Simulating membrane systems with costs and local time membrane systems with time Petri nets

B. Aman[1], P. Battyányi[2], G. Ciobanu[1], Gy. Vaszil[2]

[1]Romanian Academy, Institute of Computer Science, Iași, Romania

[2]Department of Computer Science, Faculty of Informatics, University of Debrecen, Debrecen, Hungary

Outline

Membrane systems, the model

- Multisets of objects evolve independently in each region (compartment).
- Evolution rules prescribe how the objects change or move.
- Objects can be transformed into another objects of the same region, they can penetrate the regions and move into the parent or one of the children membranes.
- Rules with additional capabilities can be present: they can dissolve the membrane in which they are placed. Priorities between computational rules add a means for governing control in the computations. We confine ourselves to the basic model, however.
- Computation finishes when the system halts, i.e. no more rule application is possible.

(Păun [2])

Configuration:

- A configuration is the finite sequence of multisets describing the contents of the regions.

Computation:

- A computational sequence is a sequence of configurations.
- A computation ends when there are no applicable rules (when no further configuration change is possible).

The result of the computation is "extracted" from the halting configuration:

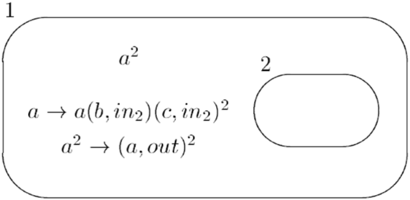
- An integer: The multipliciy of a certain object
- An integer vector: The multiplicities of several objects

A definition

Let $\Pi = (V, \mu, w_1, \ldots, w_n, R_1, \ldots, R_n)$ is a membrane system, where V is the alphabet, μ is a labelled tree conveying the membrane structure, $C_0 = (w_1, \ldots, w_n)$ is the initial configuration and R_i are the rules belonging to membrane i $(1 \leq i \leq n)$.

Example

A system with two regions:



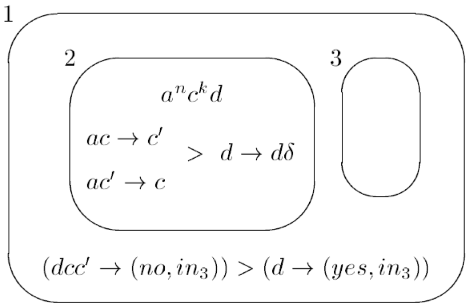
The result:

- The number of objects collected in region 2 is $\{6n \mid n \geq 0\}$.
- The result can also be looked at as the set of vectors

$$\{(2n, 4n) \mid n \geq 0\}$$

Another example

A system to decide whether k divides n



- The input n and k in the form of the multiset $a^n c^k d$ initially contained by region 1.
- δ is an operator to destroy the membrane boundary of region 2
- The result is the object *yes* or *no* appearing in region 3

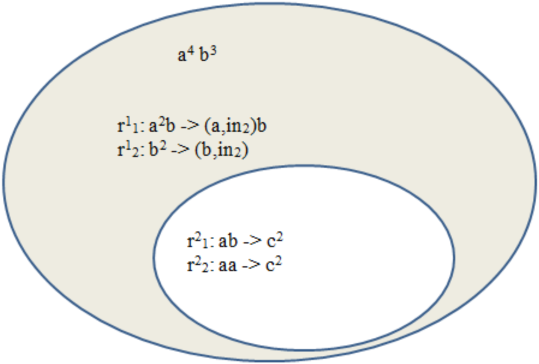
The computational power of membrane systems

- Systems with rules having one object on the left-hand side (non-cooperative systems) generate the semilinear sets of numbers
- Systems with rules having more than one object on their left-hand generate all recursively enumerable sets of numbers (like Turing machines)

- Let $\Pi = (V, \mu, w_1, \ldots, w_n, R_1, \ldots, R_n)$ be a membrane system, where V is the alphabet, μ is the membrane structure, $C_0 = (w_1, \ldots, w_n)$ is the start configuration and $R = (R_1, \ldots, R_n)$ are the set of rules.
- Let $\mathcal{C} : R \to \mathbb{N}$ be a function. Then $\mathcal{C}$ is called a cost function.

Example

Assume we have $\mathcal{C}(r_1^1) = 1$, $\mathcal{C}(r_2^1) = 2$ and $\mathcal{C}(r_1^2) = 1$, $\mathcal{C}(r_2^2) = 3$ in the membrane system below.



Example contd.

Let $\sigma = (U_1, U_2)$ be a computational sequence, where $U_1 = r_1^1 r_1^1$, $U_2 = r_2^2$. Then the total cost of the computation is 5.

Membrane systems with local time

- Let $\Pi = (V, \mu, w_1, \ldots, w_n, R_1, \ldots, R_n)$ be a membrane system, where V is the alphabet, μ is the membrane structure, $C_0 = (w_1, \ldots, w_n)$ is the start configuration and $R = (R_1, \ldots, R_n)$ are the set of rules.
- Let $\mathcal{T} : R \rightarrow IntQ$ be a function, where $IntQ$ is the set of closed intervals with nonnegative rational endpoints. Then $\mathcal{T}$ is called the active time function. We may suppose from now on that the endpoints of the intervals are nonnegative integers. We call $\Pi = (V, \mu, w_1, \ldots, w_n, R_1, \ldots, R_n, \mathcal{T})$ a membrane system with local time.

- Let $\Pi = (V, \mu, w_1, \ldots, w_n, R_1, \ldots, R_n, \mathcal{T})$ be a membrane system with local time. A timed configuration is a pair (C, s), where C is a, possibly intermediate, configuration and $s(r) \in \mathcal{T}(r)$ for every $r \in R$.

- A membrane system with local time is considered with the weak semantics, if every computational step is a maximal parallel set of rules followed by some targetting rules, which means finding the correct place for the new elements. We talk about the strong semantics if the maximal parallel step is modified as follows.

Membrane systems with local time

A selection for membrane m_i is a sequence $sel_i = (t_{j_1}, r_{j_1}, \ldots, t_{j_{k_i}}, r_{j_{k_i}})$, where all the rules are from R_i and $t_{j_l} \in \mathbb{R}_{\geq 0}$. Moreover, we have

1. $r^-_{j_{s+1}} \leq m_i - \sum_{l=1}^{s} r^-_{j_l} \qquad (1 \leq s \leq k_i - 1)$
2. $(\forall r \in R_i)(1 \leq s \leq k_i - 1)([r^- \leq m_i - \sum_{l=1}^{s} r^-_{j_l}] \wedge [\sum_{l=1}^{s} t_{j_l} \leq \mathcal{T}(r)^+] \supset [\sum_{l=1}^{s} t_{j_l} + t_{j_{s+1}} \leq \mathcal{T}(r)^+])$

In words, if a rule in a membrane is applicable we cannot skip the rule by time adjustment.

Let U be a computational step and $sel_i = (t_{j_1}, r_{j_1}, \ldots, t_{j_{k_i}}, r_{j_{k_i}})$ for m_i. Let $\tau_i(s) = \sum_{l=1}^{s} t_{j_l}$. We define the time for U as $\mathcal{T}(U) = max\{\tau_i(k_i) \mid 1 \leq i \leq n\}$. The total time for U is $\mathcal{TT}(U) = \sum_{l=1}^{k_i} \tau_i(l) \cdot \chi(l)$, where $\chi(i) = 1$, if r_i differs from r_{i-1} and 0 otherwise. The cumulative total time for U is $\mathcal{CT}(U) = \sum_{l=1}^{k_i} \tau_i(l)$. If $\sigma = (U_1, \ldots, U_k)$ is a computation, we define the similar notions for σ.

Membrane systems with local time

Remark. Membrane systems with costs are special cases of membrane systems with local time with the weak semantics.

Time Petri nets

- The Petri nets are state/transition systems: conditions are represented by transitions and information are carried by states.
- The Petri nets are bipartite graphs: arcs point from phases to transitions and from transitions to phases. Every arc possesses a multiplicity, which is a positive integer.

- A transition is ready to fire, when each of its prephases contains as many tokens as the multiplicity of the arc coming from that prephase.
- Firing a transition means removing as many tokens from the prephases as prescribed by the multiplicities of the incoming arcs and adding as many tokens to the postphases as the multiplicities of the outgoing arcs.

Formally: a Petri net is a tuple $N = (P, T, F, V, m_0)$ such that

1. P, T, F are finite, where $P \cap T = \emptyset$, $P \cup T \neq \emptyset$ and $F \subseteq (P \times T) \cup (T \times P)$,
2. $V : F \to \mathbb{N}_{>0}$,
3. $m_0 : P \to \mathbb{N}$.

The elements of P are called places and the elements of T are called transitions. The elements of F are the arcs and F is the flow relation of N. The function V is the multiplicity (weight) of the arcs and m_0 is the initial marking. We may occasionally omit the initial marking and simply refer to a Petri net as the tuple $N = (P, T, F, V)$. We stipulate that, for, every transition t, there is a place p such that $W(p, t) \neq 0$.

In the figures below we illustrate a firing sequence of a Petri net (Peterson [3]):

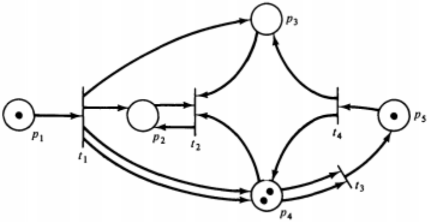


Figure 2.15 A marked Petri net to illustrate the firing rules. Transitions t_1, t_3, and t_4 are enabled.

Continuing the example:

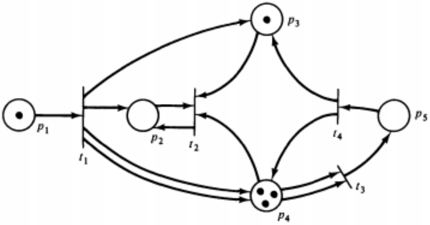


Figure 2.16 The marking resulting from firing transition t_4 in Figure 2.15.

One step more:

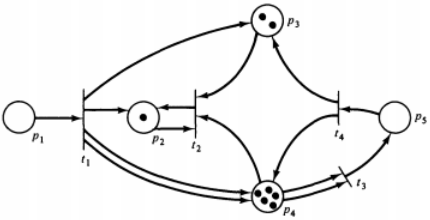


Figure 2.17 The marking resulting from firing transition t_1 in Figure 2.16.

A Time Petri net (TPN) is a 6-tuple $N = (P, T, F, V, m_0, I)$ such that

1. the 5-tuple $S(N) = (P, T, F, V, m_0)$ is a Petri net,
2. $I : T \to Q_{\geq 0} \times Q_{\geq 0}$ and, for each $t \in T$, $I(t)_1 \leq I(t)_2$ holds, where $I(t) = [I(t)_1, I(t)_2]$.

We call $I(t)_1$ and $I(t)_2$ earliest and latest firing times belonging to t, respectively. Notation: $eft(t)$, $lft(t)$.

- A transition marking (or t-marking) is a function $t : T \to \mathbb{R}_{\geq 0} \cup \{\#\}$.
- Let $N = (P, T, F, V, m_o, I)$ be a Time Petri net, m a p-marking and h a t-marking in N. A state in N is a pair $u = (m, h)$ such that
    1. $(\forall t \in T)(t^- \not\leq m \supset h(t) = \#)$,
    2. $(\forall t \in T)(t^- \leq m \supset h(t) \in \mathbb{R}_{\geq 0} \wedge h(t) \leq lft(t))$.

Let t be a transition and $u = (m, h)$ be a state such that $u \longrightarrow^{t}$. Then the result of the firing of t is a new state $u' = (m', h')$, such that $m' = m + \triangle t$ and

$$h'(\hat{t}) = \begin{cases} h(\hat{t}), & \text{if } \hat{t}^{-} \leq m, \hat{t}^{-} \leq m' \text{ and } {}^{\bullet}\hat{t} \cap {}^{\bullet}t = \emptyset, t \neq \hat{t}, \\ \# & \text{if } \hat{t}^{-} \not\leq m, \\ 0 & \text{otherwise }. \end{cases}$$

Observe that we adopt a stronger condition for h to preserve the value for a transition t: we also demand that t should be different than the transition just being fired and they should not have places common in their preplace sets.

Besides the firing of a transition there is another possibility for a state to alter, and this is the time delay step. Let t be a transition and $u = (m, h)$ be a state and $\tau \in \mathbb{R}^+$. Then elapsing of time with τ is possible for the state u (in notation: $u \longrightarrow^{\tau}$), if

$$(\forall t \in T)(h(t) \neq \# \supset h(t) + \tau \leq lft(t)).$$

Then the result of the elapsing of time by τ is defined as follows: $u \longrightarrow^{\tau} u' = (m', h')$, where $m = m'$ and

$$h'(\hat{t}) = \begin{cases} h(\hat{t}) + \tau, & \text{if } \hat{t} + \tau \leq lft(\hat{t}) \text{ for an arbitrary } \hat{t} \in T, \\ \# & \text{otherwise.} \end{cases}$$

Observe that the definition of the result of a time elapse ensures that we are not able to skip a transition when it is enabled.

Results, conjectures

- We established a translation of the basic membrane model into the set of timed Petri nets. We remark that ordinary Petri nets would not suffice for this purpose.
- We managed to formulate membrane systems with costs in terms of time Petri nets.
- We work on translating membrane systems with local time into the set of time Petri nets.

Remark. Especially the last conjecture would have ample consequences when we make use of the corresponding theory of time Petri nets. Observe that in a computation of a membrane system with local time the elapsed times can be real values. The theory of time Petri net tells us that the elapsed times could be chosen as integers when we would intend to calculate minimum or maximum reachability times or when we try to decide cost threshold problems.

- J. H. C. M. Kleijn and M. Koutny and G. Rozenberg, Towards a Petri Net Semantics for Membrane Systems. *Lecture Notes in Computer Science*, volume 3850, International Workshop on Membrane Computing, WMC 2005, 292–309, Springer Verlag, Berlin, 2005.
- G. Păun, Computing with membranes. *Journal of Computer and System Sciences*, 61(1) (2000) 108–143.
- J. L. Peterson, *Petri Net Theory and the Modelling of Systems*, Prentice Hall, N.J., 1981.
- L. Popova-Zeugmann, Essential States in Time Petri Nets, *Informatik- Berichte der HUB*, Nr. 96, 1998.
- L. Popova-Zeugmann, *Time and Petri Nets*, Springer Verlag, Berlin, 2013.