2012 14th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing

Timed Catalytic Petri Nets

Bogdan Aman[1,2], Gabriel Ciobanu[1,2] and G. Michele Pinna[3]
[1]Romanian Academy, Institute of Computer Science, Iaşi, Romania
[2]A.I.Cuza University, 700506 Iaşi, Romania
[3]Dipartimento di Matematica e Informatica, Universitá di Cagliari, Italy
baman@iit.tuiasi.ro, gabriel@info.uaic.ro, gmpinna@unica.it

Abstract

We consider catalytic membrane systems ($CatMS$) and catalytic Petri nets ($CatPN$). We add discrete timers to these classes, and prove that the timed formalisms ($tCatMS$ and $tCatPN$) can be expressed by their untimed versions if finite timers are used. We establish formal links between these formalisms, and characterize some subclasses in which various properties are decidable and can be analyzed using CPN Tools, i.e., a software tool for editing, simulating, and analyzing colored Petri nets.

I. Introduction

Membrane systems [24] and Petri nets [15], [25] are formalisms that use explicit resources, parallelism and timing. Petri nets under the usual step firing strategy are not Turing complete [12], [19], but they become Turing complete under suitable firing strategies: *maximality* (all possible transitions are fired together), or *ordering* (a transition should fire as soon as enabled) [5]. Turing completeness is obtained also if special types of arc are considered: *inhibitor* arcs (model the absence of tokens in certain places [14]), or *reset* arcs (have the characteristic of emptying the preset [11]. Thus, Petri nets become Turing complete either when suitable extensions of the model are considered (the main features of Petri nets are retained, *e.g.*, the *locality* of firing) or *heavy* assumptions on the firing rule are made (the characteristics that made Petri nets an appealing and useful model for concurrent and distributed computations are lost).

Inspired by the connections established between membrane systems and Petri nets [18], we investigated the expressivity of Petri nets looking at suitable firing strategy and at the *structure* of the nets itself [10]. We proved that the number of catalytic places needed for Turing completeness is quite limited, namely just two. Thus the expressiveness of Petri nets is increased without introducing suitable arcs or requiring transitions fired upon a complete snapshot of the system.

In this paper we consider catalytic membrane systems/catalytic Petri nets and present a relation between them. We extend them with a discrete notion of time that models the delays between consumption and appearance of reactants. Membrane systems were extended with timing aspects [6], while Petri Nets have two main extensions with time: Time Petri Nets [22] (a transition can fire within a time interval) and Timed Petri Nets [27] (a transition fires as soon as possible). We establish links between the timed and untimed versions, as depicted in Figure I.

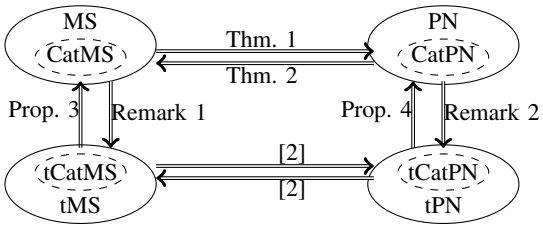

Figure 1. Correspondences

Even if the formalisms presented in this paper are Turing complete, the extensions with finite delays can be modeled in CPN Tools in order to study some quantitative properties. In particular we show that if the net can be suitably decomposed into components with at most one catalyst then various interesting properties can be checked locally and they hold in the whole net.

The paper is organized as follows: in the next two sections we review $CatMS$ and $CatPN$, that are related in Sections IV and V; then we show how to add time to these formalisms (Sections VI and VII). In section VIII we introduce a notion of decomposition for $CatPN$ and $tCatPN$ and we use this notion in section IX to give some

978-0-7695-4934-7/12 $26.00 © 2012 IEEE
DOI 10.1109/SYNASC.2012.22

properties that can be still proved using the CPN tool.

II. Catalytic Membrane Systems

Membrane systems (also called P systems) are distributed, parallel and nondeterministic systems, inspired by cell biology [24]. Several variants of membrane systems are inspired by different aspects of living cells (e.g., catalytic objects [24]). Various applications of membrane systems are presented in [9], while links between membrane systems and process calculi are presented in [7].

Let $\mathbb{N}$ be the set of positive integers, and O a finite alphabet of objects. A multiset over O is $u : O \to \mathbb{N}$. Given two multisets u, v over O, for any $a \in O$, we have $(u+v)(a) = u(a) + v(a)$ as the multiset union, and $(u - v)(a) = max\{0, u(a) - v(a)\}$ as the multiset difference. We write $u \subseteq u'$ if $u(a) \leq u'(a)$ for all $a \in O$, and $u \subset u'$ if $u \subseteq u'$ and $u \neq u'$. The cardinality of a multiset is $|u| = \sum_{a \in O} u(a)$. The scalar product of a number j with a multiset u is $(j \cdot u)(a) = j \cdot u(a)$. Sometimes a multiset u is written as $\sum_{a \in O} u(a) \cdot a$. The set O^* of all strings over O is a monoid with ϵ as its empty string. A multiset u over O is finite iff $dom(u) = \{a \in O \mid u(a) \neq 0\}$ is finite.

Since any property proven for flat membranes could be also proven for non-flat ones [1], we can use flattened catalytic membrane systems without losing generality.

We use catalytic membrane systems that consist of a skin membrane and a multiset of objects over a finite alphabet O. The object are divided into a set O_C of catalysts, and a set of non-catalytic objects. Each membrane is assigned an initial multiset of objects and a fixed set of evolution rules. The rules are applied in the minimal parallel mode: at least one rule involving each catalyst is used if possible, without any other restriction (e.g., more rules can be used, but we do not care how many) [8].

Definition 1. *A* (flat) catalytic membrane system $\Pi = (O, w^0, R, O_C, O')$ *is defined by*

- *O is a finite alphabet (its elements are called* objects*);*
- *$O_C \subseteq O$ is a finite set of* catalytic *objects;*
- *$O' \subseteq O$ is a finite set of* final *objects;*
- *w^0 is a finite multiset of objects over O, called the* initial *configuration;*
- *R is a finite set of rules having the following form: either $r = a \to v$ (non-cooperative rules) or $r = ca \to cv$ (catalytic rules), with $a \in O \backslash O_C$, $v \in \mathbb{N}^{O \backslash O_C}$ and $c \in O_C$.*

A configuration of a membrane system is any finite multiset of objects. If all the rules are of the form $r = ca \to cv$ we say that the catalytic P system is purely catalytic.

Given a rule $r = u \to v$, with $lhs(r)$ we denote the multiset u and with $rhs(r)$ the multiset v. The dynamic of a catalytic P system is formalized in the following way: given a minimal multiset of rules $\mathcal{R} \neq \emptyset$ and a configuration w, we have $w \overset{\mathcal{R}}{\Longrightarrow}_C w'$ iff $\sum_{r \in \mathcal{R}} \mathcal{R}(r) \cdot lhs(r) \subseteq w$ and $w' = w - (\sum_{r \in \mathcal{R}} \mathcal{R}(r) \cdot lhs(r)) + (\sum_{r \in \mathcal{R}} \mathcal{R}(r) \cdot rhs(r))$. The transition $w \overset{R}{\Longrightarrow}_C w'$ is called an *evolution step*. A configuration w is reachable from the initial one iff there exists a finite sequence of evolution steps $w_1 \overset{\mathcal{R}_1}{\Longrightarrow}_C w_2 \ldots w_n \overset{\mathcal{R}_n}{\Longrightarrow}_C w_{n+1}$ such that $w_1 = w^0$ and $w = w_{n+1}$. $\Rightarrow^*$ denotes a reflexive and transitive closure of the binary relation $\overset{\mathcal{R}}{\Longrightarrow}_C$.

A *computation* (sequence of evolution steps) terminates (halts) iff at the configuration w, reached by this computation, there is no possible evolution step. For an alphabet V, the *Parikh vector* is $\psi_V : V^* \to \mathbf{N}^n$ with $\psi_V(x) = (|x|_{a_1}, \ldots, |x|_{a_n})$, for all $x \in V^*$.

We have the following characterization of the reachability sets of deterministic purely catalytic P systems.

Proposition 1 ([13])**.** *The reachability set $\{\psi_V(w_1) \mid w_0 \Rightarrow^* w_1\}$ of a deterministic purely catalytic P system $\Pi = (O, w^0, R, O_C, O')$ is semilinear.*

As an immediate consequence of the previous result, we also infer that flat deterministic purely catalytic P systems are not computationally complete because their halting problem is decidable:

Proposition 2 ([13])**.** *The problem of determining whether the deterministic purely catalytic P system $\Pi = (O, w^0, R, O_C, O')$ halts is decidable.*

III. Catalytic Petri Nets

A Petri net is a tuple $\mathcal{N} = ((P, T, W, M_0), \mathcal{P})$ where P and T are finite disjoint sets of *places* and *transitions*, respectively, $W : (T \times P) \cup (P \times T) \to \mathbb{N}$ is a *weight function*, $M_0 : P \to \mathbb{N}$ is the *initial marking* and $\mathcal{P} \subseteq P$ is a set of *final* places. With $pre(x)$ (*preset*) and $post(x)$ (*postset*) we indicate the multisets $\{W(p, x) \mid p \in P\}$ and $\{W(x, p) \mid p \in P\}$. We assume that for each transition t, $dom(pre(t)) \neq 0$.

Catalytic nets were introduced in [10] to control the firing of suitable transitions locally by using special places (that maintain the same number of tokens during the whole execution). The intuition behind catalytic Petri nets is: a transition t, that uses a catalyst, fires only if there is a token in the catalytic places associated with the transition. Tokens in catalytic places may be consumed/produced by transitions, but the transitions that use these tokens as catalysts recreate the consumed tokens after firing.

Definition 2. *Let $\mathcal{N} = ((P, T, W, M_0), \mathcal{P})$ be a Petri net. $\mathcal{N}$ is* catalytic *iff the set of places P is partitioned into two disjoint sets $\mathcal{C}$ and $P \backslash \mathcal{C}$ such that $\forall t \in T$*

1) *if $dom(pre(t)) \subseteq (P \backslash \mathcal{C})$ then $|dom(pre(t)) \cap P| = 1$;*

2) *if* $dom(pre(t)) \subseteq \mathcal{C}$ *then* $|dom(pre(t)) \cap \mathcal{C}| = 1$ *and for all* $c \in dom(pre(t)) \cap \mathcal{C}$. $pre(t)(c) = post(t)(c)$.

A net $\mathcal{N}$ *is said to be* purely catalytic *iff* $\forall t \in T\ \exists p \in \mathcal{C}$ *such that either* $W(p.t) > 0$ *or* $W(t,p) > 0$.

Places in a catalytic net are: the catalytic places ($\mathcal{C}$) and the non catalytic ones ($P \backslash \mathcal{C}$). 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 made available again for further use.

Sometimes we explicitly indicate the set of catalytic places and their number, and so if $|\mathcal{C}| = k$ then a catalytic net is presented as $\mathcal{N} = ((P, T, W, M, \mathcal{C}, k), \mathcal{P})$.

It is worth to stress the difference between catalytic nets and Petri nets with read arcs [23]: in the latter case, 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 the former case if two transitions use the same catalytic place, this place must contain enough tokens for both transitions.

Next we discuss the dynamic of catalytic Petri nets. A multiset M over P is called a marking for the net. A multiset U of transitions is enabled at the marking M if $\forall p \in P$, $\sum_{t \in T} U(t) \cdot W(p,t) \leq M(p)$, and it is written as $M[U\rangle$. A multiset U of transitions is catalytically enabled at the marking M if and only if it is enabled, and $\forall p \in \mathcal{C}$ either there is a transition $t \in dom(U)$ such that $pre(t)(p) \neq 0$ or $\forall t \in dom(post(p))$ it holds that $\neg M[t\rangle$. In other words, a step is catalytically enabled whenever for each catalytic place in $\mathcal{C}$, either all the transitions using tokens from this catalyst 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 by $M[U\rangle_{\mathcal{C}} M'$ the firing of a catalytically enabled step, where U is a catalytic step. The marking M' is defined as $M'(p) = M(p) + \sum_{t \in T} U(t) \cdot (W(t,p) - W(p,t))$, for all $p \in P$. A catalytic firing sequence is defined as follows: M_0 is a catalytic firing sequence, and $M_0[U_1\rangle_{\mathcal{C}} M_1 \dots M_{n-1}[U_n\rangle_{\mathcal{C}} M_n$ is a step firing sequence. Given a net $\mathcal{N} = ((P, T, W, M_0, \mathcal{C}, k), \mathcal{P})$, a marking M is *reachable* if there is a catalytic firing sequence leading to it, namely $M_0[U_1\rangle_{\mathcal{C}} M_1 \dots M_{n-1}[U_n\rangle_{\mathcal{C}} M_n$, with $M = M_n$. The set of reachable markings of the net $\mathcal{N}$ is denoted by $\mathcal{M}_{\mathcal{C}}(\mathcal{N})$. Given a Petri net $\mathcal{N}$, it has been shown that the problem of deciding whether a given marking M is reachable from the initial one (i.e., if $M \in \mathcal{M}_{\mathcal{C}}(\mathcal{N})$) is decidable (see [12] for a survey).

The ordinary firing strategy of Petri nets (just one enabled transition is fired at each marking, regardless of how many are simultaneously enabled) is an instance of the firing strategy we have previously discussed: in this case the multiset U is such that $|(dom(U))| = 1$. We observe that, given a step $M[U\rangle_{\mathcal{C}} M'$, with $U = \{t_1, \dots, t_k\}$, it can always be written as $M[t_{i_1}\rangle_{\mathcal{C}} M_{i_1} \dots [t_{i_k}\rangle_{\mathcal{C}} M'$, with $i_j \in \{1, \dots, k\}$ for $j \in \{1, \dots, k\}$, and thus M' can be reached from M by different catalytic firing sequences.

Given a net $\mathcal{N} = ((P, T, W, M_0), \mathcal{P})$ and a marking M of $\mathcal{N}$, we say that M is a final marking iff $M \in \mathcal{M}_{\mathcal{C}}(\mathcal{N}))$ and for all $t \in T$ we have $\neg(M[t\rangle_{\mathcal{C}})$.

IV. From $CatMS$ to $CatPN$

The intuition of encoding a flat catalytic P system into a catalytic Petri net is similar to the encoding presented in [18]: an object of the membrane system is associated with a place, and each rule is associated with a transition. The links between membrane systems and Petri nets were further developed in [18] and references therein.

Definition 3. *Starting from a catalytic P system* $\Pi = (O, w^0, R, O_C, O')$*, we can associate it with a catalytic Petri net* $F(\Pi) = ((P, T, W, M_0), \mathcal{P})$ *such that*

- $P = O$ *(with* $\mathcal{C} = O_C$*) and* $T = \{t^r | r \in R\}$*;*
- $\forall t = t^r \in T$ *and* $\forall p \in P$ *with* $r = u \to v$*, we define* $W(p,t) = u(p)$ *and* $W(t,p) = v(p)$
- $M(p) = w^0(p)$ *and* $\mathcal{P} = O'$.

The multiplicity of an object a is modeled by the number of tokens in the corresponding place a. Obviously, this construction leads to a Petri net. Following [18], we introduce two functions: one (ν) associating a configuration of a membrane system with a marking of its corresponding net, and another (σ) associating the rules applied in an evolution of the membrane system with a step in the net.

Definition 4. *Let* $\Pi = (O, w^0, R, O_C, O')$ *be a P system, and* $F(\Pi) = ((P, T, W, M_0), \mathcal{P})$ *the associated Petri net.*

1) *Let* w *be a configuration of* Π*. Then* $\nu(w)$ *is the marking defined by* $\nu(w)(p) = w(p)$*, for all* $p \in P$.
2) *Let* $w \overset{\mathcal{R}}{\Longrightarrow}_C w'$ *be an evolution step of* Π*. Then* $\sigma(\mathcal{R})$ *is the multiset defined by* $\sigma(\mathcal{R})(t^j) = \mathcal{R}(r^j)$*, for all* $t^j \in T$.

We can now state the main result contained in [10].

Theorem 1. *Let* $\Pi = (O, w^0, R, O_C, O')$ *be a membrane system, and* $F(\Pi) = ((P, T, W, M_0), \mathcal{P})$ *be the associated Petri net. We have*

$$w \overset{\mathcal{R}}{\Longrightarrow}_C w' \text{ iff } \nu(w)[\sigma(\mathcal{R})\rangle_{\mathcal{C}} \nu(w').$$

This theorem states that the catalytic net associated with a catalytic P system behaves in the same way.

V. From $CatPN$ to $CatMS$

Given a catalytic Petri net, it is always possible to associate a flat catalytic P system to it: each place is associated with an object.

Definition 5. *Let $\mathcal{N} = ((P, T, W, M_0), \mathcal{P})$ be a catalytic Petri net. Then $K(\mathcal{N}) = (O, w^0, R, O_C, O')$ is the flat catalytic P system defined as follows:*

- *$O = P$, $O_C = \mathcal{C}$, $w^0(p) = M_0(p)$ and $O' = \mathcal{P}$;*
- *for all $t \in T$ define a rule $r^t = u \to v$ in R where $u = pre(t)$ and $v = post(t)$.*

Moreover, we define two mappings ξ and η which associate multisets of objects with markings and multisets of rules with multisets of transitions, respectively.

Definition 6. *Let $\mathcal{N} = ((P, T, W, M_0), \mathcal{P})$ be a catalytic Petri net and $K(\mathcal{N})$ the associated catalytic P system.*

1) *Let M be a marking of $\mathcal{N}$. Then $\xi(M)$ is the configuration defined by $\xi(M)(p) = M(p)$, $\forall p \in P$.*
2) *Let $M[U\rangle_{\mathcal{C}} M'$ be a step of $\mathcal{N}$. Then $\eta(U)$ is the evolution step defined by $\eta(U)(r^t) = U(t)$, $\forall r^t \in R$.*

We are now ready to present the following result:

Theorem 2. *If $\mathcal{N} = ((P, T, W, M_0), \mathcal{P})$ is a catalytic Petri net and $K(\mathcal{N})$ the associated catalytic P system, then*

$$M[U\rangle_{\mathcal{C}} M' \text{ iff } \xi(M) \stackrel{\eta(U)}{\Longrightarrow}_C \xi(M').$$

VI. Timed Catalytic Membrane Systems

The following definition of timed catalytic P systems is similar to that from [6], but without considering signal-promoters and output membranes.

Definition 7. *A* timed (flat) catalytic P system $t\Pi = (O, w^0, R, O_C, O', e)$ *is defined by*

- *O, w^0, R, O, O_C are as in Definition 1;*
- *$e : R \to \mathbb{N}$ is a (computable) function indicating the execution time of each rule; time evolves after applying an evolution step, according to a global clock that starts from 0.*

At each tick of the clock, the rules are applied in a minimal manner (as for catalytic P systems). An evolution rule r started at the j-th tick of the clock ends its execution at the $j+e(r)$-th tick, meaning that the newly created objects by rule r can be used starting from the $j+e(r)+1$-th tick of the clock. When a rule starts, the objects from the left hand side of the rule become unavailable for other rules.

A configuration of $tCatMS$ is a tuple $tw = (w, k)$, namely the multiset of the skin membrane and the value of the global clock. For two configurations tw and tw' of $t\Pi$, we say that there is a transition from tw to tw', and write $tw \Rightarrow_{tC} tw'$, if the minimal evolution step is applied.

The evolution of the system $t\Pi$ at time step k, from a configuration $tw = (w, k)$ to another configuration $tw' = (w', k+1)$ is made by applying a multiset of rules $\mathcal{R}$ in a minimal parallel manner. If the multiset $\mathcal{R}$ is empty, then only the clock is incremented (from k to $k+1$). Given $\mathcal{R}$, we denote by $lhs = \sum_{r \in \mathcal{R}} \mathcal{R}(r) \cdot lhs(r)$ the multiset of objects in the left hand sides of the rules in $\mathcal{R}$. In a similar way, by $rhs_j^k = \sum_{r \in \mathcal{R};\ e(r)=j} \mathcal{R}(r) \cdot\ rhs_j^k(r)$ is denoted the multiset of objects in the right hand sides of the rules in $\mathcal{R}$, applied at time k that are created after j units of time. We also denote by $m = max_{r \in \mathcal{R}} e(r)$ the maximum delay inferred by the rules of $\mathcal{R}$. We say that tw evolves to tw' by a multiset $\mathcal{R}$ of rules (this is denoted by $tw \stackrel{\mathcal{R}}{\Longrightarrow}_{tC} tw'$) if:

(i) $lhs \leq w$;

(ii) $\forall a \in O$, $w'(a) = w(a) - lhs(a) + \sum_{s=max(0,k-m)}^{k} rhs_0^s(a)$.

According to (i), a configuration tw has enough objects to enable the execution of the multiset $\mathcal{R}$ of rules. Condition (ii) describes the effect of applying the rules by adding all the objects created in the last $min(k, m)$ steps which are ready to be used in the membrane system evolution, namely the ones having the subscript $j = 0$. Before incrementing the global clock, all multisets rhs_j^s are transformed into rhs_{j-1}^s for $max(0, k-m) \leq s, j \leq k$.

Each timed P system can be simulated by an untimed P system [2], [6]. The restriction to (timed) catalytic P systems does not change this relation:

Proposition 3. *For every timed catalytic P system $t\Pi = (tO, tw^0, tR, tO_C, tO', e)$ there exists an untimed catalytic P system $\Pi = (O, w^0, R, O_C, O')$ that simulates the evolution of $t\Pi$ (restricted to the elements of tO). Formally, $\forall ta \in tO$ and $k \in \mathbb{N}$ we have $w_k(ta) = tw_k(ta)$, where w_k and tw_k are the multisets of objects Π and $t\Pi$ at step k.*

Remark 1. *It is easy to prove that the class of timed catalytic P systems includes the class of untimed catalytic P systems, since we can assign 0 to all the rules by the timing function e.*

VII. Timed Catalytic Petri Nets

An extension of catalytic Petri nets is defined by adding delays to transitions. The value of the global clock is kept in a variable gc.

Definition 8. *A timed catalytic Petri net is a tuple $t\mathcal{N} = ((P, T, W, D, M_0), \mathcal{P})$ where*

- *P, T, W are as in Definition 2;*
- *$M_0 : P \cup \{gc\} \to \mathbb{N}$ is the initial marking;*
- *$D : T \to \mathbb{N}$ is a* delay mapping.

The delay mapping D introduces a time delay to each object created by a transition; the delays indicate how long the objects cannot be used in other transitions. The initial marking M_0 assigns to each place a number of tokens, and the value 0 to the global clock gc.

Markings represent global states of the timed catalytic Petri nets, and they are defined as functions from $P \cup \{gc\}$

to $\mathbb{N}$. A Petri net $t\mathcal{N}$ evolves at a time step k from a marking tM to another marking tM' by a multiset U of transitions. If the multiset U of transitions is empty, then the only action is incrementing the global clock gc. Given a multiset of transitions U that is catalytically enabled, we denote by $pre(U)(p) = \sum_{tr \in U} U(tr) \cdot W(p, tr)$ the number of tokens associated with the input arcs $(P \times T)$ of all transitions $tr \in U$. In a similar way, by $post^k_j(p) = \sum_{tr \in U;\ D(tr)=j} U(tr) \cdot W(tr, p)$ is denoted the number of tokens associated with the output arcs $(T \times P)$ that are added to their corresponding places after j units of time (k represents the current time). We denote by $m' = max_{tr \in U} D(tr)$ the maximum delay inferred by the transitions of U. A marking tM leads to a marking tM' via a multiset U of transitions (denoted by $tM[U\rangle_{t\mathcal{C}} tM'$) if $tM'(gc) = tM(gc) + 1$ and for each place $p \in P$:

(i) $pre(U)(p) \leq tM(p)$;

$$(ii)\quad tM'(p) = tM(p) - pre(U)(p) + \sum_{s=max(0,k-m')}^{k} post^s_0(p).$$

According to (i), a marking tM has in each place p enough tokens to enable the execution of the multiset U of transitions. Condition (ii) describes the effect of applying the transitions by adding all the tokens created in the last $min(k, m')$ steps which are ready to be used in the evolution of the timed catalytic Petri nets, namely the ones having the subscript $j = 0$. Before incrementing the global clock, all the multisets $post^s_j(p)$ are transformed into $post^s_{j-1}(p)$ for $max(0, k - m') \leq s, j \leq k$.

We proved in [2], that each timed Petri net can be simulated by an untimed Petri net. The restriction to (timed) catalytic Petri nets does not change this relation:

Proposition 4. *For every timed catalytic Petri net $t\mathcal{N} = ((tP, tT, tW, D, tM_0), t\mathcal{P})$ there exists an untimed catalytic Petri net $\mathcal{N} = ((P, T, W, M_0), \mathcal{P})$ that simulates the evolution of $t\mathcal{N}$ (with respect to places of tP). Formally, for all $tp \in tP$ and $k \in \mathbb{N}$ we have $tM_k(tp) = M_k(tp)$, where tM_k and M_k are markings of $t\mathcal{N}$ and $\mathcal{N}$ at step k.*

Remark 2. *It is easy to prove that the class of timed catalytic Petri nets includes the class of catalytic Petri nets, since we can assign 0 to all values of the function D, namely all transitions fire instantaneously.*

VIII. Decomposing $CatPN$ and $tCatPN$ nets

In this section we briefly investigate on the possibility of decomposing catalytic and timed catalytic Petri nets.

Definition 9. *Let I be a finite set of indexes and $\forall i \in I$ let $\mathcal{N}^i = ((P^i, T^i, W^i, M^i_0), \mathcal{P}^i)$ be catalytic Petri nets. Assume that $\forall i, j \in I$, if $i \neq j$ then $P^i \cap P^j = \emptyset$ and $T^i \cap T^j = \emptyset$. Then*

$$((\cup_{i \in I} P^i, \cup_{i \in I} T^i, \cup_{i \in I} W^i, \cup_{i \in I} M^i_0), \cup_{i \in I} \mathcal{P}^i)$$

is called the union *of $\mathcal{N}^i$ and it is denoted with $\bigcup_{i \in I} \mathcal{N}^i$.*

The union of catalytic Petri nets is still a catalytic net.

Proposition 5. *Let $\bigcup_{i \in I} \mathcal{N}^i$ be the union of I catalytic nets satisfying definition 9. Then $\bigcup_{i \in I} \mathcal{N}^i$ is a catalytic net.*

The markings of the union of nets and those of its components are related: each reachable marking in a component is a part of a reachable marking in the whole net.

Proposition 6. *Let $\bigcup_{i \in I} \mathcal{N}^i$ be the union of I catalytic nets satisfying Definition 9. Let M^i be a reachable marking in $\mathcal{N}^i$. Then there exists a reachable marking M in $\bigcup_{i \in I} \mathcal{N}^i$ such that $\forall p \in P^i$. $M^i(p) = M(p)$.*

The above proposition can be used to prove *locally* properties concerning specific parts of the net, with the result that they hold in the whole net as well. In this respect it is useful to stress that Turing completeness arises from having at least two catalytic places, hence catalytic nets with only one catalytic places *are not* Turing complete. However, the union of several *less* powerful components may lead to a net calculating a recursive function.

Definition 10. *Let I be a finite set of indexes and $\forall i \in I$ let $\mathcal{N}^i = ((P^i, T^i, W^i, M^i_0), \mathcal{P}^i)$ be catalytic Petri nets. Assume that $\forall i, j \in I$, if $i \neq j$ then $P^i \cap P^j = \emptyset$ and $T^i \cap T^j = \emptyset$, and also that $|\mathcal{C}^i| \leq k$, for $k > 0$. Then*

$$((\cup_{i \in I} P^i, \cup_{i \in I} T^i, \cup_{i \in I} W^i, \cup_{i \in I} M^i_0), \cup_{i \in I} \mathcal{P}^i)$$

*is called the k-*union *of $\mathcal{N}^i$ and it is denoted with $\bigcup^k_{i \in I} \mathcal{N}^i$.*

Clearly the results in Propositions 5 and 6 hold for nets that are the k-union of a set of k-bounded catalytic nets.

We turn now our attention to timed catalytic net. In this case each net has a place counting the time, but this is not a limitation, as these places *count* the *same* time. Formally:

Definition 11. *Let I be a finite set of indexes and $\forall i \in I$ let $\mathcal{N}^i = ((P^i, T^i, W^i, M^i_0), \mathcal{P}^i)$ be a timed catalytic Petri nets. Assume that $\forall i, j \in I$, if $i \neq j$ then $P^i \cap P^j = \emptyset$ and $T^i \cap T^j = \emptyset$. Then*

$$((\cup_{i \in I} P^i, \cup_{i \in I} T^i, \cup_{i \in I} W^i, \cup_{i \in I} M^i_0), \cup_{i \in I} \mathcal{P}^i)$$

is called the union *of $\mathcal{N}^i$ and it is denoted with $\bigcup_{i \in I} \mathcal{N}^i$.*

The union of timed catalytic Petri nets is still a timed catalytic net. Since the time is global (is the same for all the components), the counterpart of the Proposition 5 holds:

Proposition 7. *Let $\bigcup_{i \in I} \mathcal{N}^i$ be the union of I timed catalytic nets satisfying definition 11. Then $\bigcup_{i \in I} \mathcal{N}^i$ is a timed catalytic net.*

Inspecting the firing rule it is again easy to establish the analogous of Proposition 6, but now we can also easily see that the flow of the time in each components is the same, hence by looking at the time elapsed in the component we find the time elapsed in the whole net.

Proposition 8. *Let $\bigcup_{i\in I}\mathcal{N}^i$ be the union of I timed catalytic nets satisfying definition 11. Let M^i be a reachable marking in N^i. Then there exists a reachable marking M in $\bigcup_{i\in I}\mathcal{N}^i$ such that $\forall p \in P^i$. $M^i(p) = M(p)$. Furthermore $M^i(gc) = M(gc)$.*

Again this proposition is used to prove *locally* properties concerning specific part of the net.

Definition 12. *Let I be a finite set of indexes and $\forall i \in I$ let $\mathcal{N}^i = ((P^i, T^i, W^i, M_0^i), \mathcal{P}^i)$ be timed catalytic Petri nets. Assume that $\forall i, j \in I$, if $i \neq j$ then $P^i \cap P^j = \emptyset$ and $T^i \cap T^j = \emptyset$, and also that $|\mathcal{C}^i| \leq k$, for $k > 0$. Then*

$$((\cup_{i\in I}P^i, \cup_{i\in I}T^i, \cup_{i\in I}W^i, \cup_{i\in I}M_0^i), \cup_{i\in I}\mathcal{P}^i)$$

*is called the k-*union *of $\mathcal{N}^i$ and it is denoted with $\bigcup_{i\in I}^k \mathcal{N}^i$.*

IX. Proving Properties with CPN Tools

There are numerous decidability problems that have to be answered to when using $tCatPN$ for modeling. We show that for the models described in this formalism we can check some behavioral properties using CPN Tools (cpntools.org) if they have only a catalyst. A survey of known decidability issues for Petri nets is given in [12]. CPN Tools is a tool for editing, simulation, state space analysis, and performance analysis of systems described as colored Petri nets (CPN).

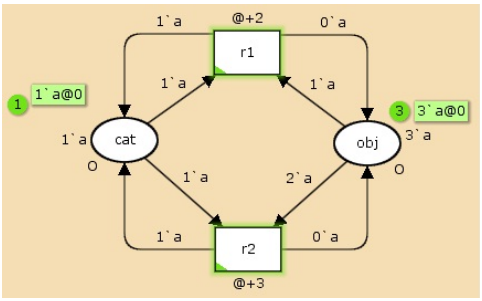

Figure 2. A Simple Timed Catalytic Petri Net

A CPN model is always created as a graphical drawing and Figure 2 contains a simple example: a deterministic catalytic Petri net with a catalytic place and a normal one, for which we have decidability results (Propositions 1 and 2). For simplicity, we consider only tokens represented by a objects. The CPN model contains places, drawn as ellipses or circles, transitions drawn as rectangular boxes, a number of directed arcs connecting places and transitions, and finally, some textual inscriptions next to the places, transitions, and arcs. The inscriptions are written in the ML programming language. Places and transitions are called nodes. An arc always connects a place to a transition or a transition to a place. The arc expressions are multisets of token colors. Next to each place is an inscription which determines the set of token colors (data values) that the tokens on that place are allowed to have. The set of possible token colors is specified by means of a color set of the place. Color sets are defined using the CPN ML declaration: e.g, *colset I = with a,b,c;*. If the tokens have timestamps, then timed color sets are defined: e.g, *colset I = with a,b,c timed;*.

The inscription of the catalytic place cat is $1'a$ specifying that the marking of this place consists of one token a. The timestamps of tokens are written after the symbol @, which is pronounced 'at'. The transitions have a separate time delay: @+t. This means that the tokens are added to the output places after t units of time. The value of the global clock in the initial marking is 0.

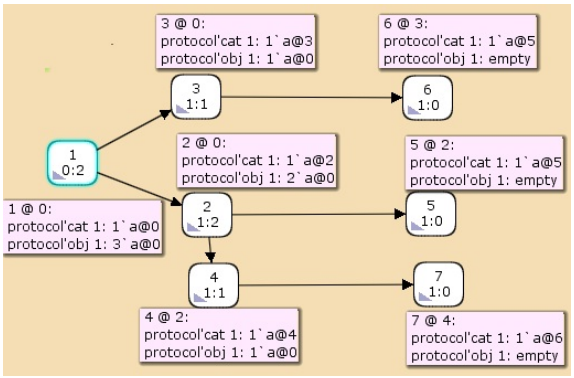

Figure 3. The State Space of a $tCatPN$

A state space is a graph where we have a node for each reachable marking and an arc for each occurring transition. The state space of the $tCatPN$ from Figure 2 can be computed automatically and is represented in Figure 3.

A. Reachability Properties

Reachability is the problem of deciding whether a system can reach a given configuration during its evolution. This problem is useful in the verification of systems, by checking if a system can reach an undesired state. When a biological system is modeled using membrane systems, reachability is also useful. Assume we are modeling a virus attack on an organism. One can be interested if, during evolution, the virus successfully infects some parts of the organism, that is, if the membrane system model reaches some undesired configuration.

The *reachability* properties are concerned with determining whether, for a net $t\mathcal{N}$ there exists an occurrence sequence starting from tM which leads to the marking tM'. There is a one-to-one correspondence between the paths in the state space and the reachable markings. This means that a marking tM' is reachable from a reachable marking tM

iff there exists a path in the state space from the node representing tM to the node representing tM'.

The reachability problem is a central issue of net theory, since many other problems were shown to be equivalent to the reachability problem [14], [17]. The reachability problem was proven to be decidable [21], and the algorithm used in the proof was improved in [19], [20]. Using the connections presented in this paper, we have the following results (the first one holds because the net *calculates* a finite set, the second one uses the decomposition results).

Proposition 9. *The reachability property is decidable in $CatPN$ and $tCatPN$ with finite delays.*

Proposition 10. *Let $\mathcal{N} \in CatPN$ ($tCatPN$, respectively) be a 1-union of $\mathcal{N}^i$ nets in $CatPN$ ($tCatPN$, respectively), for a set of indexes $i \in I$. Then the reachability problem is decidable for $\mathcal{N}$.*

We write a query which uses the state space to determine whether there exists a reachable marking for a $tCatPN$ model, that corresponds to the desired terminal state of the protocol system. For this purpose, we implement a predicate $DesiredTerminal$, which given a node n, returns true if the marking represented by n corresponds to the desired terminal state. We test if a catalytic place cat gets empty in any marking:

fun DesiredTerminal n = ((Mark.protocol'cat 1 n) == empty);

It is also possible to obtain a list containing those markings that satisfy the predicate $DesiredTerminal$. This can be done using the standard query function $PredAllNodes$, that takes a predicate as an argument, applies the predicate to all nodes in the state space, and returns the list of nodes satisfying the predicate. The query is:

PredAllNodes DesiredTerminal;

The result of this query function is the empty list. This tells that there is no marking satisfying the predicate $DesiredTerminal$.

B. Boundedness Properties

Boundedness is a property of systems whose resources may be bounded. From a biological point of view, boundedness can be interpreted as a storage limitation: e.g., in a cell only a finite amount of chemical components are allowed to be stored since the cell cannot accumulate more than a finite amount of material.

A Petri net is said to be *bounded* if its set of reachable markings is finite. The boundedness problem was proven to be decidable [16], and the algorithm used in the proof was improved in [26], [28]. Using the connections presented in this paper, we have the following results:

Proposition 11. *The boundedness property is decidable in $CatPN$ and $tCatPN$ with finite delays.*

Proposition 12. *Let $\mathcal{N} \in CatPN$ ($tCatPN$, respectively) be a 1-union of $\mathcal{N}^i$ nets in $CatPN$ ($tCatPN$, respectively), for a set of indexes $i \in I$. Then the boundedness problem is decidable for $\mathcal{N}$.*

The *boundedness* properties specify how many tokens a place may hold, when all reachable markings are considered. The best upper (lower) integer bound of a place specifies the maximal (minimal) number of tokens that can reside on that place in any reachable marking. If the best upper (lower) integer bound of a place cat is 1, this means that there is exactly one token on this place in any reachable marking. When the best upper and lower integer bounds are equal, this implies that the place is catalytic. Thus we have a method of checking if a place is catalytic.

To find the best upper and lower integer bounds for a place, the CPN state space tool searches through all of the nodes in the state space, finds the number of tokens on that place in each of these states, and returns the largest and smallest of these values. To do this, we define a function $CatNo$ that counts the tokens from place cat:

fun CatNo n = (Mark.protocol'cat 1 n);

To find the best upper and lower integer bounds, $CatNo$ is used as an argument to the query functions $UpperInteger$ and $LowerInteger$ as follows:

UpperInteger CatNo; and *LowerInteger CatNo;*

C. Liveness Properties

A liveness property asserts that a system always progresses. More precisely, when proving the liveness property of a biological dynamical, we check that the time progresses from each reachable configuration.

A dead marking is a marking in which no transitions are enabled. The liveness problem was shown to be equivalent to the reachability problem [3], and thus decidable [14].

Proposition 13. *The liveness property is decidable in $CatPN$ and $tCatPN$ with finite delays.*

Proposition 14. *Let $\mathcal{N} \in CatPN$ ($tCatPN$, respectively) be a 1-union of $\mathcal{N}^i$ nets in $CatPN$ ($tCatPN$, respectively), for a set of indexes $i \in I$. Then the liveness property is decidable for $\mathcal{N}$.*

To find the set of dead markings, the CPN state space tool traverses all nodes in the state space and finds the nodes that lack outgoing arcs. The following function shows how the list of dead markings is obtained using the standard query function $PredAllNodes$:

fun ListDeadMarkings () =
PredAllNodes (fn n => (OutArcs n) = []);

The predicate function provided as an argument to $PredAllNodes$ uses the function $OutArcs$ which lists the outgoing arcs of a node in the state space. A node

represents a dead marking if the list of outgoing arcs is empty. In our example we have three dead markings that correspond to parts of the model that can never be activated. Thus, we can find the halting configurations.

The results of different queries from in Section IX are:

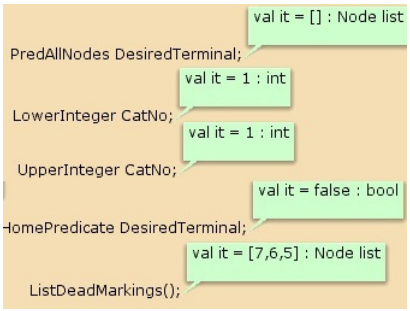

Figure 4. Simple Functions to Test a $tCatPN$

X. Conclusion

In this paper we presented catalytic membrane systems/catalytic Petri nets, that turn out to be Turing complete by using a suitable evolution strategy. We defined their discrete timed extensions (relative to rules/transitions), and established links between these timed extensions and their untimed versions. The timed extensions are useful in modeling biological systems.

Even if the formalisms presented in this paper are Turing complete when using at least two catalysts, the timed extensions can be modeled in CPN Tools (a tool for editing and simulation of systems) when the delays are finite or when the nets can be decomposed into parts where only a catalyst is used (with one catalyst the studied formalism are below Turing). In these cases some quantitative properties of the extended formalisms are studied (e.g, reachability). These properties are useful when modeling biological systems: reachability (checking if the system reaches an undesired configuration), boundedness (imposing certain limitations to the capacity of systems), liveness (checking if the system will never stop working).

We stress the fact that decomposition is quite natural when considering the fact that these formalisms are often the union of separate components and their expressiveness is achieved posing some constraints to their evolution.

Acknowledgments The work of Bogdan Aman and 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] O. Agrigoroaiei, G. Ciobanu. Flattening the Transition P Systems With Dissolution. In *11th Conference On Membrane Computing*, *LNCS*, vol.6501, 53–64, 2010.

[2] B. Aman, G. Ciobanu. Time Delays in Membrane Systems and Petri Nets. In *9th Workshop on Quantitative Aspects of Programming Languages*, *EPTCS*, vol.57, 47–60, 2011.

[3] T. Araki, T. Kasami. Some Decision Problems Related to the Reachability Problem for Petri Nets. *Theor. Comp. Sci.*, vol.3, 85–104, 1977.

[4] T. Araki, T. Kasami. Decidable Problems on the Strong Connectivity of Petri Net Reachability Sets. *Theor. Comp. Sci.*, vol.4, 99–119, 1977.

[5] H.D. Burkhard. On Priorities of Parallelism. In *Conference on Logic of Programs and Their Applications*, *LNCS*, vol.148, 86–97, 1980.

[6] M. Cavaliere, D. Sburlan. Time in Membrane Computing. *Oxford Handbook of Membrane Computing*, 594–604, 2010.

[7] G. Ciobanu. *Membrane Computing and Biologically Inspired Process Calculi*. "A.I.Cuza" University Press, Iaşi, 2010.

[8] G. Ciobanu, L. Pan, Gh. Păun, M.J. Pérez-Jiménez. P Systems With Minimal Parallelism. *Theor. Comp. Sci.*, vol.378, 117–130, 2007.

[9] G. Ciobanu, Gh. Păun, M.J. Pérez-Jiménez. *Applications of Membrane Computing*. Springer, Natural Computing Series, 2006.

[10] G. Ciobanu, G.M. Pinna. Catalytic Petri Nets are Turing Complete. In *6th International Conference on Language and Automata Theory and Applications*, *LNCS*, vol.7183, 192–203, 2012.

[11] C. Dufourd, A. Finkel, P. Schnoebelen. Reset Nets Between Decidability and Undecidability. In *25th International Colloquium on Automata, Languages and Programming*, *LNCS*, vol.1443, 103–115, 1998.

[12] J. Esparza, M. Nielsen. Decidability Issues for Petri Nets - A Survey. *Bulletin of the EATCS*, vol.52, 244–262, 1994.

[13] R. Freund, O. Ibarra, A. Păun, P. Sosík, H.-C. Yen. *Catalytic P Systems*. In [24], 83–117, 2010.

[14] M.H.T. Hack. Decidability Questions for Petri Nets. *Ph.D. Thesis*, M.I.T., 1976.

[15] K. Jensen. *Coloured Petri Nets; Basic Concepts, Analysis Methods and Practical Use. Vol. 1,2,3*. Monographs in Theoretical Computer Science, Springer, 1992, 1994, 1997.

[16] R.M. Karp, R.E. Miller. Parallel Program Schemata. *Journal of Comp. and Syst. Sci.*, vol.3, 147–195, 1969.

[17] R.M. Keller. A Fundamental Theorem of Asynchronous Parallel Computation. In *3rd Annual Sagamore Conference on Parallel Computation*, *LNCS*, vol.24, 102–112, 1975.

[18] J. Kleijn , M. Koutny. *Petri Nets and Membrane Computing*. In [24], 389–412, 2010.

[19] S.R. Kosaraju. Decidability of Reachability in Vector Addition Systems. *14th Annual ACM Symposium on Theory of Computing*, 267–281, 1982.

[20] J.L. Lambert. A Structure to Decide Reachability in Petri Nets. *Theor. Comp. Sci.*, vol.99, 79–104, 1992.

[21] E.W. Mayr. Persistence of Vector Replacement Systems is Decidable. *Acta Infor.*, vol.15, 309–318, 1981.

[22] P. M. Merlin. *A Study of the Recoverability of Computing Systems*. PhD thesis, Department of Information and Computer Science, University of California, Irvine, CA, 1974.

[23] U. Montanari, F. Rossi. Contextual Nets. *Acta Informatica*, vol.32, 545–596, 1995.

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

[25] J.L. Peterson. *Petri Net Theory and the Modelling of Systems*. Prentice-Hall, 1981.

[26] C. Rackoff. The Covering and Boundedness Problem for Vector Addition Systems. *Theor. Comp. Sci.*, vol.6, 223–231, 1978.

[27] C. Ramchandani. *Analysis of Asynchronous Concurrent Systems by Timed Petri Nets*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1974.

[28] L.E. Rosier, H.C. Yen. A Multiparameter Analysis of the Boundedness Problem for Vector Addition Systems. *Journal of Computer and System Sciences*, vol.32, 105–135, 1986.