. Follows from Theorems 1 and 2. □

As to the effective construction of synthesised net, it has been demonstrated in [9–11] that this can be easily done for non-overlapping localities in the case of

PT-nets and EN-systems with localities (and with or without inhibitor and read arcs). Similar argument can be applied also in the general setting of overlapping localities and τ-nets corresponding to PT-nets and EN-systems with localities. We omit fairly straightforward details.

4 Saturated localities

In this section, we will look closely at the relationship between control enabled steps and the degree of activation exhibited by different localities involved.

Given a step α which is resource enabled at some marking M of a net with localities $\mathcal{NP}$, a locality $\ell oc \in \mathcal{L}_\alpha$ is *globally saturated* if

$$\alpha + u \notin \mathit{enbld}_{\mathcal{NP}}(M) ,$$

for every transition $u \in \ell oc$. We denote this by $\ell oc \in \mathit{gsatloc}_M(\alpha)$. Similarly, we say that a locality $\ell oc \in \mathcal{L}_\alpha$ is *locally saturated* if

$$\mathcal{L}_u \subseteq \mathcal{L}_\alpha \implies \alpha + u \notin \mathit{enbld}_{\mathcal{NP}}(M) ,$$

for every transition $u \in \ell oc$. We denote this by $\ell oc \in \mathit{lsatloc}_M(\alpha)$. Consider, for example, the net in Figure 4(a) in the initial marking M_0. Then we have:

$$\begin{aligned}
\mathit{gsatloc}_{M_0}(\{\mathtt{a},\mathtt{b}\}) &= \{\mathtt{loc}_1,\mathtt{loc}_2,\mathtt{loc}_3\}\\
\mathit{gsatloc}_{M_0}(\{\mathtt{a}\}) &= \{\mathtt{loc}_1\}\\
\mathit{gsatloc}_{M_0}(\{\mathtt{b}\}) &= \varnothing .
\end{aligned}$$

Intuitively, globally saturated localities of a step α are those which have been 'fully active' during the execution of α. They made α control enabled or contributed to its control enabledness. The relationship between control enabledness and global saturation of localities is given by the following result.

Proposition 2. *Let M be a marking of a $\tau^{\forall}_{\mathcal{L}}$-net ($\tau^{\exists}_{\mathcal{L}}$-net) system $\mathcal{NP}$ such that the set $\mathit{enbld}_{\mathcal{NP}}(M)$ is finite. Then*

(a) $\mathit{Enbld}^{\forall}_{\mathcal{NP}}(M) = \{\alpha \in \mathit{enbld}_{\mathcal{NP}}(M) \mid \mathit{supp}(\alpha) \subseteq \bigcup \mathit{gsatloc}_M(\alpha)\}$.
(b) $\mathit{Enbld}^{\exists}_{\mathcal{NP}}(M) = \{\alpha \in \mathit{enbld}_{\mathcal{NP}}(M) \mid \mathcal{L}_\alpha = \mathit{gsatloc}_M(\alpha)\}$.

Locally saturated localities of a step α are the localities that cannot 'contribute' any more transitions to the extension of the step α (as a resource enabled step) without introducing localities that are not present in α. For the net in Figure 4(a), in the initial marking M_0, we have:

$$\begin{aligned}
\mathit{lsatloc}_{M_0}(\{\mathtt{a},\mathtt{b}\}) &= \{\mathtt{loc}_1,\mathtt{loc}_2,\mathtt{loc}_3\}\\
\mathit{lsatloc}_{M_0}(\{\mathtt{a}\}) &= \{\mathtt{loc}_1,\mathtt{loc}_2\}\\
\mathit{lsatloc}_{M_0}(\{\mathtt{b}\}) &= \{\mathtt{loc}_2,\mathtt{loc}_3\} .
\end{aligned}$$

The difference between locally saturated localities and globally saturated localities is most visible in the case of 'small' steps. Some of their localities can be locally saturated, but not yet globally saturated (see the steps $\{\mathtt{a}\}$ and $\{\mathtt{b}\}$ considered above).

The relationship between control enabledness and local saturation of localities is clarified by the next result.

Proposition 3. *Let M be a marking of a $\tau_{\mathcal{L}}^{\forall\subseteq}$-net ($\tau_{\mathcal{L}}^{\exists\subseteq}$-net) system $\mathcal{NP}$ such that the set $enbld_{\mathcal{NP}}(M)$ is finite. Then*

(a) $Enbld_{\mathcal{NP}}^{\forall\subseteq}(M) = \{\alpha \in enbld_{\mathcal{NP}}(M) \mid supp(\alpha) \subseteq \bigcup lsatloc_M(\alpha)\}$.
(b) $Enbld_{\mathcal{NP}}^{\exists\subseteq}(M) = \{\alpha \in enbld_{\mathcal{NP}}(M) \mid \mathcal{L}_\alpha = lsatloc_M(\alpha)\}$.

Consider now the net discussed in the introduction together with its initial marking M_0. It can be checked that:

$$
\begin{array}{lll}
gsatloc_{M_0}(\{\mathtt{t}_2,\mathtt{t}_3,\mathtt{t}_4\}) & = \{\mathtt{loc}_3\} & \\
gsatloc_{M_0}(\{\mathtt{t}_2,\mathtt{t}_3,\mathtt{t}_4,\mathtt{t}_5,\mathtt{t}_8,\mathtt{t}_9,\mathtt{t}_{10}\}) & = \{\mathtt{loc}_3,\mathtt{loc}_4,\mathtt{loc}_9\} & \\
gsatloc_{M_0}(\{\mathtt{t}_3\}) & = \varnothing & (*) \\
lsatloc_{M_0}(\{\mathtt{t}_3\}) & = \{\mathtt{loc}_2,\mathtt{loc}_3,\mathtt{loc}_4\} & (**) \\
lsatloc_{M_0}(\{\mathtt{t}_0,\mathtt{t}_1,\ldots,\mathtt{t}_{n-1}\}) & = \{\mathtt{loc}_0,\mathtt{loc}_1,\ldots,\mathtt{loc}_{n-1}\} & \\
lsatloc_{M_0}(\{\mathtt{t}_2,\mathtt{t}_4\}) & = \{\mathtt{loc}_1,\mathtt{loc}_2,\mathtt{loc}_4,\mathtt{loc}_5\}\ . &
\end{array}
$$

Hence the first interpretation of the overlapping localities in Figure 1 conforms to the rules of $\tau_{\mathcal{L}}^{\forall}$-net systems (but not $\tau_{\mathcal{L}}^{\exists\subseteq}$-net systems, on account of $(**)$), and the second interpretation conforms to the rules of $\tau_{\mathcal{L}}^{\exists\subseteq}$-net systems (but not $\tau_{\mathcal{L}}^{\forall}$-net systems, on account of $(*)$).

5 Towards synthesis with unknown localities

The synthesis result presented in the previous section, Theorem 3, has been obtained assuming that the locality set $\mathcal{L}$ was given. However, localities might be (partially) unknown, and part of the outcome of a successful synthesis procedure would be a suitable or, in the terminology used below, *good* locality set. Clearly, as there are only finitely many different locality sets, the synthesis procedure could simply enumerate them and check each one in turn using Theorem 3. This, however, would be impractical as the number of locality sets is double exponential in the number of transitions. We will now present our initial findings concerning possible reductions of the number of potentially good locality sets. It is worth noting that, in general, for a given τ-net there can be different locality sets yielding the same reachability graph. The example in Figure 5 shows that this holds for all the locality based policies considered in this paper as it is based on disjoint localities.

In what follows, we assume that $\mathcal{T}$ is *finite*. We also assume that we have checked that, for every state q of $\mathcal{T}$, the set of steps $enbld_{\mathcal{T},\tau}(q)$ is finite; otherwise $\mathcal{T}$ could not be isomorphic to the concurrent reachability graph of any τ-net with localities (see AXIOM II and Theorem 3). For a set Y and a finite set of sets

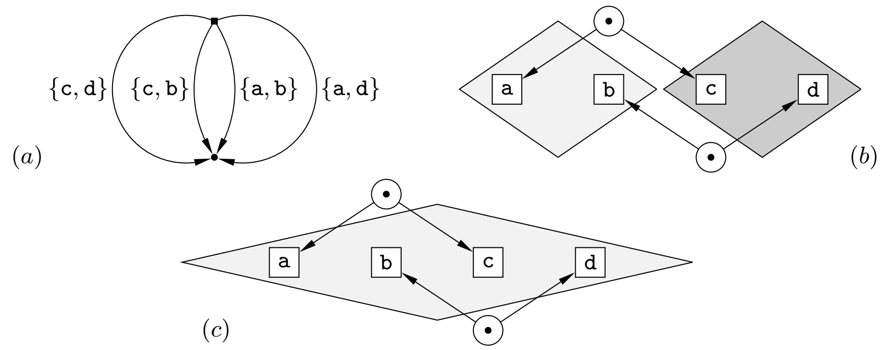


Fig. 5. Two different sets of localities, (b) and (c), for a PT-net system giving rise to the same concurrent reachability graph (a).

$\mathcal{Z} = \{Z_1, \ldots, Z_k\}$ we denote by $Y \cap \mathcal{Z}$ the set of all non-empty intersections of Y and the Z_i's, i.e., the set $\{Y \cap Z_i \mid i \leq k \wedge Y \cap Z_i \neq \varnothing\}$.

In the rest of this section, for every state q of the step transition system $\mathcal{T}$, and any two locality sets, $\mathcal{L}$ and $\mathcal{L}'$:

- $\mathit{allSteps}_q$ is the set of all steps labelling arcs outgoing from q.
- T_q is the set of all net transitions occurring in the steps of $\mathit{allSteps}_q$.
- $\mathit{clusters}_q^{\mathcal{L}}$ is the set of (locality) *clusters* at q, defined as $T_q \cap \mathcal{L}$.
- $\mathcal{L}$ and $\mathcal{L}'$ are *node-consistent* if $\mathit{clusters}_q^{\mathcal{L}} = \mathit{clusters}_q^{\mathcal{L}'}$, for every state q of the transition system $\mathcal{T}$.

Note that the clusters are all the non-empty projections of the localities onto the transitions fired at an individual state.

A major result concerning locality sets is that they are equally suitable for being good locality sets whenever they induce the same clusters in each node of the step transition system.

Theorem 4. *Let $\mathcal{L}^0$ and $\mathcal{L}^1$ be two node-consistent locality sets. Then $\mathcal{T}$ is $\tau_{\mathcal{L}^0}^{\maltese}$-transition system* iff *$\mathcal{T}$ is $\tau_{\mathcal{L}^1}^{\maltese}$-transition system, for every $\maltese \in \{\exists, \forall, \exists \subseteq, \forall \subseteq\}$.*

The above theorem implies that a good locality set can be arbitrarily modified to yield another good locality set as long as both are node-consistent (there is no need to re-check the two axioms involved in Theorem 3). This should facilitate searching for an optimal good locality set starting from some initial choice (for example, one might prefer to have as few localities per transition as possible, or as many transitions per locality as possible, or as few localities as possible, etc).

Theorem 4 leads to another important observation, namely that in order to be a good locality set, all that matters are the projections of the localities onto transition sets enabled at the states of the transition system $\mathcal{T}$. As a consequence, the construction of a good locality set can be turned into modular process, in the following way.

First, for each state q and $\maltese \in \{\exists, \forall, \exists \subseteq, \forall \subseteq\}$, we identify possible cluster-sets $\mathit{ClSets}_q^{\maltese}$ of transitions in T_q induced by hypothetical good locality sets. Each such cluster-set $\mathit{clSet} = \{C_1, \ldots, C_k\} \in \mathit{ClSets}_q^{\maltese}$ is a cover of T_q and:

$$\mathit{enbld}_{\mathcal{T}}^{\maltese}(q) = \mathit{enbld}_{\mathcal{T},\tau}(q) \setminus \mathit{cds}_{\mathit{clSet}}^{\maltese}(\mathit{enbld}_{\mathcal{T},\tau}(q))$$

where we have the following (below $\mathit{clSet}_t \stackrel{\mathrm{df}}{=} \{C_i \in \mathit{clSet} \mid t \in C_i\}$ and $\mathit{clSet}_\alpha \stackrel{\mathrm{df}}{=} \{C_i \in \mathit{clSet} \mid \exists t \in \alpha : \ t \in C_i\}$):

$$\begin{aligned}
&\mathit{cds}_{\mathit{clSet}}^{\exists}(X) \stackrel{\mathrm{df}}{=} \{\alpha \in X \mid \exists C_i \in \mathit{clSet}_\alpha \ \exists \alpha + t \in X : \ t \in C_i\} \\
&\mathit{cds}_{\mathit{clSet}}^{\exists\subseteq}(X) \stackrel{\mathrm{df}}{=} \{\alpha \in X \mid \exists C_i \in \mathit{clSet}_\alpha \ \exists \alpha + t \in X : \ t \in C_i \ \wedge \ \mathit{clSet}_t \subseteq \mathit{clSet}_\alpha\} \\
&\mathit{cds}_{\mathit{clSet}}^{\forall}(X) \stackrel{\mathrm{df}}{=} \{\alpha \in X \mid \exists v \in \alpha \ \forall C_i \in \mathit{clSet}_v \ \exists \alpha + t \in X : \ t \in C_i\} \\
&\mathit{cds}_{\mathit{clSet}}^{\forall\subseteq}(X) \stackrel{\mathrm{df}}{=} \{\alpha \in X \mid \exists v \in \alpha \ \forall C_i \in \mathit{clSet}_v \ \exists \alpha + t \in X : \\
&\qquad\qquad\qquad\qquad\qquad\qquad t \in C_i \ \wedge \ \mathit{clSet}_t \subseteq \mathit{clSet}_\alpha\} \ .
\end{aligned}$$

We can then select different cluster-sets (one per each state of the step transition system) and check whether combining them together yields a good locality set. Such a procedure was used in [11] to construct 'canonical' locality sets for the case of non-overlapping localities (and the combining of cluster-sets was based on the operation of transitive closure). This effort can be reduced by observing that some cluster-sets cannot be combined to yield a good locality set. A simple check is provided by the following result.

Proposition 4. *Let q and q' be two states of the transition system $\mathcal{T}$ and $\maltese \in \{\exists, \forall, \exists \subseteq, \forall \subseteq\}$. Moreover, let $\mathit{clSet} \in \mathit{ClSets}_q^{\maltese}$ and $\mathit{clSet}' \in \mathit{ClSets}_{q'}^{\maltese}$ be cluster-sets such that $T_q \cap \mathit{clSet}' \neq T_{q'} \cap \mathit{clSet}$. Then there is no locality set $\mathcal{L}$ which is good w.r.t. $\maltese$ as well as satisfying $T_q \cap \mathcal{L} = \mathit{clSet}$ and $T_{q'} \cap \mathcal{L} = \mathit{clSet}'$.*

It is, in general, difficult to estimate how many combinations of cluster-sets one needs to consider or how many of these yield good locality sets. One can, however, obtain important insights if one looks for solutions in a specific class of nets, or if the locality set is partially known or constrained (for example, if two specific transitions cannot share a locality).

6 Concluding remarks

In this paper, we introduced four different semantics of nets based on transition localities. In the future research, we plan to work on an efficient synthesis procedure of PT-nets with localities with unknown locality sets. The problem has been investigated in [11] for non-overlapping localities, and some initial results have been obtained in [12] for the case of $\mathit{cds}_{\mathcal{L}}^{\forall}$.

Acknowledgement We would like to thank Piotr Chrzastowski-Wachtel for his suggestion to investigate nets with overlapping localities. We are indebted to the reviewers of both [12] and this paper for their detailed and helpful comments; in particular, for a suggestion which has led to the introduction of two policies, $cds^{\exists}_{\mathcal{L}}$ and $cds^{\exists\subseteq}_{\mathcal{L}}$, as well as identifying mistakes in the treatment of some of the results. This research was supported by Epsrc Verdad project, and Nsfc Grants 60910004 and 2010CB328102.

References

1. Badouel, E., Bernardinello, L., Darondeau, Ph.: The Synthesis Problem for Elementary Net Systems is NP-complete. Theoretical Computer Science **186** (1997) 107–134
2. Badouel, E., Darondeau, Ph.: Theory of Regions. In: Reisig, W., Rozenberg, G. (eds.): Lectures on Petri Nets I: Basic Models, Advances in Petri Nets. Lecture Notes in Computer Science **1491**. Springer-Verlag, Berlin Heidelberg New York (1998) 529–586
3. Bernardinello, L.: Synthesis of Net Systems In: Marsan, M.A. (ed.): Application and Theory of Petri Nets 1993. Lecture Notes in Computer Science **691**. Springer-Verlag, Berlin Heidelberg New York (1993) 89–105
4. Darondeau, P., Koutny, M., Pietkiewicz-Koutny, M., Yakovlev, A.: Synthesis of Nets with Step Firing Policies. Fundamenta Informaticae **94** (2009) 275–303
5. Dasgupta, S., Potop-Butucaru, D., Caillaud, B., Yakovlev, A.: Moving from Weakly Endochronous Systems to Delay-Insensitive Circuits. Electronic Notes in Theoretical Computer Science **146** (2006) 81–103
6. Desel, J., Reisig, W.: The Synthesis Problem of Petri Nets. Acta Informatica **33** (1996) 297–315
7. Kleijn, J., Koutny, M.: Processes of Membrane systems with Promoters and Inhibitors. Theoretical Computer Science **404** (2008) 112–126
8. Kleijn, H.C.M., Koutny, M., Rozenberg, G.: Towards a Petri Net Semantics for Membrane Systems. In: Freund, R., Paun, G., Rozenberg, G., Salomaa, A. (eds.): WMC 2005. Lecture Notes in Computer Science **3850**. Springer-Verlag, Berlin Heidelberg New York (2006) 292–309
9. Koutny, M., Pietkiewicz-Koutny, M.: Transition Systems of Elementary Net Systems with Localities. In: Baier, C., Hermanns, H. (eds.): CONCUR 2006. Lecture Notes in Computer Science **4137**. Springer-Verlag, Berlin Heidelberg New York (2006) 173–187
10. Koutny, M., Pietkiewicz-Koutny, M.: Synthesis of Elementary Net Systems with Context Arcs and Localities. Fundamenta Informaticae **88** (2008) 307–328
11. Koutny, M., Pietkiewicz-Koutny, M.: Synthesis of Petri Nets with Localities. Scientific Annals of Computer Science **19** (2009) 1–23
12. Koutny, M., Pietkiewicz-Koutny, M.: Synthesis of General Petri Nets with Localities. In: Desel, J., Yakovlev, A. (eds.): Application of Region Theory (ART). IEEE (2010) 15–28
13. Mukund, M.: Petri Nets and Step Transition Systems. International Journal of Foundations of Computer Science **3** (1992) 443–478
14. Nielsen, M., Rozenberg, G., Thiagarajan, P.S.: Elementary transition systems. Theoretical Compututer Science **96** (1992) 3–33

15. Păun, G.: Membrane Computing, An Introduction. Springer-Verlag, Berlin Heidelberg New York (2002)
16. Pietkiewicz-Koutny, M.: The Synthesis Problem for Elementary Net Systems with Inhibitor Arcs. Fundamenta Informaticae **40** (1999) 251–283

Appendix

Proof of Theorem 2

We first show that $cds_{\mathcal{L}}^{\exists}$ is a pre-order based bounded step firing policy.

In what follows, we denote $\gamma <^{1} \gamma'$ if there is a transition t such that $\gamma' = \gamma + t$. For all steps α and β, we define $\alpha \preceq_{\mathcal{L}}^{\exists} \beta$ if $\alpha = \beta$ or if $\alpha < \beta$ and there are steps $\alpha = \alpha_1 <^{1} \ldots <^{1} \alpha_k = \beta$ such that, for all $1 \leq i < k$, $\mathcal{L}_{\alpha_i} \cap \mathcal{L}_{\alpha_{i+1}-\alpha_i} \neq \varnothing$. Clearly, $\preceq_{\mathcal{L}}^{\exists}$ is both transitive and reflexive, and so it is a pre-order.

We therefore have, for every finite set of steps X:

$$\begin{aligned} cds_{\preceq_{\mathcal{L}}^{\exists}}(X) &= \{\alpha \in X \mid \alpha \neq \varnothing \wedge \exists \beta \in X : \alpha \prec_{\mathcal{L}}^{\exists} \beta\} \\ &= \{\alpha \in X \mid \alpha \neq \varnothing \wedge \exists \beta \in X : \alpha < \beta \wedge \exists \alpha = \alpha_1 <^{1} \ldots <^{1} \alpha_k = \beta \\ &\qquad\qquad\qquad \forall 1 \leq i < k : \mathcal{L}_{\alpha_i} \cap \mathcal{L}_{\alpha_{i+1}-\alpha_i} \neq \varnothing\} \,. \end{aligned}$$

What we need to show is that:

$$\alpha \in cds_{\preceq_{\mathcal{L}}^{\exists}}(X) \Longleftrightarrow \alpha \in \{\alpha \in X \mid \exists v \in \alpha \ \exists \ell oc \in \mathcal{L}_v \ \exists \alpha + t \in X : \ t \in \ell oc\} \,.$$

($\Longleftarrow$) Suppose that $v \in \alpha$, $\ell oc \in \mathcal{L}_v$ and $\alpha + t \in X$ are such that $t \in \ell oc$. Since $v \in \alpha$, we have $\alpha \neq \varnothing$. We can take $\alpha_1 = \alpha$ and $\alpha_2 = \beta = \alpha + t \in X$. Then $\mathcal{L}_\alpha \cap \mathcal{L}_{\beta-\alpha} \neq \varnothing$ because $\beta - \alpha = t$ and $t \in \ell oc$ and $\ell oc \in \mathcal{L}_\alpha$ (as $v \in \alpha$ and $\ell oc \in \mathcal{L}_v$).

($\Longrightarrow$) Suppose that $\alpha, \beta \in X$ and $\alpha = \alpha_1 <^{1} \ldots <^{1} \alpha_k = \beta$ are such that $\mathcal{L}_{\alpha_i} \cap \mathcal{L}_{\alpha_{i+1}-\alpha_i} \neq \varnothing$, for $1 \leq i < k$. Hence there is t such that $\alpha_2 = \alpha_1 + t$ and $\mathcal{L}_{\alpha_1} \cap \mathcal{L}_{\alpha_2-\alpha_1} \neq \varnothing$. This means that there is $v \in \alpha$ ($\alpha \neq \varnothing$)such that $\mathcal{L}_v \cap \mathcal{L}_t \neq \varnothing$. So, there are $v \in \alpha$ and $\ell oc \in \mathcal{L}_v$ such that $t \in \ell oc$. All we still need to show is that $\alpha + t \in X$ which follows from $\alpha + t \leq \beta \in X$ and Assumption 1.

Next, we show that $cds_{\mathcal{L}}^{\exists\subseteq}$ is also a pre-order based bounded step firing policy. For all steps α and β, we define $\alpha \preceq_{\mathcal{L}}^{\exists\subseteq} \beta$ if $\alpha \preceq_{\mathcal{L}}^{\exists} \beta$ and $\mathcal{L}_\alpha = \mathcal{L}_\beta$. Clearly, $\preceq_{\mathcal{L}}^{\exists\subseteq}$ is both transitive and reflexive as $\preceq_{\mathcal{L}}^{\exists}$ is.

We therefore have, for every finite set of steps X:

$$\begin{aligned} cds_{\preceq_{\mathcal{L}}^{\exists\subseteq}}(X) &= \{\alpha \in X \mid \alpha \neq \varnothing \wedge \exists \beta \in X : \alpha \prec_{\mathcal{L}}^{\exists\subseteq} \beta\} \\ &= \{\alpha \in X \mid \alpha \neq \varnothing \wedge \exists \beta \in X : \alpha \prec_{\mathcal{L}}^{\exists} \beta \wedge \mathcal{L}_\alpha = \mathcal{L}_\beta\} \,. \end{aligned}$$

We need to show that:

$$\begin{aligned} &\alpha \in cds_{\preceq_{\mathcal{L}}^{\exists\subseteq}}(X) \Longleftrightarrow \\ &\qquad \alpha \in \{\alpha \in X \mid \exists v \in \alpha \ \exists \ell oc \in \mathcal{L}_v \ \exists \alpha + t \in X : \ t \in \ell oc \wedge \mathcal{L}_t \subseteq \mathcal{L}_\alpha\} \,. \end{aligned}$$

The proof is similar to that for $\preceq^{\exists}_{\mathcal{L}}$.

To show that $cds^{\forall}_{\mathcal{L}}$ is a bounded step firing policy, we need to prove that if $X \in \mathcal{X}_\tau$ is finite and $Y \subseteq X$ and $X \setminus cds^{\forall}_{\mathcal{L}}(X) \subseteq Y$ and $\alpha \in cds^{\forall}_{\mathcal{L}}(X) \cap Y$, then $\alpha \in cds^{\forall}_{\mathcal{L}}(Y)$. Before proceeding with the proof, we note that in the proofs of the key Theorems 4.1 and 4.3 of [4] from which Theorem 1 in this paper is derived, the set Y appearing in Definition 3(2) is always taken to be of the form $enbld_{\mathcal{T},\tau}(q)$. Hence, due to Assumption 1, we can assume in Definition 3(2) that Y is sub-step closed.

We first observe that $\max_{\leq}(X) \cap cds^{\forall}_{\mathcal{L}}(X) = \varnothing$ and so we have $\max_{\leq}(X) \subseteq X \setminus cds^{\forall}_{\mathcal{L}}(X) \subseteq Y$. Then we observe that since X is finite and $\alpha \in cds^{\forall}_{\mathcal{L}}(X)$, there is a transition $v \in \alpha$ such that for all $\ell oc \in \mathcal{L}_v$ there exists $\alpha+\beta \in \max_{\leq}(X) \subseteq Y$ with $\ell oc \in \mathcal{L}_\beta$. Since, as we explained above, the set Y may be assumed to be sub-step closed, there exists $t \in \beta$ such that $\alpha + t \in Y$ and $\ell oc \in \mathcal{L}_t$ ($t \in \ell oc$). This and the fact that Y is finite (as $Y \subseteq X$) means that we have $\alpha \in cds^{\forall}_{\mathcal{L}}(Y)$.

Finally, we show that $cds^{\forall\subseteq}_{\mathcal{L}}$ is also a bounded step firing policy. We need to show that if $X \in X_\tau$ is finite and $Y \subseteq X$ and $X \setminus cds^{\forall\subseteq}_{\mathcal{L}} \subseteq Y$ and $\alpha \in cds^{\forall\subseteq}_{\mathcal{L}}(X) \cap Y$, then $\alpha \in cds^{\forall\subseteq}_{\mathcal{L}}(Y)$.

We first observe that $\max_{\leq}(X) \cap cds^{\forall\subseteq}_{\mathcal{L}}(X) = \varnothing$ and so we have $\max_{\leq}(X) \subseteq X \setminus cds^{\forall\subseteq}_{\mathcal{L}}(X) \subseteq Y$. Then we observe that since X is finite and $\alpha \in cds^{\forall\subseteq}_{\mathcal{L}}(X)$, there is a transition $v \in \alpha$ such that, for all $\ell oc \in \mathcal{L}_v$, there exists $\alpha+t \leq \alpha+\beta \in \max_{\leq}(X) \subseteq Y$ with $\ell oc \in \mathcal{L}_t$ and $\mathcal{L}_t \subseteq \mathcal{L}_\alpha$. This and the fact that Y is finite (as $Y \subseteq X$) means that we have $\alpha \in cds^{\forall\subseteq}_{\mathcal{L}}(Y)$.

Proof of Proposition 2

To show (a) we observe that the following holds.

$$\begin{array}{lr}
\alpha \in \{\alpha \in enbld_{\mathcal{NP}}(M) \mid supp(\alpha) \subseteq \bigcup gsatloc_M(\alpha)\} & \Leftrightarrow \\
\alpha \in enbld_{\mathcal{NP}}(M) \wedge \forall v \in \alpha\ \exists \ell oc \in gsatloc_M(\alpha) : v \in \ell oc & \Leftrightarrow \\
\alpha \in enbld_{\mathcal{NP}}(M) \wedge \forall v \in \alpha\ \exists \ell oc \in gsatloc_M(\alpha) : \ell oc \in \mathcal{L}_v & \Leftrightarrow \\
\alpha \in enbld_{\mathcal{NP}}(M) \wedge \forall v \in \alpha\ \exists \ell oc \in \mathcal{L}_v : \ell oc \in gsatloc_M(\alpha) \ \ \Leftrightarrow & \text{(def. } gsatloc_M(\alpha)) \\
\alpha \in enbld_{\mathcal{NP}}(M) \wedge \forall v \in \alpha\ \exists \ell oc \in \mathcal{L}_v \forall t \in \ell oc : \alpha + t \notin enbld_{\mathcal{NP}}(M) & \Leftrightarrow \\
\alpha \in enbld_{\mathcal{NP}}(M) \wedge \forall v \in \alpha\ \exists \ell oc \in \mathcal{L}_v \forall \alpha + t \in enbld_{\mathcal{NP}}(M) : & \\
\hfill t \notin \ell oc \ \ \Leftrightarrow & \text{(by (1) and (3))} \\
\alpha \in Enbld^{\forall}_{\mathcal{NP}}(M)\ . &
\end{array}$$

Then, to show (b), we proceed as follows.

$$\begin{array}{lr}
\alpha \in \{\alpha \in enbld_{\mathcal{NP}}(M) \mid \mathcal{L}_\alpha = gsatloc_M(\alpha)\} & \Leftrightarrow \\
\alpha \in enbld_{\mathcal{NP}}(M) \wedge \forall \ell oc \in \mathcal{L}_\alpha\ \forall t \in \ell oc : \alpha + t \notin enbld_{\mathcal{NP}}(M) & \Leftrightarrow \\
\alpha \in enbld_{\mathcal{NP}}(M) \wedge \forall \ell oc \in \mathcal{L}_\alpha\ \forall \alpha + t \in enbld_{\mathcal{NP}}(M) : t \notin \ell oc \Leftrightarrow & \text{(by (1) and (3))} \\
\alpha \in Enbld^{\exists}_{\mathcal{NP}}(M)\ . &
\end{array}$$

Proof of Proposition 3

To show (a) we observe that the following holds.

$$
\begin{array}{ll}
\alpha \in \{\alpha \in \mathit{enbld}_{\mathcal{NP}}(M) \mid \mathit{supp}(\alpha) \subseteq \bigcup \mathit{lsatloc}_M(\alpha)\} & \Leftrightarrow \\
\alpha \in \mathit{enbld}_{\mathcal{NP}}(M) \wedge \forall v \in \alpha \; \exists \ell oc \in \mathit{lsatloc}_M(\alpha) : v \in \ell oc & \Leftrightarrow \\
\alpha \in \mathit{enbld}_{\mathcal{NP}}(M) \wedge \forall v \in \alpha \; \exists \ell oc \in \mathit{lsatloc}_M(\alpha) : \ell oc \in \mathcal{L}_v & \Leftrightarrow \\
\alpha \in \mathit{enbld}_{\mathcal{NP}}(M) \wedge \forall v \in \alpha \; \exists \ell oc \in \mathcal{L}_v : \ell oc \in \mathit{lsatloc}_M(\alpha) & \Leftrightarrow \text{(def. } \mathit{lsatloc}_M(\alpha)\text{)} \\
\alpha \in \mathit{enbld}_{\mathcal{NP}}(M) \wedge \forall v \in \alpha \; \exists \ell oc \in \mathcal{L}_v \forall t \in \ell oc : & \\
\qquad \mathcal{L}_t \subseteq \mathcal{L}_\alpha \Rightarrow \alpha + t \notin \mathit{enbld}_{\mathcal{NP}}(M) & \Leftrightarrow \\
\alpha \in \mathit{enbld}_{\mathcal{NP}}(M) \wedge \forall v \in \alpha \; \exists \ell oc \in \mathcal{L}_v \forall t \in \ell oc : & \\
\qquad \neg(\mathcal{L}_t \subseteq \mathcal{L}_\alpha \wedge \alpha + t \in \mathit{enbld}_{\mathcal{NP}}(M)) & \Leftrightarrow \\
\alpha \in \mathit{enbld}_{\mathcal{NP}}(M) \wedge \forall v \in \alpha \; \exists \ell oc \in \mathcal{L}_v \forall t \in \ell oc : & \\
\qquad \mathcal{L}_t \setminus \mathcal{L}_\alpha \neq \varnothing \vee \alpha + t \notin \mathit{enbld}_{\mathcal{NP}}(M) & \Leftrightarrow \\
\alpha \in \mathit{enbld}_{\mathcal{NP}}(M) \wedge \forall v \in \alpha \; \exists \ell oc \in \mathcal{L}_v \forall t : & \\
\qquad t \in \ell oc \Rightarrow (\mathcal{L}_t \setminus \mathcal{L}_\alpha \neq \varnothing \vee \alpha + t \notin \mathit{enbld}_{\mathcal{NP}}(M)) & \Leftrightarrow \\
\alpha \in \mathit{enbld}_{\mathcal{NP}}(M) \wedge \forall v \in \alpha \; \exists \ell oc \in \mathcal{L}_v \forall t : & \\
\qquad t \notin \ell oc \vee \mathcal{L}_t \setminus \mathcal{L}_\alpha \neq \varnothing \vee \alpha + t \notin \mathit{enbld}_{\mathcal{NP}}(M) & \Leftrightarrow \\
\alpha \in \mathit{enbld}_{\mathcal{NP}}(M) \wedge \forall v \in \alpha \; \exists \ell oc \in \mathcal{L}_v \forall t : & \\
\qquad \alpha + t \in \mathit{enbld}_{\mathcal{NP}}(M) \Rightarrow (t \notin \ell oc \vee \mathcal{L}_t \setminus \mathcal{L}_\alpha \neq \varnothing) & \Leftrightarrow \\
\alpha \in \mathit{enbld}_{\mathcal{NP}}(M) \wedge \forall v \in \alpha \; \exists \ell oc \in \mathcal{L}_v \forall \alpha + t \in \mathit{enbld}_{\mathcal{NP}}(M) : & \\
\qquad t \notin \ell oc \vee \mathcal{L}_t \setminus \mathcal{L}_\alpha \neq \varnothing & \Leftrightarrow \text{(by (1) and (3))} \\
\alpha \in \mathit{Enbld}^{\forall\subseteq}_{\mathcal{NP}}(M) \; . &
\end{array}
$$

Then, to show (b), we proceed as follows.

$$
\begin{array}{ll}
\alpha \in \{\alpha \in \mathit{enbld}_{\mathcal{NP}}(M) \mid \mathcal{L}_\alpha = \mathit{lsatloc}_M(\alpha)\} & \Leftrightarrow \\
\alpha \in \mathit{enbld}_{\mathcal{NP}}(M) \wedge \forall \ell oc \in \mathcal{L}_\alpha \; \forall t \in \ell oc : \mathcal{L}_t \subseteq \mathcal{L}_\alpha \Rightarrow \alpha + t \notin \mathit{enbld}_{\mathcal{NP}}(M) & \Leftrightarrow \\
\alpha \in \mathit{enbld}_{\mathcal{NP}}(M) \wedge \forall \ell oc \in \mathcal{L}_\alpha \; \forall \alpha + t \in \mathit{enbld}_{\mathcal{NP}}(M) : & \\
\qquad t \notin \ell oc \vee \mathcal{L}_t \not\subseteq \mathcal{L}_\alpha & \Leftrightarrow \text{(by (1) and (3))} \\
\alpha \in \mathit{Enbld}^{\exists\subseteq}_{\mathcal{NP}}(M) \; . &
\end{array}
$$

Proof of Theorem 4

Suppose that $\mathcal{T}$ is $\tau^{\maltese}_{\mathcal{L}^0}$-transition system. First we notice that AXIOM I does not depend on the locality set. For AXIOM II and $\mathcal{L}^1$ it suffices to show that, for each state q of $\mathcal{T}$:

$$\mathit{cds}^{\maltese}_{\mathcal{L}^0}(\mathit{enbld}_{\mathcal{T},\tau}(q)) = \mathit{cds}^{\maltese}_{\mathcal{L}^1}(\mathit{enbld}_{\mathcal{T},\tau}(q)) \; . \tag{4}$$

We observe that since the maximal steps in $\mathit{enbld}_{\mathcal{T},\tau}(q)$ never belong to the set $\mathit{cds}^{\maltese}_{\mathcal{L}^0}(\mathit{enbld}_{\mathcal{T},\tau}(q))$ and AXIOM II holds for $\mathcal{L}^0$ we have:

$$\forall \alpha \in \mathit{enbld}_{\mathcal{T},\tau}(q) \; \forall u \in \alpha : \; u \in T_q \; . \tag{5}$$

We now take $i \in \{0, 1\}$ and consider four cases.

Case 1: $\alpha \in cds^{\exists\subseteq}_{\mathcal{L}^i}(enbld_{\mathcal{T},\tau}(q))$. Then $\alpha \in enbld_{\mathcal{T},\tau}(q)$ and there are $v \in \alpha$, $\ell oc \in \mathcal{L}^i_v$ and $\alpha + t \in enbld_{\mathcal{T},\tau}(q)$ such that $t \in \ell oc$ and $\mathcal{L}^i_t \subseteq \mathcal{L}^i_\alpha$. Hence, by (5), we have that $v, t \in T_q$ and $v, t \in C \in clusters^{\mathcal{L}^i}_q$, where $C = T_q \cap \ell oc$. By the node-consistency of $\mathcal{L}^i$ and $\mathcal{L}^{1-i}$ we obtain that there is $\ell oc' \in \mathcal{L}^{1-i}$ such that $C = T_q \cap \ell oc'$. Hence $\ell oc' \in \mathcal{L}^{1-i}_v$ and $t \in \ell oc'$.
Now suppose that $\mathcal{L}^{1-i}_t \not\subseteq \mathcal{L}^{1-i}_\alpha$, and so there is $\ell oc'' \in \mathcal{L}^{1-i}_t \setminus \mathcal{L}^{1-i}_\alpha$. Then, there exists $C' \in clusters^{\mathcal{L}^{1-i}}_q$, where $C' = T_q \cap \ell oc''$. By the node-consistency of $\mathcal{L}^i$ and $\mathcal{L}^{1-i}$ there is $\ell oc''' \in \mathcal{L}^i$ such that $T_q \cap \ell oc''' = C'$. So $\ell oc''' \in \mathcal{L}^i_t$. Since, by (5), all $u \in \alpha$ are such that $u \in T_q$, and $\ell oc'' \notin \mathcal{L}^{1-i}_\alpha$, we have $\ell oc''' \notin \mathcal{L}^i_\alpha$. Hence $\ell oc''' \in \mathcal{L}^i_t \setminus \mathcal{L}^i_\alpha$, producing a contradiction with $\mathcal{L}^i_t \subseteq \mathcal{L}^i_\alpha$.
As a result, $\alpha \in cds^{\exists\subseteq}_{\mathcal{L}^{1-i}}(enbld_{\mathcal{T},\tau}(q))$.

Case 2: $\alpha \in cds^{\exists}_{\mathcal{L}^i}(enbld_{\mathcal{T},\tau}(q))$. We proceed as in Case 1, ignoring the parts concerned with $\mathcal{L}^i_t \subseteq \mathcal{L}^i_\alpha$ and $\mathcal{L}^{1-i}_t \subseteq \mathcal{L}^{1-i}_\alpha$.

Case 3: $\alpha \in cds^{\forall\subseteq}_{\mathcal{L}^i}(enbld_{\mathcal{T},\tau}(q))$. Then $\alpha \in enbld_{\mathcal{T},\tau}(q)$ and there is $v \in \alpha$ such that for all $\ell oc \in \mathcal{L}^i_v$ there is $\alpha + t \in enbld_{\mathcal{T},\tau}(q)$ satisfying $t \in \ell oc$ and $\mathcal{L}^i_t \subseteq \mathcal{L}^i_\alpha$.

Let us now consider any $\ell oc' \in \mathcal{L}^{1-i}_v$ and take $C = T_q \cap \ell oc'$. By (5), we have $v \in C$ and so, by the node-consistency of $\mathcal{L}^i$ and $\mathcal{L}^{1-i}$, there is $\ell oc \in \mathcal{L}^i_v$ with $C = T_q \cap \ell oc$. We know that there is $\alpha + t \in enbld_{\mathcal{T},\tau}(q)$ satisfying $t \in \ell oc$ and $\mathcal{L}^i_t \subseteq \mathcal{L}^i_\alpha$ (and also, by (5), $t \in T_q$). Hence $t \in C$. We thus have $t \in \ell oc'$.
Now suppose that $\mathcal{L}^{1-i}_t \not\subseteq \mathcal{L}^{1-i}_\alpha$, and so there is $\ell oc'' \in \mathcal{L}^{1-i}_t \setminus \mathcal{L}^{1-i}_\alpha$. Then, there exists $C' \in clusters^{\mathcal{L}^{1-i}}_q$, where $C' = T_q \cap \ell oc''$. By the node-consistency of $\mathcal{L}^i$ and $\mathcal{L}^{1-i}$ there is $\ell oc''' \in \mathcal{L}^i$ such that $T_q \cap \ell oc''' = C'$. So $\ell oc''' \in \mathcal{L}^i_t$. Since, by (5), all $u \in \alpha$ are such that $u \in T_q$, and $\ell oc'' \notin \mathcal{L}^{1-i}_\alpha$, we have $\ell oc''' \notin \mathcal{L}^i_\alpha$. Hence $\ell oc''' \in \mathcal{L}^i_t \setminus \mathcal{L}^i_\alpha$, producing a contradiction with $\mathcal{L}^i_t \subseteq \mathcal{L}^i_\alpha$.
As a result, $\alpha \in cds^{\forall\subseteq}_{\mathcal{L}^{1-i}}(enbld_{\mathcal{T},\tau}(q))$.

Case 4: $\alpha \in cds^{\forall}_{\mathcal{L}^i}(enbld_{\mathcal{T},\tau}(q))$. We proceed as in Case 3, ignoring the parts concerned with $\mathcal{L}^i_t \subseteq \mathcal{L}^i_\alpha$ and $\mathcal{L}^{1-i}_t \subseteq \mathcal{L}^{1-i}_\alpha$.