Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 316 (2015) 29–44

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

Verification of Spatial and Temporal Modalities in Biochemical Systems

Davide Chiarugi [1]

Max Planck Institute of Colloids and Interfaces, Potsdam-Golm. Germany.

Moreno Falaschi Diana Hermith

Dipartimento di Ingegneria dell'informazione e Scienze Matematiche.
Università degli Studi di Siena. Italy.

Carlos Olarte

ECT, Universidade Federal do Rio Grande do Norte. Brazil.
DECC, Universidad Javeriana Cali. Colombia.

Abstract

Biochemical systems such as metabolic and signaling pathways tend to be arranged in a physical space: the product of one reaction must be in the right place to become the reactant for the subsequent reaction in the pathway. Moreover, in some cases, the behavior of the systems can depend on both, the location of the reactants as well as on the time needed for the reaction to occur. We address the problem of specifying and verifying properties of biochemical systems that exhibit both temporal and spatial modalities at the same time. For that, we use as specification language a fragment of intuitionistic linear logic with subexponentials (SELL). The subexponential signature allows us to capture the spatial relations among the different components of the system and the timed constraints for reactions to occur. We show that our framework is general enough to give a declarative semantics to P-Systems and we show that such logical characterization has a strong level of adequacy. Hence, derivations in SELL follow exactly the behavior of the modeled system.

Keywords: Biochemical systems, linear logic, spatial and temporal modalities.

1 Introduction

One of the main difficulties of building computational models for biological systems arise from the characteristics of the available information. Indeed, even for the best-studied systems, the known data cannot describe exhaustively the properties of each molecular species; even less known are the details of spatial information and the

[1] Email: {davide.chiarugi, carlos.olarte}@gmail.com , {moreno.falaschi, diana.hermith}@unisi.it

http://dx.doi.org/10.1016/j.entcs.2015.06.009

timing of events. Thus, desirable features of a computational modeling framework should regard the capability of dealing with information often both incomplete and of non-uniform quality.

Another desirable feature for computational models is the ability to describe a biological system at different levels of abstraction. This may be useful to capture the variability of subnetworks in the topology of a biochemical reactions network, for instance, at the level of metabolic or signaling pathways.

Several computational frameworks for modeling in different ways various aspects of biological systems have been defined in the last decade (see e.g., [28,21,20,25,38,52,39,12,7,29]). However, so far we have not seen one single formalism for modeling reaction systems with both time and space, and, at the same time, with the ability to express a logic for proving properties which can depend both on time and space locations. Normally, there is one formalism and a language for the modeling and the specification of a biological system and at least another different formalism for expressing the properties of interest (e.g., a temporal logic) and for proving them (e.g., by using a model checker).

Our approach for specifying and studying biological systems grounds on Concurrent Constraint Programming (CCP) [50] and on linear logic (LL) [30]. The former is a model for concurrency where agents interact by telling and asking constraints (i.e., logical formulas) into a store of partial information; the latter, is a substructural logic where formulas are seen as *resources*. Interestingly, the language of CCP processes is flexible enough to faithfully capture different modalities of concurrent systems (e.g., temporal, spatial and epistemic modalities) while keeping a declarative semantics based on (intuitionistic) LL as shown in [27,42]. This means that CCP models can be seen as *runnable specifications*: the model can be executed to observe the traces of the systems and, more importantly, the underlying theory of CCP and all the meta theory developed for LL can be used to verify systems' properties.

Another salient characteristic of CCP is its ability to deal with partial information: constraints add information on the system variables (e.g., $x > 42$) rather than determining the value of the variables. Hence, the more information is obtained the more constraints are accumulated and more information can be deduced from the system. Constraints also provide a compact representation of the *state of the system* (as predicates on system variables). Moreover, being able to deal with partial information is certainly useful in situations where either some components of the system are not fully specified or we do not have enough quantitative information about them.

In a previous work [16] we used the `ntcc` calculus [41], a non-deterministic temporal extension of CCP, for representing reaction rules in biological systems. This language allowed us to model discrete-time, and hence biological systems where reactions have a duration over time. Later, in [36], we described a modeling strategy based on `ntcc` where starting from an abstract model, we built refinements adding further details coming from experimentation or abstract assumptions. In a following work [14], we modeled spatial distributions in biochemical reactions. This thus

$$\frac{\Gamma, F, H \longrightarrow G}{\Gamma, F \otimes H \longrightarrow G}\ \otimes_L \qquad \frac{\Gamma_1 \longrightarrow F \quad \Gamma_2 \longrightarrow H}{\Gamma_1, \Gamma_2 \longrightarrow F \otimes H}\ \otimes_R \qquad \frac{\Gamma_1 \longrightarrow F \quad \Gamma_2, H \longrightarrow G}{\Gamma_1, \Gamma_2, F \multimap H \longrightarrow G}\ \multimap_L \qquad \frac{\Gamma, F \longrightarrow H}{\Gamma \longrightarrow F \multimap H}\ \multimap_R$$

$$\frac{\Gamma, F_i \longrightarrow G}{\Gamma, F_1 \,\&\, F_2 \longrightarrow G}\ \&_{L_i} \qquad \frac{\Gamma \longrightarrow F \quad \Gamma \longrightarrow H}{\Gamma \longrightarrow F \,\&\, H}\ \&_R \qquad \frac{\Gamma, F[e/x] \longrightarrow G}{\Gamma, \exists x.F \longrightarrow G}\ \exists_L \qquad \frac{\Gamma \longrightarrow G[t/x]}{\Gamma \longrightarrow \exists x.G}\ \exists_R$$

Fig. 1. A fragment of the LL introduction rules. e is a fresh variable. Rules $\forall_{L/R}$ are dual to $\exists_{R/L}$

allowed us to deal with cell membranes, or more in general, with the possibility to express the fact that a reaction can take place only when some reactants are in the same "*location*".

In this paper we deal with the problem of specifying and verifying properties of biochemical systems that exhibit both temporal and spatial modalities at the same time. For that, we shall encode the proposed systems as formulas in linear logic with subexponentials (SELL) [23]. The corresponding CCP language able to manipulate such SELL formulas was studied in [42] and [43]. We omit this language here to focus on the logical framework that allows for proving, in a natural way, some relevant properties of the modeled system.

More precisely, we shall show that it is possible to use two kind of subexponentials for representing the two main dimensions, namely time and space, for modeling biochemical systems where reactions depend on the location of reactants and on the duration of interactions. Then, we show the expressiveness of our framework by encoding P-Systems [46,3], a general model of computation inspired on cells structures. We show that our logical characterization of P-Systems has a strong level of adequacy, which means that derivations in the logical system follow exactly the rules (reactions) defined for the modeled system. We also show how to exploit the underlying logic for expressing, and proving, properties of interest that involve temporal and spatial modalities.

The rest of the paper is structured as follows. In Section 2 we recall some concepts about subexponentials in linear logic. Section 3 defines an encoding of biochemical reactions that considers spatial and temporal modalities. In Section 4 we show how to exploit the underlying logic for expressing properties of the system and how to encode P-Systems as SELL specifications. Section 5 discusses some related work and concludes the paper. The detailed proofs of the results here presented can be found in the companion technical report [17]. In [17] the reader may also find an application of the framework developed here in the context of the TWEAK-Fn14 cell signaling pathway [5].

2 Linear Logic with Subexponentials

Linear logic (LL) with subexponentials (SELL) [23] shares with LL [30] all its connectives except the exponentials ! and ?. Figure 1 presents the introduction rules of the fragment of intuitionistic SELL that will be used here. As one can observe from these rules, in particular in the $\otimes_R$ rule, LL formulas are not always allowed to contract and weaken. These rules are controlled in LL by the exponentials $!, ?$ and in SELL by the subexponentials, written as $!^a, ?^a$, where a is a label.

Formally, a SELL system is specified by a *subexponential signature* $\Sigma = \langle I, \preceq, U \rangle$, where I is a set of labels, $U \subseteq I$ specifies which subexponentials allow weakening and contraction, and $\preceq$ is a pre-order among the elements of I. We shall use $a, b, \ldots$ to range over elements in I. For a given such subexponential signature, SELL_Σ is the system obtained by adding the following inference rules to the LL rules in Figure 1:

- For each $b \in U$, we add the following structural rules:

  $$\dfrac{\Gamma \longrightarrow G}{\Gamma, !^b F \longrightarrow G}\ W \qquad \dfrac{\Gamma, !^b F, !^b F \longrightarrow G}{\Gamma, !^b F \longrightarrow G}\ C$$

  That is, one can specify the subexponentials that behave linearly, namely those in $I \setminus U$, and those that behave classically, namely those in U.

- For each $a \in I$, we add the dereliction rule to the left and the promotion rule to the right:

  $$\dfrac{\Gamma, F \longrightarrow G}{\Gamma, !^a F \longrightarrow G}\ !^a{}_L \qquad \dfrac{!^{a_1} F_1, \ldots, !^{a_n} F_n \longrightarrow F}{!^{a_1} F_1, \ldots, !^{a_n} F_n \longrightarrow !^a F}\ !^a{}_R, \text{ provided } a \preceq a_i \text{ for } 1 \leq i \leq n.$$

Observe that provability is preserved *downwards*: if the sequent $\Gamma \longrightarrow !^a P$ is provable in SELL_Σ, then so is the sequent $\Gamma \longrightarrow !^b P$ for all $b \preceq a$. We shall omit the signature Σ when it is understood from the context.

It is known that subexponentials greatly increase the expressiveness of the system when compared to LL. For instance, in [42], it is shown how the subexponentials can be interpreted as temporal units or spatial and epistemic modalities in distributed systems. For that, in [42] universal ($\Cap$) and existential ($\Cup$) quantification on subexponentials are introduced with the following proof rules:

$$\dfrac{\Gamma, P[l/l_x] \longrightarrow G}{\Gamma, \Cap l_x : a.P \longrightarrow G}\ \Cap_L \quad \dfrac{\Gamma \longrightarrow G[l_e/l_x]}{\Gamma \longrightarrow \Cap l_x : a.G}\ \Cap_R \quad \dfrac{\Gamma, P[l_e/l_x] \longrightarrow G}{\Gamma, \Cup x : a.P \longrightarrow G}\ \Cup_L \quad \dfrac{\Gamma \longrightarrow G[l/l_x]}{\Gamma \longrightarrow \Cup l_x : a.G}\ \Cup_R$$

In these rules, l_e is fresh and it represents a subexponential variable playing a similar role as eigenvariables. The generic variable $l_x : a$, where $a \in I$ plays the role of the type of l_x, represents *any subexponential constant* l_c in the ideal of a, i.e., $l_c \preceq a$.

The above system, called $\mathrm{SELL}^{\Cap}$, enjoys good proof theoretic properties: [23] (resp. [42]) proved that SELL (resp. $\mathrm{SELL}^{\Cap}$) admits cut-elimination. Moreover, a sound and complete focused proof system [2] for those systems can be given [42]. Focusing is a powerful discipline on proofs which can be seen as normal form proofs for proof search. In fact, we shall use focusing to prove the adequacy of specifications as shown in the forthcoming sections.

3 Spatial and Temporal Dependencies in $\mathrm{SELL}^{\Cap}$

In this section we show how spatial and temporal dependencies in biochemical reactions can be neatly modeled as formulas in $\mathrm{SELL}^{\Cap}$. The encoding we propose not only gives a logical meaning to those systems but also exhibits a close correspondence between the behavior of the system and the shape of the proofs in

SELL$^{\Cap}$.

We start by describing the kind of reactions we shall consider. We assume a set of reactions of the shape:

$$r_j : \quad [c_1.A_1]_{a_1} + \cdots + [c_n.A_n]_{a_n} \longrightarrow^k [d_1.B_1]_{b_1} + \cdots + [d_m.B_n]_{b_m} \tag{1}$$

meaning that c_i units of A_i *located* in the *space domain* a_i are consumed in k *time-units* to produce d_j units of B_j in the space domain b_j.

In order to combine spatial and temporal modalities in SELL$^{\Cap}$ we need first to define a subexponential signature as the one depicted in Figure 2. The only unbounded subexponentials are t_ω (time) and s_ω (space). The former will be used to mark the set of reactions that can be used as many times as needed. The second will be used in the encoding of P-System in Section 4.2. The linear subexponentials $1, 2, 3, \cdots$ represents temporal time-units. The subexponentials $i+$ represent the time-units starting from i. For instance, a subexponential variable $l_x : 4+$ can be instantiated with any time-unit (in the future) starting from 4. Those subexponentials will be used to specify system's properties as we explain in Section 4. Finally, the linear subexponentials $s_a.i$ will be used to mark the formulas (reactants) available in the *space* domain s_a in the time-unit i.

For each reactant A in the system, we assume to have a constant symbol A in the logic. We also assume to have an uninterpreted binary predicate $\mathtt{ct}(\cdot,\cdot)$. Intuitively, the formula $!^{s_b.2}\mathtt{ct}(A, c)$ means that the concentration of A in the space domain s_b is c during the second time-unit. As usual, c is defined as the n-th application of the successor function $\mathtt{suc}$ to the constant $\mathbf{0}$. We shall use $\mathtt{suc}^n(x)$ to denote the n-th application of $\mathtt{suc}$ to x.

We model the state of the system at time-unit t (i.e., the concentration of each specie in each space at time t) as the formula

$$\mathtt{state}(t) \stackrel{\mathtt{def}}{=} \bigotimes_{s\in\mathcal{S}} \bigotimes_{A_i\in\mathcal{A}} !^{s.t}\left[\mathtt{ct}(A_i, c_i)\right]$$

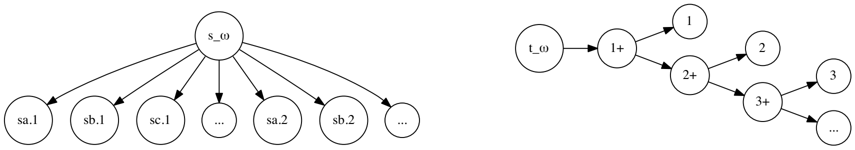
where $\mathcal{A}$ denotes the set of reactants and $\mathcal{S}$ the set of domain spaces. If there are no species of kind A_j in the space s_k, then $c_j = \mathbf{0}$. Intuitively, $!^{s.t}\left[\mathtt{ct}(A, c)\right]$ means that the formula $\mathtt{ct}(A, c)$ is available at *location* $s.t$ that represents the time-unit t in the space domain s. Hence, in location $s.t$ we can deduce that the concentration of A is c.

We shall model the set of reaction of the system as the following formula:

$$\mathtt{eqs} \stackrel{\mathtt{def}}{=} !^{t_\omega}\left[\Cap l_x : 1+ . \left[\mathtt{eq}_1(l_x) \& \cdots \& \mathtt{eq}_k(l_x)\right]\right]$$

The unbounded subexponential $!^{t_\omega}$ allows us to use the set of reactions as many times as needed. The universal quantification $\Cap l_x : +1$ says that at any time-unit the reactions are available. Moreover, the use of the LL connective & allows us to *choose* (non-deterministically) one of the reactions and then discard the others.

The model of a reaction (Equation 1) is a formula that first checks if the needed reactants are available in the specific space domains. If this is the case, the reactants are consumed and the products are added k time-units later:



(a) Spatial domain

(b) Temporal domain

Fig. 2. Subexponential structure for spatial domain (a) and time (b). $b \rightarrow a$ means $a \preceq b$. The subexponential $i+$ represent the time-units starting from i. The subexponential $s_a.i$ marks the formulas (reactants) available in the space domain s_a in the time-unit i. The only unbounded subexponentials are s_ω and t_ω.

$$\begin{aligned}\mathtt{eq}(t) \quad &\stackrel{\mathtt{def}}{=} \forall \overline{x}.\ [\mathtt{consume}(t) \multimap \mathtt{produce}(t+k)] \\ \mathtt{consume}(t) &\stackrel{\mathtt{def}}{=} \bigotimes_{s \in \mathcal{S}} \bigotimes_{A_i \in \mathcal{A}} !^{s.t}\, [\mathtt{ct}(A_i, N_i)]\end{aligned}$$

where $\overline{x} = x_1, ..., x_n$ and

$$N_i = \begin{cases} x_i & \text{if } [c_i.A_i]_s \text{ does not occur in the left-hand side of the reaction} \\ \mathtt{suc}^{c_i}(x_i) & \text{if } [c_i.A_i]_s \text{ occurs in the left-hand side of the reaction} \end{cases}$$

The formula $\mathtt{produce}(t)$ is the same as $\mathtt{consume}(\mathrm{t})$ but, in this case,

$$N_i = \begin{cases} x_i & \text{if } [d_i.A_i]_s \text{ does not occur in the right-hand side of the reaction} \\ \mathtt{suc}^{d_i}(x_i) & \text{if } [d_i.A_i]_s \text{ occurs in the right-hand side of the reaction} \end{cases}$$

The quantifier $\forall \overline{x}$ allows us to bind the current number of reactants in the system. The formula consume consumes the needed reactants and produce adds such reactants k time-units later. We note that once a rule is applied, the concentrations of the reactants that do not occur in the reaction are simply copied (without changes) to the time-unit $t+k$ (due to the first cases of N_i above).

Finally, the model of the system at a given time-unit t is:

$$\mathtt{system}(t) \stackrel{\mathtt{def}}{=} \mathtt{eqs} \otimes [\mathtt{state}(t)]$$

3.1 Behavior and correspondence

In this section we show that our model enjoys interesting properties. In particular, we shall show that a step in a focused derivation [2] corresponds exactly to one step in the evolution of the system. We shall briefly explain the focusing discipline for SELL$^{\Cap}$ but, in order to avoid technicalities, we omit the proof rules for that system. The reader may refer to [42] and [17] for a complete description of the focused SELL$^{\Cap}$ system.

Focusing [2] is a powerful discipline on proofs which can be seen as normal form proofs for proof search. In this discipline, one classifies as *negative* all formulas whose main connective is $\&, \multimap, \forall, \Cap$ and classifies the remaining formulas (both non-atomic and atomic) as *positive*. Similarly, *positive* rules are those that introduce positive formulas to the right-hand-side of sequents and negative formulas to the left-hand-side of sequents, *e.g.*, $\exists_R, \multimap_L$. *Negative* rules are those that introduce negative formulas to the right-hand-side of sequents and positive formulas to the left-hand-side of sequents, *e.g.*, $\forall_R, \otimes_L$.

This distinction between positive and negative phases is natural as all negative rules are invertible rules, that is, provability is not affected when applying such a rule. For example, the rule $\exists_L$ belongs to the negative phase, as the choice of the name used for the eigenvariable is not important for provability, as long as it is fresh. A positive rule, on the other hand, is possibly non-invertible and therefore provability may be lost. For instance, the $\exists_R$ rule belongs to the positive phase: one needs to provide a witness t for that rule. As another example, $\otimes_L$ belongs to the negative phase because this rule is invertible. On the other side, $\otimes_R$ belongs to the positive phase as this rules splits the linear context.

Let us show a simple example on how the focusing discipline allows to control the proof search procedure. Consider the following derivations:

$$\dfrac{\dfrac{}{a \longrightarrow a} \quad \dfrac{\dfrac{}{b \longrightarrow b} \quad \dfrac{}{c \longrightarrow c}}{b, b \multimap c \longrightarrow c}\multimap_L}{a, a \multimap b, b \multimap c \longrightarrow c}\multimap_L \qquad\qquad \dfrac{\dfrac{}{c \longrightarrow c} \quad \dfrac{\dfrac{}{b \longrightarrow b} \quad \dfrac{}{a \longrightarrow a}}{a, a \multimap b \longrightarrow b}\multimap_L}{a, a \multimap b, b \multimap c \longrightarrow c}\multimap_L$$

The proof on the left corresponds to a focused proof. We *choose* (i.e., we focus on) the formula $a \multimap b$. For that, a, which is an atom and hence a positive formula, must be already in the context. The same happens in the proof of the sequent $b, b \multimap c \longrightarrow c$. We choose to use $b \multimap c$ and we are forced to prove the atom b *immediately*. That is, once $\multimap_L$ is used on a formula of the shape $F \multimap G$, the focus persists on F. Moreover, if F is a positive atom, the proof of F must finish with an application of the initial rule.

The proof on the right, on the other hand, is not focused. Note that we *use* the implication $b \multimap c$ but the proof of b was delayed until b was *later* produced by $a \multimap b$. In the context of biochemical reactions, this does not correspond to what we expect: we are allowed to use a reaction whose reactants are not *yet* available but they will be *later* produced.

Now we shall state our desired result relating derivations in the logical system and steps in the biochemical system. Before that, let us introduce some needed notation.

Notation 1 (States) *We use $s_1 : [A_1 : c_1^1, ..., A_n : c_n^1], \cdots, s_m : [A_1 : c_1^m, ..., A_n : c_n^m]$ to denote a state $\mathfrak{s}$ where there are c_i^j species of the reactant j in the space domain i. If reaction r can be applied on state $\mathfrak{s}_1$ producing the state $\mathfrak{s}_2$ after k time-units, we shall write $\mathfrak{s}_1 \xrightarrow{(r,k)} \mathfrak{s}_2$. Given a state $\mathfrak{s}$ and a time-unit t, we shall denote with $[\![\mathfrak{s}]\!]_t$ the SELL$^{\Cap}$ formula $\mathtt{system}(t)$.*

Theorem 3.1 (Correspondence) *Let $\mathfrak{s}_1$ and $\mathfrak{s}_2$ be states, r a reaction and $t > 0$. Then, $\mathfrak{s}_1 \xrightarrow{(r,k)} \mathfrak{s}_2$ iff $[\![\mathfrak{s}_1]\!]_t \longrightarrow [\![\mathfrak{s}_2]\!]_{t+k}$. Moreover, such adequacy is at the level of derivations, that is, one focused logical phase corresponds exactly to the move from state $\mathfrak{s}_1$ to state $\mathfrak{s}_2$.*

Even though the detailed proof of the above theorem is in [17], let us give some intuitions about it to understand better the level of adequacy we obtained. Assume that $\mathfrak{s}_1 \xrightarrow{(r,k)} \mathfrak{s}_2$ and consider the sequent $[\![\mathfrak{s}_1]\!]_t \longrightarrow G$ where G is the property we want to verify (we shall give some examples of properties in the next section). In a focused system, the rule $!_L$ belongs to the negative phase. For that, the left-hand-side context of the sequent is organized as follows:

$$[a_1 : \Delta_1; \cdots ; a_n : \Delta_n], \Gamma \longrightarrow G$$

where a_i is a subexp. and Δ_i is a multisets of formulas. Intuitively, $a_i : \Delta_i$ represents the formula $!^i F_1 \otimes \cdots \otimes !^i F_m$ if $\Delta_i = \{F_1, ..., F_m\}$. Then, the rule $!_L$ can be written as

$$\frac{[a_1 : \Delta_1; \cdots ; a_i : \Delta_i \uplus \{F\}; \cdots ; a_n : \Delta_n], \Gamma \longrightarrow G}{[a_1 : \Delta_1; \cdots ; a_i : \Delta_i; \cdots ; a_n : \Delta_n], \Gamma, !^{a_i} F \longrightarrow G}\ !_L$$

i.e, the formula F is *stored* in the context a_i.

Consider the formula $[\![\mathfrak{s}_1]\!]_t = \texttt{eqs} \otimes [\texttt{state}(t)]$. In a negative phase, all the ! and $\otimes$ connectives in the formula can be eagerly introduced as follows:

$$\begin{array}{c} [t_\omega : \texttt{eqs}'; s_i.t : \{\texttt{ct}(A_1, c_1), ..., \texttt{ct}(A_n, c_n)\}] \longrightarrow G \\ \hline\hline [\![\mathfrak{s}_1]\!]_t \longrightarrow G \end{array}\ !_L, \otimes_L$$

where $\texttt{eqs}' = \Cap l_x : 1+. [\texttt{eq}_1(l_x) \& \cdots \& \texttt{eq}_k(l_x)]$ and $s_i.t$ represents the context for the formulas of the form $!^{s_i.t}F$. Note that the negative phase ends here since $\texttt{eqs}'$ is a negative formula (that must be introduced in the positive phase) and $\texttt{ct}(\cdot, \cdot)$ is an atom.

If we decide to focus on the formulas on the left, we only have one possibility: to focus on $\texttt{eqs}'$. In the following, we shall show that after the positive phase of the derivation, we end up with a formula of the shape $[\![\mathfrak{s}_2]\!]_{t+k}$.

The positive phase then begins by deciding to focus on the formula $\texttt{eqs}'$. Therefore, since $\Cap$ and & are negative connectives, the focus persists:

$$\dfrac{\dfrac{\dfrac{[t_\omega : \texttt{eqs}'; s_i.t : \{\texttt{ct}(A_1, c_1), ..., \texttt{ct}(A_n, c_n)\}], \texttt{eq}_i(t') \longrightarrow G}{[t_\omega : \texttt{eqs}'; s_i.t : \{\texttt{ct}(A_1, c_1), ..., \texttt{ct}(A_n, c_n)\}], \texttt{eq}(t') \& ... \& \texttt{eq}(t') \longrightarrow G}\ \&_L}{[t_\omega : \texttt{eqs}'; s_i.t : \{\texttt{ct}(A_1, c_1), ..., \texttt{ct}(A_n, c_n)\}], \texttt{eqs}' \longrightarrow G}\ \Cap_L}{[t_\omega : \texttt{eqs}'; s_i.t : \{\texttt{ct}(A_1, c_1), ..., \texttt{ct}(A_n, c_n)\}] \longrightarrow G}\ \mathit{decide}$$

Again, the main connective of $\texttt{eq}_i$ is negative ($\forall$, and then $\multimap$) and the focus persists:

$$\dfrac{\dfrac{\dfrac{\pi}{\Gamma_1 \longrightarrow \texttt{consume}_i(t')[\overline{k}/\overline{x}]} \quad \dfrac{\psi}{\Gamma_2, \texttt{produce}_i(t' + n)[\overline{k}/\overline{x}]}}{[t_\omega : \texttt{eqs}'; s_i.t : \{\texttt{ct}(A_1, c_1), ..., \texttt{ct}(A_n, c_n)\}], \texttt{eq}_i'(t')[\overline{k}/\overline{x}] \longrightarrow G}\ \multimap_L}{[t_\omega : \texttt{eqs}'; s_i.t : \{\texttt{ct}(A_1, c_1), ..., \texttt{ct}(A_n, c_n)\}], \texttt{eq}_i(t') \longrightarrow G}\ \forall_L$$

Here Γ_1 and Γ_2 correspond to the splitting of the context due to the rule $\multimap_L$. The interesting point here is that focusing allows us to reduce the nondeterminism on how to split such context. Note that $\texttt{consume}$ is a conjunction ($\otimes$) of formulas of the shape $!^{s_i.t'}\texttt{ct}(A_i, c_i)$ and $\otimes_R$ belongs to the positive phase. Hence, the focus persist on $\texttt{consume}_i(t')$ and the derivation π splits further the context Γ_1 to prove each of the formulas of the shape $!^{s_i.t'}\texttt{ct}(A_i, c_i)$. Consider one of such sequents, e.g.,

$\Gamma_1' \longrightarrow !^{s_i.t'}\mathtt{ct}(A_i, c_i)$. Since $s_i.t$ is not related to any other subexponential, the rule $!_R$ restricts Γ_1' to be of the shape $[s_i.t' : \Delta]$. Moreover, since $\mathtt{ct}(A_i, c_i)$ is an atom, the proof must finish with an application of the initial rule after introducing $!^{s_i.t'}$. This last step has two important consequences: first, the focusing discipline forces the rule $\Cap_L$ to chose $t' = t$, in other case, the formula $!^{s_i.t'}\mathtt{ct}(A_i, c_i)$ is not provable; second, rule $\&L$ must choose the encoding of a reaction whose reactants are already in the context.

Now let's take a look on the derivation on the right (ψ) where we are focused on $\mathtt{produce}$. We note that the main connective of such formula is $\otimes$ and then, the positive phase terminates (since $\otimes_L$ belongs to the negative phase). Derivation ψ starts then a negative phase where all the formulas representing the reactants are stored in the context as we already explained. Hence, what we observe is that the change of polarity from the positive to the negative phase corresponds exactly to the behavior of the system, i.e., $\mathfrak{s}_1 \xrightarrow{(r,k)} \mathfrak{s}_2$.

4 System Properties and Applications

This section is devoted to show some examples of properties that can be verified with our framework. Moreover, to give a more general picture of our developments, we show how to encode P-Systems [46,3] and some properties of such systems.

4.1 Properties of interest

We can prove reachability properties in our system by proving sequents of the shape $\mathtt{system}(1) \longrightarrow \Cup l : 1+ . \left[!^{a.l}\mathtt{ct}(A, n)\right]$. Such a sequent can be read as "given the initial state of the system, there exists a location (time-unit) where there are n copies of A in the space domain a". We note that rule $\Cup_R$ belongs to the positive phase. If we decide to focus on it, we need to introduce $!^{a.l}$ and the focus is lost. As a matter of fact, it must be the case that $\mathtt{ct}(A, n)$ is already in the context since $a.l$ is not related to $1+$ and then, the set of reactions cannot be used to finish the proof (see the explanation about the π derivation in the previous section). Moreover, due to Theorem 3.1, the proof of such a sequent can be directly traced to the moves the system has to perform to reach the state $\mathtt{ct}(A, n)$.

Now consider the problem of verifying whether the system reaches a stable state, i.e., a state where no rule can be applied. Detecting in a logical system that a given configuration cannot proceed is usually difficult. In our case, it would require to check that none of the $\mathtt{eq}_i$ in the formula $\mathtt{eq}_1\&...\&\mathtt{eq}_n$ can be chosen. One possible solution is to add a *dummy* formula introducing the atom $\mathtt{stable}$ as follows: consider two reactions $r_1 : [2.a]_s + [b]_s \longrightarrow [c]_s$ and $r_2 : [c]_s \longrightarrow [a]_s$ and let

$$\mathtt{eqs}_d \stackrel{\mathtt{def}}{=} !^{\omega}\left[\Cap l_x : 1+ . \left[\mathtt{eq}_1(l_x)\& \cdots \&\mathtt{eq}_k(l_x)\&\mathtt{eq}_d(l_x)\right]\right]$$

$$\mathtt{eq}_d(t) \stackrel{\mathtt{def}}{=} \left[\left[!^{s.t}\mathtt{ct}(a, \mathbf{0}) \oplus !^{s.t}\mathtt{ct}(\mathbf{a}, \mathtt{suc}(\mathbf{0})) \oplus !^{s.t}\mathtt{ct}(\mathbf{b}, \mathbf{0})\right] \& \left[!^{s.t}\mathtt{ct}(c, \mathbf{0})\right]\right] \multimap !^{t}\mathtt{stable}$$

If none of the $\mathtt{eq}_i$ succeed, then $\mathtt{eq}_d$ must be chosen. We note that rule $\&_L$ belongs to the negative phase and then, we cannot guarantee that the proof has to finish *immediately* proving from the context the atoms $\mathtt{ct}(\cdot,\cdot)$. Nevertheless, since the subexponential $s.t$ is unrelated to all other subexponentials, we do guarantee that for proving such atoms the set of reactions cannot be used again (as in the derivation π of the previous section). Intuitively, $\mathtt{eq}_d$ checks whether there are not enough reactants to trigger any of the rules. In that case, the atom $\mathtt{stable}$ is added to the context. Hence, the system reaches a stable state iff the sequent $\mathtt{system}(1) \longrightarrow \Cup l : 1 + .!^{l}\mathtt{stable}$ is provable.

4.2 Encoding P-Systems

P-Systems [46,3] is a model of computation that interprets the processes taking place in the compartmentalized structure of a biological cell as computations. The main abstraction is the notion of a cell-like membrane structure. Several membranes placed in an outermost membrane, called "*the skin membrane*", determine the configuration of the system. This structural shape defines compartments where multisets of objects (components) are placed and evolve according to a set of rules. More precisely,

Definition 4.1 [P-System] A P-System is a structure $\Pi = (V, \mu_0, R, O)$ where V is an *alphabet* of symbols; μ_0 is the *initial* configuration; O is the label of the *observable membrane*; and R is a finite set of rewriting rules of the following forms:

- *Communication:* $xx'[_i\ y'y \longrightarrow xy'[_i\ x'y$; for $x, y, x', y' \in V^*$.
- *Transformation*: $[_i\ y \longrightarrow [_i\ y'$; for $y, y' \in V^*$.

Intuitively, a transformation rule consumes the objects in the multiset y to produce the multiset y' in the membrane i. A communication rule is similar but allows us to *move* objects through membranes: the multiset x' (resp. y') is moved inside (resp. outside) the membrane i. Given two configurations (states) $\mathfrak{s}_1$ and $\mathfrak{s}_2$, we shall write $\mathfrak{s}_1 \Longrightarrow \mathfrak{s}_2$ if $\mathfrak{s}_1$ moves to $\mathfrak{s}_2$ applying the rules in R.

Communication and transformation rules can be interpreted as SELL$^{\Cap}$ formulas mostly as we did in the previous section. Figure 3 depicts the proposed encoding. Assuming a set of n different components in the system, the current state of the system in the membrane i is defined as $!^{s_i}\mathtt{p}(a_1, ..., a_n)$. Rules manipulate the state of the system by consuming elements in the current time-unit and then, producing them in the future time-unit. Hence, we shall use the formula $!^{s_i}\mathtt{f}(a_1, ..., a_n)$ to model that, in the next time-unit, there will be a_i additional species of the component A_i in the membrane s_i.

Consider the formula $[\![r_j]\!]_t$ for a given rule r_j. The first implication, that we shall call *positive rule*, is similar to the one we have in the previous section. We note that the elements are consumed in the current time-unit but the products are "stored" in the predicate $\mathtt{f}(\cdot)$. The new part is the second implication that we call the *negative rule*. Here we check whether there are not enough resources to fire the rule (as we did with the dummy formula $\mathtt{eq}_d(t)$ above). This is done by proving formulas of the shape $\mathtt{lt}(x, a)$ (i.e., $x < a$). For that, the axiom in Figure 3 is used: $!^{s_j}\mathtt{lt}(x, 3)$ can be deduced , for instance, if there are two or less resources of type x. Note also that either the positive rule or the negative one are fired but not both due to the & connective and also because they are mutually exclusive. Moreover, both rules add the formula $\mathtt{ok}$ which is needed as we explain in brief.

We recall that in P-systems, all the rules must be applied in a non-deterministic and maximally parallel manner. For that, we shall use the predicates $\mathtt{tk}$ and $\mathtt{ok}_i$ as follows. Once we focus on the formula $F = [\![r_j]\!]_t$, F is decomposed and it adds in the end of the negative phase, the formula $\mathtt{ok}_j$ into the context t. As we already explain, if F can be *positively* fired, it also changes the current state of the system by consuming and producing the corresponding $\mathtt{p}(\cdot)$ and $\mathtt{f}(\cdot)$ predicates. Note also that, unlike the encoding of the previous section, here the encoding of each rule is glued with the $\otimes$ connective (while in the previous section we used &). This allows us to fire all the rules during the current time-unit. Once all the rules are fired (either modifying the state or not) the formula $\mathtt{next}$ can be focused on to propagate the changes to the next time-unit. When this happens, we can say that the time-unit t ends and we start the computations of the time-unit $t+1$.

Theorem 4.2 (Adequacy) *Let $\mathfrak{s}_1$ and $\mathfrak{s}_2$ be states, $t > 0$ and $[\![\mathfrak{s}]\!] = \mathtt{state}(t)$ as in Figure 3. Then, $\mathfrak{s}_1 \Longrightarrow \mathfrak{s}_2$ iff $\mathtt{system}(t), [\![\mathfrak{s}_1]\!]_t \longrightarrow [\![\mathfrak{s}_2]\!]_{t+1}$*

The encoding we have here does not exhibit an adequacy at the level of derivations as in the previous section. The reason is simple. Each time we fire a rule, we change from a negative phase to a positive one. Then, applying the k rules of the system at the time-unit t requires flipping $k+1$ times the polarity of the proof (the "+1" is due to the extra phase needed to fire the implication in the formula $\mathtt{next}$). However, the focusing discipline and the subexponentials allow us to control correctly the proof. In particular, if a rule is fired, then needed reactants must be already available in the context. Moreover, if the rule cannot be applied, it must be the case that the *negative* part of the rule is applied. The reason is that the encoding does not increase the number of components in the current state. Then, if a rule cannot be applied now, it cannot be applied after executing some other rules (during the same time-unit). Hence, what we observe is that the rules are applied non-deterministically and once all of them are fired, the system moves to the next time-unit.

Besides reachability properties as those stated in the previous section, we can also check the *periodicity* of the system, i.e., whether the system exhibits the behavior $\mathfrak{s}_1 \Longrightarrow \mathfrak{s}_{x_1} \Longrightarrow \cdots \Longrightarrow \mathfrak{s}_{x_n} \Longrightarrow \mathfrak{s}_1$ where $\mathfrak{s}_{x_i}$ is different from $\mathfrak{s}_1$. This means that, after n time-units, there is a cycle in the system going back to the state $\mathfrak{s}_1$. This property holds iff the sequent $\mathtt{system}(1), [\![\mathfrak{s}_1]\!]_1 \longrightarrow [\![\mathfrak{s}_1]\!]_{n+1}$ is provable. More generally, we can find such a periodicity by using existential quantification on subexponentials, i.e., by looking at the final instantiation of the subexponential variable l in the proof of the sequent $\mathtt{system}(1), [\![\mathfrak{s}_1]\!]_1 \longrightarrow \Cup l : 2 + [\![\mathfrak{s}_1]\!]_l$.

5 Concluding Remarks

In this paper we presented a framework to specify and verify computational biochemical systems. We have dealt with the problem of representing both spatial

Predicates

- $\mathtt{p}(x_1, ..., x_n)$: currently, there are x_i species of reactant A_i.
- $\mathtt{f}(x_1, ..., x_n)$: in the next time-unit, there will be x_i species of reactant A_i
- $\mathtt{tk}$: enables the set of reaction in the current time-unit. •$\mathtt{ok}_j$: Rule j was applied.

Axioms

$\Cap s_x : s_\omega . [!^{s_x}\mathtt{p}(x_1, ..., x_i, \cdots, x_n) \multimap !^{s_x}\mathtt{p}(x_1, ..., x_n) \otimes !^{s_x}\mathtt{lt}(A_i, \mathtt{suc}(x_i))]$

Rules

- Transformation rule: $[\![r_w : [_i S \longrightarrow [_i S' \,]\!]_t =$

$\left[!^{l_i}\mathtt{f}(x_1, ..., x_n) \otimes !^{l_i}\mathtt{p}(\mathtt{suc}^{a_1}y_1, ..., \mathtt{suc}^{a_n}y_n) \multimap !^{l_i}\mathtt{f}(\mathtt{suc}^{b_1}x'_1, ..., \mathtt{suc}^{b_n}x'_n) \otimes !^{l_i}\mathtt{p}(y'_1, ..., y'_n) \otimes !^t\mathtt{ok}_w\right] \&$
$!^{l_i}(\mathtt{lt}(a_1) \oplus \cdots \oplus \mathtt{lt}(a_n)) \multimap !^t\mathtt{ok}_w$

- Communication rule: $[\![r_w : [_j T[_i S \longrightarrow [_j T'[_i S' \,]\!]_t =$

$[!^{l_i}\mathtt{f}(x_1, ..., x_n) \otimes !^{l_i}\mathtt{p}(\mathtt{suc}^{a_1}y_1, ..., \mathtt{suc}^{a_n}y_n) \otimes !^{l_j}\mathtt{f}(z_1, ..., z_n) \otimes !^{l_j}\mathtt{p}(\mathtt{suc}^{c_1}w_1, ..., \mathtt{suc}^{c_n}w_n) \multimap$
$!^{l_i}\mathtt{f}(\mathtt{suc}^{b_1}x'_1, ..., \mathtt{suc}^{b_n}x'_n) \otimes !^{l_i}\mathtt{p}(y'_1, ..., y'_n) \otimes !^{l_j}\mathtt{f}(\mathtt{suc}^{d_1}z'_1, ..., \mathtt{suc}^{d_n}z'_n) \otimes !^{l_j}\mathtt{p}(w'_1, ..., w'_n) \otimes !^t\mathtt{ok}_w]\&$
$\left[!^{l_i}(\mathtt{lt}(a_1) \oplus \cdots \oplus \mathtt{lt}(a_n))\&!^{l_j}(\mathtt{lt}(c_1) \oplus \cdots \oplus \mathtt{lt}(c_n))\right] \multimap !^t\mathtt{ok}_w$

Auxiliary Formulas

$$\mathtt{next}(t) \stackrel{\mathtt{def}}{=} !^t\mathtt{ok}_1 \otimes \cdots \otimes !^t\mathtt{ok}_m \otimes \bigotimes_{s_i \in \mathcal{L}} (!^{s_i}\mathtt{p}(xi_1, ..., xi_n) \otimes !^{s_i}\mathtt{f}(yi_1, ..., yi_n)) \multimap$$
$$!^{t+1}\mathtt{tk} \otimes \bigotimes_{s_i \in \mathcal{L}} (!^{s_i}\mathtt{p}(xi_1 + yi_1, ..., xi_n + yi_1) \otimes !^{s_i}\mathtt{f}(\mathbf{0}, ..., \mathbf{0}))$$
$$\mathtt{system}(t) \stackrel{\mathtt{def}}{=} !^{t_\omega}\left[\Cap t_x : 1 + . \left[\mathtt{next}(t_x) \otimes \left(!^{t_x}\mathtt{tk} \multimap \bigotimes_{r_j \in \mathcal{R}} [\![r_j]\!]_{t_x}\right)\right]\right]$$
$$\mathtt{state}(t) \stackrel{\mathtt{def}}{=} !^t\mathtt{tk} \otimes \bigotimes_{s_i \in \mathcal{L}} !^{s_i}\mathtt{p}(ai_1, ..., ai_n) \otimes !^{s_i}\mathtt{f}(\mathbf{0}, ..., \mathbf{0})$$

Fig. 3. Encoding of P-systems into SELL$^\Cap$. We assume a set of reactants $\mathcal{A} = \{A_1, ..., A_n\}$. All the variables in axioms and in the encoding of rules ($\boldsymbol{x}$ and $\boldsymbol{y}$) are implicitly universally quantified. $\mathcal{R}$ denotes the set of rules in the system.

and time dependent partial information. Our framework relies on linear logic [30] with subexponentials (SELL) [23]. We have shown that our framework is general enough to give a declarative semantics to P-Systems and we have shown that such a logical characterization has a strong level of adequacy: derivations in the logical system follow exactly the rules (reactions) of the modeled system. This embedding shows that SELL$^\Cap$ is expressive enough to give a logical interpretation to such systems, thus opening the possibility to use all the meta theory of LL to reason about the behavior of biochemical systems that exhibit temporal and spatial modalities. The next step will be implementing our framework in a functional logic language (e.g., lambda Prolog) and use a framework for assisted theorem proving like Coq for making semi automatic the process of verification (see e.g, [13]).

Related work. A close related work to ours is [13] and [7]. In [13], the authors show that temporal properties of biochemical systems can be expressed in HyLL (hybrid linear logic). Basically, in HyLL, a partial order structure on *words* is defined to model *locations* where the formulas hold. This is much like the subexponential structure in SELL$^\Cap$. Here we show that time and spatial modalities can be considered in the same framework. Another difference is that the work in [13] considers gene networks where signals can be activated or not. Here, inspired in our previous works on modeling biological systems [16,15,14] in CCP, we consider quantitative information of the system, i.e., the concentration of the molecules in each state. This is also akin to the work in [7], where variants of CCP are considered in order to specify biological systems.

Nowadays, formal methods have been extensively used for the analysis, simulation and verification of biochemical systems at different levels of abstraction. Essentially, these methods aim at reasoning about molecular interactions (i.e., reactions rules) and spatial information (i.e., compartmentalization and local interactions). Iterative application of rules to a set of seed species may be used to generate a network by using an ordinary differential equations (ODEs) semantics on molecular concentrations (i.e., by numerically solving ODEs) or a stochastic semantics on number of molecules (i.e., by implementing the Gillespie stochastic simulation algorithm). Generally, the timing behavior is tackled with kinetic expressions in the reaction scheme. In [28,19,26,45,8,48,24,25] models of biochemical systems can be associated with a continuous-time Markov chain (CTMC) process or an ODE process. While in Pathway Logic (PL) [1] each rule is associated with a scalar value called affinity. This one can be bounded with a time-dependent interpretation either by using exponential random or deterministic amortized variables. On the other hand, models of biochemical systems in timed-π-calculus [49] can deal with time-stamps and clocks handled as other names and transmit them through channels.

So as to deal with spatial information, formalisms such as Bio-Pepa [20], BioNetGen [35], BioAmbients [47], and Brane Calculi [10] have been equipped with a tree representation of the hierarchical structure of cellular compartments. Whereas in Biocham [11,18], PL [51], and Beta-Binders [34], cellular compartments can be abstracted as symbolic locations by assigning *labels* to molecular compounds. In the π@-calculus [53], restricted names are exploited to model compartments.

The above mentioned frameworks allow to reason about biological properties by using different types of logics and techniques. Properties of Biocham models [28], for instance, can be formalized within the boolean, differential and stochastic semantics by using (probabilistic) temporal logics. Bio-Pepa's models [20] can be translated into PRISM [37], a probabilistic model checker. Processes in PL [1] can be analyzed by using the Maude system [22]. Bounded Linear Temporal Logic [31] and statistical model checking are used in BioNetGen to express and to verify system properties. Temporal properties for BioAmbients processes can be analyzed by using state formulas [32] or modal logics to express spatial and time modalities [9]. Similarly, modal logic can be used to express spatial and temporal properties over membranes and systems, which is known as Brane Logic [40]. In the case of Beta-Binders models, causality properties [33] as well as flow control analyses [6] can be performed.

References

[1] Alessandro Abate, Yu Bai, Nathalie Sznajder, Carolyn L. Talcott, and Ashish Tiwari. Quantitative and probabilistic modeling in pathway logic. In *Proceedings of BIBE 2007*, pages 922–929. IEEE, 2007.

[2] Jean-Marc Andreoli. Logic programming with focusing proofs in linear logic. *J. Log. Comput.*, 2(3):297–347, 1992.

[3] Francesco Bernardini and Vincenzo Manca. Dynamical aspects of p systems. *Biosystems*, 70(2):85 – 93, 2003. Cell Computing.

[4] Marco Bernardo, Pierpaolo Degano, and Gianluigi Zavattaro, editors. *SFM 2008*, volume 5016 of *LNCS*. Springer, 2008.

[5] Mitali Bhattacharjee, Rajesh Raju, Aneesha Radhakrishnan, and et al. A bioinformatics resource for tweak-fn14 signaling pathway. *Journal of Signal Transduction*, 2012, 2012.

[6] Chiara Bodei. A control flow analysis for beta-binders with and without static compartments. *Theor. Comput. Sci.*, 410(33-34):3110–3127, 2009.

[7] Luca Bortolussi and Alberto Policriti. Modeling biological systems in stochastic concurrent constraint programming. *Constraints*, 13(1-2):66–90, 2008.

[8] Linda Brodo, Pierpaolo Degano, and Corrado Priami. A stochastic semantics for bioambients. In Victor Malyshkin, editor, *Parallel Computing Technologies*, volume 4671 of *LNCS*, pages 22–34. Springer Berlin Heidelberg, 2007.

[9] Luís Caires and Luca Cardelli. A spatial logic for concurrency (part I). *Inf. Comput.*, 186(2):194–235, 2003.

[10] Luca Cardelli. Brane calculi. In Vincent Danos and Vincent Schachter, editors, *CMSB*, volume 3082 of *LNCS*, pages 257–278. Springer Berlin Heidelberg, 2005.

[11] Nathalie Chabrier-Rivier, François Fages, and Sylvain Soliman. The biochemical abstract machine BIOCHAM. In Vincent Danos and Vincent Schächter, editors, *CMSB 2004*, volume 3082 of *LNCS*, pages 172–191. Springer, 2004.

[12] Claudine Chaouiya. Petri net modelling of biological networks. *Brief in Bioinformatics*, 8(4):210–219, 2007.

[13] Kaustuv Chaudhuri and Joelle Despeyroux. A Hybrid Linear Logic for Constrained Transition Systems with Applications to Molecular Biology. Research Report, 2009.

[14] D. Chiarugi, M. Falaschi, D. Hermith, and C. Olarte. A framework for modelling spatially dependent interactions of biological systems in ccp. In *Proc. of IWBBIO'14*, To appear, 2014.

[15] Davide Chiarugi, Moreno Falaschi, Diana Hermith, Michell Guzman, and Carlos Olarte. Simulating signalling pathways with bioways. *Electr. Notes Theor. Comput. Sci.*, 293:17–34, 2013.

[16] Davide Chiarugi, Moreno Falaschi, Carlos Olarte, and Catuscia Palamidessi. Compositional modelling of signalling pathways in timed concurrent constraint programming. In Aidong Zhang, Mark Borodovsky, Gultekin Özsoyoglu, and Armin R. Mikler, editors, *Proceedings of BCB 2010*, pages 414–417. ACM, 2010.

[17] Davide Chiarugi, Diana Hermith, Moreno Falaschi, and Carlos Olarte. Verification of spatial and temporal modalities in biochemical systems. Technical report, ECT, Universidade Federal do Rio Grande do Norte, 2014. Available at https://sites.google.com/site/carlosolarte/.

[18] Andrea Ciliberto, Béla Novak, and John J. Tyson. Steady states and oscillations in the p53/mdm2 network. *Cell Cycle*, 4(3):488–493, 2005.

[19] Federica Ciocchetta, Andrea Degasperi, JohnK. Heath, and Jane Hillston. Modelling and analysis of the nf-?b pathway in bio-pepa. In Corrado Priami, Rainer Breitling, David Gilbert, Monika Heiner, and AdelindeM. Uhrmacher, editors, *Transactions on Computational Systems Biology XII*, volume 5945 of *LNCS*, pages 229–262. Springer Berlin Heidelberg, 2010.

[20] Federica Ciocchetta and Maria Luisa Guerriero. Modelling biological compartments in bio-pepa. *Electronic Notes in Theoretical Computer Science*, 227(0):77 – 95, 2009.

[21] Federica Ciocchetta and Jane Hillston. Bio-pepa: An extension of the process algebra $\{PEPA\}$ for biochemical networks. *Electronic Notes in Theoretical Computer Science*, 194(3):103 – 117, 2008.

[22] Manuel Clavel, Francisco Durán, Steven Eker, Patrick Lincoln, Narciso Martí-Oliet, José Meseguer, and Jose F. Quesada. The maude system. In Paliath Narendran and Michaël Rusinowitch, editors, *Proc. of RTA-99*, volume 1631 of *LNCS*, pages 240–243. Springer, 1999.

[23] Vincent Danos, Jean-Baptiste Joinet, and Harold Schellinx. The structure of exponentials: Uncovering the dynamics of linear logic proofs. In Georg Gottlob, Alexander Leitsch, and Daniele Mundici, editors, *Kurt Gödel Colloquium*, volume 713 of *LNCS*, pages 159–171. Springer, 1993.

[24] Pierpaolo Degano, Davide Prandi, Corrado Priami, and Paola Quaglia. Beta-binders for biological quantitative experiments. *Electr. Notes Theor. Comput. Sci.*, 164(3):101–117, 2006.

[25] Lorenzo Dematté, Corrado Priami, and Alessandro Romanel. The blenx language: A tutorial. In Marco Bernardo, Pierpaolo Degano, and Gianluigi Zavattaro, editors, *Formal Methods for Computational Systems Biology*, volume 5016 of *LNCS*, pages 313–365. Springer Berlin Heidelberg, 2008.

[26] James R. Faeder, MichaelL. Blinov, and William S. Hlavacek. Rule-based modeling of biochemical systems with bionetgen. In Ivan V. Maly, editor, *Systems Biology*, volume 500 of *Methods in Molecular Biology*, pages 113–167. Humana Press, 2009.

[27] François Fages, Paul Ruet, and Sylvain Soliman. Linear concurrent constraint programming: Operational and phase semantics. *Inf. Comput.*, 165(1):14–41, 2001.

[28] François Fages and Sylvain Soliman. Formal cell biology in biocham. In Bernardo et al. [4], pages 54–80.

[29] Jasmin Fisher and Thomas A. Henzinger. Executable biology. In Perrone et al. [44], pages 1675–1682.

[30] Jean-Yves Girard. Linear logic. *Theor. Comput. Sci.*, 50:1–102, 1987.

[31] Haijun Gong, Paolo Zuliani, Anvesh Komuravelli, James R. Faeder, and Edmund M. Clarke. Computational modeling and verification of signaling pathways in cancer. In *ANB*, pages 117–135, 2010.

[32] Roberta Gori and Francesca Levi. An analysis for proving temporal properties of biological systems. In Naoki Kobayashi, editor, *Proc. of APLAS 2006*, volume 4279 of *LNCS*, pages 234–252. Springer, 2006.

[33] Maria Luisa Guerriero and Corrado Priami. Causality and concurrency in beta-binders. Technical report, University of Trento, 2006.

[34] Maria Luisa Guerriero, Corrado Priami, and Alessandro Romanel. Modeling static biological compartments with beta-binders. In Hirokazu Anai, Katsuhisa Horimoto, and Temur Kutsia, editors, *Proc. of AB 2007*, volume 4545 of *LNCS*, pages 247–261. Springer, 2007.

[35] Leonard A. Harris, Justin S. Hogg, and James R. Faeder. Compartmental rule-based modeling of biochemical systems. In Ann Dunkin, Ricki G. Ingalls, Enver Yücesan, Manuel D. Rossetti, Ray Hill, and Björn Johansson, editors, *Proc. of WSC 2009*, pages 908–919. WSC, 2009.

[36] Diana Hermith, Carlos Olarte, Camilo Rueda, and Frank D. Valencia. Modeling cellular signaling systems: An abstraction-refinement approach. In Miguel P. Rocha, Juan M. Corchado Rodríguez, Florentino Fdez-Riverola, and Alfonso Valencia, editors, *Proc. of PACBB 2011*, volume 93 of *Advances in Intelligent and Soft Computing*, pages 321–328. Springer, 2011.

[37] Andrew Hinton, Marta Kwiatkowska, Gethin Norman, and David Parker. Prism: A tool for automatic verification of probabilistic systems. In Holger Hermanns and Jens Palsberg, editors, *TACAS 2006*, volume 3920 of *LNCS*, pages 441–444. Springer Berlin Heidelberg, 2006.

[38] Patrick Lincoln and Ashish Tiwari. Symbolic systems biology: Hybrid modeling and analysis of biological networks. In Rajeev Alur and George J. Pappas, editors, *Proc. of HSCC 2004*, volume 2993 of *LNCS*, pages 660–672. Springer, 2004.

[39] Nagasaki M, Doi A, Matsuno H, and Miyano S. A versatile petri net based architecture for modeling and simulation of complex biological processes. *Genome Inform.*, 15(1):180–197, 2004.

[40] Marino Miculan and Giorgio Bacci. Modal logics for brane calculus. In Corrado Priami, editor, *CMSB 2006*, volume 4210 of *LNCS*, pages 1–16. Springer Berlin Heidelberg, 2006.

[41] M. Nielsen, C. Palamidessi, and F. Valencia. Temporal concurrent constraint programming: Denotation, logic and applications. *Nordic Journal of Computing*, 9(1):145–188, 2002.

[42] Vivek Nigam, Carlos Olarte, and Elaine Pimentel. A general proof system for modalities in concurrent constraint programming. In Pedro R. D'Argenio and Hernán C. Melgratti, editors, *CONCUR*, volume 8052 of *LNCS*, pages 410–424. Springer, 2013.

[43] Carlos Olarte, Vivek Nigam, and Elaine Pimentel. Dynamic spaces in concurrent constraint programming. *Electr. Notes Theor. Comput. Sci.*, 305:103–121, 2014.

[44] L. Felipe Perrone, Barry Lawson, Jason Liu, and Frederick P. Wieland, editors. *Proc. of WSC 2006.*

[45] Andrew Phillips and Luca Cardelli. Efficient, correct simulation of biological processes in the stochastic pi-calculus. In Muffy Calder and Stephen Gilmore, editors, *CMSB*, volume 4695 of *LNCS*, pages 184–199. Springer Berlin Heidelberg, 2007.

[46] Gheorghe Păun. From cells to computers: computing with membranes (p systems). *Biosystems*, 59(3):139–158, 2001.

[47] Aviv Regev, Ekaterina M. Panina, William Silverman, Luca Cardelli, and Ehud Shapiro. Bioambients: an abstraction for biological compartments. *Theoretical Computer Science*, 325(1):141 – 167, 2004.

[48] Jorma J. De Ronde and Cecile Petmi Ndjehan. Modelling networks and pathways in systems biology. Technical report, Dublin City University, School of Computing, 2005/2006.

[49] Neda Saeedloei and Gopal Gupta. Timed π-calculus. In Martín Abadi and Alberto Lluch-Lafuente, editors, *TGC 2013*, volume 8358 of *LNCS*, pages 119–135. Springer, 2013.

[50] Vijay A. Saraswat, Martin C. Rinard, and Prakash Panangaden. Semantic foundations of concurrent constraint programming. In David S. Wise, editor, *POPL*, pages 333–352. ACM Press, 1991.

[51] Carolyn Talcott. Pathway logic. In Marco Bernardo, Pierpaolo Degano, and Gianluigi Zavattaro, editors, *Formal Methods for Computational Systems Biology*, volume 5016 of *LNCS*, pages 21–53. Springer Berlin Heidelberg, 2008.

[52] Carolyn L. Talcott. Symbolic modeling of signal transduction in pathway logic. In Perrone et al. [44], pages 1656–1665.

[53] Cristian Versari and Roberto Gorrieri. pi@: A pi-based process calculus for the implementation of compartmentalised bio-inspired calculi. In Bernardo et al. [4], pages 449–506.