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...
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
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\( _\Sigma \) 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:

\[
\frac{\Gamma \rightarrow G}{\Gamma, !bF \rightarrow G} \quad \frac{\Gamma, !bF \rightarrow G}{\Gamma, !bF \rightarrow G} \]

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:

\[
\frac{\Gamma, F \rightarrow G}{\Gamma, !aF \rightarrow G} \quad \frac{\Gamma, F \rightarrow G}{\Gamma, !aF \rightarrow G} \]

\[
\frac{\Gamma, F_1, \ldots, !a_nF_n \rightarrow F}{\Gamma, !aF \rightarrow F} \quad \frac{\Gamma, F_1, \ldots, !a_nF_n \rightarrow F}{\Gamma, !aF \rightarrow F} \]

provided \( a \preceq a_i \) for \( 1 \leq i \leq n \).

Observe that provability is preserved downwards: if the sequent \( \Gamma \rightarrow !aP \) is provable in SELL\( _\Sigma \), then so is the sequent \( \Gamma \rightarrow !bP \) for all \( b \preceq a \). We shall omit the signature \( \Sigma \) 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 (\( \bowtie \)) and existential (\( \bowcup \)) quantification on subexponentials are introduced with the following proof rules:

\[
\frac{\Gamma, P[l/l_x] \rightarrow G}{\Gamma, \bowtimes x : a.P \rightarrow G} \quad \frac{\Gamma, \bowtimes x : a.G \rightarrow G}{\Gamma, P[l/l_x] \rightarrow G} \quad \frac{\Gamma, P[l/l_x] \rightarrow G}{\Gamma, \bowtimes x : a.P \rightarrow G} \quad \frac{\Gamma, \bowtimes x : a.G \rightarrow G}{\Gamma, P[l/l_x] \rightarrow G}
\]

In these rules, \( l_c \) 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 SELL\( ^{\bowtimes} \), enjoys good proof theoretic properties: [23] (resp. [42]) proved that SELL (resp. SELL\( ^{\bowtimes} \)) 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 SELL\( ^{\bowtimes} \)

In this section we show how spatial and temporal dependencies in biochemical reactions can be neatly modeled as formulas in SELL\( ^{\bowtimes} \). 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...
We start by describing the kind of reactions we shall consider. We assume a set of reactions of the shape:

\[ r_j : [c_1.A_1]_{a_1} + \cdots + [c_n.A_n]_{a_n} \rightarrow^k [d_1.B_1]_{b_1} + \cdots + [d_m.B_m]_{b_m} \]  

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, we need first to define a subexponential signature as the one depicted in Figure 2. The only unbounded subexponentials are \( t_\omega \) (time) and \( s_\omega \) (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 \( \text{ct}(\cdot, \cdot) \). Intuitively, the formula \( !^s t \text{ct}(A, c) \) means that the concentration of \( A \) in the space domain \( s_t \) is \( c \) during the second time-unit. As usual, \( c \) is defined as the \( n \)-th application of the successor function \( \text{suc} \) to the constant \( 0 \). We shall use \( \text{suc}^n(x) \) to denote the \( n \)-th application of \( \text{suc} \) 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

\[
\text{state}(t) \overset{\text{def}}{=} \bigotimes_{s \in S} \bigotimes_{A_i \in A} !^s t [\text{ct}(A_i, c_i)]
\]

where \( A \) denotes the set of reactants and \( S \) the set of domain spaces. If there are no species of kind \( A_j \) in the space \( s_k \), then \( c_j = 0 \). Intuitively, \( !^s t [\text{ct}(A, c)] \) means that the formula \( \text{ct}(A, c) \) 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:

\[
\text{eqs} \overset{\text{def}}{=} !^t \omega [\exists l_x : 1 + . [\text{eq}_1(l_x) \& \cdots \& \text{eq}_k(l_x)]]
\]

The unbounded subexponential \( !^t \omega \) allows us to use the set of reactions as many times as needed. The universal quantification \( \exists l_x : +1 \) 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:
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 $s_\omega$ and $t_\omega$.

\[
\begin{align*}
\text{eq}(t) & \overset{\text{def}}{=} \forall \bar{x}. [(\text{consume}(t) \land \text{produce}(t+k)] \\
\text{consume}(t) & \overset{\text{def}}{=} \bigotimes_{s \in S} \bigotimes_{A_i \in A} \left[ct(A_i, N_i)\right] \\
\text{produce}(t) & \overset{\text{def}}{=} \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \neg \n
\end{align*}
\]

where $\bar{x} = x_1, \ldots, x_n$ and

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

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

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

The quantifier $\forall \bar{x}$ allows us to bind the current number of reactants in the system. The formula $\text{consume}$ consumes the needed reactants and $\text{produce}$ 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:

\[
\text{system}(t) \overset{\text{def}}{=} \text{eqs} \otimes \text{state}(t)
\]

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 SELLe 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 SELLe 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 $\&$, $\neg$, $\forall$, $\exists$, $\oplus$ 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., $\exists R, \neg L$. 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 rule $\exists R$ 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:

The proof on the left corresponds to a focused proof. We choose (i.e., we focus on) the formula $a \rightarrow b$. 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 $b, b \rightarrow c \rightarrow c$. We choose to use $b \rightarrow c$ and we are forced to prove the atom $b$ immediately. That is, once $\otimes L$ is used on a formula of the shape $F \rightarrow G$, 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 $b \rightarrow c$ but the proof of $b$ was delayed until $b$ was later produced by $a \rightarrow b$. 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 $s_1 : [A_1 : c_1^1, ..., A_n : c_n^1]$, $s_m : [A_1 : c_1^m, ..., A_n : c_n^m]$ to denote a state $s$ where there are $c_j^i$ species of the reactant $j$ in the space domain $i$. If reaction $r$ can be applied on state $s_1$ producing the state $s_2$ after $k$ time-units, we shall write $s_1 \xrightarrow{(r,k)} s_2$. Given a state $s$ and a time-unit $t$, we shall denote with $[s]_t$ the SELL$_i^m$ formula system$(t)$.

**Theorem 3.1 (Correspondence)** Let $s_1$ and $s_2$ be states, $r$ a reaction and $t > 0$. Then, $s_1 \xrightarrow{(r,k)} s_2$ iff $[s_1]_t \rightarrow [s_2]_{t+k}$. Moreover, such adequacy is at the level of derivations, that is, one focused logical phase corresponds exactly to the move from state $s_1$ to state $s_2$.

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 $s_1 \xrightarrow{(r,k)} s_2$ and consider the sequent $[s_1]_t \rightarrow G$ 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:

$$[a_1 : \Delta_1; \ldots ; a_n : \Delta_n], \Gamma \rightarrow G$$

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

$$[a_1 : \Delta_1; \ldots ; a_i : \Delta_i \cup \{ F \}; \ldots ; a_n : \Delta_n], \Gamma \rightarrow G$$

$$[a_1 : \Delta_1; \ldots ; a_i : \Delta_i; \ldots ; a_n : \Delta_n], \Gamma, !^\alpha F \rightarrow G \quad !_L$$

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

Consider the formula $[s_1]_t = \text{eqs} \otimes [\text{state}(t)]$. In a negative phase, all the $!$ and $\otimes$ connectives in the formula can be eagerly introduced as follows:

$$[t_w : \text{eqs}' ; s_i , t : \{ \text{ct}(A_1, c_1), \ldots , \text{ct}(A_n, c_n) \}] \rightarrow G$$

$$[s_1]_t \rightarrow G \quad !_L, \otimes_L$$

where $\text{eqs}' = \otimes_{l_x} : 1 + [\text{eq}_1(l_x) \& \cdots \& \text{eq}_k(l_x)]$ 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 $\text{eqs}'$ is a negative formula (that must be introduced in the positive phase) and $\text{ct}(\cdot, \cdot)$ is an atom.

If we decide to focus on the formulas on the left, we only have one possibility: to focus on $\text{eqs}'$. In the following, we shall show that after the positive phase of the derivation, we end up with a formula of the shape $[s_2]_{t+k}$.

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

$$[t_w : \text{eqs}' ; s_i , t : \{ \text{ct}(A_1, c_1), \ldots , \text{ct}(A_n, c_n) \}], \text{eq}^t \rightarrow G$$

$$[t_w : \text{eqs}' ; s_i , t : \{ \text{ct}(A_1, c_1), \ldots , \text{ct}(A_n, c_n) \}], \text{eq}^t \& \cdots \& \text{eq}^t \rightarrow G \quad \&_L$$

$$[t_w : \text{eqs}' ; s_i , t : \{ \text{ct}(A_1, c_1), \ldots , \text{ct}(A_n, c_n) \}], \text{eqs}' \rightarrow G \quad \otimes_L$$

$$[t_w : \text{eqs}' ; s_i , t : \{ \text{ct}(A_1, c_1), \ldots , \text{ct}(A_n, c_n) \}], \text{eq}^t \rightarrow G \quad \otimes_L$$

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

$$\Gamma_1 \rightarrow \text{consume}_{i}(t')_{\overline{F}/\overline{\sigma}} \quad \Gamma_2 \rightarrow \text{produce}_{i}(t' + n)_{\overline{F}/\overline{\sigma}}$$

$$[t_w : \text{eqs}' ; s_i , t : \{ \text{ct}(A_1, c_1), \ldots , \text{ct}(A_n, c_n) \}], \text{eq}^t(\overline{F}/\overline{\sigma}) \rightarrow G \quad \forall_L$$

$$[t_w : \text{eqs}' ; s_i , t : \{ \text{ct}(A_1, c_1), \ldots , \text{ct}(A_n, c_n) \}], \text{eq}^t(\overline{F}/\overline{\sigma}) \rightarrow G \quad \forall_L$$

Here $\Gamma_1$ and $\Gamma_2$ correspond to the splitting of the context due to the rule $\neg \sigma_L$. The interesting point here is that focusing allows us to reduce the nondeterminism on how to split such context. Note that $\text{consume}$ is a conjunction ($\otimes$) of formulas of the shape $!^s_{i,t} \text{ct}(A_1, c_1)$ and $\otimes_R$ belongs to the positive phase. Hence, the focus persist on $\text{consume}_{i}(t')$ and the derivation $\pi$ splits further the context $\Gamma_1$ to prove each of the formulas of the shape $!^s_{i,t'} \text{ct}(A_1, c_1)$. Consider one of such sequents, e.g.,
Since $s_i.t$ is not related to any other subexponential, the rule $!R$ restricts $\Gamma_1'$ to be of the shape $[s_i.t' : \Delta]$. Moreover, since $\text{ct}(A_i, c_i)$ 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 $\bowtie R$ to chose $t' = t$; in other case, the formula $!s_i.t' \text{ct}(A_i, c_i)$ 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 ($\psi$) where we are focused on 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 $\psi$ 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., $s_1 \overrightarrow{(r,k)} s_2$.

## 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 $\text{system}(1) \rightarrow \psi_1 : 1 + \lceil !a.l \text{ct}(A,n) \rceil$. 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 $\text{ct}(A,n)$ 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 $\pi$ 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 $\text{ct}(A,n)$.

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 $\text{eq}_i$ in the formula $\text{eq}_1 \& \cdots \& \text{eq}_n$ can be chosen. One possible solution is to add a dummy formula introducing the atom $\text{stable}$ as follows: consider two reactions $r_1 : [2.a]_s + [b]_s \rightarrow [c]_s$ and $r_2 : [c]_s \rightarrow [a]_s$ and let

$$
\begin{align*}
\text{eqs}_d & \triangleq \lceil !w[\text{loc} : 1 + \lceil \text{eq}_1(l_x) \& \cdots \& \text{eq}_k(l_x) \& \text{eq}_d(l_x) \rceil] 
\text{eq}_d(t) & \triangleq \lceil ![s.t \text{ct}(a,0) \oplus !s.t \text{ct}(a,\text{suc}(0)) \oplus !s.t \text{ct}(b,0)] \& [!s.t \text{ct}(c,0)] \rceil \rightarrow !\text{stable}
\end{align*}
$$

If none of the $\text{eq}_i$ succeed, then $\text{eq}_d(t)$ 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 $\text{ct}(\cdot,\cdot)$. 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 $\pi$ of the previous section). Intuitively, $\text{eq}_d$ checks whether there are not enough reactants to trigger any of the rules. In that case, the atom $\text{stable}$ is added to the context. Hence, the system reaches a stable state iff the sequent $\text{system}(1) \rightarrow \psi! : 1 + .!\text{stable}$ 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; $\mu_0$ 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:** $xx'_i | y'y \rightarrow xy'_i | x' y$; for $x, y, x', y' \in V^*$.
- **Transformation:** $[i | y \rightarrow [i | y'$; 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) $s_1$ and $s_2$, we shall write $s_1 \Rightarrow s_2$ if $s_1$ moves to $s_2$ applying the rules in $R$.

Communication and transformation rules can be interpreted as SELL$^0$ 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 $1^i \hat{\text{p}}(a_1, \ldots, a_n)$. 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 $!1^i \hat{\text{f}}(a_1, \ldots, a_n)$ 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 $\hat{\text{f}}(\cdot)$. 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 $\text{eq}_d(t)$ above). This is done by proving formulas of the shape $1t(x, a)$ (i.e., $x < a$). For that, the axiom in Figure 3 is used: $!1^i 1t(x, 3)$ 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 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 tk and ok 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 ok 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 $p(\cdot)$ and $f(\cdot)$ 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 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 $s_1$ and $s_2$ be states, $t > 0$ and $[s] = state(t)$ as in Figure 3. Then, $s_1 \rightarrow s_2$ iff system($t$), $[s_1]_t \rightarrow [s_2]_{t+1}$

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 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 $s_1 \rightarrow s_{x_1} \rightarrow \cdots \rightarrow s_{x_n} \rightarrow s_1$ where $s_{x_i}$ is different from $s_1$. This means that, after $n$ time-units, there is a cycle in the system going back to the state $s_1$. This property holds if the sequent system(1), $[s_1]_1 \rightarrow [s_1]_{n+1}$ 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 system(1), $[s_1]_1 \rightarrow \forall l : 2 + [s_1]_t$.

### 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
A close related work to ours is [13] and [7]. In [13], the authors describe the behavior of biochemical systems that exhibit temporal and spatial modalities. They embed the set of rules in the system follow exactly the rules (reactions) of the modeled system. This embedding is expressive 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 can easily express temporal and spatial properties.

Axioms

\[ \vdash_{s} x : s.w \quad \vdash_{s} p(x_{1}, ..., x_{n}) \rightarrow \vdash_{s} p(x_{1}, ..., x_{n}) \otimes \vdash_{s} \text{lt}(A_{i}, \text{suc}(x_{i})) \]

Rules

- Transformation rule: \[ \vdash_{s} S : \vdash_{s} S' \quad \vdash_{s} t = \]
  \[ \vdash_{s} f(x_{1}, ..., x_{n}) \otimes \vdash_{s} f(y_{1}, ..., y_{n}) \rightarrow \vdash_{s} f(y_{1}, ..., y_{n}) \otimes \vdash_{s} \text{ok}_{w} \]
- Communication rule: \[ \vdash_{s} r_{w} : [r_{w} : [i, S \rightarrow [i, S']]_{t} = \]
  \[ \vdash_{s} r_{w} : [i, S \rightarrow [i, S']]_{t} = \]

Auxiliary Formulas

\[ \text{next}(l) \equiv \vdash_{t_{w}} \text{ok}_{1} \otimes \cdots \vdash_{t_{w}} \text{ok}_{m} \otimes \bigotimes_{s \in S} \vdash_{s} \text{p}(x_{1}, ..., x_{n}) \otimes \vdash_{s} \text{f}(y_{1}, ..., y_{n}) \rightarrow \vdash_{s} \text{f}(y_{1}, ..., y_{n}) \]

\[ \text{system}(l) \equiv \vdash_{t_{w}} \left[ \bigotimes_{s \in S} \vdash_{s} \text{p}(x_{1}, ..., x_{n}) \right] \]

\[ \text{state}(l) \equiv \vdash_{t_{w}} \bigotimes_{s \in S} \vdash_{s} \text{p}(a_{1}, ..., a_{n}) \otimes \vdash_{s} \text{f}(0, ..., 0) \]

Fig. 3. Encoding of P-systems into SELL\textsuperscript{m}. We assume a set of reactants \( A = \{A_{1}, ..., A_{n}\} \). All the variables in axioms and in the encoding of rules (\( x \) and \( y \)) are implicitly universally quantified. \( R \) 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\textsuperscript{m} 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 \textit{locations} where the formulas hold. This is much like the subexponential structure in SELL\textsuperscript{m}. 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-$\pi$-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], BioNet-Gen [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 \textit{labels} to molecular compounds. In the $\pi@$-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


[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.