The Microsoft Research - University of Trento
Centre for Computational
and Systems Biology

Technical Report CoSBi 15/2008

Cell Cycle and Tumor Growth in Membrane Systems with Peripheral Proteins

Tommaso Mazza
The Microsoft Research - University of Trento Centre for Computational and Systems Biology
mazza@cosbi.eu

Matteo Cavaliere
The Microsoft Research - University of Trento Centre for Computational and Systems Biology
cavaliere@cosbi.eu

This is the preliminary version of a paper that will appear in ***Electronic Notes in Theoretical Computer Science 227(4) January 2009, Pages 127-141*** *available at*
http://dx.doi.org/10.1016/j.entcs.2008.12.108

Cell Cycle and Tumor Growth in Membrane Systems with Peripheral Proteins

Tommaso Mazza & Matteo Cavaliere

Abstract

Membrane systems with peripheral proteins are essentially standard membrane systems with the possibility of having multisets of objects (proteins) embedded in the membranes and with the presence of rules that control the transport and the change of configurations of these proteins. The model intends to abstract the activities of the receptors embedded in the cellular membranes. Here we use an extension of this paradigm to model and simulate some of the mechanisms underlying cell cycle and breast tumor growth. In particular we have defined a membrane system that abstracts the `G2/M` cell cycle phase transition and extends the corresponding *Reactome* model. The model has been then simulated by using the software Cyto-Sim and we have monitored the interplay between activators and inhibitors of the cell cycle.

1 Introduction

In the membrane systems area (also referred to as P systems), it is usual to represent a membrane (that models a biological membrane) by a pair of square brackets, [ ]. As done in [19], to each topological *side* of a membrane, we associate the multisets u and v (over a particular alphabet V) and this is denoted by $[\ _u]_v$. We say that the membrane is *marked* by u and v; v is called the *external marking* and u the *internal marking*; in general, we refer to them as *markings* of the membrane. The objects of the alphabet V are called *proteins* or, simply, *objects*. An object is called *free* if it is not attached to the sides of a membrane, so is not part of a marking.

Each membrane encloses a *region* and the *contents* of a region can consist of free objects and/or other membranes (we also say that the region *contains* free objects and/or other membranes).

Moreover, each membrane has an associated label that is written as a superscript of the square brackets. If a membrane is associated to the label i we call it membrane i. Each membrane encloses a unique region, so we also say region i to identify the region enclosed by membrane i. The *set of all labels* is denoted by Lab.

For instance, in the system $[abbbbc[abb\ _{ba}]^2_b\ _{ab}]^1_{ab}$, the external membrane, labelled by 1, is marked by ab (internal an external marking). The contents of the region enclosed by the external membrane is composed by the free objects a, b, b, b, b, c and the membrane $[abb\ {}_{ba}]_{b}$. .

We consider rules that model the attachment of objects to the sides of the membranes ([19]).

$$\begin{aligned} \mathit{attach}:\ & [\ a\ {}_{u}]^{i}_{v} \rightarrow [\ {}_{ua}]^{i}_{v}, \quad a[\ {}_{u}]^{i}_{v} \rightarrow [\ {}_{u}]^{i}_{va} \\ \mathit{detach}:\ & [\ \ {}_{ua}]^{i}_{v} \rightarrow [a\ {}_{u}]^{i}_{v}, \quad [\ {}_{u}]^{i}_{va} \rightarrow [\ {}_{u}]^{i}_{v}a \end{aligned}$$

with $a \in V$, $u, v \in V^*$ and $i \in Lab$.

The semantics of the attachment rules (*attach*) is as follows.

For the first case, the rule is *applicable* to the membrane i if the membrane is marked by multisets *containing* the multisets u and v on the appropriate sides, and region i contains an object a. In the second case, the rule is applicable to membrane i if it is marked by multisets containing the multisets u and v, as before, and is contained in a region (or in the environment) that contains an object a. If the rule is applicable we say that the objects defined by u, v and a can be *assigned* to the rule (so that it may be executed).

In both cases, if a rule is applicable and the objects given in u, v and a are assigned to the rule, then the rule can be executed (applied) and the object a is added to the appropriate marking in the way specified. The objects not involved in the application of a rule are left unchanged in their original positions.

The semantics of the detachment rule (*detach*) is similar, with the difference that the attached object a is detached from the specified marking and added to the contents of either the internal or external region.

As it is biologically relevant, we also consider rules associated to the membranes that control the passage of objects across the membranes (again, from [19]). Precisely:

$$\begin{aligned} \mathit{move}_{\mathit{in}}:\ & a[\ \ {}_{u}]^{i}_{v} \rightarrow [\ a\ {}_{u}]^{i}_{v} \\ \mathit{move}_{\mathit{out}}:\ & [\ a\ {}_{u}]^{i}_{v} \rightarrow a[\ \ {}_{u}]^{i}_{v} \end{aligned}$$

with $a \in V$, $u, v \in V^*$ and $i \in Lab$.

The semantics of the rules is as follows.

In the first case, the rule is applicable to membrane i if it is marked by multisets containing the multisets u and v, on the appropriate sides, and the membrane is contained in a region containing an object a. The objects defined by u, v and a can thus be assigned to the rule. If the rule is applicable and the objects a, u and v are assigned to the rule then the rule can be executed (applied) and, in this case, the object a is removed from the contents of the region surrounding membrane i and added to the contents of region i.

In the second case, the semantics is similar, but here the object a is moved from region i to its surrounding region (or environment).

The rules of attach, detach, move$_{in}$, move$_{out}$ are generally called *membrane rules* (denoted collectively as mem_{rul}) over the alphabet V and the set of labels Lab. Several restrictions have been defined in [19]. In particular, membrane rules for which $|uv| \geq 2$ are called *cooperative* membrane rules (in short, coo_{mem}). Membrane rules for which $|uv| = 1$ are called *non-cooperative* membrane rules (in short, $ncoo_{mem}$). Membrane rules for which $|uv| = 0$ are called *simple* membrane rules (in short, sim_m).

We also admit *evolution rules* that involve objects but not membranes. These can be considered to model the biochemical reactions that take place inside the compartments of the cell. They are evolution rules over the alphabet V and set of labels Lab (no indication on the destination of the produced objects is present). We define

$$evol: \ [u \to v]^i$$

with $u \in V^+, v \in V^*$ and $i \in Lab$. An evolution rule is called *cooperative* (in short, coo_e) if $|u| > 1$, otherwise the rule is called *non-cooperative* ($ncoo_e$).

The rule is applicable to region i if the region *contains* a multiset of free objects that *includes* the multiset u. The objects defined by u can thus be assigned to the rule.

If the rule is applicable and the objects defined by u are assigned to the rule, then the rule can be executed. In this case the objects specified by u are subtracted from the contents of region i while the objects specified by v are added to the contents of the region i.

A *membrane system with peripheral proteins* (in short, a P_{pp} system) and n membranes, is then a construct, [19]

$$\Pi = (V, \mu, (u_1, v_1), \ldots, (u_n, v_n), w_1, \ldots, w_n, R, R^m)$$

where:

- V is a finite, non-empty alphabet of objects (proteins).

- μ is a membrane structure with $n \geq 1$ membranes, injectively labelled by $1, 2, \ldots, n$.

- $(u_1, v_1), \cdots, (u_n, v_n) \in V^* \times V^*$ are the markings associated, at the beginning of any evolution, to the membranes $1, 2, \cdots, n$, respectively. They are called *initial markings* of Π; the first element of each pair specifies the internal marking, while the second one specifies the external marking.

- $w_1, \cdots, w_n$ specify the multisets of free objects contained in regions $1, 2, \cdots, n$, respectively, at the beginning of any evolution and they are called *initial contents* of the regions.

- R is a finite set of evolution rules over V and the set of labels $Lab = \{1, \ldots, n\}$.

- R^m is a finite set of membrane rules over the alphabet V and set of labels $Lab = \{1, \ldots, n\}$.

A *configuration* of Π consists of a membrane structure, the markings of the membranes (internal and external) and the multisets of free objects present inside the regions. In what follows, configurations are denoted by writing the markings as subscripts (internal and external) of the parentheses which identify the membranes. The labels of the membranes are written as superscripts and the contents of the regions as string, e.g.,

$[\,[\ {}_{aa}]^4_{ab}\ [aaa\ {}_{aa}]^2_{b}\ [\ b\ ]^3_{bb}\ a\ ]^1_{a}$

We suppose a standard labelling: 0 is the label of the *environment* that surrounds the entire system Π; 1 is the label of the *skin* membrane that separates Π from the *environment*.

The *initial configuration* consists of the membrane structure μ, the initial markings of the membranes and the initial contents of the regions; the environment is empty at the beginning of the evolution.

We denote by $\mathbb{C}(\Pi)$ the *set of all possible configurations* of Π.

We assume the existence of a clock which marks the timing of steps (single *transitions*) for the whole system.

A *transition* from a configuration $C \in \mathbb{C}(\Pi)$ to a new one is obtained by assigning the objects present in the configuration to the rules of the system and then executing the rules as previously described. One can define several ways of assigning the objects to the rules. In [19] and [20] two different ways of assigning the objects have been defined and investigated: free-parallel and maximally-parallel. In the *free parallel mode*, in each region and for each marking, *an arbitrary number* of applicable rules is executed (this mode is also called asynchronous in the P systems area). In the *maximally parallel way*, in each region and for each marking, applicable rules chosen in a non-deterministic way are assigned objects, also chosen in a non-deterministic way, such that after the assignment no further rule is applicable using the unassigned objects. These two ways, conceptualize two ways of abstracting the application of biochemical reactions. Equivalence with Petri nets, counter machines and decision problems concerning these two classes of systems have been studied in [20]. We only mention here the following results: In the free-parallel case it is decidable whether or not an arbitrary membrane system with peripheral proteins can reach an arbitrary configuration or marking; the same problem becomes undecidable when the systems evolves in the maximally parallel way (the proofs of such results and other intermediate cases can be found in [20]).

It is known that membrane proteins can cluster and form more complex molecules whose activity is very distinct from the original components; moreover proteins can cross sides of a membrane and proteins on opposite sides can influence each other, in a “synchronized” manner. To capture all these aspects we extend the considered paradigm by admitting evolution rules also for the proteins embedded in the membranes.

This can be done in a rather natural manner since membrane proteins are represented as multisets of objects, and then we can still use multiset rewriting rules to represent these membrane processes.

Precisely, we can introduce *membrane-evolution rules* in this form:

$$mem - evol : \ [\ _{u}]^{i}_{v} \rightarrow [\ _{u'}]^{i}_{v'}$$

with $u, v, u', v' \in V^*$ and $i \in Lab$; if $u = \lambda$ or $v = \lambda$ then $u' = \lambda$ or $v' = \lambda$, respectively.

The rule is applicable to membrane i if the internal marking of the membrane *contains* the multiset of proteins u and the external marking contains the multiset v. The proteins defined by u and v can thus be assigned to the rule. If the rule is applicable and the objects defined by u and v are assigned to the rule, then the rule can be executed. In this case the objects specified by u are subtracted from the internal marking of membrane i, the objects specified by v are subtracted from the external marking of membrane i, while the objects specified by u' are added to the internal marking of membrane i and the objects specified by v' are added to the external marking of membrane i.

Looking into the details of the proof of Theorem 6.2 in [20], one can see that is easy to extend the result and prove that is possible to decide the reachability of arbitrary configurations and markings for membrane systems with peripheral proteins and membrane-evolution rules, when the systems work in a free-parallel manner. In fact, in Theorem 6.2, all floating and attached objects are indexed with the labels of the membranes in which they float or to which they are attached. Membrane-evolution rules can be then rewritten as cooperative evolution rules acting only on the attached objects.

However, from a computational point of view, it is not clear if the inclusion of membrane-evolution rules lead to higher complexity algorithms. The computational study of membrane systems with peripheral proteins and membrane evolution-rules is not the goal of this paper and is then left as research topic. The proposed membrane evolution rules can also be seen as a generalization of the protein rules used in [22], where only one single protein can be rewritten, on one side of the membrane. Moreover, similar types of rules have been included in the stochastic simulator presented in [21], [17]: in that case the attachment of an object can allow the rewriting of the multiset of embedded proteins. A survey of membrane systems with embedded proteins is [18].

2 Concluding Remarks

Membrane systems with peripheral proteins can specify cellular processes where the role of cellular receptors is important. We have presented an example by studying the pathways underlying cell cycle and breast tumor growth. In such a case protein binding to membrane receptors is a fundamental activity accomplished by cells to trigger the responses to extracellular or endogenous stresses. Membrane systems with peripheral proteins, working in the free-parallel manner, have been shown to be equivalent to Petri nets (see [20]). Such results should be extended to the model with the introduction of membrane-evolution rules, introduced in Section 1. Moreover, the stochastic variant of the model should also be investigated, and, possibly, in view of the results in [20] equivalence with stochastic Petri nets could be found.

References

[1] A. Benzinger, N. Muster, H. B. Koch, J. R. Yates, H. Hermeking, *Targeted proteomic analysis of 14-3-3 sigma, a p53 effector commonly silenced in cancer.* Mol. Cell. Proteomics, **4(6)** (2005), 785–95.

[2] H.-Y. Yang, Y.-Y. Wen, C.-H. Chen, G. Lozano, M.-H. Lee, *14-3-3sigma positively regulates p53 and suppresses tumor growth.* Mol. Cell. Biol., **23(20)** (2003), 7096–7107.

[3] K. Ikeda, A. Orimo, Y. Higashi, M. Muramatsu, S. Inoue, *Efp as a primary estrogen-responsive gene in human breast cancer.* FEBS Lett., **472(1)** (2000), 9–13.

[4] D. Zivadinovic, B. Gametchu, C.S. Watson, *Membrane estrogen receptor-alpha levels in MCF-7 breast cancer cells predict cAMP and proliferation responses.* Breast Cancer Res., **7(1)** (2005), R101–12.

[5] T. Ouchi, A. N. A. Monteiro, A. August, S. A. Aaronson, H. Hanafusa, *BRCA1 regulates p53-dependent gene expression.* Proc. Natl. Acad. Sci. USA, **95(5)** (1998), 2302–2306.

[6] R. E. Shackelford, W. K. Kaufmann, R. S. Paules, *Cell cycle control, checkpoint mechanisms, and genotoxic stress.* Environ. Health Perspect. Suppl., **107(S1)** (1999), 5–24.

[7] B. Novák, J. C. Sible, T. T. Tyson, *Checkpoints in the cell cycle.* in Encyclopedia of Life Sciences, London, Nature Publishing Group (2002), 1–8.

[8] T. Urano, T. Saito, T. Tsukui, M. Fujita, T. Hosoi, M. Muramatsu, Y. Ouchi, S. Inoue, *Efp targets 14-3-3 sigma for proteolysis and promotes breast tumour growth.* Nature, **417(6891)** (2002), 871–875.

[9] Y. Haupt, R. Maya, A. Kazaz, M. Oren, *Mdm2 promotes the rapid degradation of p53.* Nature, **387** (1997), 296–299.

[10] N. Mailand, J. Falck, C. Lukas, R.G. Syljuasen, M. Welcker, J. Bartek, and J. Lukas, *Rapid destruction of human cdc25a in response to dna damage.* Science **288(5470)** (2000), 1425–1429.

[11] D. P. Lane, *p53, guardian of the genome.* Nature, **358(6381)** (1992), 15–16.

[12] I. Vastrik, P. D'Eustachio, E. Schmidt, G. Joshi-Tope, G. Gopinath, D. Croft, B. de Bono, M. Gillespie, B. Jassal, S. Lewis, L. Matthews, G. Wu, E. Birney, L. Stein, *Reactome: a knowledge base of biologic pathways and processes.* Genome Biology, **8(3)** (2007), R39.

[13] M. Hucka, A. Finney, et al, *The Systems Biology Markup Language (SBML): a medium for representation and exchange of biochemical network models.* Bioinformatics, **5(19)** (2003), 524–531.

[14] G. Franco, P.H. Guzzi, T. Mazza, V. Manca, *Mitotic oscillators as MP graphs.* in Proceedings of the seventh Workshop on Membrane Computing (WMC7), LNCS, **4361** (2006), 382–394.

[15] T. Mazza, *Towards a complete covering of SBML functionalities.* in Proceedings of the eight Workshop on Membrane Computing (WMC8), LNCS, **4860** (2007), 425–444.

[16] T. Mazza, A. Nocera, *A formal lightweight view of the p53-dependent G1/S checkpoint control.* in Proceedings of Prague International Workshop on Membrane Computing (2008), 35–46.

[17] S. Sedwards, T. Mazza, *Cyto-Sim: a formal language model and stochastic simulator of membrane-enclosed biochemical processes.* Bioinformatics, **20(23)** (2007), 2800–2802.

[18] M. Cavaliere, S.N. Krishna, Gh. Păun, A. Păun, *P systems with objects on membranes.* In *The Handbook of Membrane Computing*, Oxford University Press, to appear.

[19] M. Cavaliere, S. Sedwards, *Membrane systems with peripheral proteins: transport and evolution*, CoSBi Technical Report **04/2006**, www.cosbi.eu, and Electronic Notes in Theoretical Computer Science, ENTCS, **171(2)** (2007), 37–53.

[20] M. Cavaliere, S. Sedwards, *Decision problems in membrane systems with peripheral proteins, transport and evolution*, CoSBi Technical Report **12/2006**, www.cosbi.eu, and Theoretical Computer Science, to appear.

[21] M. Cavaliere, S. Sedwards, *Modelling cellular processes using membrane systems with peripheral and integral proteins*, Computational Methods in Systems Biology, Intern. Conference, Trento (2006), LNCS - LNBI **4210**, Springer (2006), 108–126.

[22] A. Păun, B. Popa, *P systems with proteins on membranes*, Fundamentae Informaticae, **72(4)** (2006), 467–483.

[23] Cyto-Sim is available at http://www.cosbi.eu/