Catalytic and Communicating Petri Nets are Turing Complete[☆]

Gabriel Ciobanu[a], G. Michele Pinna[b]

[a] Institute of Computer Science, Romanian Academy, Iaşi, Romania
[b] Dipartimento di Matematica e Informatica, Università di Cagliari, Italy

Abstract

In most studies about the expressiveness of Petri nets, the focus has been put either on adding suitable arcs or on assuring that a complete snapshot of the system can be obtained. While the former still complies with the intuition on Petri nets, the second is somehow an orthogonal approach, as Petri nets are *distributed* in nature. Here, inspired by membrane computing, we study some classes of Petri nets where the distribution is partially kept and which are still Turing complete.

1. Introduction

Soon after their introduction in the early 60's, Petri nets have been acknowledged as a formalism for modeling distributed and concurrent computations and, from a formal language theoretic point of view, their *expressivity* has been investigated. On the one side, the classes of languages defined by Petri nets, i.e., sets of sequences of labeled or unlabeled transitions, have been studied from the beginning of the 70's (*e.g.* in [18] and references therein) and research on this topic is still in progress. On the other side, the question of how *expressive* Petri nets are has been asked, and the answers can be summarized as follows: Petri nets under the usual firing strategy are not Turing complete (see [22] and the

[☆] Work partially supported by the *CGR Coarse Grain Recommendation*-project, PIA "Industria, Artigianato e Servizi", Regione Autonoma della Sardegna, TESLA, L.R. 7/2007, Regione Autonoma della Sardegna and TRICS, L.R. 7/2007, Regione Autonoma della Sardegna

surveys [12, 13] for a comprehensive review), while they can be Turing complete under suitable assumptions on the firing strategy, either *maximality* (all the possible transitions are fired together) or *ordering* (a transition should fire as soon as it is enabled), as shown in [4] and [5].

In order to make Petri nets Turing complete, different approaches have been taken, most notably by *extending* the arcs which are considered. The classic extensions are given by *inhibitor* arcs, *i.e.*, arcs where the absence of tokens in certain places is modeled ([17]), and by *reset* arcs, *i.e.*, arcs with the characteristic of emptying the preset, regardless of the number of tokens present in the place ([11]). Other extensions of Petri Nets, not necessarily always Turing complete, allow the introduction of *non blocking* arcs or *transfer* arcs ([14] and [16]), making the transitions *marking* dependent [6], the introduction of other kinds of elements, *e.g.* the so called *zero-safe* places [3] where the notion of transaction is endorsed, or *weighted inhibitor* arcs [19]. Some of the extensions have been introduced to make Petri nets more expressive, others to facilitate the modeling of certain phenomena.

In this paper we take a different approach: we try to identify if there are *minimal* assumptions that make Petri nets Turing complete without the introduction of suitable places or arcs, and trying to preserve the intuition behind Petri nets. The tight relationship with *membrane computing* ([26]) suggests that this route can be pursued, thus we investigate the expressivity of Petri nets by looking at suitable firing strategy exploiting the *structure* of the nets itself, namely how transitions and places are connected.

The correspondence between membrane computing and Petri nets has already been established (see chapter 15 of the Oxford Handbook of Membrane Computing [27] and references therein), where it is shown that to each kind of membrane systems it is possible to associate a suitable *labeled* Petri net, where the labeling of transitions is used to model the membrane structure.

In this paper we go a step further by establishing a relation between Petri nets and membrane systems, showing that to each Petri net we can associate a membrane system, and the various firing strategies definable on Petri nets can be matched by corresponding evolution steps in membrane systems. Membrane systems come naturally equipped with a *multi-layer* structure modeled by *nested membranes*, a structure which is not usual in Petri nets. In fact, when associating a net to a membrane system, a suitable labeling is considered, the so called *location* mapping associated to transitions. The notion of locality in Petri nets belongs to the original intuition, where transitions depend on the neighborhood and not on the whole distributed state, an intuition which has only recently (and in connection with membrane computing) been promoted to a part of the definition ([21]) (and then studied, *e.g.* [20] and [23]). Here we reconsider this notion of *locality* in Petri nets by taking into account not only the transitions as elements to be *located* (as in [21]), but also the places of the net. This shift in perspective is able to capture the correspondence between Petri nets and membrane systems more accurately, by also stressing where resources are allocated. Moreover, it promotes further reflection on the relevance of suitable topologies in the Petri nets setting, which we consider to be one of the founding ideas behind Petri nets themselves.

Inspired by the expressiveness results of membrane systems, we focus on two classes of nets: the class of *catalytic* nets, and the class of *communicating* nets. In the first class the firing strategy depends on suitable places, and it requires that if a transition using tokens from these places is enabled, then it must be executed. This class is tailored onto catalytic membrane systems, where rules are applied depending on the presence of *catalysts*, *i.e.*, objects needed for applying the rule but never *consumed*[1]. The second class we consider is characterized by the fact that transitions *move* tokens from a location to another (and never to the same location), exploiting the notion of location we use in this paper.

We then use the relationship between membrane systems and Petri nets with localities to establish the Turing completeness of these two classes of nets. In the case of the first class, it turns out that the locality mapping is not needed

[1] This notion is quite different from the one of *read* arc on Petri nets.

(or better, it can be somehow *encoded*) and the number of catalytic places needed for Turing completeness is quite limited, namely just 2 catalytic places are necessary. In the second case, the surprising result is that it is enough to guarantee Turing completeness by just considering that a transition is fired for each *location*.

The merit of the results presented in this paper is that they show that the expressiveness of Petri nets can be increased without introducing suitable arcs, without requiring that transitions are fired based on a complete snapshot of the system, and without totally losing the *confinement* of the firing of transitions which is one of the main features of Petri nets. Furthermore, the relationship established between Petri nets and membrane systems suggests that other classes of Petri nets with *minimal parallelism* (*i.e.*, where a minimal number of concurrent transitions are considered) could also be Turing complete.

The paper is organized as follows: in the next section we recall the basic notions on Petri nets and their firing strategies, then introduce the notion of locality and, beside suitable firing strategies, we point out which conditions the locality mappings should satisfy. In section 3 we introduce a new format for membrane systems which encompass the membrane models we are interested in, and in section 4 we relate membrane systems to Petri nets and *vice versa*. To establish this relationship, we use a suitable equivalence on places (the intuition is that various places correspond to the same objects in different membranes). In section 5 we recall the main definitions and the expressiveness results concerning membrane systems, whereas in section 6 we introduce two classes of Petri nets: *catalytic* and *communicating* Petri nets. The relation defined in section 4 is then used in section 7 to show that these classes of Petri nets are Turing complete.

This paper is an extended and revised version of [9], where we discussed only the case of flat catalytic membrane systems and catalytic nets.

2. Petri Nets and Firing Strategies

Notations. With $\mathbb{N}$ we denote the set of *natural numbers* including zero, and with $\mathbb{N}^+$ the set of positive natural numbers. Given a set X, with $\mathbf{2}^X$ we indicate the set of subsets of X and with $\mathbf{2}\,^X_{\mathit{fin}}$ the set of *finite* subsets of X.

Given a set X, a partial order $\sqsubseteq$ on X is a reflexive, transitive and antisymmetric relation. Let $(X, \sqsubseteq)$ be a partially ordered set and $Y \subseteq X$, we say that Y has a *minimum* iff there exists $x \in X$ such that $\forall y \in Y$ it holds that $x \sqsubseteq y$. Dually it has a *maximum* iff there exists $x \in X$ such that $\forall y \in Y$ it holds that $y \sqsubseteq x$. The elements of $Y \subseteq X$ are referred to as *incomparable* iff $\forall y, y' \in Y.\ y \neq y'$ implies that $y \not\sqsubseteq y'$ and $y' \not\sqsubseteq y$. Given two elements $x, y \in X$ such that $x \sqsubseteq y$, we say that x is an *immediate* predecessor of y iff $x \neq y$ and $\forall z \in X.\ x \sqsubseteq z \sqsubseteq y$ either $x = z$ or $z = y$.

A partial order $(X, \sqsubseteq)$ is a *tree* if $\sqsubseteq$ is such that each subset $Y \subseteq X$ of incomparable elements has no maximum, and each subset $Y \subseteq X$ has a minimum. The minimum of X is called the *root* of the tree, and it is denoted as $\mathit{root}_{\sqsubseteq}(X)$. A *leaf* of the tree is an element x such that $\forall y \in X.\ x \sqsubseteq y$, it holds that $x = y$. If x is the immediate predecessor of y, we indicate this with $x \mathrel{\hat{\sqsubseteq}} y$. We define some auxiliary partial functions over trees. Given a tree $(X, \sqsubseteq)$, we define the partial function $\mathit{father} : X \to X$ by $\mathit{father}(x) = y$ whenever $y \mathrel{\hat{\sqsubseteq}} x$. Clearly, the root of a tree has no father. The function $\mathit{children} : X \to \mathbf{2}\,^X$ is defined by $\mathit{children}(x) = \{y \in X \mid x \mathrel{\hat{\sqsubseteq}} y\}$. If x is a leaf, then $\mathit{children}(x) = \emptyset$.

Multisets. Given a set S, a *multiset* over S is a function $m : S \to \mathbb{N}$; we denote by ∂S the set of multisets of S. The *multiplicity* of an element s in m is given by $m(s)$. A multiset m over S is *finite* iff the set $\mathit{dom}(m) = \{s \in S \,|\, m(s) \neq 0\}$ is finite. We always consider finite multisets. A multiset m such that $\mathit{dom}(m) = \emptyset$ is called *empty*, and it is denoted by $\mathbf{0}$. The cardinality of a multiset is defined as $\#(m) = \sum_{s \in S} m(s)$. Given a multiset in ∂S and a subset $\hat{S}$ of S, by $m|_{\hat{S}}$ we denote the multiset over $\hat{S}$ such that $m|_{\hat{S}}(s) = m(s)$. We write $m \subseteq m'$ if $m(s) \leq m'(s)$ for all $s \in S$, and $m \subset m'$ if $m \subseteq m'$ and $m \neq m'$. The operator $\oplus$ denotes *multiset union*: $m \oplus m'(s) = m(s) + m'(s)$. The operator $\ominus$ denotes *multiset difference*: $m \ominus m'(s) =$ if $m(s) > m'(s)$ then $m(s) - m'(s)$ else 0. The *scalar product* of a number j with a multiset m is $(j \cdot m)(s) = j \cdot (m(s))$. Consider now a vector of multisets $(m_1, \ldots, m_n)$; by $\hat{\circ}$ we denote the component-wise extension of the operation $\circ \in \{\oplus, \ominus, \cdot\}$ defined over multisets. Sometimes a multiset $m \in \partial S$ is written as $\oplus_{s \in S} m(s) \cdot s$, we omit the summands whenever $m(s)$ is equal to 0. If $m \in \partial S$, we denote by $[\![m]\!]$ the multiset defined as $[\![m]\!](s) = 1$ if $m(s) > 0$ and $[\![m]\!](s) = 0$ otherwise; sometimes $[\![m]\!]$ is identified (and used interchangeably) with the corresponding subset $A = \{s \in S \mid [\![m]\!](s) = 1\}$ of S.

Petri Nets. A Petri net is a tuple $N = (S, T, F, m_0, \mathfrak{S})$, where S is a set of *places*, T is a set of *transitions*, $F : (S \times T) \cup (T \times S) \to \mathbb{N}$ is a *flow relation*, $m_0 \in \partial S$ is the initial marking, and $\mathfrak{S} \subseteq S$ is a (possibly empty) set of *final* places. Furthermore, $S \cap T = \emptyset$. Places are usually drawn as circles, transitions as boxes, and the flow relation is depicted as annotated and directed arcs from transitions to places or from places to transitions. Given a net $N = (S, T, F, m_0, \mathfrak{S})$, if $F(x, y) = 0$ then no arc is drawn from x to y, and if $F(x, y) > 0$ the arc is annotated with $F(x, y)$. With ${}^{\bullet}x$ and $x^{\bullet}$, respectively, we indicate the multiset $F(\_, x)$ and $F(x, \_)$, respectively; they are indicated as the *preset* and *postset* of x, respectively. We assume that for each transition t, $dom({}^{\bullet}t) \neq \emptyset$.

Given a net $N = (S, T, F, m_0, \mathfrak{S})$, N is called a *state machine* iff $\forall t \in T.\ |dom({}^{\bullet}t)| = |dom(t^{\bullet})| = 1$. N is called an *input state machine* iff $\forall t \in T.\ |dom({}^{\bullet}t)| = 1$ and ${}^{\bullet}t = [\![{}^{\bullet}t]\!]$.

Given a net $N = (S, T, F, m_0, \mathfrak{S})$, and a subset S' of places ($S' \subseteq S$), the *subnet* of N generated by S' is the net $N@S' = (S', T@S', F@S', m_0@S', \mathfrak{S} \cap S')$, where $T@S' = \{t \in T \mid \exists s \in S'$ and either $F(s, t) > 0$ or $F(t, s) > 0\}$, $F@S'$ is the restriction of F to S' and $T@S'$, and $m_0@S' = m_0|_{S'}$.

Two nets $N = (S, T, F, m_0, \mathfrak{S})$ and $N' = (S', T', F', m_0', \mathfrak{S}')$ are *fully compatible*, denoted by $N \cong N'$, iff there exists a bijection $\gamma : T \to T'$ and an injective embedding $\delta : S \hookrightarrow S'$ such that $\forall t \in T$ and $\forall s \in S$ it holds that $F(s, t) = F'(\delta(s), \gamma(t))$ and $F(t, s) = F'(\gamma(t), \delta(s))$; moreover, $\forall s \in S' \setminus \delta(S)$, $\forall t \in T'.\ F'(s, t) = 0 = F'(t, s)$.

We define now the *dynamics* of Petri nets. Let $m \in \partial S$ be a marking of a net, a finite multiset $U \in \partial T$ of transitions is *enabled* under m (written as

$m\,[U\rangle_{st}\,$) if for all $s \in S$ $\sum_{t\in T} U(t)\cdot F(s,t) \leq m(s)$. If a finite multiset $U \in \partial T$ is *enabled* at a marking m ($m\,[U\rangle_{st}\,$), then U may *fire* reaching a new marking m' defined as $m'(s) = m(s) + \sum_{t\in T} U(t)\cdot(F(t,s) - F(s,t))$, for all $s \in S$. We write $m\,[U\rangle_{st}\,m'$, and call U as a *step*. When considering steps, we often omit the subscript st, thus we write simply $m\,[U\rangle$ and $m\,[U\rangle\,m'$.
A *step firing sequence* is defined as follows:

- m_0 is a step firing sequence, and
- if $m_0\,[U_1\rangle\,m_1\ \ldots\ m_{n-1}\,[U_n\rangle\,m_n$ is a step firing sequence and $m_n\,[U_{n+1}\rangle\,m'$ is a step, then also $m_0\,[U_1\rangle\,m_1 \ldots m_n\,[U_n\rangle\,m'$ is a step firing sequence.

The empty sequence is associated to the step firing sequence m_0.

Given a net $N = (S, T, F, m_0, \mathfrak{S})$, a marking m is *reachable* if there is a step firing sequence leading to it, *i.e.*, $m_0\,[U_1\rangle\,m_1 \ldots m_{n-1}\,[U_n\rangle\,m_n$ with $m = m_n$. The set of reachable markings of the net N is denoted by $\mathfrak{M}(N)$. Given a Petri net N, it has been shown that the problem of deciding whether a given marking m is reachable from the initial one (*i.e.*, if $m \in \mathfrak{M}(N)$) is decidable (see [13] for a quite complete survey).

The *ordinary* firing rule of Petri nets (just one enabled transition is fired at each marking, regardless of how many may be simultaneously enabled) is an instance of the step firing rule we have defined previously: in this case, the multiset U is such that $U = [\![U]\!]$ and $\#(dom(U)) = 1$. Given a step $m\,[U\rangle\,m'$, it can always be *linearized* and m' can be reached from m with a possibly different step firing sequence. In fact, we assume that $U \neq [\![U]\!]$ (otherwise the claim is trivial). Then U can be written as the sum of various steps, *i.e.*, there exist $U_1, \ldots, U_n$ such that $\oplus_i^n U_i = U$, and $m\,[U_1\rangle\,m_1\,[U_2\rangle\,m_2 \ldots m_{n-1}\,[U_n\rangle\,m'$ is a step firing sequence. The claim can be proved by induction on n. The basis is trivial. Assume it holds for n, let us show for $(n+1)$. Write $U = \oplus_i^{n+1} U_i = U$, and take $U' = \oplus_i^n U_i$. By induction, we have the step firing sequence $m\,[U_1\rangle\,m_1\,[U_2\rangle\,m_2 \ldots m_{n-1}\,[U_n\rangle\,m_n$ which corresponds to the step $m\,[\oplus_i^n U_i\rangle\,m_n$. As $m\,[U\rangle$, then also $m\,[U_{n+1}\rangle$; thus it is enough to prove that $m_n\,[U_{n+1}\rangle$. Assume it is not, then there is a place s such that for some

$t \in T$ with $t \in dom(U_{n+1})$ such that $m_n(s) \leq \sum U_{n+1}(t) \cdot F(s,t)$, and clearly $m(s) \geq \sum U_{n+1}(t) \cdot F(s,t)$, thus there must be a transition $t' \in T$ and an index j with $1 \leq j \leq n$ such that $U_j(t') > 0$ and $F(s,t') > 0$. But this contradicts the assumption that $m\,[U\rangle$.

Let $m\,[U_1\rangle\, m_1\,[U_2\rangle\, m_2 \ldots m_{n-1}\,[U_n\rangle\, m'$ be a step firing sequence and assume that each U_i is bounded by a constant k, namely the maximal cardinality of the multisets U_i is k. We write $m\,[U_1\rangle_{k-b}\, m_1\,[U_2\rangle_{k-b}\, m_2 \ldots m_{n-1}\,[U_n\rangle_{k-b}\, m'$ to indicate that each step in this firing sequence is such that the cardinality of the step is at most k, and call these firing sequences *k-bounded*. The set of reachable markings with k-bounded step firing sequence is denoted by $\mathfrak{M}_{k-b}(N)$. With this notation, the usual set of reachable markings is $\mathfrak{M}_1(N)$.

Another crucial definition is that of a *maximal* step. A step U enabled at a marking m is maximal iff each step U' such that $U \subset U'$ is such that $\neg\ (m\,[U'\rangle\,)$. A maximal step is denoted as $m\,[U\rangle_{max}\, m'$, and a maximal step firing sequence is a step firing sequence where each step is maximal. The set of reachable markings of a net N with maximal step firing sequences is $\mathfrak{M}_{max}(N)$. In this case it holds that $\mathfrak{M}_{max}(N) \subseteq \mathfrak{M}(N)$, and the inclusion may be proper (it is often so, as reachability in the case of this firing rule is undecidable [5]).

Given a net N, we may be interested in firing rules where it is guaranteed that if the tokens from specific places can be *used* (i.e., there are transitions enabled at that marking which consume these tokens), then they are actually used. More formally, consider the net $N = (S, T, F, m_0, \mathfrak{S})$ and a subset of places $\mathcal{S} \subseteq S$. A step U enabled at a marking m is $\mathcal{S}$-enabled iff for all $s \in \mathcal{S}$ either there exists a $t \in dom(s^\bullet)$ and $U(t) \neq 0$ or $\forall t \in dom(s^\bullet)$ it holds that $\neg m\,[t\rangle$. We write $m\,[U\rangle_{\mathcal{S}}$ to indicate this and the corresponding step is $m\,[U\rangle_{\mathcal{S}}\, m'$, and it is called an $\mathcal{S}$-step. Firing sequences where each step is an $\mathcal{S}$-step are defined as usual and the set of reachable markings under this firing rule is $\mathfrak{M}_{\mathcal{S}}(N)$.

To simplify the notation we use a subscript to indicate which firing rule is used: *step*, *maximal*, *k-bounded*, *$\mathcal{S}$-step*. Some others are introduce later. When no subscript is used, we assume that the step firing rule is used.

Given a net N and a marking m of $N = (S, T, F, m_0, \mathfrak{S})$, we say that m is a *final* marking iff $m \in \mathfrak{M}(N)$ and $\forall t \in T.\ \neg(m\,[t\rangle\ )$. We are interested in the reachable markings (under a firing rule $fr \in \{step, max, k-b, \mathcal{S}\}$) that are also final: $\mathfrak{F}_{fr}(N) = \{m \in \mathfrak{M}_{fr}(N) \mid m \text{ is final}\}$. Among the final markings, we may be interested in considering either the numbers of tokens contained in certain places, or the marking in these places. Let $N = (S, T, F, m_0, \mathfrak{S})$ be a net, and $\hat{S} \subseteq S$ be a set of distinguished places; by $\mathfrak{F}^{\hat{S}}_{fr}(N)$ and $\mathfrak{F}^{\#\hat{S}}_{fr}(N)$ we denote the sets $\{m|_{\hat{S}} \mid m \in \mathfrak{F}_{fr}(N)\}$ and $\{\#(m|_{\hat{S}}) \mid m \in \mathfrak{F}_{fr}(N)\}$, respectively.

Petri Nets with Localities. We introduce the notion of *locality* in Petri nets. The rationale behind this definition is to be able to *identify* regions where the firings of transitions happen. Localities are essentially indexes, namely natural numbers. With respect to other approaches in literature, *e.g.* [20, 21] and [23], our notion of locality is defined over places and transitions, whereas in these papers localities are considered just for transitions.

Definition 1. *Let $N = (S, T, F, m_0, \mathfrak{S})$ be a Petri net. A* locality mapping *is a total function $\mathcal{L} : T \cup S \to \mathbb{N}^+$. A* Petri net with localities *is the pair $N = ((S, T, F, m_0, \mathfrak{S}), \mathcal{L})$ where $(S, T, F, m_0, \mathfrak{S})$ is a Petri net and $\mathcal{L}$ is a locality mapping.*

Petri nets with localities will be denoted by PNL. Obviously each Petri net can be seen as a PNL: it is enough to assign the same locality (*i.e.*, an index) to each transition and place. However, we are interested in Petri nets where localities are organized at least in a tree-like fashion, and where transitions use places either in the same locality or in neighboring localities (*i.e.*, in the father or in the children).

Definition 2. *Let $N = ((S, T, F, m_0, \mathfrak{S}), \mathcal{L})$ be a PNL. We say that N is* correctly located *iff*

1. *there exists a relation $\prec \subseteq \mathcal{L}(S \cup T) \times \mathcal{L}(S \cup T)$ such that $(\mathcal{L}(S \cup T), \prec)$ is a tree,*
2. *$\forall t \in T$, $\forall s \in S$ such that $F(s, t) > 0$ or $F(t, s) > 0$, only one of the following holds:*

(a) $\mathcal{L}(s) = \mathcal{L}(t)$,

(b) $\mathcal{L}(s) \mathrel{\hat{\prec}} \mathcal{L}(t)$,

(c) $\mathcal{L}(t) \mathrel{\hat{\prec}} \mathcal{L}(s)$.

Condition 1 states that the location assignment of the net is a tree, whereas condition 2 indicates that a transition produces and consumes tokens from places either in the same locality, or in a locality in the neighborhood (of the locality of the transition).

More than one relation can be associated to a correctly located net. Consider the net in Figure 1; then, given the labeling $\mathcal{L}(t) = 1$, $\mathcal{L}(t') = 2$, $\mathcal{L}(t'') = 3$, $\mathcal{L}(s) = 2$, $\mathcal{L}(s'') = 3$ and $\mathcal{L}(s''') = \mathcal{L}(s') = 1$, the following relations satisfy the requirements of Definition 2: either $1 \prec 2$ and $1 \prec 3$, or $2 \prec' 1$ and $1 \prec' 3$, or $3 \prec'' 1$ and $1 \prec'' 2$. This issue will be discussed later in the paper.

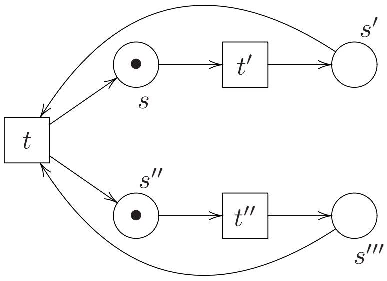


Figure 1: A Petri net with more than one partial order relation on localities.

In case we need to fix a relation, we explicitly indicate the partial order relation on labels by writing $N = ((S, T, F, m_0, \mathfrak{S}), \mathcal{L}, \prec)$.

The added information on locations may be used to introduce other firing rules on nets. Let $N = ((S, T, F, m_0, \mathfrak{S}), \mathcal{L})$ be a PNL, and U be a step enabled at the marking m. We say that U is *locally maximal* enabled if $\forall t, t' \in dom(U)$. $\mathcal{L}(t) = \mathcal{L}(t')$ and each step U' with $U \leq U'$ such that $m\,[U'\rangle$, then U' is not *locally maximal* enabled. The firing of a locally maximal enabled step U is denoted by $m\,[U\rangle_{lmax}\,m'$, and the step U is said to be locally maximal. In a locally maximal enabled step all the enabled transitions belonging to a location have to be executed. A locally maximal step firing sequence is a step firing sequence where each step is locally maximal. We say that U is *locally k-bounded* enabled if $\forall t, t' \in dom(U)$. $\mathcal{L}(t) = \mathcal{L}(t')$ and $\#(U) \leq k$. The firing of a locally k-bounded enabled step U is denoted by $m\,[U\rangle_{l-kb}\,m'$, and the step U is said to be locally k-bounded. A locally k-bounded step firing sequence is a step firing sequence where each step is locally k-bounded. Consider now a subset of places $\mathcal{S} \subseteq S$ such that $\forall s, s' \in \mathcal{S}$ it holds that $\mathcal{L}(s) = \mathcal{L}(s')$. A step U enabled at a marking m is $l - \mathcal{S}$-enabled iff for all $s \in \mathcal{S}$ either there exists a $t \in dom(s^\bullet)$ and $U(t) \neq 0$, or for all $t \in dom(s^\bullet)$ it holds that $\neg m\,[t\rangle$. We write $m\,[U\rangle_{l-\mathcal{S}}$ to indicate this, and the corresponding step is $m\,[U\rangle_{\mathcal{S}}\,m'$; it is called an $l - \mathcal{S}$-step. Finally we say that U is *locality complete* enabled at the marking m if, considering the sets $T_i = \{t \in T \mid \mathcal{L}(t) = i\}$, then $\forall i \in \mathcal{L}(T)$ either $\#(U|_{T_i}) > 0$ or $\forall t \in T_i$. $\neg m\,[t\rangle$. Thus all the localities where there is an enabled transition are *involved* in a locality complete step. The firing of a locality complete step U is denoted by $m\,[U\rangle_{lcomp}\,m'$, and the step U is said to be locality complete. A locality complete step firing sequence is a step firing sequence where each step is locality complete.

Corresponding to these rules we have the set of markings reachable with firing sequences of the proper kind: $\mathfrak{M}_{fr}(N)$ with $fr \in \{lmax, l - kb, l{-}\mathcal{S}, lcomp\}$, and consequently the sets $\mathfrak{F}_{fr}(N)$, $\mathfrak{F}^{\hat{S}}_{fr}(N)$ and $\mathfrak{F}^{\#\hat{S}}_{fr}(N)$.

3. Membrane Systems and Evolution Strategies

Membrane systems (also called P systems) represent abstract computing models based on rules over multisets. They are inspired by the structure of (eukaryotic) cells, namely by the role of their membranes in delimiting different compartments, and thus can help to understand information processes in the nature. The most comprehensive recent monograph is the Oxford Handbook of Membrane Computing [27]. Membrane computing was introduced in [26], volume [8] presents various applications, while the P Systems webpage `ppage.psystems.eu` presents recent developments.

Starting from the observation that there is an obvious parallelism in cell biochemistry [2], and relying on the assumption that “if we wait enough, then all reactions which may take place will take place”, a basic feature of the P systems is the maximally parallel way of using the rules: in each step, in each region of a system, we have to use a maximal multiset of rules. This maximal parallelism decreases the non-determinism of the systems evolution, and it provides enough power to get computational Turing completeness.

There are several types of membrane systems (defined by various ingredients inspired from cell biology); here we introduce a *format* which is common to many of them, in particular to the two classes we are interested in: catalytic membrane systems and communicating P systems (also called symport/antiport membrane systems).

Membranes. The language of *membrane structure*, denoted by MS, is a language over parentheses $\{[,]\}$ whose strings are defined either as $[\ ] \in MS$, or $[\mu_1 \dots \mu_n] \in MS$ whenever $\mu_1, \dots, \mu_n \in MS$ with $n \geq 1$, and nothing else is in MS. Given a string $\mu \in MS$, a *matching pair* of parentheses is any substring of μ which is again a string in MS. We call *membrane* each matching pair of parentheses appearing in the membrane structure. The same membrane structure can be represented by several equivalent strings (the equivalence being that $\mu_1\mu_2\mu_3\mu_4 \equiv \mu_1\mu_3\mu_2\mu_4$, for $\mu_1, \mu_4 \in MS$ and $\mu_2, \mu_3 \in MS$), hence we assume that its canonical representation is given by a tree-like structure. The number of membranes appearing in a membrane structure μ is calculated as follows:

$$\#_{MS}(\mu) = \begin{cases} 1 & \text{if } \mu = [\ ] \\ 1 + \sum_{i=1}^{k} \#_{MS}(\mu_i) & \text{if } \mu = [\mu_1 \dots \mu_k] \end{cases}$$

Thus each membrane appearing in a membrane structure μ can be uniquely labelled with an index in $\{1, \dots, \#_{MS}(\mu)\}$. We assume that the index 1 is given to the root. The index of a membrane μ_i appearing in μ is given by $index(\mu_i) \in \{1, \dots, \#_{MS}(\mu)\}$. The set $(\{1, \dots, \#_{MS}(\mu)\}, \sqsubseteq^*)$ is a tree, where $index(\mu) \sqsubset index(\mu_i)$ whenever $\mu = [\mu_1 \dots \mu_k]$ and $i \in \{1, \dots, k\}$.

Membrane Systems. Let O be a finite alphabet of (names of) objects (also called molecules), then a *membrane system* over O is a construct $\Pi = (O, \mu, w_1^0, \ldots, w_n^0, R_1, \ldots, R_n, i_o)$ where:

- μ is a *membrane structure* with n membranes indexed $1, \ldots, \#_{MS}(\mu)$, and $n = \#_{MS}(\mu)$,
- each w_i^0 is a multiset over O associated with membrane i, and
- each R_i is a finite set of *reaction (or evolution) rules* r associated with the membrane i, each rule having the form $u \rightarrow v$, where u and v are finite multisets over $O \times (\{\mathit{here}, \mathit{out}\} \cup \{\mathit{in}_j \mid \mathit{father}(j) = i\})$, such that $u \neq \mathbf{0}$;
- $i_o \in \{1, \ldots, n\}$ is the output membrane.

With respect to the usual definition of membrane system, we allow that the left hand side of a rule may use objects in neighboring membranes, and not only in the same membrane. This is done to make communicating P system a special case of this more general format, as it will be given later.

Given a rule r of form $u \rightarrow v$, u is the left hand side of r and v is the right hand side of r, and they are denoted with $\mathit{lhs}(r)$ and $\mathit{rhs}(r)$, respectively. To simplify the notation, given a multiset z over $O \times (\{\mathit{here}, \mathit{out}\} \cup \{\mathit{in}_j \mid \mathit{father}(j) = i\})$, with $\pi(z)|_\alpha$ we denote the multiset on O obtained from z by considering all the elements with the second component equal to α, where $\alpha \in \{\mathit{here}, \mathit{out}, \mathit{in}_1, \ldots, \mathit{in}_n\}$. *here* obviously means that the objects in $\mathit{rhs}(r)$ have to be taken from or put in the same membrane, *out* that the objects have to be taken from or put in the *father* membrane, and in_j that the objects have to be taken from or put in the j-th child membrane. We keep the usual notation, as it conveys an intuition about the direction the objects take during their evolution.

Membrane Systems Evolution. A membrane system Π evolves from a configuration to another configuration as a consequence of the application of (multisets of) rules in each region. The rules have to be applied *simultaneously.* We start formalizing the notion of configuration of a membrane system. If Π is a membrane system, then a *configuration* is a tuple $C = (w_1, \ldots, w_n)$ where each w_i is a multiset over O. $C_0 = (w_1^0, \ldots, w_n^0)$ is the *initial* configuration of Π. The set of configurations of a membrane system Π is denoted with Conf_Π. If $\Pi = (O, \mu, w_1^0, \ldots, w_n^0, R_1, \ldots, R_n, i_o)$ is a membrane system, then a *multi-rule vector* $\vec{R}$ is the n-uple $(\widehat{R}_1, \ldots, \widehat{R}_n)$, where $\widehat{R}_i$ is a multiset over R_i. The set of multi-rule vectors is denoted by $\mathcal{R}$. The multi-rule vector $\vec{R}$ contains all the rules that have to be applied simultaneously to a configuration of a membrane system, with their proper multiplicities.

The enabling condition is slightly more difficult to handle, as for all regions (membranes) there must be enough objects that can be used by several rules in the same membrane or in neighboring ones. Given a multi-rule vector $\vec{R}$ by $\mathit{LHS}(\vec{R})$, we denote a tuple of n multisets over O, where for each i between 1 and n we denote by $\mathit{LHS}(\vec{R})_i$ the multiset over O defined as follows: $(\bigoplus_{r \in R_i} \widehat{R}_i(r) \cdot \pi(\mathit{lhs}(r))|_{here})$ are the objects to be consumed by rules in the same membrane, $(\bigoplus_{r \in R_{father(i)}} \widehat{R}_{father(i)}(r) \cdot \pi(\mathit{lhs}(r))|_{in_i})$ are those of the rules of the father membrane, and $(\bigoplus_{j \in children(i)} (\bigoplus_{r \in R_j} \widehat{R}_j(r) \cdot \pi(\mathit{lhs}(r))|_{out}))$ those to be consumed by rules from each child membrane; these three parts have to be combined by using $\oplus$. Thus a vector multi-rule $\vec{R}$ is enabled at a configuration $C = (w_1, \ldots, w_n)$ if $\forall i.\ \mathit{LHS}(\vec{R})_i \subseteq w_i$. The effects of the *application* of a multi-rule vector $\vec{R}$ in the membrane i are the following: $(\bigoplus_{r \in R_i} \widehat{R}_i(r) \cdot \pi(\mathit{rhs}(r))|_{here})$ the effect of the rule in the same membrane, $(\bigoplus_{r \in R_{father(i)}} \widehat{R}_{father(i)}(r) \cdot \pi(\mathit{rhs}(r))|_{in_i})$ those of the rules in the father membrane, and finally $(\bigoplus_{j \in children(i)} (\bigoplus_{r \in R_j} \widehat{R}_i(r) \cdot \pi(\mathit{rhs}(r))|_{out}))$ those from the children membranes. Like previously, these three parts are combined by using $\oplus$. For each membrane, we denote the effects by $\mathit{RHS}(\vec{R})_i$.

Once a multi-rule vector $\vec{R}$ is enabled at $C = (w_1, \ldots, w_n)$, we can describe the *evolution* in each membrane by $\forall i.\ w_i' = w_i \ominus \mathit{LHS}(\vec{R})_i \oplus \mathit{RHS}(\vec{R})_i$. Given a configuration $C = (w_0, \ldots, w_n)$ and a multi-rule vector $\vec{R}$ enabled at C, then the application of the multi-rule vector produces the configuration $C' = (w_0 \ominus \mathit{LHS}(\vec{R})_0 \oplus \mathit{RHS}(\vec{R})_0, \ldots, w_n \ominus \mathit{LHS}(\vec{R})_n \oplus \mathit{RHS}(\vec{R})_n)$. This is denoted by

$C \overset{\vec{R}}{\Longrightarrow}_{step} C'$, and it is called a *reaction step.*

We now formalize the *chain* of "reactions" for a given membrane system: C_0 is a reaction sequence, and if $C_0 \overset{\vec{R}_1}{\Longrightarrow}_{step} C_1 \dots C_{n-1} \overset{\vec{R}_n}{\Longrightarrow}_{step} C_n$ is a reaction sequence, $C_n \overset{\vec{R}}{\Longrightarrow}_{step} C$, then $C_0 \overset{\vec{R}_1}{\Longrightarrow}_{step} C_1 \dots C_n \overset{\vec{R}}{\Longrightarrow}_{step} C_{n+1}$ is also a reaction sequence. A configuration C is said to be *reachable* if there is a reaction sequence starting from the initial configuration and leading to C, *i.e.* $C_0 \overset{\vec{R}_1}{\Longrightarrow}_{step} C_1 \dots C_{n-1} \overset{\vec{R}_n}{\Longrightarrow}_{step} C_n$ with $C = C_n$. Given a membrane system Π, the set of reachable configurations is denoted by $\mathfrak{H}_{step}(\Pi)$.

The evolution of membrane systems may also have several strategies, though usually it is assumed that in each membrane all the applicable rules are actually applied in a *maximally parallel* way. However, other strategies may be used, and we recall some of them here. The first strategy we present is the *maximal* one: given a configuration C, a multi-rule vector $\vec{R}$ is *maximally* enabled if it is impossible to add any instance of a rule to any $\widehat{R}_i$; consequently, this reaction step is denoted by $C \overset{\vec{R}}{\Longrightarrow}_{max} C'$, and a maximal reaction sequence is a reaction sequence where each step is maximal. The set of the reachable configurations under this strategy is denoted by $\mathfrak{H}_{max}(\Pi)$.

The whole spectrum devised for Petri nets can be instantiated here: k-boundedness means the that the overall number of rules (counting the instances) is bounded by k; object-awareness means that if certain objects are present and there is a rule that may use them, this rule should be used; locally maximal means that only rules from a membrane are used in maximal way; locally k-bounded means that the rules from a single membrane are used, and that the overall number (counting the instances) is bounded by k; local object awareness means that if certain objects are present and there is a rule that may use them, they should be used by rules from a membrane; locality completeness means that at least one rule from each membrane is used (if present, and if enabled). We adopt the same subscripts we introduced for the Petri nets firing strategies.

Similarly for what we have done for Petri nets (with localities), we can say that a configuration of a membrane system is final if no further evolution is possible. We introduce the set $\mathfrak{G}_{fr}(\Pi) = \{C \mid C \in \mathfrak{H}_{fr}(\Pi) \text{ and } C \text{ is final}\}$. In membrane systems, the output is usually located in a specific membrane, hence with $\mathfrak{G}^i_{fr}(\Pi)$ we denote the sets $\{w_i| \ (w_0, \ldots, w_i, \ldots, w_n) \in \mathfrak{G}_{fr}(\Pi)\}$, and with $\mathfrak{G}^{\#i}_{fr}(\Pi)$ we denote the set $\{\#(w_i) \mid (w_0, \ldots, w_i, \ldots, w_n) \in \mathfrak{G}_{fr}(\Pi)\}$.

It should be stressed that the notion of membrane system we have introduced here encompasses many different variants of membrane systems, and each variant is obtained by imposing suitable constraints on the kind of rules allowed. Nevertheless, these variants can be easily related to Petri nets with localities, as will be described in the next section.

4. Relating Petri Nets and Membrane Systems

From Membrane System to Petri Nets with Localities. The intuition of encoding of a membrane system into a Petri net with localities is exactly the same as that presented in [21] and further developed (see [27]): a place is associated to each object and each index of the membrane system, and a transition is associated to each rule belonging to a membrane with the locality being equal to the index of the membrane.

Definition 3. *If $\Pi = (O, \mu, w^0_1, \ldots, w^0_n, R_1, \ldots, R_n, i_0)$ is a membrane system, then we associate to it the structure $\mathcal{F}(\Pi) = ((S, T, F, m_0, \mathfrak{S}), \mathcal{L})$, where:*

- *$S = O \times \{1, \ldots, n\}$, and $T = \bigcup_{i=1}^n \{t^r_i \mid r \in R_i\}$,*
- *for all transitions $t = t^r_i \in T$ and places $s \in S$ with $r = u \to v$, we define*

$$F(s,t) = \begin{cases} u(a, here)) & \text{if } s = (a,i) \\ u((a, out)) & \text{if } j = father(i) \text{ and } s = (a,j) \\ u((a, in_j)) & \text{if } j \in children(i) \text{ and } s = (a,j) \\ 0 & \text{otherwise} \end{cases}$$

$$F(t,s) = \begin{cases} v((a, here)) & \text{if } s = (a,i) \\ v((a, out)) & \text{if } j = father(i) \text{ and } s = (a,j) \\ v((a, in_j)) & \text{if } j \in children(i) \text{ and } s = (a,j) \\ 0 & \text{otherwise} \end{cases}$$

- $m_0(s) = \begin{cases} w_i^0(a) & \textit{if } s = (a,i), \quad i \in \{1,\dots,n\} \\ 0 & \textit{otherwise} \end{cases}$
- $\mathfrak{S} = \{(a, i_0) \mid a \in O\}$

and $\mathcal{L}(t_i^r) = i$ *and* $\mathcal{L}(s) = i$ *if* $s = (a,i)$.

The multiplicity of the object a in the membrane i is modeled by the number of tokens in the place (a,i).

Proposition 1. *Let* $\Pi = (O, \mu, w_1^0, \dots, w_n^0, R_1, \dots, R_n, i_0)$ *be a membrane system, and* $\mathcal{F}(\Pi) = ((S,T,F,m_0,\mathfrak{S}), \mathcal{L})$ *be the associated structure. Then* $\mathcal{F}(\Pi)$ *is a correctly located PNL net.*

Proof: $(S,T,F,m_0,\mathfrak{S})$ is clearly a Petri net. We have to verify that $\mathcal{F}(\Pi)$ is correctly located, *i.e* that $(\mathcal{L}(S \cup T), \prec)$ is a tree for a suitable partial order $\prec$, and furthermore the other conditions are satisfied. μ is a membrane structure with indexes $\{1, \dots, \#_{MS}(\mu)\}$ and with $\#_{MS}(\mu) = n$; we assume without loss of generality that 1 is the root. Consider an index $i \in \{1,\dots,n\}$ with $i \neq 1$, and assume that $index(\mu') = i$ with $\mu' = [\mu_{i_1} \dots \mu_{i_j}]$. Set $i \hat{\prec} index(\mu_{i_k})$ with $1 \leq k \leq j$ and $\prec = \hat{\prec}^*$. $(\mathcal{L}(S \cup T), \prec)$ is a partial order, and obviously $(\mathcal{L}(S \cup T), \prec)$ is a tree. Consider now $s \in S$ and $t \in T$ such that $F(s,t) > 0$. The cases are the following:

- $F(s,t) = u((a, here))$ for a rule $r = u \to v$ and then $\mathcal{L}(s) = \mathcal{L}(t^r)$,
- $F(s,t) = u((a, in_j))$ and $j \in children(i)$ with $s = (a,j)$ and $\mathcal{L}(t^r) = i$. By definition, $\mathcal{L}(t^r) \hat{\prec} \mathcal{L}(s)$,
- $F(s,t) = u((a, out))$ and $j = father(i)$ with $s = (a,j)$ and $\mathcal{L}(t^r) = i$. By definition, $\mathcal{L}(s) \hat{\prec} \mathcal{L}(t^r)$.

The cases where $F(t,s) > 0$ are similar.

Thus we can conclude that $\mathcal{F}(\Pi)$ is a correctly located PNL net. □

Following [21], we introduce two functions: one associating to a configuration of a membrane system a marking of the corresponding net, and one associating to the rules applied in an evolution of the membrane system a step in the net. Observe that in the following two definitions, the locality mapping and membrane structure play a small rôle.

Definition 4. *Let* $\Pi = (O, \mu, w_1^0, \dots, w_n^0, R_1, \dots, R_n, i_0)$ *be a membrane system, and* $\mathcal{F}(\Pi) = ((S, T, F, m_0, \mathfrak{S}), \mathcal{L})$ *be the associated PNL. If* $C = (w_1, \dots, w_n)$ *is a configuration of* Π*, then* $\nu(C)$ *is the marking defined as* $\nu(C)((v, i)) = w_i(v)$ *for all* $v \in O$ *and membranes* i*,* $1 \leq i \leq n$*.*

Given a membrane system $\Pi = (O, \mu, w_1^0, \dots, w_n^0, R_1, \dots, R_n, i_0)$, we assume without loss of generality that all the rules have a different index: the j-th rule of the membrane i (i.e., a rule in R_i) is denoted with r_i^j and the corresponding transition t_i^j. To denote the multiplicity of a given rule r_i^j in a multi-rule vector $\vec{R} = (\widehat{R}_1, \dots, \widehat{R}_n)$ we will often write $\vec{R}(r_i^j)$ rather than $\widehat{R}_i(r_i^j)$.

Definition 5. *Let* $\Pi = (O, \mu, w_1^0, \dots, w_n^0, R_1, \dots, R_n, i_0)$ *be a membrane system, and* $\mathcal{F}(\Pi) = ((S, T, F, m_0, \mathfrak{S}), \mathcal{L})$ *be the associated PNL. If* $\vec{R} \in \mathcal{R}$ *is a multi-rule vector of* Π*, then* $\sigma(\vec{R})$ *is the multiset of transitions of* $\mathcal{F}(\Pi)$ *defined as* $\sigma(\vec{R})(t_i^j) = \vec{R}(r_i^j)$ *for all* $t_i^j \in T$*.*

These functions define a bijection among the reachable markings and the reachable configurations (using a certain firing - evolution strategy).

We can now state the main result presented in [21]. The authors do not prove it for all the firing strategies introduced previously, but each of them can be proved using the same argument. We refer to [21] for the proof of the result. We just emphasize that the added generality in membrane systems does not change the reasoning. In fact, being $\mathcal{F}(\Pi)$ correctly located, $\nu(LHS(r))$ has non empty entries only in the neighboring localities of $\mathcal{L}(\sigma(r))$ (with respect to the partial order defined by the membrane structure μ), hence the enabling in the net and in the membrane structure properly correspond.

Theorem 1. *Let* $\Pi = (O, \mu, w_1^0, \dots, w_n^0, R_1, \dots, R_n, i_0)$ *be a membrane system, and* $\mathcal{F}(\Pi) = ((S, T, F, m_0, \mathfrak{S}), \mathcal{L})$ *be the associated PNL.* $C \stackrel{\vec{R}}{\Longrightarrow}_{fr} C'$ *iff* $\nu(C)\,[\sigma(\vec{R})\rangle_{fr}\,\nu(C')$ *with* $fr \in \{step, max, k-b, \mathcal{S}, lmax, l-kb, l-\mathcal{S}, lcomp\}$*.*

This theorem essentially says that, provided that the net steps are performed accordingly in the same way as the evolution steps of the membrane system are defined, the net associated to a membrane system behaves in the same way as the membrane system it is derived from.

From Correctly Located PNL to Membrane Systems. Given a correctly located Petri net, it is always possible to associate a membrane system to it. The membrane structure is derived by the locality mapping, rules are obtained from transitions and objects from places. In the translation from membrane systems to Petri nets we have associated to each object and each membrane a place; now we do not have enough information to *guess* the minimal set of objects, and so we obtain a membrane system with a set of objects which may not be *optimal* (i.e., minimal). To partially solve this problem, we introduce a suitable equivalence relation on places indicating when two different places *represent* the same object (in two different membranes).

The intuition on how to obtain a membrane system out of a PNL is slightly different from the one that guided the other way. From the locality mapping it is possible to obtain a membrane structure, and then the objects of the membrane system are just the places (though, as suggested above, the same object in two different membranes may result in two different objects). To try to solve this problem we introduce an equivalence relation on places of a correctly located net, and this equivalence relation should satisfy the minimal requirement that two places in the same location cannot be equivalent. Thus, given a correctly located Petri net $N = ((S, T, F, m_0, \mathfrak{S}), \mathcal{L})$, we define an equivalence relation $\equiv$ on S such that $\forall s, s' \in S$, $s \neq s'$ and $s \equiv s'$ implies $\mathcal{L}(s) \neq \mathcal{L}(s')$. Such a relation is called an *object* relation.

We first show how to obtain a membrane structure out of a correctly located net.

Definition 6. *Let $N = ((S, T, F, m_0, \mathfrak{S}), \mathcal{L})$ be a correctly located Petri net, let $\prec$ be a relation on $\mathcal{L}(T \cup S)$ satisfying definition 2, and let $\equiv$ be an object relation. Let $a \in \mathcal{L}(T \cup S)$. Then $\alpha_{\prec}(a)$ is the membrane structure μ_a defined* *as follows:* $\alpha_{\prec}(a) = [\alpha_{\prec}(b_1), \ldots, \alpha_{\prec}(b_m)]_a$ *if* $children(a) = \{b_1, \ldots, b_m\}$ *and* $\alpha_{\prec}(a) = [\ ]_a$ *if* $children(a) = \emptyset$*.*

Proposition 2. *Let* $N = ((S, T, F, m_0, \mathfrak{S}), \mathcal{L})$ *be a correctly located Petri net, let* $\prec$ *be a relation on* $\mathcal{L}(T \cup S)$ *satisfying definition 2, and let* $\equiv$ *be an object relation. Then* $\sigma_{\prec}(N)$ *defined as* $\sigma_{\prec}(N) = \alpha_{\prec}(root_{\prec}(\mathcal{L}(T \cup S)))$ *is a membrane structure.*

Proof: Trivial. □

The correctly located net we are interested in are nets where all the final places have the same location.

Definition 7. *Let* $N_{\prec} = ((S, T, F, m_0, \mathfrak{S}), \mathcal{L})$ *be a correctly located Petri net. We say that* N *is a* membrane *net iff* $\forall s, s' \in \mathfrak{S}$*,* $\mathcal{L}(s) = \mathcal{L}(s')$*.*

We show how to associate a membrane system to a correctly labeled net. Observe that the choice of the relation making a net correctly located is arbitrary, as well as the equivalence relation on places, hence the mapping is made dependent on these parameters.

Definition 8. *Let* $N_{\prec} = ((S, T, F, m_0, \mathfrak{S}), \mathcal{L})$ *be a membrane net, let* $\prec$ *be a relation on* $\mathcal{L}(T \cup S)$ *satisfying definition 2, and let* $\equiv$ *be an object relation. Then* $\mathcal{K}^{\equiv}_{\prec}(N) = (O, \mu, w^0_1, \ldots, w^0_n, R_1, \ldots, R_n, i_0)$ *is the membrane system defined as follows:*

- $O = S/_{\equiv}$,
- $\mu = \sigma_{\prec}(N)$,
- $w^0_i(a) = m(s)$*, if* $a = [s]_{\equiv}$ *and* $\mathcal{L}(s) = i$*, for all* $i \in \mathcal{L}(T \cup S)$*,*
- *for all* $i \in \mathcal{L}(T)$ *and all* $t \in T$ *such that* $\mathcal{L}(t) = i$ *we define a rule* $r^t_i = u \to v$ *in* R_i *where* u *and* v *are the multiset on* $O \times \{here, out, in_j\}$ *defined by*
  - $\forall s \in dom({}^{\bullet}t)$

    - $*$ *if* $\mathcal{L}(s) = \mathcal{L}(t)$ *then* $u(a, here) = {}^{\bullet}t(s)$*, where* $a = [s]_{\equiv}$*,*
    - $*$ *if* $\mathcal{L}(s)\hat{\prec}\mathcal{L}(t)$ *then* $u(a, out) = {}^{\bullet}t(s)$*, where* $a = [s]_{\equiv}$*, and*
    - $*$ *if* $\mathcal{L}(t)\hat{\prec}\mathcal{L}(s)$ *then* $u(a, in_{\mathcal{L}(s)}) = {}^{\bullet}t(s)$*, where* $a = [s]_{\equiv}$*,*
  - $\forall s \in dom(t^{\bullet})$
    - $*$ *if* $\mathcal{L}(s) = \mathcal{L}(t)$ *then* $v(a, here) = t^{\bullet}(s)$*, where* $a = [s]_{\equiv}$*,*
    - $*$ *if* $\mathcal{L}(s)\hat{\prec}\mathcal{L}(t)$ *then* $v(a, out) = t^{\bullet}(s)$*, where* $a = [s]_{\equiv}$*, and*
    - $*$ *if* $\mathcal{L}(t)\hat{\prec}\mathcal{L}(s)$ *then* $v(a, in_{\mathcal{L}(s)}) = t^{\bullet}(s)$*, where* $a = [s]_{\equiv}$*, and*
- $i_0 = n$*, where* n *is the location of the final places.*

The intuition is as follows: to each transition corresponds a rule. As the transition consumes tokens from places in the neighboring localities and produces tokens in places of the neighboring localities as well, the associated rule uses *objects* from the neighboring membranes (corresponding to the localities) and produces objects in the neighboring membranes, mimicking the transition. The equivalence on places is used to find the correct object *name*, and the localities are used to establish which among the tags *here*, *out* and in_i have to be used.

A simple observation is that $\mathcal{K}^{\equiv}_{\prec}(N)$ is indeed a membrane system.

Theorem 2. *Let* $N_{\prec} = ((S, T, F, m_0, \mathfrak{S}), \mathcal{L})$ *be a membrane net, let* $\prec$ *be a relation on* $\mathcal{L}(T \cup S)$ *satisfying definition 2, and let* $\equiv$ *be an object relation. Then* $\mathcal{K}^{\equiv}_{\prec}(N)$ *is a membrane system.*

Another simple observation is that the net associated to $\mathcal{K}^{\equiv}_{\prec}(N)$ is indeed N.

Proposition 3. *Let* $N_{\prec} = ((S, T, F, m_0, \mathfrak{S}), \mathcal{L})$ *be a membrane net, let* $\prec$ *be a relation on* $\mathcal{L}(T \cup S)$ *satisfying definition 2, and let* $\equiv$ *be an object relation. Then* $\mathcal{F}(\mathcal{K}^{\equiv}_{\prec}(N)) \cong N$*.*

Proof: The only point to check is the set of places. Consider $O = S/_{\equiv}$ (the objects in $\mathcal{K}^{\equiv}_{\prec}(N)$). To these objects correspond the places $S' = O \times \{1, \ldots, \#_{MS}(\sigma_{\prec}(N))\}$ representing a set (trivially) isomorphic to the subset of non isolated places of N. □

The vice versa does not hold, namely $\mathcal{K}^{\equiv}_{\prec}(\mathcal{F}(\Pi))$ is in general *syntactically* different from Π, because when associating a net to a membrane system we lose two relevant pieces of information: one on the objects (to each object and each membrane index a place is associated, going back this information is lost, and the object relation is arbitrary), and the membrane structure can be different, as the example of section 1 shows (again the choice of the $\prec$ relation is arbitrary).

Analogously to what we have done for membrane systems, we define the two mappings, ξ and η, the former relating the markings of a membrane net to the configurations of the associated membrane system, and the latter giving the multi-rule vector associated to a multisets of transitions. These mappings are used to relate steps in nets to evolution steps in membrane systems.

Definition 9. *Let* $N_{\prec} = ((S, T, F, m_0, \mathfrak{S}), \mathcal{L})$ *be a membrane net, let* $\prec$ *be a relation on* $\mathcal{L}(T \cup S)$ *satisfying definition 2, let* $\equiv$ *be an object relation, and* $\mathcal{K}^{\equiv}_{\prec}(N)$ *be the associate membrane system. If* m *is a marking of* $N_{\prec}$*, then* $\xi(m)$ *is the configuration defined by* $\xi(m) = (w_1, \ldots, w_n)$*, where* $w_i(a) = m(s)$ *if* $\mathcal{L}(s) = i$ *and* $a = [s]_{\equiv}$*, or* $w_i(a) = 0$ *if* $\forall s \in S.\ \mathcal{L}(s) = i$ *then* $a \neq [s]_{\equiv}$*, for all* $i \in \mathcal{L}(T \cup S)$*.*

Definition 10. *Let* $N_{\prec} = ((S, T, F, m_0, \mathfrak{S}), \mathcal{L})$ *be a membrane net, let* $\prec$ *be a relation on* $\mathcal{L}(T \cup S)$ *satisfying definition 2, let* $\equiv$ *be an object relation, and* $\mathcal{K}^{\equiv}_{\prec}(N)$ *be the associate membrane system. If* $m\,[U\rangle\, m'$ *is a step of* $N_{\prec}$*, then* $\eta(U)$ *is the step defined by* $\eta(U)(r^t_i) = U(t)$ *for all* $r^t_i \in R_i$ *and all* i*.*

We are now ready to present the main result of this section.

Theorem 3. *Let* $N_{\prec} = ((S, T, F, m_0, \mathfrak{S}), \mathcal{L})$ *be a membrane net, let* $\prec$ *be a relation on* $\mathcal{L}(T \cup S)$ *satisfying definition 2, let* $\equiv$ *be an object relation, and* $\mathcal{K}^{\equiv}_{\prec}(N)$ *be the associate membrane system.* $m\,[U\rangle_{fr}\, m''$ *iff* $\xi(m) \xrightarrow{\eta(U)}_{fr} \xi(m')$ *with* $fr \in \{step, max, k-b, \mathcal{S}, lmax, l-kb, l-\mathcal{S}, lcomp\}$*.*

Proof: We first show that the effects of executing a step in both models correspond, and then we show that also the enabling part is satisfied. As in the first instance, we reason by assuming that whenever a step U is enabled at a marking m, the corresponding rules $\eta(U)$ are enabled at $\xi(m)$.

Assume $m\,[U\rangle\,m'$ in $N_{\prec}$. Then $\xi(m) \overset{\eta(U)}{\Longrightarrow}_{fr} \xi(m')$. $m' = m \ominus {}^{\bullet}U \oplus U^{\bullet}$, and clearly $\xi(m') = \xi(m)\hat{\ominus}\xi({}^{\bullet}U)\hat{\oplus}\xi(U^{\bullet})$. It remains to show that $\xi(m) \overset{\eta(U)}{\Longrightarrow}_{fr} \xi(m')$; however, this is a direct consequence of how the rules of the membrane system $\mathcal{K}^{\equiv}_{\prec}(N)$ are associated to the transitions of the net $N_{\prec}$. Assume now that $\xi(m) \overset{\eta(U)}{\Longrightarrow}_{fr} C'$, and show that $m\,[U\rangle\,m'$ and $\xi(m') = C'$. Consider $\xi(m) = (w_1, \ldots, w_n)$. We recall that w'_i is obtained by C_i as follows: $w'_i = w_i \ominus \mathit{LHS}(\vec{R})_i \oplus \mathit{RHS}(\vec{R})_i$. Take a transition $t \in \mathit{dom}(U)$ of the net $N_{\prec}$ such that $\mathcal{L}(t) = i$. We have that $\mathit{lhs}(\eta(t))$ is $\xi({}^{\bullet}t)$, and $\mathit{rhs}(\eta(t))$ is $\xi(t^{\bullet})$ (with non empty entries in the neighborhood of i). It is straightforward to conclude that m' is reached from m by executing U such that $\xi(m') = C'$. Hence, assuming that the step U is enabled at m, it is obvious that $\eta(U)$ is enabled at $\xi(m)$, and the effects correspond.

It remain to show that whenever $m\,[U\rangle_{fr}$ then also $\eta(U)$ is *fr*-enabled at $\xi(m)$, and vice versa. This can be easily proved by observing that if a multiset U is *fr*-enabled at m and $\eta(U)$ is not at $\xi(m)$, then it can be shown that indeed $\neg m\,[U\rangle_{fr}$. This ends the proof. □

A consequence of the theorem 3 is the following result.

Theorem 4. *Let* $N_{\prec} = ((S, T, F, m_0, \mathfrak{S}), \mathcal{L})$ *be membrane net and* $\mathcal{K}^{\equiv}_{\prec}(N)$ *be the associate membrane system. Then* $\mathfrak{F}^{\mathcal{S}}_{fr}(\mathcal{F}(\Pi)) = \mathcal{P}s_{fr}(\Pi)$ *and* $\mathfrak{F}^{\mathcal{S}}_{fr}(\mathcal{F}(\Pi)) = \mathcal{N}_{fr}(\Pi)$*, with* $fr \in \{\mathit{step}, \mathit{max}, k-b, \mathcal{S}, \mathit{lmax}, l-kb, l-\mathcal{S}, \mathit{lcomp}\}$*.*

5. Catalytic and Communicating Membrane Systems

A research topic in membrane computing is to find P systems which are more realistic from a biological point of view, and one target in this respect is to relax the condition of using the rules in a maximally parallel way. *Minimal* parallelism was introduced in [7], describing another way of applying the rules: if at least a rule from a set of rules associated with a membrane (or a region) *can* be used, then at least one rule from that membrane (or region) *must* be used, without any other restriction (e.g., more rules can be used, but we do not care how many). Even if it might look weak, this minimal parallelism still leads to Turing completeness for certain classes of membrane systems which will be presented later in this section. The *minimal parallelism* stems out from the consideration that this mode of using the rules ensures that all compartments (or regions) of the system evolve in parallel by using at least one rule, whenever such a rule is applicable. Another slightly different version of *minimal* parallelism involves the use of specific rules, as in the case of catalytic P systems.

Therefore we are now interested in two classes of P systems, namely catalytic and symport/antiport. These classes are characterized by the kind of evolution rules they allow. We first revise the class of catalytic P systems.

Catalysts. A catalytic P system consists of a hierarchical membrane structure with uniquely labeled membranes, the whole structure being embedded in a single skin membrane; each membrane contains a multiset of abstract objects (from a finite alphabet) which is divided into two parts: the set $\mathcal{C}$ of catalysts, and the set of non-catalytic objects. Each membrane is associated with an initial multiset of objects and a fixed set of evolution rules of two possible types: catalytic rules $ca \to cv$, and non-cooperative rules $a \to v$, where c is a *catalyst object*, a is a non-catalyst object, and v is a (possibly empty) multiset of objects. A catalytic P system is called *purely* catalytic if it contains only catalytic rules. The rules are (usually) applied in the maximally parallel mode: at each computational step and in each membrane, the selected multiset of applicable rules must be maximal, i.e., unused objects are not sufficient to enable an additional rule.

Considering the class of catalytic P systems, Turing completeness can be achieved by relaxing the maximal parallelism requirement with the weaker one stating that for each catalyst c at least one catalytic rule $r = ca \to cv$ is used, if possible. Thus the question on *minimality* can be posed as follows: "how many catalysts are needed to obtain Turing completeness?". With RE we denote the class of recursively enumerable sets of numbers and with $PsRE$ the class of recursively enumerable sets of Parikh vectors.

A catalytic P system with a single membrane and only two catalysts has the power of a Turing machine, *i.e.*, it can generate all the computable enumerable sets of (vectors of) natural numbers [15]. With $([p-]cat, n)$ we denote the class of flat (purely) catalytic P system with at least n catalysts. Assuming that the computations obey to the *minimal* parallel strategy (*i.e.*, at least one rule involving each catalyst is used, if possible), the following theorems present the results used in Section 7.

To state precisely that catalytic P systems are Turing equivalent, we need to introduce some notation. By $NO_{min}(cat, n)$ we denote the class of sets $\{\mathfrak{G}^{\#1}(\Pi) \mid \Pi$ is a catalytic P system with just one membrane, $\mathcal{C}$ is the set of catalysts and $|\mathcal{C}| \geq\ n\}$, and analogously by $PsO_{min}(cat, n)$ the classes of sets $\{\mathfrak{G}^{1}(\Pi) \mid \Pi$ is a catalytic P system with just one membrane, $\mathcal{C}$ is the set of catalysts and $|\mathcal{C}| \geq\ n\}$.

Theorem 5. *$PsO_{min}(cat, n) = PsRE$ and $NO_{min}(cat, n) = RE$, for $n \geq\ 2$.*

Three catalysts are needed in the case of purely catalytic systems. $NO_{min}(p - cat, n)$ and $PsO_{min}(p - cat, n)$ are the classes of sets $\{\mathfrak{G}^{\#1}(\Pi) \mid \Pi$ is a purely catalytic P system with just one membrane, $\mathcal{CC}$ is the set of catalysts and $|\mathcal{C}| \geq\ n\}$ and $\{\mathfrak{G}^{1}(\Pi) \mid \Pi$ is a purely catalytic P system with just one membrane, $\mathcal{C}$ is the set of catalysts and $|\mathcal{C}| \geq\ n\}$, respectively.

Theorem 6. *$PsO_{min}(p-cat, n) = PsRE$ and $NO_{min}(p-cat, n) = RE$, $n \geq\ 3$.*

We end this part devoted to catalytic P system by reminding that chapter 4 of the handbook provides a good survey of the computability power of such P systems [27]. The above results are presented in that chapter, together with other interesting results.

Before introducing Symport/Antiport P systems, we want to stress that catalytic P systems use a set of catalysts, and it is enough to consider just one membrane and at least 2 catalysts to achieve Turing completeness. Hence just a *local* check has to be performed.

Symport/Antiport P Systems. Symport/antiport P systems are P systems where the existence of more than one membrane is required, as the rules work across the membranes [25].

Symport rules are rules of the following form: either $lhs(r) \subseteq O \times \{out\}$ and $rhs(r) \subseteq O \times \{here\}$, and furthermore $lhs(r)|_{out} = rhs(r)|_{here}$, or $lhs(r) \subseteq O \times \{here\}$ and $rhs(r) \subseteq O \times \{out\}$ and $lhs(r)|_{here} = rhs(r)|_{out}$. Thus the effect of the application of one of these rules is that either a multiset in the father membrane is *imported* in the membrane the rule belongs to, or a multiset in the membrane is sent to the father membrane. *Antiport rules* move objects across the boundaries (membranes) in both directions: $lhs(r), rhs(r) \subseteq O \times \{out, here\}$, $lhs(r)|_{here} = rhs(r)|_{out}$ and $lhs(r)|_{out} = rhs(r)|_{here}$. The *weight* of a rule r is given by $max\{\#(lhs(r), \#(rhs(r)\}$.

With $P_n(sym_p, anti_q)$ we denote the class of symport/antiport P systems $\Pi = (O, \mu, w_1^0, \ldots, w_n^0, R_1, \ldots, R_n, i_o)$, where the symport rules have weight at most p and antiport rules weight at most q. With $NO_{min}(P_n(sym_p, anti_q))$ we denote the class of sets $\{X \mid$ there exists a symport/antiport P system $\Pi \in P_n(sym_p, anti_q)$ with at most n membranes such that $X = \mathfrak{G}^{\#i_o}(\Pi)\}$.

Minimal parallelism in symport/antiport P systems requires that at least one rule is applied in each membrane, provided that this is possible: it may happen that, depending on the choice of which rules to apply, some membranes do not have any *enabled* rule. The following theorem is presented in [7].

Theorem 7. *$NO_{min}(P_n(sym_p, anti_q)) = RE$ for all $n \geq 3, p \geq 2$ and $q \geq 2$.*

We end this part devoted to symport/antiport P systems by reminding that chapter 5 of the handbook provides a good survey of the computability power of such P systems [27].

6. Catalytic and Communicating Petri Nets

In this section we introduce two new classes of Petri nets corresponding to catalytic and communicating membrane systems.

Catalytic Nets. Catalytic nets are Petri nets where the firing of transitions is *controlled* by tokens in suitable places (that maintain the same number of tokens during the whole execution).

Definition 11. *Let $N = (S, T, F, m_0, \mathfrak{S})$ be a Petri net. N is* catalytic *iff the set of places S is partitioned into two sets $\mathcal{C}$ and $\mathcal{V}$ such that*

1. *the subnet $N@\mathcal{V} = (\mathcal{V}, T@\mathcal{V}, F@\mathcal{V}, m_0@\mathcal{V}, \mathcal{V} \cap \mathfrak{S})$ is an input state machine, and*
2. *the subnet of $N@\mathcal{C} = (\mathcal{C}, T@\mathcal{C}, F@\mathcal{C}, m_0@\mathcal{C}, \emptyset)$ is a state machine, i.e., $\forall t \in T@\mathcal{C}.\ {}^\bullet t = t^\bullet$ and $\#({}^\bullet t) = 1$.*

The places in $\mathcal{C}$ will be called catalytic *places.*

A net $N = (S, T, F, m_0, \mathfrak{S})$ is said purely catalytic *iff $T = T@\mathcal{C}$.*

Places in a catalytic net are partitioned into two subsets: the catalytic places ($\mathcal{C}$) and the non catalytic ones ($S \setminus \mathcal{C}$). Condition 1 states that each transition may consume tokens from only one non catalytic place, whereas condition 2 says that each transition may use at most one token from a catalytic place, and the used token is again made available for further use. Since the number of tokens in catalytic places remains constant, it is meaningless to consider them as output places.

Catalytic Petri nets $N = (S, T, F, m_0, \mathfrak{S})$ are abbreviated as CPN; when we explicitly indicate the set of catalytic places, a catalytic Petri net is presented formally by $N = (S, T, F, m_0, \mathcal{C}, \mathfrak{S})$. The intuition behind catalytic Petri nets is the following: a transition t which uses a catalyst fires only if there is a token in the catalytic places associated to the transition. Tokens in catalytic places may be consumed/produced by transitions, but the transition using these tokens as catalysts should ideally leave the token in the place.

It is worth noting the difference between catalytic nets and Petri nets with read arcs of Montanari and Rossi [24]: in the latter nets, if two transitions test for the presence of a token in a place (with read arcs) it is enough to have a token in that place, whereas in catalytic nets if two transitions use the *same* catalytic place, this must contain enough tokens.

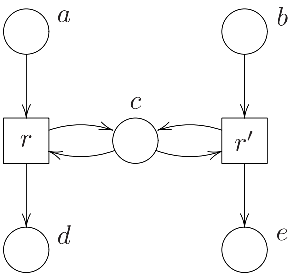
We recall the notion of enabling relevant for this class (which is now dependent on catalysts). Let $N = (S, T, F, m_0, \mathcal{C}, \mathfrak{S})$ be a CPN. The step U is catalytically enabled at the marking m iff it is enabled at m and $\forall c \in \mathcal{C}$ either there is a transition $t \in dom(U)$ such that ${}^{\bullet}t(c) \neq 0$, or $\forall t \in dom(c^{\bullet})$ it holds that $\neg m\,[t\rangle$. In other words, a step is a catalytically enabled whenever for each catalytic place in $\mathcal{C}$, either all the transitions using tokens from this catalytic place are not enabled, or there is at least one transition using a token from a catalytic place. We write $m\,[U\rangle_{\mathcal{C}}$ to denote that U is catalytically enabled, and we denote with $m\,[U\rangle_{\mathcal{C}}\, m'$ the firing of a catalytically enabled step; U is called a catalytic step. A catalytic firing sequence is a step firing sequence where each step is a catalytic one. The set of reachable markings is defined accordingly, and denoted by $\mathfrak{M}_{\mathcal{C}}(N)$. The sets $\mathfrak{F}_{\mathcal{C}}(N)$, $\mathfrak{F}_{\mathcal{C}}^{\hat{S}}(N)$ and $\mathfrak{F}_{\mathcal{C}}^{\#\hat{S}}(N)$ are defined in a similar way.

Catalytic Located Nets. We now introduce a class of nets containing both localities and catalysts.

Definition 12. *Let* $N = ((S, T, F, m_0, \mathfrak{S}), \mathcal{L})$ *be a PNL. We say that* N *is* lcatalytic *iff* $(S, T, F, m_0, , \mathfrak{S})$ *is a catalytic net,* N *is correctly located and* $\forall t \in T.\ \forall s \in dom({}^{\bullet}t)$ *it holds that* $\mathcal{L}(s) = \mathcal{L}(t)$*.*

Lcatalytic Petri nets $N = ((S, T, F, m_0, \mathfrak{S}), \mathcal{L})$ are abbreviated by CPNL; when we explicitly indicate the set of catalytic places, an lcatalytic net is presented formally as $N = ((S, T, F, m, \mathcal{C}, \mathfrak{S}), \mathcal{L})$.

Example. The following net is a catalytic one, the catalytic place being c.



Observe that tokens in the places c are consumed and produced, hence their number remains constant. Here we have not indicated the locality mapping, which should be the same for all the places and transitions.

The various notions of firing we have presented so far can obviously be applied to this new class, providing the sets of reachable markings and final markings under these firing rules.

Communicating Petri Nets. The second class we consider is defined on Petri nets with localities (which play a major rôle here).

Definition 13. *Let $N = ((S, T, F, m_0, \mathfrak{S}), \mathcal{L})$ be a correctly located PNL, $\prec$ be a partial order on $\mathcal{L}(T \cup S)$ satisfying definition 2, and $\equiv$ an object relation on S. N is a* communicating *Petri net iff $\forall t \in T$, $\forall s \in S$,*

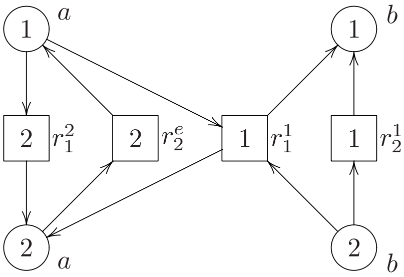
1. *if $F(s,t) > 0$ or $F(t,s) > 0$, then one of the following holds: $\mathcal{L}(s)\hat{\prec}\mathcal{L}(t)$, or $\mathcal{L}(t)\hat{\prec}\mathcal{L}(s)$, or $\mathcal{L}(t) = \mathcal{L}(s)$;*
2. *$\forall t \in T$, $\forall s \in S$. if $F(t,s) > 0$ and $\mathcal{L}(t) = \mathcal{L}(s)$, then $\exists s' \in S$. such that $F(s',t) = F(t,s)$, $\mathcal{L}(s')\hat{\prec}\mathcal{L}(t)$ and $s \equiv s'$;*
3. *$\forall t \in T$, $\forall s \in S$. if $F(t,s) > 0$ and $\mathcal{L}(t)\hat{\prec}\mathcal{L}(s)$, then $\exists s' \in S$. such that $F(s',t) = F(t,s)$, $\mathcal{L}(s') = \mathcal{L}(t)$ and $s \equiv s'$;*
4. *$\forall t \in T$, $\forall s \in S$. if $F(s,t) > 0$ and $\mathcal{L}(t) = \mathcal{L}(s)$, then $\exists s' \in S$. such that $F(t,s') = F(s,t)$, $\mathcal{L}(s')\hat{\prec}\mathcal{L}(t)$ and $s \equiv s'$, and*
5. *$\forall t \in T$, $\forall s \in S$. if $F(s,t) > 0$ and $\mathcal{L}(s)\hat{\prec}\mathcal{L}(t)$, then $\exists s' \in S$. such that $F(t,s') = F(s,t)$, $\mathcal{L}(s') = \mathcal{L}(t)$ and $s \equiv s'$.*

Communicating Petri nets are presented as $N = (((S, T, F, m_0, \mathfrak{S}), \mathcal{L}), \equiv)$, and the class of communicating Petri nets is denoted by $CommPNL$.

Condition 1 guarantees that the transitions use tokens from places in neighboring localities, whereas the others guarantee that tokens are always moved across the boundaries with the proper multiplicity.

In this definition the notion of locality is crucial: without its labeling, it is impossible to establish when a boundary is crossed or not.

Example. Consider the net



The numbers inside the places and transitions indicate the locations (here 1 and 2). Each transition moves the same number of tokens (in this case just one) from a place in a location with a place to the neighboring location. Clearly here the relation $\prec$ could be either $1 \prec 2$ or $2 \prec 1$; the only feasible equivalence is $(a, 1) \equiv (a, 2)$ and $(b, 1) \equiv (b, 2)$.

As we did before for catalytic (located) net, we specialize the firing rules to the new class. In particular, we require that for each location (index), if a transition labelled with this index is enabled, then it must happen. Formally, let U be a step and m be a marking such that $m\,[U\rangle$; assuming that there exists a transition t such that $\mathcal{L}(t) \notin \mathcal{L}(dom(U))$, then $\neg m\,[U \oplus \{t\}\rangle$. We denote by *lmin* this kind of enabling. We write $m\,[U\rangle_{lmin}$ to denote that U is *lmin*-enabled, and we denote by $m\,[U\rangle_{\mathcal{C}}\, m'$ the firing of a *lmin*-enabled; U is called a *lmin* step. A location minimal firing sequence is a step firing sequence where each step is a *lmin* one. The set of reachable markings is defined accordingly, and denoted by $\mathfrak{M}_{lmin}(N)$. The sets $\mathfrak{F}_{lmin}(N)$, $\mathfrak{F}^{\hat{S}}_{lmin}(N)$ and $\mathfrak{F}^{\#\hat{S}}_{lmin}(N)$ are defined in a similar way.

7. Catalytic and Communicating Petri Nets are Turing Complete

In this section we present the main results of the paper, namely that catalytic Petri Nets with at least two catalysts and communicating Petri nets using *specific* firing strategies, have the same computational power of Turing machines.

For catalytic nets we provide a simplified version of the result proved in [15], considering *flattened* membrane system. We start formalizing the notion of *flat* P system and of *flat* catalytic P system. It is shown in [1] that the flattened version of a transition P system (with promoters and inhibitors) has the same computational power as a non flattened one; thus we can use the flattened one without loss of generality.

Definition 14. *A* (flat) P system *is the 4-tuple* $\Pi_f = (O, w^0, R, O')$ *where*

- O *is a finite set of* objects*, and* $O' \subseteq O$ *are the* final *objects,*
- $w^0 \in \partial O$ *is a finite multiset of objects, called the* initial *configuration, and*
- R *is a finite set of rules of the form* $r = u \to v$*, with* $u, v \in \partial O$ *and* $u \neq \mathbf{0}$.

A configuration *of a membrane system is any finite multiset of objects.*

A flat membrane system Π_f *is called* catalytic *iff there is a designated subset* $O_C \subset O$ *of* catalysts *and the rules have the following form: either* $r = a \to v$ *with* $a \in O \setminus O_C$ *and* $v \in \partial(O \setminus O_C)$ *or* $r = ca \to cv$ *with* $a \in O \setminus O_C$, $v \in \partial(O \setminus O_C)$ *and* $c \in O_C$*. If all the rules are of the form* $r = ca \to cv$ *we say that the catalytic P system is* purely *catalytic. We denote catalytic P systems by* $C\Pi$*, and purely catalytic ones by* $CP\Pi$.

According to [1], any property proved for flat membrane systems can essentially be proved also for non-flat ones.

Catalytic P Systems and Catalytic Nets. The results of Section 4 can be extended from catalytic P systems to catalytic nets. In particular, we can state the following two results.

Proposition 4. *Let* $\Pi = (O, \mathcal{C}, \mu, w_1^0, \ldots, w_n^0, R_1, \ldots, R_n, i_o)$ *be a catalytic P system, and* $\mathcal{F}(\Pi)$ *be its associated structure. Then* $\mathcal{F}(\Pi)$ *is a lcatalytic net.*

Proof: Consider the catalytic P system $\Pi = (O, \mu, w_1^0, \ldots, w_n^0, R_1, \ldots, R_n, i_o)$, with $\mathcal{C}$ being the catalysts. Each rule $r \in R_i$ either is such that

- $\pi(lhs(r))|_{here} = ca$, with $c \in \mathcal{C}$ and $a \in O \setminus \mathcal{C}$, and $\pi(lhs(r))|_{out} = \mathbf{0} = \pi(lhs(r))|_{in_j}$, and
- $\pi(rhs(r))|_{here} = cv_{here}$, with v_{here} be a multiset (possibly empty) over $O \setminus \mathcal{C}$, and $\pi(rhs(r))|_{out} = v_{out}$, $\pi(rhs(r))|_{in_j} = v_{in_j}$, with v_{out}, v_{in_j} be multisets (possibly empty) over $O \setminus \mathcal{C}$,

or is such that

- $\pi(lhs(r))|_{here} = a$, with $a \in O \setminus \mathcal{C}$, and $\pi(lhs(r))|_{out} = \mathbf{0} = \pi(lhs(r))|_{in_j}$, and
- $\pi(rhs(r))|_{here} = v_{here}$, with v_{here} be a multiset (possibly empty) over $O \setminus \mathcal{C}$, and $\pi(rhs(r))|_{out} = v_{out}$, $\pi(rhs(r))|_{in_j} = v_{in_j}$, with v_{out}, v_{in_j} be multisets (possibly empty) over $O \setminus \mathcal{C}$.

Consider now $\mathcal{F}(\Pi)$. The catalytic places are those of the form $\mathcal{C} \times \{1, \ldots, \#_{MS}(\mu)\}$. Then an easy inspection shows that $\mathcal{F}(\Pi)@\mathcal{C} \times \{1, \ldots, \#_{MS}(\mu)\}$ is a state machine (only rules using catalysts have to be considered), and the net $\mathcal{F}(\Pi)@(O \setminus \mathcal{C}) \times \{1, \ldots, \#_{MS}(\mu)\}$ is clearly an input state machine. □

Proposition 5. *Let* $N = ((S, T, F, m_0, \mathcal{C}, \mathfrak{S}), \mathcal{L})$ *be a lcatalytic Petri net, and let* $\prec$ *be a relation on* $\mathcal{L}(T \cup S)$ *satisfying definition 2. Then* $\mathcal{K}_\prec(N)$ *is a catalytic P system.*

Proof: We first observe that all the transitions in a catalytic Petri net consume tokens from places in the same location, and furthermore the maximal number of input places can be 2. Observing that N is by assumption correctly located, the conclusion follows easily. □

The following two results are essentially corollaries of the previous propositions, and tell us that we can use flat P systems.

Corollary 1. *Let* $\Pi = (O, \mathcal{C}, w^0 O')$ *be a flat catalytic P system, and* $\mathcal{F}(\Pi)$ *be its associated structure. Then* $\mathcal{F}(\Pi)$ *is a catalytic net.*

Corollary 2. *Let* $N = (S, T, F, m_0, \mathcal{C}, \mathfrak{S})$ *be a catalytic Petri net. Then* $\mathcal{K}^{id}(N)$ *is a flat catalytic P system.*

Just note that the identity equivalence relation on places is enough to guarantee flatness.

Turing Completeness under the Catalytic Firing Strategy. We will denote by $CPN(n)$ the class of catalytic Petri nets with at most n catalytic places, whereas the class of purely catalytic Petri nets with at most n catalytic places will be denoted by $PCPN(n)$. Then Theorems 5 and 6 can be specialized as follows:

Theorem 8.

$$\{\mathfrak{F}_{\mathcal{C}}^{\#\mathfrak{S}}(N) \mid N \in CPN(n)\} = RE \quad \textit{and}$$
$$\{\mathfrak{F}_{\mathcal{C}}^{\mathfrak{S}}(N) \mid N \in CPN(n)\} = PsRE, \textit{ for } n \geq 2.$$

Theorem 9.

$$\{\mathfrak{F}_{\mathcal{C}}^{\#\mathfrak{S}}(N) \mid N \in PCPN(n)\} = RE \quad \textit{and}$$
$$\{\mathfrak{F}_{\mathcal{C}}^{\mathfrak{S}}(N) \mid N \in PCPN(n)\} = PsRE \textit{ for } n \geq 3.$$

Communicating P System and Communicating Nets. Also in this case we can use the expressivity result for membrane system in a suitable class of Petri nets. In the case of communicating Petri nets, the equivalence relation on objects is already a part of the definition. Despite this added information, it is still not possible to prove $\Pi = \mathcal{F}(\mathcal{K}_{\prec}^{\equiv}(N))$, where $\equiv$ is the natural equivalence on places induced by objects. This is because the tree structure can still be different.

Proposition 6. *Let* $\Pi = (O, \mu, w_1^0, \ldots, w_n^0, R_1, \ldots, R_n, i_o)$ *be a communicating P system, and* $\mathcal{F}(\Pi)$ *be its associated structure. Then* $(\mathcal{F}(\Pi), \equiv)$ *is a communicating Petri net.*

Proof: Just observe that the symport/antiport rules are perfectly captured by conditions 2-5 of definition 13, and the object relation $\equiv$ is defined as follows: $(a, i) \equiv (a, j)$ for all $a \in O$ and $i, j \in \{1, \ldots, \#_{MS}(\mu)\}$. □

Proposition 7. *Let* $N = (((S, T, F, m_0, \mathfrak{S}), \mathcal{L}), \equiv)$ *be a communicating Petri net, and let* $\prec$ *be a relation on* $\mathcal{L}(T \cup S)$ *satisfying definition 2. Let* $\mathcal{K}_{\prec}^{\equiv}(N)$ *be the associated structure. Then* $\mathcal{K}_{\prec}^{\equiv}(N)$ *is a communicating P system.*

Proof: A straightforward inspection of the conditions on transitions of a communicating Petri net guarantees that the evolutions rules are either symport or antiport. □

Turing Completeness under the Minimal Firing Strategy. The class of communicating Petri nets with n localities, where the symport transitions have at most p input places and the antiport transitions have at most q input places, is denoted by $ComPN(n, p, q)$.

Theorem 7 can be then specialized as follows:

Theorem 10.

$\{\mathfrak{F}_{\mathcal{C}}^{\#\mathfrak{S}}(N) \mid N \in ComPN(n, p, q)\} = RE$*, for* $n \geq 3$*,* $p \geq 2$ *and* $q \geq 2$*.*

8. Conclusion

In this paper we have introduced and studied two new classes of Petri nets that are Turing complete. We have investigated their computability power by relating them to appropriate membrane systems. The outcome is that Petri nets can be more expressive by imposing conditions on the firing strategies, without neglecting the distribution in space that is one of the key feature of Petri nets. In fact, in [10], the principle of *locality* in Petri nets is formulated as follows

- the conditions for enabling a transition, in a certain mode if applicable, only depend on local states of (some) places in its immediate vicinity, and

- the occurrence of an enabled transition only changes the local state of (some) places in its immediate vicinity.

This principle is certainly true in our case, provided that the part concerning the enabling of a transition is somehow reformulated in this way:

- the conditions for enabling a transition, in a certain mode if applicable, only depend on local states of (some) places and *transitions* in its immediate vicinity.

The kind of locality mapping we use is rather different from the one which is normally considered, as we promote places as *subjects* to be located *a priori* and not, as it is usually done, *a posteriori*, *i.e.*, the location is somehow induced by the transitions using these places.

The problem of finding algorithms to equip, when possible, Petri nets with a suitable locality mapping is an open problem. Clearly one can assign locations to places and transitions, guess a partial order on locations, and then check if the resulting net is correctly located net; this direction was not investigated yet, as we have concentrated on the expressivity results.

We consider that our effort of emphasizing the existence of new classes of Petri nets which are more expressive is a promising one.

By establishing a really tight connection among the membrane systems and Petri nets, we believe that we can gain in both directions. For instance, being able to introduce for membrane systems the whole verification apparatus which has been developed for Petri nets; on the other side, by looking differently at Petri nets languages or studying the expressiveness of new classes of nets.

Acknowledgements. We would like to thank the anonymous reviewers for their helpful remarks and suggestions. The work of Gabriel Ciobanu was supported by a grant of the Romanian National Authority for Scientific Research, CNCS-UEFISCDI, project number PN-II-ID-PCE-2011-3-0919.

References

[1] Agrigoroaiei, O., Ciobanu, G., 2010. Flattening the transition P systems with dissolution. In: Gheorghe, M., Hinze, T., Păun, G., Rozenberg, G., Salomaa, A. (Eds.), 11th Int. Conf. on Membrane Computing. Vol. 6501 of Lecture Notes in Computer Science. Springer, pp. 53–64.

[2] Alberts, B., Johnson, A., Walter, P., Lewis, J., 2002. Molecular Biology of the Cell. Garland Publishing, New York.

[3] Bruni, R., Montanari, U., 1997. Zero-safe nets, or transition synchronization made simple. Electronic Notes Theoretical Computer Science 7, 55–74.

[4] Burkhard, H.-D., 1980. On priorities of parallelism. In: Salwicki, A. (Ed.), Proceedings of Logics of Programs and Their Applications. Vol. 148 of Lecture Notes in Computer Science. Springer, pp. 86–97.

[5] Burkhard, H.-D., 1981. Ordered firing in Petri nets. Elektronische Informationsverarbeitung und Kybernetik 17 (2/3), 71–86.

[6] Ciardo, G., 1994. Petri nets with marking-dependent arc cardinality: Properties and analysis. In: Valette, R. (Ed.), Proceeding of the 15th International Conference on Application and Theory of Petri Nets. Vol. 815 of Lecture Notes in Computer Science. Springer, pp. 179–198.

[7] Ciobanu, G., Pan, L., Păun, G., Pérez-Jiménez, M. J., 2007. P systems with minimal parallelism. Theoretical Computer Science 378 (1), 117–130.

[8] Ciobanu, G., Pérez-Jiménez, M. J., Păun, G. (Eds.), 2006. Applications of Membrane Computing. Natural Computing Series. Springer.

[9] Ciobanu, G., Pinna, G. M., 2012. Catalytic Petri nets are Turing complete. In: Dediu, A. H., Martín-Vide, C. (Eds.), Proceedings of the 6th International Conference on Language and Automata Theory and Applications. Vol. 7183 of Lecture Notes in Computer Science. Springer, pp. 192–203.

[10] Desel, J., Juhás, G., 2001. ”What is a Petri net?”. In: Ehrig, H., Juhás, G., Padberg, J., Rozenberg, G. (Eds.), Unifying Petri Nets. Vol. 2128 of Lecture Notes in Computer Science. Springer, pp. 1–25.

[11] Dufourd, C., Finkel, A., Schnoebelen, P., 1998. Reset nets between decidability and undecidability. In: Larsen, K. G., Skyum, S., Winskel, G. (Eds.), Proceedings of the 25th International Colloquium on Automata, Languages and Programming. Vol. 1443 of Lecture Notes in Computer Science. Springer, pp. 103–115.

[12] Esparza, J., 1996. Decidability and complexity of Petri net problems - an introduction. In: Reisig, W., Rozenberg, G. (Eds.), Lectures on Petri Nets I: Basic Models - Advances in Petri Nets 1996. Vol. 1491 of Lecture Notes in Computer Science. Springer, pp. 374–428.

[13] Esparza, J., Nielsen, M., 1994. Decidability issues for Petri nets - a survey. Bulletin of the EATCS 52, 244–262.

[14] Finkel, A., Geeraerts, G., Raskin, J.-F., Begin, L. V., 2006. On the *ω*-language expressive power of extended Petri nets. Theoretical Computer Science 356 (3), 374–386.

[15] Freund, R., Kari, L., Oswald, M., Sosík, P., 2005. Computationally universal P systems without priorities: two catalysts are sufficient. Theoretical Computer Science 330 (2), 251–266.

[16] Geeraerts, G., 2007. On the expressive power of Petri nets with transfer arcs vs. Petri nets with reset arcs. Tech. Rep. 572, Université Libre de Bruxelles.

[17] Hack, M., 1975. Decidability Questions for Petri Nets. Outstanding Dissertations in the Computer Sciences. Garland Publishing, New York.

[18] Jantzen, M., Zetzsche, G., 2008. Labeled step sequences in Petri nets. In: van Hee, K. M., Valk, R. (Eds.), Proceedings of the 32rd International

Conference on Application and Theory of Petri Nets. Vol. 5062 of Lecture Notes in Computer Science. Springer, pp. 270–287.

[19] Kleijn, H. C. M., Koutny, M., 2002. Causality semantics of Petri nets with weighted inhibitor arcs. In: Brim, L., Jancar, P., Kretínský, M., Kucera, A. (Eds.), CONCUR 2002 - Concurrency Theory, 13th International Conference, Brno, Czech Republic, August 20-23, 2002, Proceedings. Vol. 2421 of Lecture Notes in Computer Science. Springer, pp. 531–546.

[20] Kleijn, J., Koutny, M., 2010. Petri nets with localities and testing. In: Lilius, J., Penczek, W. (Eds.), Proceedings of the 31st International Conference on Applications and Theory of Petri Nets. Vol. 6128 of Lecture Notes in Computer Science. Springer, pp. 19–38.

[21] Kleijn, J., Koutny, M., Rozenberg, G., 2005. Towards a Petri net semantics for membrane systems. In: Proceedings of the 6th International Workshop on Membrane Computing. Vol. 3850 of Lecture Notes in Computer Science. Springer, pp. 292–309.

[22] Kosaraju, S. R., 1982. Decidability of reachability in vector addition systems (preliminary version). In: Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing. ACM, pp. 267–281.

[23] Koutny, M., Pietkiewicz-Koutny, M., 2012. Synthesis problem for Petri nets with localities. Transactions on Petri Nets and Other Models of Concurrency 5, 160–180.

[24] Montanari, U., Rossi, F., 1995. Contextual nets. Acta Informatica 32 (6).

[25] Păun, A., Păun, G., 2002. The power of communication: P systems with Symport/Antiport. New Generation Computing 20 (3), 295–306.

[26] Păun, G., 2002. Membrane Computing. An Introduction. Natural Computing Series. Springer.

[27] Păun, G., Rozenberg, G., Salomaa, A., 2010. The Oxford Handbook of Membrane Computing. Oxford University Press.