Processes of Membrane Systems with Promoters and Inhibitors

Jetty Kleijn [a] Maciej Koutny [b,*]

[a] *LIACS, Leiden University,P.O.Box 9512, NL-2300 RA Leiden, The Netherlands*

[b] *School of Computing Science, Newcastle University, Newcastle upon Tyne, NE1 7RU, United Kingdom*

Abstract

Membrane systems (with promoters and inhibitors) are a computational model inspired by the way living cells are divided by membranes into compartments where chemical reactions may take place. We consider synchrony and asynchrony between executed reactions in the computations of such systems using Petri nets and their processes as a formal behavioural model. We first discuss different definitions of individual computational steps, and show how they can be rendered within the Petri net domain by assigning all transitions localities corresponding to the compartments, and using activator and inhibitor arcs. The non-sequential semantics of the resulting nets is formalised through processes based on occurrence nets augmented with additional information about localities and activator/inhibitor arcs. Such processes provide a convenient tool for analysing synchrony and asynchrony in the executions of membrane systems and shed light on the causal relationships between the reactions taking place.

Key words: membrane system, promoter, inhibitor, Petri net, locality

1 Introduction

Membrane systems, also known as P systems, have become a prominent new computational model [1,19–21] inspired by the way living cells are divided by membranes into compartments where chemical reactions may take place.

* corresponding author, tel: +44 191 222 7982, fax: +44 191 222 8232

Email addresses: kleijn@liacs.nl (Jetty Kleijn), maciej.koutny@ncl.ac.uk (Maciej Koutny).

These reactions transform multisets of objects (molecules) present in the compartments into new objects, possibly transferring objects to neighbouring compartments, including the environment. Consequently, the behavioural aspects of membrane systems are based on sets of *reaction* or *evolution rules* defined for each compartment. A distinguishing feature of many models of membrane systems is that they evolve in a synchronous fashion: within each time unit (of a global clock), the system is transformed by a maximally concurrent execution of its reaction rules, i.e., no more rules in any compartment could have been applied in the same time unit. These transformations, or computation steps, are applied starting from an initial distribution of objects. Depending on the exact formalisation of the model, the notion of a successful (or halting) computation is defined together with its output, e.g., no evolution rule can be applied anymore and the output is the number of objects sent to the environment. This describes the functionality of the basic membrane system model, according to [20,21]. In addition, many different extensions and modifications of that basic model have been proposed and studied, mostly focusing on the outcomes of the computations of membrane systems and their computational power, including various aspects of complexity.

In [16], a Petri net model (see, e.g., [8,24]) has been proposed as a means to describe what is actually going on *during* a computation of a membrane system. *Petri nets* are bipartite directed graphs consisting of two kinds of nodes, called *places* and *transitions*. Places indicate the local availability of resources (represented by so-called tokens) and thus can be used to represent objects in specific compartments, whereas transitions are actions which can occur depending on local conditions related to the availability of resources and thus can be used to represent reaction rules associated with specific compartments. When a transition occurs it consumes resources from its input places and produces items in its output places, thus mimicking the effect of a reaction rule.

Since multiset calculus is basic for membrane systems as well as for computing the token distribution in Petri nets [5], some connections between the two models were already established including interpretations of reaction rules of membrane systems using Petri net transitions (see, e.g., [7,23]). In [16], it was demonstrated that a direct structural relationship between Petri nets and membrane systems can be established at the system level. A formal translation has been given for the basic class of membrane systems into a class of Petri nets. In these Petri nets, called *Place/Transition nets with localities* (PTL-nets), each transition has a location, similar to the distribution of the reaction rules over the compartments in a membrane system. It has been shown how the computations (sequences of computation steps) of membrane systems are faithfully reflected in the maximally concurrent step sequence semantics of their corresponding PTL-nets. Note that for the definition of maximal concurrency localities are not relevant, as the net supports the local aspects of resources consumed and produced by transitions. Localities are primarily a modelling tool in that co-located transitions correspond to reaction rules in a single compartment and, e.g., allow to identify the active parts of a system in the course of a computation. However, transitions with associated localities can be used to restrict synchronicity to certain locations within a system: in each step, and for each locality actively involved in that step, as many transitions belonging to this locality as possible are executed. Interestingly, the original strict global synchronicity of membrane systems is not always justifiable from a biological point of view as already observed in [20], but see also [6,7,10]. Thus the PTL-net model and its *locally* maximal concurrent step semantics make it possible to investigate membrane systems working subject to the natural assumption that synchronicity is restricted to the compartments of the system as delineated by the membranes.

Step sequence semantics of Petri nets provide important insights into concurrency aspects of the systems they are intended to model. Such semantics are, however, by definition sequential in nature in the sense that steps (of concurrently occurring transitions) are ordered which obscures the true causal relationships between the occurrences of transitions. Still information on causal relationships is often of high importance for system analysis and/or design. Petri nets can easily support a formal approach where this information is readily available as was recognised a long time ago; see [18] where it was proposed to unfold behaviours into structures allowing an explicit representation of causality, conflict and concurrency. For this purpose, labelled occurrence nets, called *processes* are used (see, e.g., [2,3,11,25]). In a nutshell, a process of a *Place/Transition net* (or PT-net) is a labelled partial order which records the essential relationships between the occurrences of transitions in its execution.

As noted in [16], the unfolding strategy defined for PT-nets does not work in the PTL-net case as the standard approach does not provide enough information about the potential executability of transitions which is relevant for the local maximality of executed steps. To address this problem, [16] introduced *barb-processes* where, in addition to the events which have actually occurred, also some potential events are represented. In this paper, we show that the idea of a barb-process can be extended to membrane systems with promoters and inhibitors.

In the first part of this paper, we will show how membrane systems with promoters and inhibitors can be modelled in a direct way using a class of Petri nets supporting localities as well as activator and inhibitor arcs. Crucially, the semantics of promoters and inhibitors turns out to be that of activator and inhibitor arcs working according to the *a priori* semantics which was used, e.g., in [12] to give a concurrency semantics to nets with inhibitor arcs. In the second part, we define a process semantics for the class of nets used in the translation. In the discussion of the process semantics, we will use (a fragment of) the general semantical framework developed in [12], which allows a systematic presentation of the process and causality semantics for various types of Petri nets. Here we would be particularly interested in justifying our process definition by establishing the consistency of the operational (step sequence) semantics of nets and the operational behaviour of their processes.

2 Preliminaries

We use the standard mathematical notation. In particular, $\uplus$ denotes disjoint set union, $\mathbb{N}$ the set of natural numbers (including 0) and $\mathbb{N}^+$ the set of positive natural numbers.

Functions. Let $\mathbb{P}(V)$ denote the powerset of a set V. The standard notation for the composition of functions is used also in the special case of two functions, $f : X \to \mathbb{P}(Y)$ and $g : Y \to \mathbb{P}(Z)$, for which $(g \circ f) : X \to \mathbb{P}(Z)$ is defined by $g \circ f(x) \stackrel{\text{df}}{=} \bigcup_{y \in f(x)} g(y)$, for all $x \in X$. The restriction of a function $f : X \to Y$ to a set $Z \subseteq X$ is denoted by $f|_Z$.

Binary relations. For a binary relation $P \subseteq X \times Y$ we will sometimes use an infix notation and write xPy rather than $(x, y) \in P$. Moreover, $dom_P \stackrel{\text{df}}{=} \{x \mid (x, y) \in P\}$. The composition of two binary relations, $P \subseteq X \times Y$ and $Q \subseteq Y \times Z$, is given by $P \circ Q \stackrel{\text{df}}{=} \{(x, z) \mid \exists y \in Y : (x, y) \in P \wedge (y, z) \in Q\}$. The restriction of a relation $P \subseteq X \times Y$ to a set $Z \subseteq X \times Y$ is denoted by $P|_Z$. By id_X we denote the identity relation on a set X. Relation $P \subseteq X \times X$ is reflexive if $id_X \subseteq P$; irreflexive if $id_X \cap P = \varnothing$; and transitive if $P \circ P \subseteq P$. The transitive closure of P is denoted by P^+, and the transitive and reflexive closure by $P^\star$.

Multisets. A multiset over a set X is a function $\mathsf{m} : X \to \mathbb{N}$ and an extended multiset over X is a function $\mathsf{m} : X \to \mathbb{N} \cup \{\infty\}$. The set of all multisets over X is denoted by $\mathbb{N}^X$. Any subset of X may be viewed through its characteristic function as a multiset (or an extended multiset) over X. A multiset m is finite (empty) if there are finitely many (no) $x \in \mathsf{m}$ by which we mean that $x \in X$ and $\mathsf{m}(x) \geq 1$; the cardinality of m is then defined as $|\mathsf{m}| \stackrel{\text{df}}{=} \sum_{x \in X} \mathsf{m}(x)$. For two multisets m and m' over X, the sum is given by $(\mathsf{m}+\mathsf{m}')(x) \stackrel{\text{df}}{=} \mathsf{m}(x)+\mathsf{m}'(x)$ for all $x \in X$, and $\mathsf{m} \leq \mathsf{m}'$ if $\mathsf{m}(x) \leq \mathsf{m}'(x)$ for all $x \in X$.

Labellings. A *labelling* for a set X is a function $\ell : X \to Z$, where Z is a set of *labels*, and we say that $x \in X$ is z–labelled if $\ell(x) = z$. Labelling ℓ can be lifted in a special way for a multiset m over X to an extended multiset $\ell\langle\mathsf{m}\rangle$, in the following way: for each $z \in Z$, $\ell\langle\mathsf{m}\rangle(z) = \infty$ if there are infinitely many $x \in \mathsf{m}$ such that $\ell(x) = z$; otherwise $\ell\langle\mathsf{m}\rangle(z) \stackrel{\text{df}}{=} \sum_{\{x \in X \mid \ell(x) = z\}} \mathsf{m}(x)$. If $\infty \notin \ell\langle\mathsf{m}\rangle(Z)$ then $\ell\langle\mathsf{m}\rangle$ can be treated of as a multiset over Z. For example, if $\ell(p) = \ell(q) = a$ and $\ell(r) = b$ then $\ell\langle\{p,p,q,r,r\}\rangle = \{a,a,a,b,b\}$ and $\ell\langle\{p,q\}\rangle = \{a,a\}$.

If X_i, $i \in \mathcal{I}$, are sets and for each X_i we have a labelling ℓ_i, such that $\ell_i(x) = \ell_j(x)$ whenever $x \in X_i \cap X_j$, then $\ell = \bigcup_{k\in\mathcal{I}} \ell_k$ is the function defined by $\ell(x) \stackrel{\mathrm{df}}{=} \ell_i(x)$ if $x \in X_i$.

Sequences. We use the notation $\sigma = \langle x_i \rangle_{\mathcal{I}}$ to represent an infinite $x_1x_2\ldots$ or finite $x_1x_2\ldots x_n$ sequence σ, including the empty one ε, where in the former case $\mathcal{I} = \mathbb{N}^+$ and in the latter $\mathcal{I} = \{1,2,\ldots,n\}$ or $\mathcal{I} = \varnothing$, respectively. For example, $\langle xyz \rangle_{\mathbb{N}^+} = xyzxyzxyz\ldots$. We will also write $\mathcal{I}_0 \stackrel{\mathrm{df}}{=} \mathcal{I} \cup \{0\}$. If all the x_i's are sets then $\bigcup\sigma \stackrel{\mathrm{df}}{=} \bigcup_{i\in\mathcal{I}} x_i$. If each x_i is a multiset over a set X and ℓ is a labelling for X, then $\ell\langle\sigma\rangle \stackrel{\mathrm{df}}{=} \langle \ell\langle x_i \rangle\rangle_{\mathcal{I}}$.

Step sequences and labelled step sequences. A step sequence (over a set X) is a possibly infinite sequence of finite multisets (over X). In this paper, we will denote by $\mathcal{STS}$ the set of all step sequences. A labelled step sequence is a pair $\varpi \stackrel{\mathrm{df}}{=} (\sigma, \ell)$, where σ is a step sequence consisting of mutually disjoint sets and ℓ is a labelling for the set $\bigcup\sigma$. With such ϖ we associate the step sequence $\phi(\varpi) \stackrel{\mathrm{df}}{=} \ell\langle\sigma\rangle$. The set of all labelled step sequences will be denoted by $\mathcal{LSTS}$.

3 Membrane systems with promoters and inhibitors

In this section, we formalise the notion of a membrane system. What follows extends the basic model introduced in [19,21] with promoters and inhibitors proposed in [4].

Definition 3.1 *A* membrane system (with promoters and inhibitors) *is a construct* $\Pi \stackrel{\mathrm{df}}{=} (V, \mu, w_1^0, \ldots, w_m^0, R_1, \ldots, R_m)$*, where:*

- V *is a finite* alphabet *consisting of (names of) objects or molecules;*
- μ *is a* membrane structure *given by a rooted tree with* m *nodes, representing the membranes — we assume that the nodes are given as the integers* $1,\ldots,m$*, and* $(i,j) \in \mu$ *will mean that there is an edge from* i *(parent) to* j *(child) in the tree of* μ*;*
- *each* w_i^0 *is a multiset of objects initially associated with membrane* i*;*
- *each* R_i *is a finite set of* reaction (or evolution) rules r *associated with membrane* i*, of the form* $lhs^r \to rhs^r|_{prom^r,inh^r}$*, where* lhs^r *(the left hand side of* r*),* $prom^r$ *(the promoters of* r*) and* inh^r *(the inhibitors of* r*) are multisets over* V*, and* rhs^r *(the right hand side of* r*) is a possibly empty multiset over* $V \cup \{a_{out} \mid a \in V\} \cup \{a_{in_j} \mid a \in V$ *and* $(i,j) \in \mu\}$*. It is assumed that no evolution rule* r *associated with the root of the membrane structure*

uses any a_{out} in rhs^r, and that the lhs^r are non-empty.

The nodes of a membrane structure represent membranes which in their turn determine the compartments: node j represents membrane m_j which defines c_j as the compartment enclosed by m_j and in-between m_j and its children if any. In the above, symbols a_{in_j} represent objects a that will be sent to (the compartment defined by) the child node j and a_{out} stands for an a that will be sent out to the parent's compartment. The role of $prom^r$ and inh^r is to constrain the applicability of r so that it can only be executed if its compartment currently holds at least $prom^r(a)$ and less than $inh^r(a)$ copies of each object a. Both promoters and inhibitors have interesting biological interpretations, for example, inhibitors correspond to substances which may block certain reactions even though there are sufficient resources for their execution. Note that if $prom^r$ and/or inh^r is the empty multiset, then there are no restrictions on executing the reaction due to promoting/inhibiting elements.

Figure 1 shows a membrane system over the alphabet $V = \{a, b, c, d\}$ comprising two membranes, m_1 and m_2, and five reaction rules, $r1, \ldots, r5$. For example, $r1$ can be executed if the inner compartment contains one copy of a and three copies of b; when executed, $r1$ consumes these four molecules and produces two copies of a: one is retained in the inner compartment and the other is sent to the outer one. Another, rather more complicated, rule $r4$ consumes one b and produces three copies of b (one retained in the outer and two sent to the inner compartment), but can only be executed if there is at least one c and no a in the outer compartment.

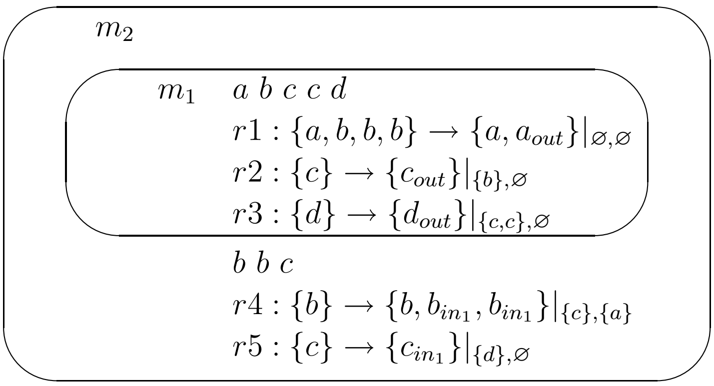


Fig. 1. A membrane system with promoters and inhibitors.

A membrane system Π as above evolves from configuration to configuration as a consequence of the application of (multisets of) evolution rules in each compartment. Formally, a *configuration* is a tuple $C \overset{\text{df}}{=} (w_1, \ldots, w_m)$ where each w_i is a multiset of object names, and we define a *vector multi-rule* $\vec{R}$ as an element of $\mathbb{N}^{R_1} \times \cdots \times \mathbb{N}^{R_m}$. Vector multi-rule $\vec{R} = \langle \widehat{R}_1, \ldots, \widehat{R}_m \rangle$ is said to be *empty* if each of the $\widehat{R}_i$ is empty. Given a vector multi-rule $\vec{R} = \langle \widehat{R}_1, \ldots, \widehat{R}_m \rangle$, we use as additional notations: $lhs_i = \sum_{r \in R_i} \widehat{R}_i(r) \cdot lhs^r$ and

$rhs_i = \sum_{r\in R_i} \widehat{R}_i(r)\cdot rhs^r$ for the multisets of all objects (possibly indexed) in, respectively, the left and right hand sides of the rules in the multiset $\widehat{R}_i$.

The execution semantics of a membrane system can vary, depending on the balance between *synchrony* and *asynchrony* in the allowed behaviours. We will consider four such variants that have been extensively investigated in the area of membrane systems, viz. *maximal* parallelism, *locally maximal* parallelism [16,17], *minimal* parallelism [9], and *free* parallelism [22].

Under free parallelism any multiset of reaction rules can be executed as a synchronous step provided that enough resources are available, enough promoters are present to support the reactions, and too few inhibitors are present to block the reactions. More precisely, configuration $C = (w_1, \ldots, w_m)$ *free-evolves* into configuration $C' = (w'_1, \ldots, w'_m)$ by a vector multi-rule $\vec{R} = \langle \widehat{R}_1, \ldots, \widehat{R}_m\rangle$ if the following hold, for every $1 \le i \le m$:

- $lhs_i \le w_i$;
- $prom^r(a) \le w_i(a)$, for all $r \in \widehat{R}_i$ and $a \in prom^r$;
- $w_i(a) < inh^r(a)$, for all $r \in \widehat{R}_i$ and $a \in inh^r$; and
- $w'_i(a) = w_i(a) - lhs_i(a) + rhs_i(a) + rhs_{parent(i)}(a_{in_i}) + \sum_{(i,j)\in\mu} rhs_j(a_{out})$, for every $a \in V$, where $parent(i)$ is the father membrane of i unless i is the root in which case $parent(i)$ is undefined and $rhs_{parent(i)}(a_{in_i})$ is omitted. Note that any j in the last term must be a child of i.

We denote this by $C \overset{\vec{R}}{\Longrightarrow}_{free} C'$ (or $C \overset{\vec{R}}{\Longrightarrow}_{free}$). Note that the second part of the above definition describes the effect of the application of the rules in $\vec{R}$.

The other three execution semantics can be seen as restrictions of the free parallelism paradigm. Given $C \overset{\vec{R}}{\Longrightarrow}_{free} C'$ as above, we say that C:

- *min-evolves* into C' by $\vec{R}$ (or $C \overset{\vec{R}}{\Longrightarrow}_{min} C'$) if $|\widehat{R}_1| + \cdots + |\widehat{R}_m| = 1$;
- *max-evolves* into C' by $\vec{R}$ (or $C \overset{\vec{R}}{\Longrightarrow}_{max} C'$) if there is no $\widehat{R}_i$ and rule r in R_i such that $C \overset{\vec{R}'}{\Longrightarrow}_{free}$ where $\vec{R}'$ is obtained from $\vec{R}$ by adding r to $\widehat{R}_i$; and
- *lmax-evolves* into C' by $\vec{R}$ (or $C \overset{\vec{R}}{\Longrightarrow}_{lmax} C'$) if there is no $\widehat{R}_i$ with $|\widehat{R}_i| \ge 1$, and rule r in R_i such that $C \overset{\vec{R}'}{\Longrightarrow}_{free}$ where $\vec{R}'$ is obtained from $\vec{R}$ by adding r to $\widehat{R}_i$.

A *free/min/max/lmax-computation* of Π is then defined to be a finite or infinite sequence of free/min/max/lmax-evolutions by non-empty multi-rules starting from $C_0 \overset{\mathrm{df}}{=} (w^0_1, \ldots, w^0_m)$, the initial configuration of Π.

We have a clear relationship between the four execution modes of membrane systems, which stem from the following inclusions (no other inclusions hold in general): $\xrightarrow{\vec{R}}_{min} \cup \xrightarrow{\vec{R}}_{max} \cup \xrightarrow{\vec{R}}_{lmax} \subseteq \xrightarrow{\vec{R}}_{free}$ and $\xrightarrow{\vec{R}}_{max} \subseteq \xrightarrow{\vec{R}}_{lmax}$. For the membrane system in Figure 1, we have the following:

$$
\begin{aligned}
(\{a,b,c,c,d\},\{b,b,c\}) &\xrightarrow{(\{r2,r2,r3\},\{r4,r4\})}_{max} (\{a,b,b,b,b,b\},\{b,b,c,c,c,d\})\\
(\{a,b,c,c,d\},\{b,b,c\}) &\xrightarrow{(\{r2,r2,r3\},\varnothing)}_{lmax} (\{a,b\},\{b,b,c,c,c,d\})\\
(\{a,b,c,c,d\},\{b,b,c\}) &\xrightarrow{(\{r2\},\{r4\})}_{free} (\{a,b,b,b,c,d\},\{b,b,c,c\})\\
(\{a,b,c,c,d\},\{b,b,c\}) &\xrightarrow{(\varnothing,\{r4\})}_{min} (\{a,b,b,b,c,c,d\},\{b,b,c\})
\end{aligned}
$$

4 Petri nets

A *net* is a triple $N \stackrel{\text{df}}{=} (P,T,W)$ such that P and T are disjoint sets, and $W : (T\times P)\cup(P\times T) \to \mathbb{N}$ is a multiset. The elements of P and T are respectively the *places* and *transitions*, and W is the *weight function*. In diagrams, places are drawn as circles, and transitions as boxes. If $W(x,y) \geq 1$ for some $(x,y) \in (T\times P)\cup(P\times T)$, then (x,y) is an *arc* leading from x to y. An arc is annotated with its weight if the latter is greater than one. The net N is *finite* (*countable*) if both P and T are finite (countable) sets.

The *pre-* and *post-multiset* of a transition (or place) x are multisets of places (resp. transitions), $\text{PRE}_N(x)$ and $\text{POST}_N(x)$, respectively given by $\text{PRE}_N(x)(y) \stackrel{\text{df}}{=} W(y,x)$ and $\text{POST}_N(x)(y) \stackrel{\text{df}}{=} W(x,y)$, for each place (resp. transition) y. We assume that $\text{PRE}_N(x)$ is finite for every place x, and that $\text{PRE}_N(x)$ is non-empty for every transition x.

A *marking* is a multiset M of places.[1] In diagrams, it is represented by drawing in each place p exactly $M(p)$ tokens (small black dots). In general, we will consider nets with explicit or implicit *initial* markings.

A *step* is a finite multiset U of transitions. It is *enabled* at a marking M if $M(p) \geq \sum_{t\in T} U(t)\cdot \text{PRE}_N(t)(p)$ for all $p \in P$. We denote this by $M[U\rangle$. An enabled step U can be *executed* leading to the marking M' given by $M'(p) \stackrel{\text{df}}{=} M(p) - \sum_{t\in T} U(t)\cdot \text{PRE}_N(t)(p) + \sum_{t\in T} U(t)\cdot \text{POST}_N(t)(p)$, for all $p \in P$. We denote this by $M[U\rangle M'$.

A (possibly infinite) sequence $\sigma = \langle U_i\rangle_{\mathcal{I}}$ of non-empty steps is a *step sequence* from a marking M_0 if there are markings $\langle M_i\rangle_{\mathcal{I}}$ satisfying $M_{i-1}[U_i\rangle M_i$ for every $i \in \mathcal{I}$. Moreover, the sequence of alternating markings and steps, $\mu = M_0\langle U_i M_i\rangle_{\mathcal{I}}$ will be called a *mixed step sequence* from M_0. If $\mathcal{I}$ is finite then

[1] For technical reasons, we do not require that M be finite.

σ (μ) is a (mixed) step sequence from M_0 to M_n, where n is the largest index in $\mathcal{I}_0$. If $\mathcal{I} = \varnothing$, then $\sigma = \varepsilon$ is the empty sequence and $\mu = M_0$.

If σ is a step sequence from M we write $M[\sigma\rangle$, and if σ is a step sequence from M to some M' we write $M[\sigma\rangle M'$, calling M' *reachable* from M. Note that $M[\varepsilon\rangle M$. If we want to make it clear which net we are dealing with, we may add a subscript N and write $[\cdot\rangle_N$ rather than $[\cdot\rangle$.

A *Place/Transition net* (or *PT-net*) is a marked finite net (P, T, W, M_0) consisting of a finite net (P, T, W) together with an initial marking M_0.

4.1 Petri nets with localities and activator/inhibitor arcs

We now introduce the class of Petri nets to be used for a direct behaviour preserving translation from membrane systems with promoters and inhibitors. Each reaction rule (associated with a membrane i) will be represented by a transition (belonging to the *locality* i). A *locality mapping* $\mathfrak{D}$ partitions the transition set by associating with each transition a locality, given by an integer. Thus, each non-empty inverse image $\mathfrak{D}^{-1}(i)$ determines a set of co-located transitions. Note that the locality mapping is never considered as a multiset nor as a labelling. In diagrams, boxes representing transitions with localities are shaded with the actual locality being shown in the middle (see Figure 2).

Definition 4.1 *A* PT-net with localities, and weighted activator and inhibitor arcs *(or PTLAI-net) is a tuple NLAI* $\stackrel{\text{df}}{=} (P, T, W, \mathfrak{D}, \mathfrak{A}, \mathfrak{I}, M_0)$*, where:*

- $\text{UND}(NLAI) \stackrel{\text{df}}{=} (P, T, W)$ *is a finite net underlying NLAI;*
- $\mathfrak{D} : T \to \mathbb{N}$ *is a locality mapping;*
- $\mathfrak{A} : P \times T \to \mathbb{N}$ *is a multiset for specifying activator arcs;*
- $\mathfrak{I} : P \times T \to \mathbb{N} \cup \{\infty\}$ *is an extended multiset for specifying inhibitor arcs;*
- M_0 *is the initial marking.*

We denote this by NLAI $\in \mathcal{PNLAI}$.

If $\mathfrak{A}(p, t) = k \geq 1$, then (p, t) is an *activator arc* with weight k, and p is an *activator place of* t; the latter can only be executed if the former contains at least k tokens. In diagrams, we draw an arrow from p to t with a small black circle as arrowhead and annotated with its weight k whenever $k > 1$. If $\mathfrak{I}(p, t) = k \in \mathbb{N}$, then (p, t) is an *inhibitor arc* with weight k, and p is an *inhibitor place of* t; the latter can only be executed if p does not contain more than k tokens. In that case, we draw an arrow from p to t with a small (open) circle as arrowhead and annotated with its weight k whenever $k > 0$.

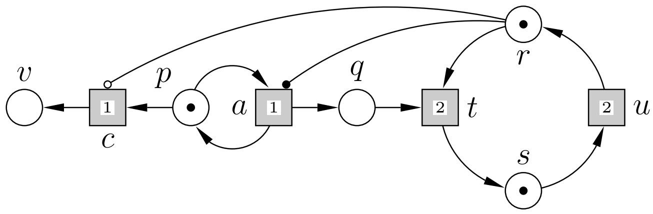


Fig. 2. PTLAI-net of a one-producer / two-consumers system.

If $\mathfrak{I}(p,t) = \infty$, for all p and t, then the occurrence of transitions is never inhibited by the presence of too many tokens in some of the places. In this case, the diagram has no arrows for inhibitor arcs and we can specify *NLAI* as a tuple $(P, T, W, \mathfrak{D}, \mathfrak{A}, \infty, M_0)$. Finally, if *NLAI* has neither inhibitor arcs nor activator arcs ($\mathfrak{A}(p,t) = 0$ for all p and t), then it is a *PT-net with localities* [16,17]. In this case, it may be simply specified as $(P, T, W, \mathfrak{D}, M_0)$.

For each $t \in T$, we define a multiset of places $\text{ACT}_{NLAI}(t)$ and an extended multiset of places $\text{INH}_{NLAI}(t)$ in such a way that, for every $p \in P$: $\text{ACT}_{NLAI}(t)(p) \overset{\text{df}}{=} \mathfrak{A}(p,t)$ and $\text{INH}_{NLAI}(t)(p) \overset{\text{df}}{=} \mathfrak{I}(p,t)$.

Figure 2 shows a PTLAI-net modelling a system consisting of one producer and two consumers. Transitions a and c correspond to adding new items to the buffer place q and cancelling of this operation by the producers, while transitions t and u correspond to taking and using the deposited items by the two consumers. The way transitions' localities are assigned reflects the view that producers operate away (at location 1) from consumers (location 2). The activator arc between r and a encodes the assumption that producers only produce items if there is at least one consumer waiting for them. On the other hand, the inhibitor arc between r and c means that a producer can cancel the production of items only if there is no consumer waiting for them.

All notations and notions introduced before for nets, are defined for *NLAI* through its underlying net.
A step $U : T \to \mathbb{N}$ is *free-enabled* at a marking M (denoted as $M[U\rangle_{free}$) if $M[U\rangle_{\text{UND}(NLAI)}$ and $\text{ACT}_{NLAI}(t) \leq M \leq \text{INH}_{NLAI}(t)$, for every $t \in U$. Thus, in order for U to be free-enabled at M, it should be enabled at M and moreover, for every transition t appearing in U, no place p may contain less than $\mathfrak{A}(p,t)$ tokens, and no place q may contain more than $\mathfrak{I}(q,t)$ tokens.
As special cases of free-enabledness we distinguish min-enabledness when only singleton steps can be enabled and max-enabledness when no more transitions can be added to steps. Moreover, localities come in use in case of locally max-enabledness or lmax-enabledness, when for no locality actively involved in a step, more transitions can be added to that step. Thus U is:

- *min-enabled* at M (or $M[U\rangle_{min}$) if $|U| = 1$;
- *max-enabled* at M (or $M[U\rangle_{max}$) if there is no $t \in T$ such that $M[U +$

$\{t\}\rangle_{free}$; and

- *lmax-enabled* at M (or $M[U\rangle_{lmax}$) if there is no transition t such that we have $M[U + \{t\}\rangle_{free}$ and $\mathfrak{D}(t) \in \mathfrak{D}(U)$.

Let $\mathfrak{m} \in \{free, min, max, lmax\}$ be a mode of execution. If a step U is $\mathfrak{m}$-enabled at M, then it can be executed as before (the activator and inhibitor arcs have no effect on the execution itself) leading to the marking M' such that $M[U\rangle_{\text{UND}(NLAI)}M'$. We denote this by $M[U\rangle_{\mathfrak{m}}M'$. We then obtain the notions of a (finite or infinite) $\mathfrak{m}$-*step sequence*, $\mathfrak{m}$-*mixed step sequence* and $\mathfrak{m}$-*reachability* of markings as in the case of ordinary Petri nets, by replacing the standard enabledness with $\mathfrak{m}$-enabledness.

It is thus immediate that all execution-modes are restricted versions of the (non-activated, non-inhibited) step sequence semantics of the underlying net and, moreover, we have as before a clear relationship between the four modes of execution: $[U\rangle_{min} \cup [U\rangle_{max} \cup [U\rangle_{lmax} \subseteq [U\rangle_{free} \subseteq [U\rangle_{\text{UND}(NLAI)}$ and $[U\rangle_{max} \subseteq [U\rangle_{lmax}$. For the PTLAI-net in Figure 2, we have the following: examples of step sequences under different modes:

$$M_0[\{a,u\}\{a,t\}\{a,t,u\}\rangle_{max} \qquad M_0[\{u\}\{a\}\{a\}\{t,t\}\{c\}\rangle_{lmax}$$

$$M_0[\{a\}\{u\}\{a,t\}\{a,t\}\rangle_{free} \qquad M_0[\{a\}\{u\}\{t\}\{a\}\{t\}\{c\}\rangle_{min}$$

In what follows, we will use the notation $\omega(NLAI)$ to denote the set of all (finite and infinite) lmax-step sequences defined by the PTLAI-net $NLAI$.

4.2 PTLAI-nets with complemented inhibitor places

A special class of PTLAI-nets is that of PTLAI-nets with *complemented inhibitor places* (or PTLACI-nets). In each such net $NLAI = (P, T, W, \mathfrak{D}, \mathfrak{A}, \mathfrak{I}, M_0)$, every inhibitor place $p \in P$ has a complement place in P, denoted by p^{cpl}, such that $\text{PRE}_{NLAI}(p) = \text{POST}_{NLAI}(p^{cpl})$ and $\text{POST}_{NLAI}(p) = \text{PRE}_{NLAI}(p^{cpl})$. For technical convenience we assume that $p \neq p^{cpl}$ and $(p^{cpl})^{cpl} = p$.[2] Thus the total number of tokens in p and p^{cpl} will always be the same, whatever step sequence has been executed and we can associate with both a bound on the number of tokens they can ever have: $\text{BND}_{NLAI}(p) = \text{BND}_{NLAI}(p^{cpl}) \stackrel{\text{df}}{=} M_0(p) + M_0(p^{cpl})$. Thus testing whether p has no more than k tokens can be considered as testing whether p^{cpl} has at least $\text{BND}_{NLAI}(p) - k$ tokens. Consequently, for PTLACI-nets, there is a straightforward behaviour-preserving

[2] Note that we can always copy places in a PTLAI-net with their initial marking and each of their incoming and outgoing arcs, without affecting the step sequence semantics of the net.

translation into PTLAI-nets without inhibitor arcs which are relatively easy to deal with.

Let $NLAI = (P, T, W, \mathfrak{D}, \mathfrak{A}, \mathfrak{I}, M_0)$ be a PTLACI-net. Then we define $\Delta(NLAI) \stackrel{\text{df}}{=} (P, T, W, \mathfrak{D}, \mathfrak{A}', \infty, M_0)$, where $\mathfrak{A}'(p,t) \stackrel{\text{df}}{=} \max\{\mathfrak{A}(p,t), \text{BND}_{NLAI}(p^{cpl}) - \mathfrak{I}(p^{cpl}, t)\}$ if p^{cpl} is an inhibitor place of t, and $\mathfrak{A}'(p,t) \stackrel{\text{df}}{=} \mathfrak{A}(p,t)$ otherwise, for every $p \in P$ and $t \in T$.

Proposition 4.2 *$\omega(NLAI) = \omega(\Delta(NLAI))$.*

Proof Follows directly from the definition of Δ. □

As an example, consider the PTLACI-net in Figure 2 and note that s is the complement place of r with common bound 2. It corresponds through the Δ mapping to the PTLAI-net without inhibitor arcs in Figure 5.

5 From membrane systems to Petri nets

To model a membrane system with inhibitor arcs as a PTLAI-net, we introduce a separate place (x, j) for each kind of molecule x and compartment c_j. For each rule r associated with a compartment c_i we introduce a separate transition t_i^r with locality i. If the transformation described by a rule r of compartment c_i consumes k copies of molecule x from compartment c_j, then we introduce a k weighted arc from place (x, j) to transition t_i^r, and similarly for molecules produced by transformations. If the rule has exactly k occurrences of molecule x in $prom^r$ then we introduce a k weighted activator arc from (x, i) to transition t_i^r. Similarly, if the rule has exactly k occurrences of molecule x in inh^r then we introduce a $k-1$ weighted inhibitor arc from (x, i) to t_i^r. Finally, assuming that, initially, compartment c_j contained n copies of molecule x, we introduce n tokens into place (x, j).

Definition 5.1 *Let $\Pi = (V, \mu, w_1^0, \ldots, w_m^0, R_1, \ldots, R_m)$ be a membrane system with promoters and inhibitors. Then the corresponding net is $NLAI_\Pi \stackrel{\text{df}}{=} (P, T, W, \mathfrak{D}, \mathfrak{A}, \mathfrak{I}, M_0)$, where the various components are defined thus:*

- *$P \stackrel{\text{df}}{=} V \times \{1, \ldots, m\}$ and $T \stackrel{\text{df}}{=} T_1 \cup \ldots \cup T_m$ where each T_i contains a distinct transition t_i^r for every reaction rule $r \in R_i$;*

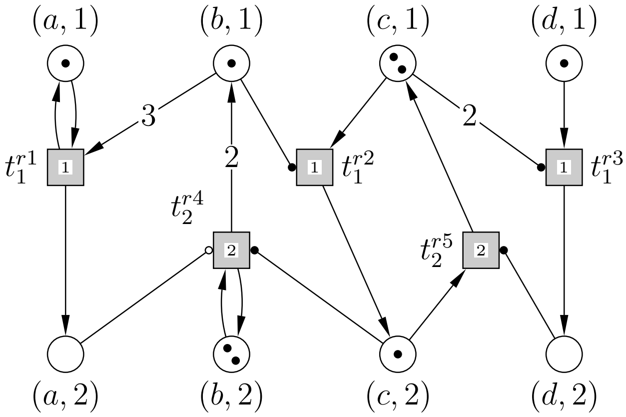


Fig. 3. PTLAI-net corresponding to the membrane system in Figure 1.

- *for every place* $p = (a, j) \in P$ *and every transition* $t = t_i^r \in T$,

$$W(p,t) \stackrel{\text{df}}{=} \begin{cases} lhs^r(a) & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \quad W(t,p) \stackrel{\text{df}}{=} \begin{cases} rhs^r(a) & \text{if } i = j \\ rhs^r(a_{out}) & \text{if } (j,i) \in \mu \\ rhs^r(a_{in_j}) & \text{if } (i,j) \in \mu \\ 0 & \text{otherwise} \end{cases}$$

$$\mathfrak{A}(p,t) \stackrel{\text{df}}{=} \begin{cases} prom^r(a) & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \quad \mathfrak{I}(p,t) \stackrel{\text{df}}{=} \begin{cases} inh^r(a){-}1 & \text{if } i = j \wedge a \in inh^r \\ \infty & \text{otherwise} \end{cases}$$

- *for every place* $p = (a, j) \in P$, *its initial marking is* $M_0(p) \stackrel{\text{df}}{=} w_j(a)$.
- *for every transition* $t = t_i^r \in T$, *its locality is* $\mathfrak{D}(t) \stackrel{\text{df}}{=} i$.

It is a matter of a simple check that $NLAI_\Pi$ is a PTLAI-net. Figure 3 shows the application of the last definition to the membrane system in Figure 1.

To capture the very tight correspondence between the membrane system Π and the PTLAI-net $NLAI_\Pi$, we introduce a straightforward bijection between configurations of Π and markings of $NLAI_\Pi$, based on the correspondence of the locations of objects and places.

Let $C = (w_1, \ldots, w_m)$ be a configuration of Π. Then the corresponding marking $\nu(C)$ of $NLAI_\Pi$ is given by $\nu(C)(a,i) \stackrel{\text{df}}{=} w_i(a)$, for every place (a,i) of $NLAI_\Pi$. Similarly, for any vector multi-rule $\vec{R} = \langle \widehat{R}_1, \ldots, \widehat{R}_m \rangle$ of Π, we define a multiset $\rho(\vec{R})$ of transitions of $NLAI_\Pi$ such that $\rho(\vec{R})(t_i^r) \stackrel{\text{df}}{=} \widehat{R}_i(r)$ for every $t_i^r \in T$. It is clear that ν is a bijection from the configurations of Π to the markings of $NLAI_\Pi$, and that ρ is a bijection from vector multi-rules of Π to the steps of $NLAI_\Pi$. Moreover, $\vec{R}$ is $\mathfrak{m}$-enabled at configuration C if and only if $\rho(\vec{R})$ is $\mathfrak{m}$-enabled at the marking $\nu(C)$, for every $\mathfrak{m} \in \{free, min, max, lmax\}$.

We now can formulate a fundamental property concerning the relationship between the dynamics of a membrane system and the corresponding PTLAI-net.

Theorem 5.2 *Let* $\mathfrak{m} \in \{free, min, max, lmax\}$ *be a mode of execution of membrane systems. Then* $C \stackrel{\vec{R}}{\Longrightarrow}_{\mathfrak{m}} C'$ *if and only if* $\nu(C)\ [\rho(\vec{R})\rangle_{\mathfrak{m}}\ \nu(C')$.

Since the initial configuration of Π corresponds through ν to the initial marking of $NLAI_{\Pi}$, the above immediately implies that the $\mathfrak{m}$-computations of the membrane system with promoters and inhibitors Π coincide with the $\mathfrak{m}$-step sequences of the PTLAI-net $NLAI_{\Pi}$. What is more, due to the bijective nature of the translation captured by Definition 5.1, it should be intuitively clear that the causal relationships in the behaviours of a membrane system are properly reflected in the behaviours of the corresponding PTLAI-net, and so the latter provide a convenient way of dealing with the former.

6 Semantical framework

We have shown how computations of membrane systems with promoters and inhibitors can be modelled by step sequences of the corresponding PTLAI-nets. This allows one to employ various techniques developed for the latter to analyse the behaviours of the former. For example, invariant based techniques can be used to explore the distribution of molecules among and within the compartments. Any step sequence semantics, however, is based on ordered occurrences (sequences) of steps which may obscure the causal relationship between executions of transitions. To deal with causality related aspects of Petri nets, one can resort to another well-established approach and consider labelled occurrence nets of Petri nets, called *processes* (see, e.g., [2,3,11,25]). Processes may be defined operationally by unfoldings based on step sequences through unravelling their steps while registering the production and consumption of tokens (resources), i.e., the changing of markings (configurations). The resulting processes are structures which explicitly represent causality and concurrency:

- *Causality.* The causality relationships among the executed transitions can be read-off by following directed paths in the process net.
- *Concurrency.* Executed transitions for which there is no directed path from one to another are concurrent.
- *Reachability.* Any maximal set of places of the process net for which there are no directed paths from one to the other corresponds to a reachable marking of the original net.

In what follows, we will focus only on one of the execution modes, namely locally maximal concurrency. The three remaining modes will be briefly discussed in the concluding section.

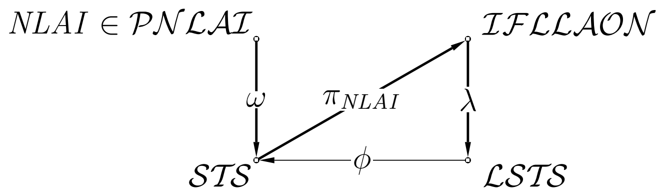


Fig. 4. The semantical framework for PTLAI-nets, where the bold arcs indicate mappings to powersets.

Figure 4 shows the concrete setup that we will follow. The semantical domains we are concerned with are:

- PTLAI-nets ($\mathcal{PNLAI}$), step sequences ($\mathcal{STS}$) and labelled step sequences ($\mathcal{LSTS}$);
- $\mathcal{IFLLAON}$ is a domain still to be defined, consisting of initially finite labelled occurrence nets with localities and activator arcs, providing the basis for processes of PTLAI-nets.

The intended roles of the mappings in Figure 4 are as follows:

- $\omega : \mathcal{PNLAI} \to \mathbb{P}(\mathcal{STS})$ yields the set of lmax-step sequences defined by the PTLAI-net $NLAI$;
- $\pi_{NLAI} : \mathcal{STS} \to \mathbb{P}(\mathcal{IFLLAON})$ is a partial mapping which defines a set of occurrence nets, for each lmax-step sequence generated by $NLAI$;
- each occurrence net from $\mathcal{IFLLAON}$ is given an operational labelled step sequence semantics via $\lambda : \mathcal{IFLLAON} \to \mathbb{P}(\mathcal{LSTS})$; and
- labelled step sequences can be re-interpreted as step sequences through the total function ϕ.

Two of these mappings, viz. ϕ and ω, have already been defined, and the other two will be introduced in due course.

An overall goal is to show that this setup is consistent in the sense that processes (as given by $\mathcal{IFLLAON}$) describe the causal relationships between events in a way which is in accordance with the lmax-step sequence semantics of $\mathcal{PNLAI}$. Formally, the consistency between the process semantics (defined by $\pi_{NLAI} \circ \omega$) of a PTLAI-net $NLAI$ and its operational semantics as given by ω, is provided by the following result.

Theorem 6.1 $\omega = \phi \circ \lambda \circ \pi_{NLAI} \circ \omega$.

Clearly, both ϕ and ω are total mappings, moreover, ω never returns the empty set (since every PTLAI-net has ε as an lmax-step sequence). Therefore, as shown in [12], Theorem 6.1 holds whenever the two properties given below are satisfied. [3]

Property 1 *The functions (i)* $\pi_{NLAI}|_{\omega(NLAI)}$ *and (ii)* $\lambda|_{\pi_{NLAI}\circ\omega(NLAI)}$ *are total and never return the empty set.*

Property 2 *For all* $\xi \in \omega(NLAI)$ *and* $LLAON \in \pi_{NLAI}(\xi)$*, we have* $\xi \in \phi(\lambda(LLAON))$ *and* $\phi(\lambda(LLAON)) \subseteq \omega(NLAI)$*.*

Note that the second property captures two interesting features:

- *Representation.* In any process associated with a step sequence of $NLAI$ through π_{NLAI}, this step sequence can be executed from the implicit initial marking of the process.
- *Executability.* Any labelled step sequence from the implicit initial marking of a process represents a legal step sequence of the original net.

In what follows, we will introduce the fourth semantical domain as well as the two remaining mappings. After that, we will establish Property 1 and Property 2.

7 Occurrence nets with localities and activator arcs

The nets in $\mathcal{IFLLAON}$ form the domain for the processes of PTLAI-nets, and are based on the *initially finite labelled occurrence nets with activator arcs* (ifao-nets) used in [13] for the process semantics of PT-nets with inhibitor arcs. Moreover, just as in the *locality occurrence nets* (loco-nets) defined in [17], localities and special barb-events are used to properly treat the locally maximal concurrency semantics. Note that barb-events signal potential executability of transitions.

Definition 7.1 *An* initially finite labelled occurrence net with activator arcs and localities *(or* iflao-net*) is a tuple* $LLAON \stackrel{\text{df}}{=} (B, E, R, A, \mathcal{L}, \mathfrak{E}, \mathfrak{R}, \mathfrak{A}, \mathfrak{L}, \ell)$ *such that the following hold:*

- $(B, E \cup \mathfrak{E}, R \cup \mathfrak{R})$ *is a countable net. Its places (i.e., the elements of* B*) are called* conditions*, its transitions in* E *are called* events*, and its transitions in* $\mathfrak{E}$ *are called* barb-events*. The sets of events and barb-events are disjoint.*
- $R \subseteq (B \times E) \cup (E \times B)$ *and* $\mathfrak{R} \subseteq B \times \mathfrak{E}$*.* [4]
- $A \subseteq B \times E$ *and* $\mathfrak{A} \subseteq B \times \mathfrak{E}$ *are sets of activator arcs.*

[3] In [12], λ instead of $\lambda|_{\pi_{NLAI}\circ\omega(NLAI)}$ is used, but this does not change the validity of the result.

[4] We treat the weight functions R and $\mathfrak{R}$ as binary relations since they always return 0 or 1, and similarly for A and $\mathfrak{A}$.

- *For every* $b \in B$*, there is at most one* $e \in E$ *such that* $(e, b) \in R$*, and at most one* $f \in E$ *such that* $(b, f) \in R$*.*
- *For every* $e \in E$*, there is* $b \in B$ *such that* $(e, b) \in R$*.*
- $\mathcal{L}$ *is a locality mapping for* E *and* $\mathfrak{L}$ *is a locality mapping for* $\mathfrak{E}$*.*
- ℓ *is a labelling for* $B \cup E$*.*
- *Let* $\prec_{LLAON}$ *and* $\sqsubset_{LLAON}$ *be two relations on* E *defined by:*

  $$\prec_{LLAON} \stackrel{\mathrm{df}}{=} (\prec \cup \sqsubset)^{\star} \circ \prec \circ (\prec \cup \sqsubset)^{\star} \quad and \quad \sqsubset_{LLAON} \stackrel{\mathrm{df}}{=} (\prec \cup \sqsubset)^{\star} \backslash id_E$$

  where $\prec \stackrel{\mathrm{df}}{=} (R \circ R)|_{E \times E} \cup (R \circ A)$ *and* $\sqsubset \stackrel{\mathrm{df}}{=} A^{-1} \circ R$*.*
  It is assumed that the relation $\prec_{LLAON}$ *is irreflexive, and there are only finitely many* $f \in E$ *such that* $f \sqsubset_{LLAON} e$*, for every* $e \in E \cup \mathfrak{E}$*.*

We denote this by $LLAON \in \mathcal{IFLLAON}$*.*

In diagrams, we will show the labels of conditions and events rather than their identities; moreover, barb-events are depicted using dark boxes (see Figure 5). In the rest of this section, we assume an iflao-net $LLAON$ together with the auxiliary relations, as in the above definition.

The implicit *initial* marking MIN_{LLAON} of $LLAON$ consists of all conditions without incoming arcs, i.e., $\text{MIN}_{LLAON} \stackrel{\mathrm{df}}{=} B \backslash dom_R$. We now introduce the notions of enabledness and executability of steps for iflao-nets under a locally maximal concurrency semantics. As in [17], the former involves barb-events but the steps themselves contain only ordinary (non-barb) events.

A non-empty multiset U over E is free-enabled at a marking M of $LLAON$ if $M[U\rangle$ in (B, E, R) and $M(b) \geq A(b, e)$, for all $e \in U$ and $b \in B$. It is *barb-enabled* at M if, in addition, there is no event $e \in E$ such that $\mathcal{L}(e) \in \mathcal{L}(U)$ and $U + \{e\}$ is free-enabled at M in $LLAON$, nor a barb-event $f \in \mathfrak{E}$ such that $\mathfrak{L}(f) \in \mathcal{L}(U)$, $M(b) \geq \mathfrak{A}(b, f)$ for all $b \in B$, and $U + \{f\}$ is enabled at M in $(B, E \cup \mathfrak{E}, R \cup \mathfrak{R})$. In other words, putting aside the possibility that $LLAON$ may be infinite, U is barb-enabled at M in $LLAON$ if it is lmax-enabled at M in the PTLAI-net $(B, E \cup \mathfrak{E}, R \cup \mathfrak{R}, A \cup \mathfrak{A}, \infty, \mathcal{L} \cup \mathfrak{L}, \text{MIN}_{LLAON})$. The notions of barb-step sequence and mixed barb-step sequence as well as barb-reachability are then defined with barb-enabledness replacing the standard notion of enabledness.

It is worth mentioning that $(E, \prec_{LLAON}, \sqsubset_{LLAON}, \ell)$ is a relational structure which captures causality between events. More precisely, $\prec_{LLAON}$ captures causality and $\sqsubset_{LLAON}$ weak causality. In essence, if b and c have been executed and $e \prec_{LLAON} f$ then e was executed *before* f, and if $e \sqsubset_{LLAON} f$ then e was executed *before or together with* f. Note that the global causality relations are induced by the two *local* (or immediate) relationships, $\prec$ and $\sqsubset$.

Ignoring all elements relating to localities and barb-events leads to $\text{UND}(LLAON) \stackrel{\text{df}}{=} (B, E, R, A, \ell)$ which is an ifao-net of [13] for which a number of results have already been established. To start with, the enabledness of steps in $\text{UND}(LLAON)$ coincides with the free-enabledness in $LLAON$. Next, if $\langle E_i \rangle_{\mathcal{I}}$ is a step sequence of $\text{UND}(LLAON)$ from MIN_{LLAON}, then the E_i's are mutually disjoint finite sets and each marking reachable from MIN_{LLAON} is a set (cf. Propositions 3 and 6 of [13]). A relation which characterises causally related conditions is defined as $\text{SLIN}(\text{UND}(LLAON)) \stackrel{\text{df}}{=} (R\circ \prec^{\star} \circ R)|_{B\times B}$. To characterise reachable markings, we define $\text{SSL}(\text{UND}(LLAON))$ to be the set of all $S \subseteq B$ which are maximal w.r.t. set inclusion and such that $(S \times S) \cap \text{SSL}(AON) = \varnothing$, and there are only finitely many events $e \in E$ satisfying $(e, b) \in R^+$ for some $b \in S$. Proposition 6 of [13] then states that the set of markings reachable from MIN_{LLAON} in $\text{UND}(LLAON)$ coincides with $\text{SSL}(\text{UND}(LLAON))$. Moreover, each set of events executable through a finite step sequence from MIN_{LLAON} belongs to $\text{SCNF}(\text{UND}(LLAON))$ which comprises all finite sets $D \subseteq E$ such that $e \in D$ and $(f, e) \in \prec^+ \Rightarrow f \in D$. We will denote $\text{SLIN}(LLAON) \stackrel{\text{df}}{=} \text{SLIN}(\text{UND}(LLAON))$, etc.

Proposition 7.2 *For every mixed barb-step sequence of $LLAON$ from MIN_{LLAON}, each marking occurring in it is a set in $\text{SSL}(LLAON)$, and the steps occurring in it are mutually disjoint sets. Moreover, if the sequence is finite, then the set D of events it contains belongs to $\text{SCNF}(LLAON)$ and leads to the marking*

$$\text{MAR}_D \stackrel{\text{df}}{=} \text{MIN}_{AON} \cup \{b \mid \exists e \in D : (e, b) \in R\} \setminus \{b \mid \exists e \in D : (b, e) \in R\} .$$

Proof Follows from the fact that such a sequence is also a mixed step sequence of $\text{UND}(LLAON)$ from the marking MIN_{LLAON}, and the results of [13] mentioned above. □

We define the mapping $\lambda : \mathcal{IFLLAON} \to \mathbb{P}(\mathcal{LSTS})$ from Figure 4, by associating with each iflao-net the labelled step sequences defined by those barb-step sequences which include *all* events of the net. Formally, the set $\lambda(LLAON)$ of *labelled barb-step sequences* of $LLAON$ comprises all labelled step sequences $\xi \stackrel{\text{df}}{=} (\sigma, \ell)$ such that $\sigma = \langle E_i \rangle_{\mathcal{I}}$ is a barb-step sequence of $LLAON$ from MIN_{LLAON} satisfying $E = \bigcup_{i\in\mathcal{I}} E_i$. We denote this by $\xi \in \lambda(LLAON)$ (ξ is well-defined due to Proposition 7.2). Note that λ is a total mapping, but it may happen that $\lambda(LLAON) = \varnothing$ as shown in [17] for locality occurrence nets. However, as we shall demonstrate, if the iflao-net has been defined operationally from an lmax-step sequence of a PTLAI-net, then $\lambda(LLAON) \neq \varnothing$ and so Property 1(ii) holds.

8 Processes of PTLAI-nets

We first deal with PTLAI-nets without inhibitor arcs, using a relatively simple construction and indicate how this treatment can be readily extended to PTLAI-nets with complemented inhibitor places.

For technical reasons (see also Definition 7.1, where it is postulated that every event has a post-condition), we will assume for the rest of the paper that each transition in a PTLAI-net has at least one outgoing arc. Though a net resulting from the translation of a membrane system does not need to satisfy this, we may always introduce a dummy output place which has no impact on the possible behaviours.

8.1 PTLAI-nets without inhibitor arcs

Let $NLAI \overset{\mathrm{df}}{=} (P, T, W, \mathfrak{D}, \mathfrak{A}, \infty, M_0)$ be a PTLAI-net without inhibitor arcs, fixed for the rest of this subsection. The next definition takes a possibly infinite lmax-step sequence of $NLAI$ and constructs a corresponding iflao-net. The construction combines that proposed in [13] for PT-nets with complemented inhibitor places, with the treatment of [17] which uses barb-events to signal the enabledness of transitions.

Definition 8.1 *Let $\sigma = \langle U_i \rangle_{\mathcal{I}}$ be an lmax-step sequence of $NLAI$. A* barb-activator process *(*ba-process*) of $NLAI$* generated by σ *is an iflao-net*

$$\begin{aligned} LLAON &= (B, E, R, A, \mathcal{L}, \mathfrak{E}, \mathfrak{R}, \mathfrak{A}', \mathfrak{L}, \ell) \\ &\overset{\mathrm{df}}{=} \Big(\textstyle\bigcup_{k\in\mathcal{I}_0} B_k, \bigcup_{k\in\mathcal{I}_0} E_k, \bigcup_{k\in\mathcal{I}_0} R_k, \bigcup_{k\in\mathcal{I}_0} A_k, \bigcup_{k\in\mathcal{I}_0} \mathcal{L}_k, \bigcup_{k\in\mathcal{I}_0} \bigcap_{j\geq k} \mathfrak{E}_j, \\ &\qquad \textstyle\bigcup_{k\in\mathcal{I}_0} \bigcap_{j\geq k} \mathfrak{R}_j, \bigcup_{k\in\mathcal{I}_0} \bigcap_{j\geq k} \mathfrak{A}_j, \bigcup_{k\in\mathcal{I}_0} \bigcap_{j\geq k} \mathfrak{L}_j, \bigcup_{k\in\mathcal{I}_0} \ell_k\Big) \end{aligned}$$

where for $k \in \mathcal{I}_0$:

$$B_k = \biguplus_{i=0}^{k} B^i \quad E_k = \biguplus_{i=0}^{k} E^i \quad R_k = \biguplus_{i=0}^{k} R^i \quad A_k = \biguplus_{i=0}^{k} A^i \quad \mathcal{L}_k = \biguplus_{i=0}^{k} \mathcal{L}^i \quad \ell_k = \biguplus_{i=0}^{k} \ell^i$$

and the various sets used above are constructed in the following way (it is assumed that they do not contain any elements other than those specified explicitly):

(1) $E^0 = \varnothing$ and for all $i \in \mathcal{I}$, E^i comprises a distinct event for each transition occurrence in U_i. The event corresponding to the j-th occurrence of t in

U_i is denoted by $t^{i,j}$; we set $\ell^i(t^{i,j}) \stackrel{\mathrm{df}}{=} t$ and $\mathcal{L}^i(t^{i,j}) \stackrel{\mathrm{df}}{=} \mathfrak{D}(t)$.

(2) B^0 comprises a distinct condition for each place occurrence in M_0. The condition corresponding to the j-th occurrence of s in M_0 is denoted by s^j; we set $\ell^0(s^j) \stackrel{\mathrm{df}}{=} s$.

(3) For all $i \in \mathcal{I}$ and $e \in E^i$, B^i comprises a distinct condition for each place occurrence in $\textsc{post}_{NLAI}(\ell_i(e))$. The condition corresponding to the j-th occurrence of p in $\textsc{post}_{NLAI}(\ell_i(e))$ is denoted by $p^{e,j}$; we set $\ell^i(p^{e,j}) \stackrel{\mathrm{df}}{=} p$.

(4) $R^0 = \varnothing$, and for all $i \in \mathcal{I}$ and $e \in E^i$:

- *We choose a disjoint (i.e., $B_f \cap B_g = \varnothing$ whenever $f \neq g$) set of conditions $B_e \subseteq B_{i-1} \backslash dom_{R_{i-1}}$ such that $\ell_i\langle B_e\rangle = \textsc{pre}_{NLAI}(\ell_i(e))$. After that we add an arc (b, e) to R^i for each $b \in B_e$.*
- *We add an arc $(e, p^{e,j})$ to R^i for each $p^{e,j} \in B^i$.*

(5) $A^0 = \varnothing$, and for $i \in \mathcal{I}$ and every $e \in E^i$, we choose a set A_e of conditions in $B_{i-1} \backslash dom_{R_{i-1}}$ such that $\ell_i\langle A_e\rangle = \textsc{act}_{NLAI}(\ell_i(e))$. After that we add an activator arc (b, e) to A^i for each $b \in A_e$.

(6) $\mathfrak{E}_{-1} = \varnothing$, and for all $i \in \mathcal{I}_0$ we construct $\mathfrak{E}_i$ from $\mathfrak{E}_{i-1}$ as follows:

- *We first form a set of candidate barb-events $\mathfrak{O}_i$ consisting of all $e^l_{C,D}$, where $C \cup D \subseteq B_i$ and $l \in \mathbb{N}$, such that for some $t \in T$ the following are satisfied:*
  - $\cdot$ *$\mathfrak{D}(t) = l$ and $\textsc{pre}_{NLAI}(t) = \ell_i\langle C\rangle$ and $\textsc{act}_{NLAI}(t) = \ell_i\langle D\rangle$,*
  - $\cdot$ *$((C \cup D) \times (C \cup D)) \cap \textsc{slin}((B_i, E_i, R_i, A_i, \ell_i)) = \varnothing$,*
  - $\cdot$ *$(C \cup D) \cap B^i \neq \varnothing$.*
- *We then obtain $\mathfrak{E}_i$ from $\mathfrak{E}_{i-1} \cup \mathfrak{O}_i$ by removing every barb-event $e^l_{C,D}$ for which one of the following holds:*
  - $\cdot$ *there is $f \in E_i$ satisfying $\mathcal{L}(f) = l$ and*

$$\{b \mid (b, f) \in R_i\} \subseteq C \quad \text{and} \quad \{b \mid (b, f) \in A_i\} \subseteq C \cup D\,,$$

  - $\cdot$ *there is $e^l_{C',D'} \in \mathfrak{E}_{i-1} \cup \mathfrak{O}_i$ satisfying*

$$(C' \subset C) \wedge (D' \subseteq C \cup D) \quad \text{or} \quad (C' = C) \wedge (D' \subset D)\,.$$

(7) $\mathfrak{R}_i$ comprises all directed arcs $(b, e^l_{C,D})$ such that $e^l_{C,D} \in \mathfrak{E}_i$ and $b \in C$.

(8) $\mathfrak{A}_i$ comprises all activator arcs $(b, e^l_{C,D})$ such that $e^l_{C,D} \in \mathfrak{E}_i$ and $b \in D$.

(9) $\mathfrak{L}_i(e^l_{C,D}) \stackrel{\mathrm{df}}{=} l$, for each $e^l_{C,D} \in \mathfrak{E}_i$.

We denote this by $LLAON \in \pi_{NLAI}(\sigma)$.

It is easy to see that $LLAON$ in the above definition is indeed an iflao-net (the same holds for the intermediate nets constructed in successive stages). Note that if σ is finite then $\mathfrak{E} \stackrel{\mathrm{df}}{=} \mathfrak{E}_m$, $\mathfrak{R} \stackrel{\mathrm{df}}{=} \mathfrak{R}_m$, $\mathfrak{A}' \stackrel{\mathrm{df}}{=} \mathfrak{A}_m$ and $\mathfrak{L} \stackrel{\mathrm{df}}{=} \mathfrak{L}_m$, where m is the greatest integer in $\mathcal{I}_0$.

Throughout the rest of this subsection we assume the notation as in Definition 8.1 and, furthermore, denote $\textsc{max}_i \stackrel{\mathrm{df}}{=} B_i \backslash dom_{R_i}$, for every $i \in \mathcal{I}_0$.

In the definition, items 1 through 4 are exactly as for PT-nets with complemented inhibitor places (see [13]), while in item 5 we can now directly introduce new activator arcs, rather than first interpreting inhibitor arcs as activator arcs to complement places. The construction is well-defined since items 4 and 5 can always be executed (see Corollary 8.4).

Items 6 through 9 all relate to barb-events similar to those used in [17]. Now, however, activator arcs have to be taken into account as well. In each stage of the construction, the candidate barb-events $\mathfrak{O}_i$ indicate the potential existence of newly enabled transitions of the PTLAI-net (first part of item 6). However, a candidate barb-event does not become an actual barb-event (second part of item 6) if there is an old or new event with the same locality whose input conditions are contained in those of the candidate (implying that this candidate is superfluous), or if there is an existing barb-event or another candidate from the same locality whose input conditions are strictly contained in those of the current candidate (which thus is not needed to signal enabledness). The first case of removing superfluous barb-events is also applied to old barb-events. Items 7 and 8 add arcs and activator arcs to the new barb-events corresponding to arcs and activator arcs in the PTLAI-net leading to the original transition, and item 9 labels new barb-events.

Figure 5 shows a PTLAI-net without inhibitor arcs and illustrates the generation of a ba-process for the step sequence $\sigma = \{a\}\{t, u\}$. First, in addition to the conditions representing the initial marking of the PTLAI-net, we have two barb-events representing the transitions which can be executed at the initial marking (the upper one corresponds to a and the lower one to u). In the second stage, one of these barb-events has disappeared due to the occurrence of transition a, leading to a new barb-event with locality 1, again corresponding to a. Another barb-event with pre-conditions labelled by r and q and locality 2 corresponds to t. In the third stage, two barb-events have disappeared (since the corresponding transitions were executed) and five new barb-events were added.

We will now establish key properties of ba-processes. First, if we ignore the effect of localities and barb-events (which implies that the execution mode is that of free-enabledness), Definition 8.1 and other supporting notions reduce (after minor notational changes) to Definition 1 in [15], and so we may recall some useful results.

Proposition 8.2 ([15]) *Let* $\mu = \text{MAX}_0\langle E^i \text{MAX}_i\rangle_{\mathcal{I}}$*, and* θ *be a (mixed) free-step sequence not involving barb-events of LLAON from* MIN_{LLAON}*.*

(1) μ *is a mixed free-step sequence of LLAON.*
(2) $\ell\langle\theta\rangle$ *is a (mixed) free-step sequence of NLAI from* M_0*.*

The above still hold if we switch to lmax-enabledness and barb-enabledness.

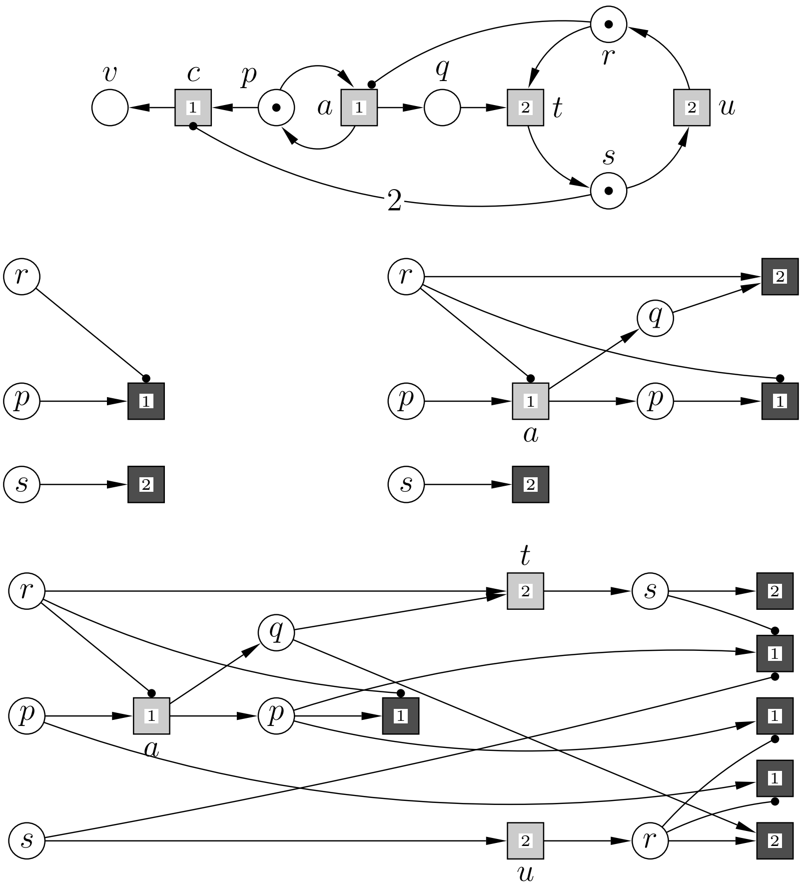


Fig. 5. A PTLAI-net without inhibitor arcs and a ba-process constructed for the lmax-step sequence $\sigma = \{a\}\{t, u\}$.

Proposition 8.3 *Let* $\mu = \text{MAX}_0 \langle E^i \text{MAX}_i \rangle_{\mathcal{I}}$*, and* θ *be a (mixed) barb-step sequence of* $LLAON$ *from* MIN_{LLAON}*.*

(1) μ *is a mixed barb-step sequence of* $LLAON$*.*
(2) $\ell\langle\theta\rangle$ *is a (mixed) lmax-step sequence of* $NLAI$ *from* M_0*.*

Proof If (1) does not hold then, since $\text{MAX}_0 = B^0$ is a mixed barb-step sequence of $LLAON$, there is $m \geq 1$ such that $\mu' = \text{MAX}_0 \langle E^i \text{MAX}_i \rangle_{\{1,\dots,m-1\}}$ is a mixed barb-step sequence of $LLAON$, but $\mu' E^m \text{MAX}_m$ is not. By Proposition 8.2(1), this means that E^m is free-enabled at MAX_{m-1}, but it is not barb-enabled. Hence one of the following holds:

Case 1: There is an event $e \in E \setminus E_m$ such that $\mathcal{L}(e) \in \mathcal{L}(E^m)$ and $E^m \cup \{e\}$ is free-enabled at MAX_{m-1}. Hence, by Proposition 8.2(2) and $\ell\langle\langle E^i \rangle_{\mathcal{I}}\rangle = \sigma$, $\langle U_i \rangle_{\{1,\dots,m-1\}}(U_m + \{\ell(e)\})$ is a free-step sequence of $NLAI$. Since $\mathfrak{D}(\ell(e)) \in \mathfrak{D}(U_m)$, this contradicts the lmax-enabledness of U_m at $\ell\langle \text{MAX}_{m-1} \rangle$ in $NLAI$.

Case 2: There is a barb-event $e = e^l_{C,D} \in \mathfrak{E}$ such that $l \in \mathcal{L}(E^m)$ and $E^m \cup \{e\}$ is free-enabled at MAX_{m-1}. Hence, by Proposition 8.2(2) and $\ell\langle\langle E^i \rangle_{\mathcal{I}}\rangle = \sigma$,

$\langle U_i\rangle_{\{1,\ldots,m-1\}}(U_m+\{t\})$, for some $t \in T$ such that $\mathfrak{D}(t)=l$ and $\textsc{pre}_{NLAI}(t) = \ell\langle C\rangle$ and $\textsc{act}_{NLAI}(t) = \ell\langle D\rangle$, is a free-step sequence of $NLAI$. Again, since $\mathfrak{D}(t) \in \mathfrak{D}(U_m)$, this contradicts the lmax-enabledness of U_m at $\ell\langle\textsc{max}_{m-1}\rangle$ in $NLAI$. Thus (1) holds.

It suffices to show (2) for mixed barb-step sequences. Suppose that it does not hold. Then, by $\ell\langle\textsc{max}_0\rangle = \ell\langle B^0\rangle = M_0$, there is a mixed barb-step sequence $\xi MFM'$ of $LLAON$ from $\textsc{min}_{LLAON}$ such that: $\ell\langle\xi M\rangle$ is a mixed lmax-step sequence of $NLAI$ from M_0, but $\ell\langle\xi MFM'\rangle$ is not. By Proposition 8.2(2), this means that $\ell\langle F\rangle$ is free-enabled at $\ell\langle M\rangle$ in $NLAI$, but not lmax-enabled. Therefore there exists $t \in T$ such that $l = \mathfrak{D}(t) \in \mathfrak{D}(\ell\langle F\rangle)$ and $\ell\langle F\rangle + \{t\}$ is free-enabled at $\ell\langle M\rangle$ in $NLAI$, and so there are $C \subseteq M \setminus \textsc{pre}_{LLAON}(F)$ and $D \subseteq M$ such that $\textsc{pre}_{NLAI}(t) = \ell\langle C\rangle$ and $\textsc{act}_{NLAI}(t) = \ell\langle D\rangle$. Moreover, by Proposition 7.2, $M \in \textsc{ssl}(LLAON)$. Let i be the minimal index such that $C \cup D \subseteq B_i$. For such an i a barb-event $e^l_{C,D}$ was included in the set of candidate barb-events $\mathfrak{O}_i$ during the construction of $\mathfrak{E}_i$. If $e^l_{C,D} \in \mathfrak{E}$, then F is not barb-enabled at M in $LLAON$, a contradiction. Thus $e^l_{C,D} \notin \mathfrak{E}$. Let $j \geq i$ be the smallest index such that $e^l_{C,D} \notin \mathfrak{E}_j$. This means that one of the following holds: (i) there exists $f \in E_j$ satisfying $\mathcal{L}(f) = l$ and such that $\{b \mid (b,f) \in R_j\} \subseteq C$ and $\{b \mid (b,f) \in A_j\} \subseteq C \cup D$; or (ii) there is $e^l_{C',D'} \in \mathfrak{E}_{j-1} \cup \mathfrak{O}_j$ satisfying $(C' \subset C) \wedge (D' \subseteq C \cup D)$ or $(C' = C) \wedge (D' \subset D)$.

In the case of (i), $F \uplus \{f\}$ is free-enabled in $LLAON$ at M, contradicting the barb-enabledness of F. Therefore (ii) must hold. If $e^l_{C',D'} \in \mathfrak{E}$ then, as before, we have a contradiction with the barb-enabledness of F. If $e^l_{C',D'} \notin \mathfrak{E}$ then we iterate the argument and this iteration has to eventually stop since the sets C and D are finite. Consequently, it must be the case that there exists an event g satisfying $\mathcal{L}(g) = l$ and $\{b \mid (b,g) \in R_k\} \subseteq C'$ and $\{b \mid (b,g) \in A_k\} \subseteq C' \cup D'$, or there exists a barb-event $e^l_{C'',D''}$ such that $C'' \subseteq C$ and $D'' \subseteq C \cup D$. In either case we obtain a contradiction with the barb-enabledness of F. Hence (2) is also satisfied. □

As a direct consequence it follows that items 4 and 5 of Definition 8.1 can always be carried out as the sets $\textsc{max}_i$ contain enough conditions, which follows from the following.

Corollary 8.4 *For every $m \in \mathcal{I}_0$, $M_0[U_0 \ldots U_m\rangle_{lmax}\ell\langle\textsc{max}_m\rangle$.*

Thus Property 1(i) is satisfied. Moreover, also as a consequence of the first part of the last result, Property 1(ii) is satisfied. As a matter of fact, any ba-process generated by σ will have a labelled step sequence corresponding to σ (after forgetting about the identities of the underlying events through the function ϕ). Formally,

Corollary 8.5 *If $\sigma \in \omega(NLAI)$ and $LLAON \in \pi_{NLAI}(\sigma)$, then it is the case* *that* $\sigma \in \phi(\lambda(LLAON))$.

Furthermore, the barb-step sequences of a ba-process correspond to lmax-step sequences of the original PTLAI-net.

Corollary 8.6 *If* $\sigma \in \omega(NLAI)$ *and* $LLAON \in \pi_{NLAI}(\sigma)$*, then it is the case that* $\phi(\lambda(LLAON)) \subseteq \omega(NLAI)$.

Together, the last two corollaries imply Property 2, which completes the proof of the consistency between the lmax-step sequence semantics and the ba-process semantics of PTLAI-nets without inhibitor arcs.

The construction from Definition 8.1 readily extends to PTLACI-nets through the mapping Δ (cf. [12,13]); in other words, for each such net $NLAI$ we can define $\pi_{NLAI}(\sigma) = \pi_{\Delta(NLAI)}(\sigma)$ which, together with Proposition 4.2 and the results shown in this section, yields the consistency between the lmax-step sequence semantics and the ba-process semantics of PTLACI-nets. In this way, the ba-process in Figure 5 is also a ba-process of the PTLACI-net in Figure 2.

Finally, we note that thanks to Proposition 8.3 every lmax-reachable marking of $NLAI$ can be recovered through the labelling as a barb-reachable marking of some ba-process.

8.2 The general case

Since, in general, we cannot rely on complements of inhibitor places, another feature is needed to test that an inhibitor place does not contain too many tokens. The solution in [12,13,15] was to add 'on demand' new artificial conditions (labelled by the special symbol $\curlywedge$) with activator arcs to fulfill this role. We will use the same device here.

Let $NLAI \overset{\text{df}}{=} (P, T, W, \mathfrak{D}, \mathfrak{A}, \mathfrak{I}, M_0)$ be a PTLAI-net fixed for the rest of this section. If $p \in P$ and $t, w \in T$ are such that $\text{INH}_{NLAI}(t)(p) \neq \infty$ and $\text{PRE}_{NLAI}(w)(p) + \text{POST}_{NLAI}(w)(p) \neq 0$, then we write $w \multimap t$. The key idea behind the next construction is to ensure that if $w \multimap t$ then any two occurrences, f of w and e of t, are adjacent to a common condition to reflect the relationship which holds for w and t. To simplify the presentation, on this occasion superfluous barb-events are not removed.

Definition 8.7 *Let* $\sigma = \langle U_i \rangle_{\mathcal{I}}$ *be an lmax-step sequence of NLAI. A* barb-activator process with auxiliary conditions *(*baa-process*)* generated by σ *is* *an iflao-net*

$$LLAON = (B, E, R, A, \mathcal{L}, \mathfrak{E}, \mathfrak{R}, \mathfrak{A}', \mathfrak{L}, \ell)$$

$$\stackrel{\text{df}}{=} \Big(\bigcup_{k\in\mathcal{I}_0}(B_k \cup \widetilde{B}_k \cup \mathfrak{B}_k),\ \bigcup_{k\in\mathcal{I}_0}E_k,\ \bigcup_{k\in\mathcal{I}_0}R_k,\ \bigcup_{k\in\mathcal{I}_0}A_k,$$

$$\bigcup_{k\in\mathcal{I}_0}\mathcal{L}_k,\ \bigcup_{k\in\mathcal{I}_0}\mathfrak{E}_k,\ \bigcup_{k\in\mathcal{I}_0}\mathfrak{R}_k,\ \bigcup_{k\in\mathcal{I}_0}\mathfrak{A}_k,\ \bigcup_{k\in\mathcal{I}_0}\mathfrak{L}_k,\ \bigcup_{k\in\mathcal{I}_0}\ell_k\Big)$$

where for $k \in \mathcal{I}_0$*:*

$$B_k = \biguplus_{i=0}^{k} B^i \qquad \widetilde{B}_k = \biguplus_{i=0}^{k} \widetilde{B}^i \qquad \mathfrak{B}_k = \biguplus_{i=0}^{k} \mathfrak{B}^i \qquad E_k = \biguplus_{i=0}^{k} E^i$$

$$R_k = \biguplus_{i=0}^{k} R^i \qquad \mathfrak{E}_k = \biguplus_{i=0}^{k} \mathfrak{E}^i \qquad \mathfrak{R}_k = \biguplus_{i=0}^{k} \mathfrak{R}^i \qquad A_k = \biguplus_{i=0}^{k} A^i$$

$$\mathcal{L}_k = \biguplus_{i=0}^{k} \mathcal{L}^i \qquad \mathfrak{A}_k = \biguplus_{i=0}^{k} \mathfrak{A}^i \qquad \mathfrak{L}_k = \biguplus_{i=0}^{k} \mathfrak{L}^i \qquad \ell_k = \biguplus_{i=0}^{k} \ell^i$$

and the various sets used above are constructed as in Definition 8.1 except that $\widetilde{B}^0 = \mathfrak{B}^0 = \mathfrak{E}^0 = \mathfrak{R}^0 = \mathfrak{A}^0 = \mathfrak{L}^0 \stackrel{\text{df}}{=} \varnothing$ *and, for* $k \in \mathcal{I}$*:*

(1) If $e \in E^k$ *and* $f \in E^j$ *(for* $j < k$*) are such that* $\ell_k(f) \multimap \ell_k(e)$ *then we create exactly one condition* $b \in \widetilde{B}^k$ *and add two arcs:* $(f, b) \in R^k$ *and* $(b, e) \in A^k$*.*

(2) If $f \in E^k$ *and* $e \in E^j$ *(for* $j \le k$*) are such that* $\ell_k(f) \multimap \ell_k(e)$ *then we create exactly one condition* $b \in \widetilde{B}^k$ *and add two arcs:* $(b, f) \in R^k$ *and* $(b, e) \in A^k$*.*

(3) $\mathfrak{E}^k$ *comprises all* $e^t_{H,C,D}$ *where* $H \in \mathrm{SCNF}((B_k, E_k, R_k, A_k, \ell_k))$*,* $C \cup D \subseteq B_k$ *and* $t \in T$ *are such that:* $C \cup D \subseteq \mathrm{MAR}_H$*,* $H \cap E^k \neq \varnothing$*,* $\mathrm{PRE}_{NLAI}(t) = \ell_k\langle C\rangle$*,* $\mathrm{ACT}_{NLAI}(t) = \ell_k\langle D\rangle$ *and* $\ell_k\langle\widehat{\mathrm{MAR}_H}\rangle \le \mathrm{INH}_{NLAI}(t)$*; we also set* $\mathfrak{L}_i(e^t_{H,C,D}) \stackrel{\text{df}}{=} \mathfrak{D}(t)$*.*

(4) $\mathfrak{R}^k$ *comprises all directed arcs* $(b, e^t_{H,C,D})$ *such that* $e^t_{H,C,D} \in \mathfrak{E}^k$ *and* $b \in C$*.*

(5) $\mathfrak{A}^k$ *comprises all activator arcs* $(b, e^t_{H,C,D})$ *such that* $e^t_{H,C,D} \in \mathfrak{E}^k$ *and* $b \in D$*.*

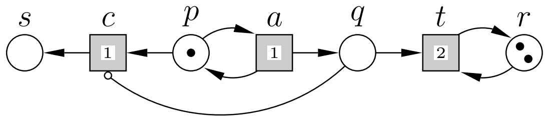
(6) If $e = e^t_{H,C,D} \in \mathfrak{E}^k$ *and* $f \in H$ *are such that* $\ell_k(f) \multimap t$ *then we create exactly one condition* $b \in \mathfrak{B}^k$ *and add two arcs:* $(f, b) \in R^k$ *and* $(b, e) \in \mathfrak{A}^k$*.*

(7) If $e = e^t_{H,C,D} \in \mathfrak{E}^k$ *and* $f \in E_k \setminus H$ *are such that* $\ell_k(f) \multimap t$ *then we create exactly one condition* $b \in \mathfrak{B}^k$ *and add two arcs:* $(b, f) \in R^k$ *and* $(b, e) \in \mathfrak{A}^k$*.*

(8) If $f \in E^k$ *and* $e = e^t_{H,C,D} \in \mathfrak{E}_{k-1}$ *are such that* $\ell_k(f) \multimap t$ *then we create exactly one condition* $b \in \mathfrak{B}^k$ *and add two arcs:* $(b, f) \in R^k$ *and* $(b, e) \in \mathfrak{A}^k$*.*

(9) $\ell^k(b) \stackrel{\text{df}}{=} \curlywedge$ *for all* $b \in \widetilde{B}^k \cup \mathfrak{B}^k$*.*

We will denote this by $LLAON \in \pi^a_{NLAI}(\sigma)$*.*



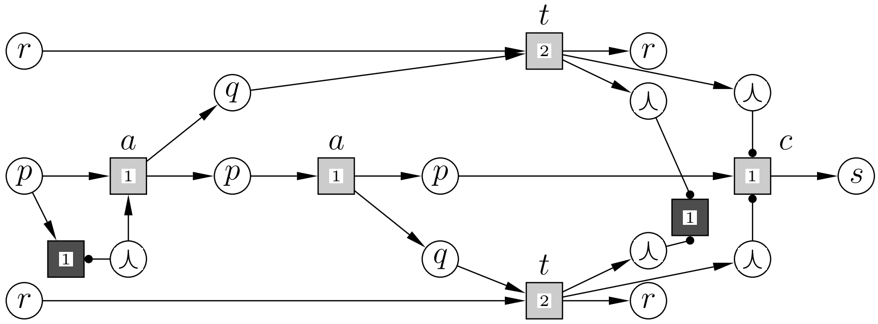


Fig. 6. A PTLAI-net and a baa-process generated by its lmax-step sequence $\sigma = \{a\}\{a\}\{t,t\}\{c\}$.

It is easy to see that $LLAON$ in the above definition is indeed an iflao-net (the same holds for the intermediate nets constructed in successive stages).

Definition 8.7 is illustrated in Figure 6. For clarity, we omitted $\curlywedge$-labelled conditions contributing causality relationships which can be deduced from other existing arcs. For example, let x be the left barb-event and z the left a-labelled event, joined by a $\curlywedge$-labelled condition inducing a weak precedence $x \sqsubset z$. Definition 8.7 creates, in fact, a $\curlywedge$-labelled condition for each other event which induces a similar weak precedence. However, only one such precedence, $x \sqsubset z$, is needed as we also have strong precedences induced by the directed arcs between z and each remaining event y.

In the rest of this section we assume the notation as in Definition 8.7. Moreover, we denote $\widehat{B} \stackrel{\text{df}}{=} \bigcup_{k\in\mathcal{I}_0} B_k$, $\mathfrak{B} \stackrel{\text{df}}{=} \bigcup_{k\in\mathcal{I}_0} \mathfrak{B}_k$, $\textsc{max}_i \stackrel{\text{df}}{=} (B_i \cup \widetilde{B}_i \cup \mathfrak{B}_i)\setminus dom_{R_i}$ for every $i \in \mathcal{I}_0$, and $\widehat{M} \stackrel{\text{df}}{=} M \cap \widehat{B}$ for every $M \subseteq B$.

Similarly as before, if we ignore the effect of localities, barb-events and conditions in $\mathfrak{B}$ (which implies that the execution mode is that of free-enabledness), Definition 8.7 reduces (after minor notational changes) to Definition 3 in [15], hence we have the following.

Proposition 8.8 ([15]) *Let $\mu = \textsc{max}_0\langle E^i \textsc{max}_i\rangle_{\mathcal{I}}$, and $\theta = \textsc{min}_{LLAON}\langle F^i M_i\rangle_{\mathcal{J}}$ be a mixed free-step sequence of LLAON not involving barb-events.*

(1) μ is a mixed free-step sequence of LLAON.
(2) $\ell\langle\widehat{\textsc{min}}_{LLAON}\langle F^i \widehat{M}_i\rangle_{\mathcal{J}}\rangle$ is a mixed free-step sequence of NLAI from M_0.

Proof The results from [15] are not directly applicable since $\textsc{und}(LLAON)$ contains an additional set of auxiliary conditions $\mathfrak{B}$. These, however, have no impact on the free-executability as each has a single incoming or outgoing arc, and no other adjacent arcs. $\square$

The above still hold if we switch to lmax-enabledness and barb-enabledness.

Proposition 8.9 *Let* $\mu = \text{MAX}_0\langle E^i\text{MAX}_i\rangle_{\mathcal{I}}$*, and* $\theta = \text{MIN}_{LLAON}\langle F^iM_i\rangle_{\mathcal{J}}$ *be a mixed barb-step sequence of LLAON not involving barb-events.*

(1) μ *is a mixed barb-step sequence of LLAON.*
(2) $\ell\langle\widehat{\text{MIN}}_{LLAON}\langle F^i\widehat{M}_i\rangle_{\mathcal{J}}\rangle$ *is a mixed lmax-step sequence of NLAI from* M_0*.*

Proof If (1) does not hold then, since $\text{MAX}_0 = B^0$ is a mixed barb-step sequence of $LLAON$, there is $m \geq 1$ such that $\mu' = \text{MAX}_0\langle E^i\text{MAX}_i\rangle_{\{1,\ldots,m-1\}}$ is a barb-step sequence of $LLAON$, but $\mu' E^m\text{MAX}_m$ is not. By Proposition 8.8(1), this means that E^m is free-enabled at MAX_{m-1}, but it is not barb-enabled. Hence one of the following holds:

Case 1: There is an event $e \in E \setminus E_m$ such that $\mathcal{L}(e) \in \mathcal{L}(E^m)$ and $E^m \cup \{e\}$ is free-enabled at MAX_{m-1}. Hence, by Proposition 8.8(2) and $\ell\langle\langle E^i\rangle_{\mathcal{I}}\rangle = \sigma$, $\langle U_i\rangle_{\{1,\ldots,m-1\}}(U_m + \{\ell(e)\})$ is a free-step sequence of $NLAI$. Since $\mathfrak{D}(\ell(e)) \in \mathfrak{D}(U_m)$, this contradicts the lmax-enabledness of U_m at $\ell\langle\widehat{\text{MAX}}_{m-1}\rangle$ in $NLAI$. Hence (1) holds.

Case 2: There is a barb-event $e = e^t_{H,C,D} \in \mathfrak{E}$ such that $\mathfrak{D}(t) \in \mathcal{L}(E^m)$ and $E^m \cup \{e\}$ is free-enabled at MAX_{m-1}. We first observe that $H \subseteq E_{m-1}$ and that if $f \in E_{m-1} \setminus H$ then, by $\text{MAX}_{m-1} = \text{MAR}_{E_{m-1}}$, it is not possible that $\ell(f) \multimap t$ since then e would not be enabled at MAX_{m-1}. Moreover, $\ell\langle\widehat{\text{MAR}}_H\rangle \leq \text{INH}_{NLAI}(t)$. Hence we also have $\ell\langle\widehat{\text{MAX}}_{m-1}\rangle \leq \text{INH}_{NLAI}(t)$. As a result, $\langle U_i\rangle_{\{1,\ldots,m-1\}}(U_m + \{t\})$ is a free-step sequence of $NLAI$. Since $\mathfrak{D}(t) \in \mathfrak{D}(U_m)$, this contradicts the lmax-enabledness of U_m at $\ell\langle\widehat{\text{MAX}}_{m-1}\rangle$ in $NLAI$. Thus (1) holds.

Suppose now that (2) does not hold. Then, by $\ell\langle\widehat{\text{MIN}}_{LLAON}\rangle = \ell\langle B^0\rangle = M_0$, there is $m \in \mathcal{J}$ such that $\ell\langle\widehat{\text{MIN}}_{LLAON}\langle F^i\widehat{M}_i\rangle_{\{1,\ldots,m-1\}}\rangle$ is a mixed lmax-step sequence of $NLAI$ from M_0 but $\ell\langle\widehat{\text{MIN}}_{LLAON}\langle F^i\widehat{M}_i\rangle_{\{1,\ldots,m\}}\rangle$ is not. By Proposition 8.8(2), this means that $\ell\langle F^m\rangle$ is free-enabled at $\ell\langle\widehat{M}_{m-1}\rangle$ in $NLAI$, but not lmax-enabled. Therefore there exists $t \in T$ such that $\mathfrak{D}(t) \in \mathfrak{D}(\ell\langle F\rangle)$ and $\ell\langle F^m\rangle + \{t\}$ is free-enabled at $\ell\langle\widehat{M}_{m-1}\rangle$ in $NLAI$. Hence there must be $C \subseteq M_{m-1} \setminus \text{PRE}_{LLAON}(F^m)$ and $D \subseteq M_{m-1}$ such that $\text{PRE}_{NLAI}(t) = \ell\langle C\rangle$ and $\text{ACT}_{NLAI}(t) = \ell\langle D\rangle$ and $\ell\langle\widehat{\text{MAR}}_H\rangle \leq \text{INH}_{NLAI}(t)$, where $H = F^1 \cup \ldots F^{m-1}$. Thus the following barb-event has been constructed: $e = e^t_{H,C,D}$. Therefore $F^m \cup \{e\}$ is free-enabled at M_{m-1}, and so F^m is not barb-enabled in $LLAON$ at M_{m-1}, a contradiction. Hence (2) is also satisfied. $\square$

As a result, Properties 1 and 2 are satisfied as before, and so the consistency between the lmax-step sequence semantics and the baa-process semantics of PTLAI-nets holds.

Finally, by Proposition 8.9 every lmax-reachable marking of $NLAI$ can be recovered as a barb-reachable marking of some baa-process of $NLAI$ after applying the labelling and restricted to conditions labelled by places in the original PTLAI-net.

9 Concluding remarks

Sections 7 and 8 considered the execution mode induced by lmax-enabledness. As far as free-enabledness is concerned, the results of [15] cover this case, and max-enabledness can be reduced to lmax-enabledness after assuming that all transitions belong to the same locality. The min-enabledness can be dealt with in a similar way as free-enabledness after assuming that all transitions are connected by a self-loop with a special place marked initially with a single token.

In this paper, we introduced processes of PTLAI-nets by following the generic scheme proposed in [12]. To complete the development of process semantics, one still needs to provide an axiomatic definition of ba(a)-processes and the causality structures they induce. Though in the case of free-enabledness (and min-enabledness), [15] provides solutions to both problems, treating lmax-enabledness and max-enabledness is a subject of an ongoing investigation.

A Petri net semantics for the basic class of membrane systems was provided in [16,17]. A striking feature of this approach was the one-to-one correspondence between transitions and reaction rules, as well as between tokens in places (markings) and local availability of resources (configurations). This paper achieved a similar translation for membrane systems in which resources (molecules) are not only produced/consumed but can trigger/inhibit reactions. The reader might wonder whether similar one-to-one translations are also possible for other extensions of the basic membrane system model. An initial investigation is reported in [14] where membrane systems with reaction rules that may become obsolete or available depending on the changing structure of the cells, as well as membrane systems with permeable or with dissolving membranes, are discussed. It appears that such extensions do not lend themselves to one-to-one translations to Petri nets, although there is a way of modelling of their salient features using the translation to PTLAI-nets in combination with special control structures. The process semantics developed in this paper should therefore be applicable to a wider class of membrane systems.

Acknowledgment

This research was supported by the EPSRC project CASINO.

References

[1] Membrane systems web page: `http://psystems.disco.unimib.it/`

[2] E. Best, R. Devillers, Sequential and concurrent behaviour in Petri net theory, Theoretical Computer Science 55 (1988) 87–136.

[3] E. Best, C. Fernández, Nonsequential Processes. A Petri Net View, EATCS Monographs on Theoretical Computer Science, Springer-Verlag, 1988.

[4] P. Bottoni, C. Martín-Vide, Gh. Păun, G. Rozenberg, Membrane systems with promoters/inhibitors, Acta Informatica 38 (2002) 695–720.

[5] C. S. Calude, Gh. Păun, G. Rozenberg, A. Salomaa (Eds.), Multiset Processing. Mathematical, Computer Science, and Molecular Computing Points of View, Springer-Verlag, LNCS 2235, 2001.

[6] M. Cavaliere, D. Sburlan, Time-independent P systems, in: G. Mauri et al. (Eds.), WMC 2004, Springer-Verlag, LNCS 3365, 2005, 239–258.

[7] S. Dal Zilio, E. Formenti, On the dynamics of PB systems: a Petri net view, in: C. Martín-Vide et al. (Eds.), WMC 2003, Springer-Verlag, LNCS 2933, 2004, 153–167.

[8] J. Desel, W. Reisig, G. Rozenberg (Eds.), Lectures on Concurrency and Petri Nets, Springer-Verlag, LNCS 3098, 2004.

[9] R. Freund, Sequential P systems, Romanian Journal of Information Science and Technology 4 (2001) 77–88.

[10] R. Freund, Asynchronous P systems and P systems working in the sequential mode, in: G. Mauri et al. (Eds.), WMC 2004, Springer-Verlag, LNCS 3365, 2005, 36–62.

[11] U. Goltz, W. Reisig, The non-sequential behaviour of Petri nets, Information and Control 57 (1983) 125–147.

[12] H.C.M. Kleijn, M. Koutny, Process semantics of general inhibitor nets, Information and Computation 190 (2004) 18–69.

[13] H.C.M. Kleijn, M. Koutny, Infinite process semantics of inhibitor nets, in: S. Donatelli, P.S. Thiagarajan (Eds.), ICATPN 2006, Springer-Verlag, LNCS 4024, 2006, 282–301.

[14] H.C.M. Kleijn, M. Koutny, Synchrony and asynchrony in membrane systems, in: H. J. Hoogeboom, Gh. Păun, G. Rozenberg, A. Salomaa (Eds.), WMC'06, Springer-Verlag, LNCS 4361, 2006, 66-85.

[15] J. Kleijn, M. Koutny, Processes of Petri nets with range testing, Fundamenta Informaticae 80 (2007) 199-219.

[16] J. Kleijn, M. Koutny, G. Rozenberg, Towards a Petri net semantics for membrane systems, in: R. Freund, Gh. Paun, G. Rozenberg, A. Salomaa (Eds.), WMC'05, Springer-Verlag, LNCS 3850, 2006, 292–309.

[17] J. Kleijn, M. Koutny, G. Rozenberg, Process semantics for membrane systems, Journal of Automata, Languages and Combinatorics 11 (2006) 321–340.

[18] M. Nielsen, G. Plotkin, G. Winskel, Petri nets, event structures and domains, Part I, Theoretical Computer Science 13 (1980) 85–108.

[19] Gh. Păun, Computing with membranes, Journal of Computer and System Sciences 61 (2000) 108–143.

[20] Gh. Păun, Membrane Computing. An Introduction, Springer-Verlag, 2002.

[21] Gh. Păun, G. Rozenberg, A Guide to membrane computing, Theoretical Computer Science 287 (2002) 73–100.

[22] Gh. Păun, S. Yu, On synchronization in P systems, Fundamenta Informaticae 38 (1999) 397–410.

[23] Z. Qi, J. You, H. Mao, P systems and Petri nets, in: C. Martín-Vide et al. (Eds.), WMC 2003, Springer-Verlag, LNCS 2933, 2004, 286–303.

[24] W. Reisig, G. Rozenberg (Eds.), Lectures on Petri Nets, Springer-Verlag, LNCS 1491, 1492, 1998.

[25] G. Rozenberg, J. Engelfriet, Elementary net systems, in: W. Reisig, G. Rozenberg (Eds.), Advances in Petri Nets. Lectures on Petri Nets I: Basic Models, Springer-Verlag, LNCS 1491, 1998, pp. 12–121.