Theoretical Computer Science 641 (2016) 25–42

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

A proof theoretic view of spatial and temporal dependencies in biochemical systems

C. Olarte [a], D. Chiarugi [b], M. Falaschi [c,*], D. Hermith [c]

[a] ECT, Universidade Federal do Rio Grande do Norte, Brazil
[b] Max Planck Institute of Colloids and Interfaces, Potsdam-Golm, Germany
[c] Dept. of Information Engineering and Mathematics, Via Roma 56, Siena, Italy

ARTICLE INFO

Article history:
Received 6 November 2015
Received in revised form 17 March 2016
Accepted 28 March 2016
Available online 4 April 2016
Communicated by I. Petre

Keywords:
Biochemical systems
Linear logic
Spatial and temporal modalities
Verification of P systems

ABSTRACT

The behavior of biochemical systems such as metabolic and signaling pathways may depend on either the location of the reactants or on the time needed for a reaction to occur. In this paper we propose a formalism for specifying and verifying properties of biochemical systems that combines, coherently, temporal and spatial modalities. To this aim, we consider 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. We illustrate our approach by specifying some well-known biological systems and verifying properties of them. Moreover, we show that our framework is general enough to give a logic-based semantics to P systems. We show that the proposed logical characterizations have a strong level of adequacy. Hence, derivations in SELL follow exactly the behavior of the modeled system.

1. Introduction

In the last decade, the impressive enhancement of experimental techniques in biology has made available a huge amount of information concerning living organisms. In this way, the knowledge about the components of biological systems is becoming increasingly detailed and accurate. Nevertheless, determining how these components interact in living entities is a task that is still beyond the reach of the current laboratory methodologies. Understanding these interactions in the context of biological networks such as, e.g., cellular signaling pathways, is a relevant problem in biology.

Various approaches based on computer science have proven to be useful for addressing these issues. Formal models, for instance, allow us to make precise statements about the properties of biological systems, classifying them and, possibly, deducing other properties which are hard to discover by intuition or experimentally. It is worth noticing that, in general, the features of biochemical systems are often expressed informally, thus making it difficult or impossible to reason about them.

Several frameworks have been used for modeling various aspects of biological systems (see e.g., [1–5]). The characteristics of each model are mainly shaped by the features of the formalism it relies upon. Indeed, for a given model, some properties can be straightforwardly expressed while others must be abstracted away, due to the inherent limitations of the used language.

* Corresponding author.
E-mail addresses: carlos.olarte@gmail.com (C. Olarte), davide.chiarugi@gmail.com (D. Chiarugi), moreno.falaschi@unisi.it (M. Falaschi), dhermith@gmail.com (D. Hermith).

http://dx.doi.org/10.1016/j.tcs.2016.03.029
0304-3975/

In this paper we focus on biological networks whose interactions depend on both time and spatial locations. We study how to give a declarative meaning to those interactions by providing encodings of them into intuitionistic linear logic (ILL) [6] with subexponentials [7–9] (SELL). ILL is a substructural logic where formulas are seen as resources and then, its proof system(s) controls the number of times a formula can be used during a proof. In SELL, such control is much richer and it allows us to define different modalities (e.g., locations) where formulas can be stored. Hence, the role of the subexponentials, as clarified later, is to specify the two biologically relevant main dimensions in our study, namely time and space. We show that our method is general enough to encode P systems [10,11], a general model of computation inspired by cells structures. We show that different semantics for P systems, such as minimal [12] and maximal parallelism [10] semantics, can be logically characterized in the same framework.

The encodings we propose are shown to have a strong level of adequacy. More precisely, biological steps correspond one-to-one to (focused [13]) derivations in the SELL system. Hence, a proof of a given property can be directly traced to steps in the biological system. Our proof-theoretic characterization of spatial and timed dependencies in biochemical systems thus has at least two main benefits:

- From the biological point of view, our encodings open the possibility to use all the meta theory and tools developed for linear logic to specify and verify biochemical systems featuring spatial and temporal modalities. One salient characteristic of our approach is that both, the system and its properties, are specified in the same logical framework. This is particularly relevant since in many works related to ours (see Sections 5 and 6 for details), usually one formalism is used for specifying the model while at least one different formalism is used for expressing the properties of interest (e.g., a temporal logic) and for proving them (e.g., by using a model checker).
- From the computational point of view, we give a further step in showing that linear logic (with subexponentials) is a general framework to specify and verify concurrent systems. Other studies relating concurrent formalisms and linear logic can be found, e.g., in [14–17,8].

A preliminary short version of this paper appeared in [18]. Here, we significantly review, enhance and refine our previous work. In particular, we present the full set of proofs, we clarify crucial technical details and we introduce many more explanations and examples, including an application of our framework in the context of the TWEAK-Fn14 cell signaling pathway [19].

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. We also show how to exploit the underlying logic for expressing some properties of the system. Section 4 provides a logical characterization of P systems as SELL formulas. Section 5 highlights how the proof theory of SELL may be used to draw conclusions about the studied biochemical systems. Section 6 discusses related work and concludes the paper.

2. Linear logic with subexponentials

In this section we recall the proof theory of intuitionistic linear logic (ILL) [6] with subexponentials [7,8]. Although this review should suffice to understand the developments in the forthcoming sections, we assume that the reader is familiar with logic and proof theory (see e.g., [20]).

Linear logic [6] is a substructural logic where formulas can be seen as resources. Hence, there is an explicit control over the number of times a formula can be used in a proof. More precisely, formulas can be split into two sets: classical (those that can be used as many times as needed) or linear (those that are consumed after being used). Classical formulas are marked with the modal operators !, ?, called exponentials. For instance, the formula $!F$ allows F to be used arbitrarily many times.

Intuitionistic linear logic with subexponentials [7] (SELL) shares with linear logic all its connectives except the exponentials. The subexponentials ($!^a$, $?^a$) add an index to the exponentials, thus allowing for splitting the formulas into many sets, each of which can then be specified to be classical (i.e., unbounded) or linear. In this case, the formula $!^a F$ can be interpreted as F holding in a given modality a, e.g., in the space location a.

As we shall see, the subexponentials provide a finer control on proofs and they allow for the specification of different modalities such as time-units or spatial locations. In this paper we shall use the system $\text{SELL}^{\Uparrow}$ proposed in [8,9] that enjoys good proof theoretic properties: it admits cut-elimination and a sound and complete focused proof system [8,9]. Focusing [13] is a discipline on proofs to reduce the non-determinism during proof search. Hence, focused proofs can be interpreted as the normal form proofs for proof search. Although the proof rules of the focused system may look more involved at first glance, we shall rely on them to prove the adequacy results in Section 3. We thus introduce here the focused system and, for the sake of readability, we simplify a bit the notation. We also confine ourselves to the proof rules needed in the forthcoming sections, for instance, we do not introduce the rules for $?^a$, a connective not used in our encodings. The reader may refer to [8,9] for deeper technical details.

Connectives are separated into *negative* $\multimap, \&, \top, \forall, \Uparrow, \top$ and *positive* $\otimes, \oplus, \exists, \Downarrow, !^s$. The polarity of non-atomic formulas is inherited from its outermost connective (e.g., $F \multimap G$ is a negative formula while $F \otimes G$ is a positive one). Although the bias assigned to atoms does not interfere with provability [21], it changes considerably the shape of proofs. Here we require atoms to have a positive behavior.

The proof rules of SELL$^{\Uparrow}$ introduced here consider three kind of sequents[1]:

(i) $[\mathscr{K} : \Gamma], \Delta \longrightarrow \mathscr{R}$ is an unfocused sequent.
(ii) $[\mathscr{K} : \Gamma] -_F\!\rightarrow$ is a sequent focused on the right.
(iii) $[\mathscr{K} : \Gamma] \xrightarrow{F} G$ is a sequent focused on the left.

The meaning of the context $[\mathscr{K} : \Gamma]$ will be clear soon. We start with the proof rules for multiplicative ($\otimes$) and additive (&) conjunction, linear implication ($\multimap$), additive disjunction ($\oplus$), the first-order quantifiers ($\forall, \exists$) and the additive version of truth ($\top$).

Negative Phase

$$\frac{[\mathscr{K} : \Gamma], \Delta, F, G \longrightarrow \mathscr{R}}{[\mathscr{K} : \Gamma], \Delta, F \otimes G \longrightarrow \mathscr{R}}\ \otimes_L \qquad \frac{[\mathscr{K} : \Gamma], \Delta, F \longrightarrow G}{[\mathscr{K} : \Gamma], \Delta \longrightarrow F \multimap G}\ \multimap_R$$

$$\frac{[\mathscr{K} : \Gamma], \Delta \longrightarrow G[x_e/x]}{[\mathscr{K} : \Gamma], \Delta \longrightarrow \forall x.G}\ \forall_R \qquad \frac{[\mathscr{K} : \Gamma], \Delta \longrightarrow F \quad [\mathscr{K} : \Gamma], \Delta \longrightarrow G}{[\mathscr{K} : \Gamma], \Delta \longrightarrow F \& G}\ \&_R$$

$$\frac{}{[\mathscr{K} : \Gamma], \Delta \longrightarrow \top}\ \top_R \qquad \frac{[\mathscr{K} : \Gamma], \Delta, F \longrightarrow \mathscr{R} \quad [\mathscr{K} : \Gamma], \Delta, H \longrightarrow \mathscr{R}}{[\mathscr{K} : \Gamma], \Delta, F \oplus H \longrightarrow \mathscr{R}}\ \oplus_L$$

The proof rule $\exists_R$ is similar to $\forall_L$ and x_e is assumed to be fresh.

First notice that the negative connectives have invertible *right* rules, while the positive connectives have invertible *left* rules. As an example, consider the rule $\forall_R$: the choice of the name used for the eigenvariable x_e is not important for provability, as long as it is fresh. Hence, in a negative phase of the proof, no *backtracking* on the selection of inference rules is necessary. Moreover, without loosing provability, we can eagerly introduce all the negative non-atomic formulas on the right and all the positive non-atomic formulas on the left. Such part of the proof is represented by sequents of the shape (*i*) above.

The change of phase's polarity and the manipulation of the context $[\mathscr{K} : \Gamma]$ are governed by the following structural rules:

Structural Rules

$$\frac{[\mathscr{K} : \Gamma], F \longrightarrow G}{[\mathscr{K} : \Gamma] \xrightarrow{F} G}\ R_L \qquad \frac{[\mathscr{K} : \Gamma] \longrightarrow F}{[\mathscr{K} : \Gamma] -_F\!\rightarrow}\ R_R \qquad \frac{[\mathscr{K} : \Gamma] -_G\!\rightarrow}{[\mathscr{K} : \Gamma] \longrightarrow G}\ D_R$$

$$\frac{[\mathscr{K}, F : \Gamma] \xrightarrow{F} G}{[\mathscr{K}, F : \Gamma] \longrightarrow G}\ D_{L1} \qquad \frac{[\mathscr{K} : \Gamma] \xrightarrow{F} G}{[\mathscr{K} : \Gamma, F] \longrightarrow G}\ D_{L2} \qquad \frac{[\mathscr{K} : \Gamma, N_a], \Delta \longrightarrow G}{[\mathscr{K} : \Gamma], \Delta, N_a \longrightarrow G}\ St_L$$

Rules R_L and R_R mark the end of the positive phase. In R_L (resp. R_R), F is a positive (resp. negative) formula and then, the positive phase must finish. A positive phase begins by choosing a formula on which to focus enabling sequents of the forms (*ii*) or (*iii*). Note that in such sequents, the multiset of formulas Δ must be empty. Rule D_R is used to *decide* to focus on the formula G (on the right). Rule D_{L1} (resp. D_{L2}) focuses on the formula F in the unbounded $\mathscr{K}$ (resp. linear Γ) context explained below. Finally, in rule St_L, N_a is a negative or atomic formula. Then, during the negative phase, this rule stores in the context Γ the formulas that cannot be introduced (on the left) during the negative phase (e.g., $F \multimap G$).

Now we introduce some of the proof rules that belong to the positive phase:

Positive Phase

$$\frac{[\mathscr{K} : \Gamma] -_{G_i}\!\rightarrow}{[\mathscr{K} : \Gamma] -_{G_1 \oplus G_2}\!\rightarrow}\ \oplus_{R_i} \qquad \frac{[\mathscr{K} : \Gamma] \xrightarrow{F_i} G}{[\mathscr{K} : \Gamma] \xrightarrow{F_1 \& F_2} G}\ \&_{L_i} \qquad \frac{[\mathscr{K} : \Gamma] \xrightarrow{F[t/x]} G}{[\mathscr{K} : \Gamma] \xrightarrow{\forall x.F} G}\ \forall_L$$

The above rules are introduced in the positive phase since they require a *decision* to continue the proof and then, backtracking may be needed. Rule $\oplus_R$, for instance, needs to chose a formula G_i ($i \in \{1, 2\}$) to continue the proof. Similarly, rule $\forall_L$ decides on the term t. Note also that in the above rules, the focusing is not lost and the proof must continue decomposing the selected formula (for instance, in $\oplus_R$, the focusing persists on G_i). This procedure continues until one is focused either on a negative formula on the right or a positive formula on the left (see structural rules R_L and R_R). This point marks the end of the positive phase.

[1] The SELL$^{\Uparrow}$ system in fact considers also a fourth kind of sequent of the shape $[\mathscr{K} : \Gamma], \Delta \longrightarrow [F]$ to represent the end of the negative phase. In our presentation, we shall omit this kind of sequent and also the notation of bracket formulas $[F]$.

A SELL$^{\Uparrow}$ 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 are unbounded and $\preceq$ is a pre-order among the elements of I. We shall use the letters a, b, l and s to denote elements in I. Intuitively, $!^a F$ means that F is marked with a given modality a. As shown in [8], the formula $!^a F$ can be interpreted in several ways, for instance, it may represent the fact that F holds in the space location a or that F holds in the time-unit a. Moreover, if a is unbounded (or classical), then F can be used as many times as needed.

Let us now introduce some other proof rules and clarify the meaning of the context $[\mathscr{K} : \Gamma]$ and the subexponential $!^a$.

Negative Phase

$$\frac{[\mathscr{K} +_s F : \Gamma], \Delta \longrightarrow \mathscr{R}}{[\mathscr{K} : \Gamma], \Delta, !^s F \longrightarrow \mathscr{R}}\ !^s_L \qquad \frac{[\mathscr{K} : \Gamma], \Delta \longrightarrow G[l_e/l_x]}{[\mathscr{K} : \Gamma], \Delta \longrightarrow \Uparrow l_x : a.G}\ \Uparrow_R$$

Positive Phase

$$\frac{[\mathscr{K}_1 : \Gamma_1]{-_F\!\!\rightarrow} \quad [\mathscr{K}_2 : \Gamma_2]{-_G\!\!\rightarrow}}{[\mathscr{K}_1 \otimes \mathscr{K}_2 : \Gamma_1, \Gamma_2]{-_{F\otimes G}\!\!\rightarrow}}\ \otimes_R\ \blacklozenge_1 \qquad \frac{[\mathscr{K}_1 : \Gamma_1]{-_F\!\!\rightarrow} \quad [\mathscr{K}_2 : \Gamma_2] \xrightarrow{H} G}{[\mathscr{K}_1 \otimes \mathscr{K}_2 : \Gamma_1, \Gamma_2] \xrightarrow{F\multimap H} G}\ \multimap_L\ \blacklozenge_1$$

$$\frac{}{[\mathscr{K} : \Gamma]{-_A\!\!\rightarrow}}\ I_R\ \blacklozenge_2 \qquad \frac{[\mathscr{K} \preceq_s : \cdot] \longrightarrow F}{[\mathscr{K} : \cdot]{-_{!^s F}\!\!\rightarrow}}\ !^s_R \qquad \frac{[\mathscr{K} : \Gamma] \xrightarrow{P[l/l_x]} G}{[\mathscr{K} : \Gamma] \xrightarrow{\Uparrow l_x : a.P} G}\ \Uparrow_L$$

In the context $[\mathscr{K} : \Gamma]$, Γ contains only atomic or negative formulas that must be introduced in the positive phase of the proof (see Rule St_L). $\mathscr{K}$ is used to map a subexponential index to a multiset of formulas, notation $s \mapsto \{F_1, \ldots, F_n\}$. The multiset $\mathscr{K}[s] = \{F_1, \ldots, F_n\}$ (the image of $\mathscr{K}$ in s) represents all the formulas marked with $!^s$. That is, $\mathscr{K}[s]$ should be interpreted as the multiset of formulas $!^s F_1, \ldots, !^s F_n$. This explains the rule $!^s_L$ which belongs to the negative phase and it simply stores F into the context $\mathscr{K}[s]$ (notation, $\mathscr{K} +_s F$).

Since formulas in linear logic are *resources*, rules $\otimes_R$ and $\multimap_L$ split the context. For instance, in a proof of $F \otimes G$, some resources must be used to prove F and the remaining ones must be used to prove G. However, the unbounded resources, i.e., those formulas of the shape $!^a F$ where $a \in U$, can be used as many times as needed. Hence, side condition $\blacklozenge_1$ says that the contexts $\mathscr{K}_1$ and $\mathscr{K}_2$ must agree on those formulas, i.e., $\mathscr{K}_1[a] = \mathscr{K}_2[a]$ for any $a \in U$. Notation $\mathscr{K}_1 \otimes \mathscr{K}_1$ represents the fact that the context on the conclusion of rules $\otimes_R$ and $\multimap_L$ can be decomposed into $\mathscr{K}_1$ and $\mathscr{K}_2$. More precisely, $\mathscr{K}_1 \otimes \mathscr{K}_2[s]$ is the multiset union (resp. set union) of $\mathscr{K}_1[s]$ and $\mathscr{K}_2[s]$ if $s \notin U$ (resp. $s \in U$).

The implication $F \multimap H$ can be read as "*consume* F *to produce* H". Then, rule $\multimap_L$ consumes some resources to prove F and then, focusing persists on H.

The initial rule I_R says that a proof for an atom A (a positive formula) can finish if A is already in the context. Side condition $\blacklozenge_2$ says that either 1) the linear context (formulas in Γ and those in $\mathscr{K}$ marked with subexponentials not in U) is empty and A belongs to the unbounded context, i.e., $A \in \mathscr{K}[a]$ and $a \in U$; or 2) the linear context is $\{A\}$. This guarantees that either A is an unbounded resource (and then, we can prove A) or the only linear formula in the context is A (i.e., there is only one copy of A).

Rule $!^s_R$ says that, in order to introduce the formula $!^s F$, the linear context Γ must be empty (notation $[\mathscr{K} : \cdot]$). Moreover, for all $a \in I$, if $s \not\preceq a$ then $\mathscr{K}'[a] = \emptyset$, i.e., all the formulas marked with an unrelated or smaller subexponential a, must be weakened (notation $[\mathscr{K} \preceq_s : \cdot]$). This can be done, of course, if a is an unbounded subexponential. Roughly, if the formula $!^s F$ needs to be proved, one can only keep the resources (formulas) marked with a higher subexponential. The other resources must be dropped (weakened) from the context.

Two other observations on rule $!^s_R$ are in order: (1) provability is preserved *downwards*, i.e., if $[\mathscr{K} : \Gamma], \Delta \longrightarrow !^a F$ is provable then so is the sequent $[\mathscr{K} : \Gamma], \Delta \longrightarrow !^b F$ for all $b \preceq a$; and (2), the focusing does not persist on F.

In [8,9], universal ($\Uparrow$) and existential ($\Downarrow$) quantification on subexponentials are introduced for the specification of modalities in distributed systems. Similar to the first-order quantifiers, the rule $\Uparrow_R$ belongs to the negative phase and it simply introduces a fresh subexponential variable. The generic variable $l_x : a$, where $a \in I$, plays the role of the type of l_x representing *any subexponential constant* l_c in the ideal of a, i.e., $l_c \preceq a$. Rule $\Uparrow_L$ belongs to the positive phase since a subexponential $l \preceq a$ must be chosen. Rules for the existential quantifier $\Downarrow$ are similar.

Let us introduce two simple examples showing how focusing and the subexponentials allow us to control the proofs in SELL$^{\Uparrow}$. This may help to understand the level of adequacy we achieve in Section 3.

Example 1 *(Focusing).* Consider the derivations in Fig. 1 where F, G, H are atoms. The proof on the left corresponds to a focused proof. First, in the negative phase, we introduce all the $\otimes$ on the left and we use the rule St_L to store the atom F and the negative formulas ($F \multimap G$ and $G \multimap H$) into the context. Then we *decide* to focus on the formula $F \multimap G$ (rule D_{L2}). According to rule $\multimap_L$, the focusing persists on the atom F and then, the proof must finish with the initial rule I_R. Since G is an atom, focusing is lost (rule R_L) and such atom is added later into the context (rule St_L). In a new positive phase, we

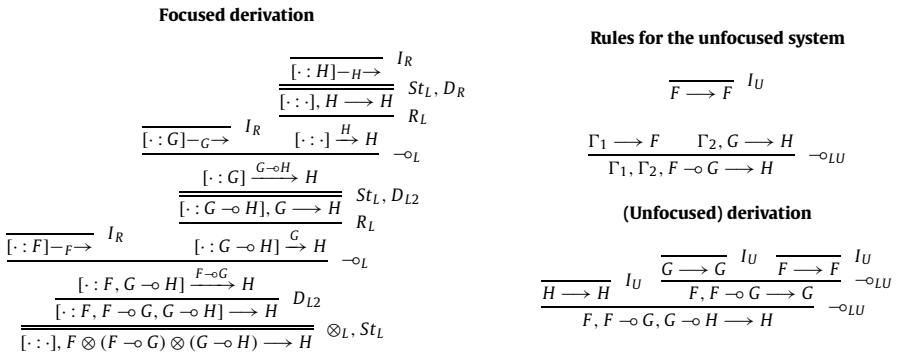

Fig. 1. Examples of focused and unfocused derivations (see Example 1).

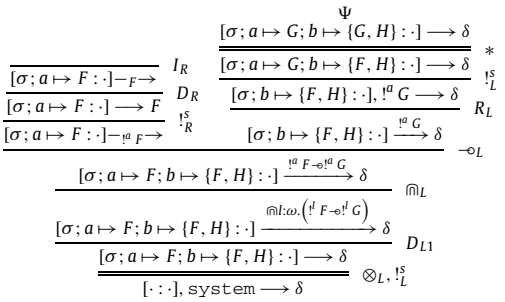

Fig. 2. Derivation in Example 2. σ is the context $\omega \mapsto \Cap l : \omega.\left(!^l F \multimap !^l G\right)$ and δ is the formula $!^a G \otimes !^b G \otimes !^b H$.

focus on $G \multimap H$ (rule D_{L2}) and rules $\multimap_L$, I_R and R_L are applied as before until they produce the sequent $[\cdot : \cdot], H \longrightarrow H$. At this point, we store H into the context and decide to focus on H (on the right, D_R) and the derivation ends with I_R.

Derivation on the right corresponds to an unfocused proof using the standard rules of linear logic (I_U and LU in the figure). Note that we "use" the implication $G \multimap H$ but the proof of G was delayed until G was *later* "produced" by $F \multimap G$. The behavior of this proof, in the context of biochemical reactions, 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. We observe that this cannot happen in the focused system: it is not possible to focus on $G \multimap H$ in the beginning of the derivation because G is not in the context.

Example 2 *(Subexponentials).* Consider an unbounded subexponential ω (i.e., $\omega \in U$) and two linear subexponentials a, b s.t. $a, b \preceq \omega$ and $a \not\preceq b$. Let F, G, H be atoms and $\mathtt{system} \stackrel{\mathtt{def}}{=} !^\omega \Cap l : \omega.\left(!^l F \multimap !^l G\right) \otimes !^a F \otimes !^b F \otimes !^b H$. The subexponentials a and b can be interpreted as two different locations (or spatial domains) in a cell. Hence, $!^a F$ (resp. $!^b F$, $!^b H$) represents that in a (resp. b) we can find a copy of some given reactant F (resp. F,H). The formula $\Cap l : \omega.\left(!^l F \multimap !^l G\right)$ represents a reaction that, in a given location $l \preceq \omega$, consumes F to produce G. Note that this formula is marked with $!^\omega$, thus allowing us to use the reaction as many times as needed in the space domains a and b. Hence, the system may evolve to a state where the location a contains one copy of G and the location b stores one copy of G and H as proved in Fig. 2.

The first step in the derivation corresponds to the negative phase introducing the connectives $!^s$ and $\otimes$ on the left. Then we decide to focus on the formula representing the reaction (D_{L1}). Note that focusing persists and then, a subexponential l in $\Cap l : \omega$ must be chosen (in this case, $[a/l]$) to later decompose the formula $!^a F \multimap !^a G$. It is also interesting to note that rule $!^s_R$ forces the derivation to consider only the contexts ω and a to prove F since $a \not\preceq b$ (see rule $!^s_R$). This intuitively means that information on the location b cannot be used to finish the proof of F in location a.

The steps in $*$ correspond to the proof of G in the context b by consuming F, similarly as we did in the context a. Derivation Ψ uses $\otimes_R$, $!^s{}_R$ and I_R to prove G in the context a and G and H in the context b.

3. Spatial and temporal dependencies as SELL$^{\Cap}$ formulas

In this section we show how spatial and temporal dependencies in biochemical systems can be declaratively characterized as formulas in SELL$^{\Cap}$. We prove that the proposed encoding exhibits the highest level of adequacy: each step in the evolution of the system corresponds, one-to-one, to a change of polarity in a focused derivation. We shall exemplify our framework by verifying some basic properties of the TWEAK-Fn14 cell signaling pathway [19].

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 Fig. 3. The only unbounded subexponentials are t_ω and i^+. The former is used to mark the set of reactions that can be used as many times as needed. The subexponential i^+ represents the time-units starting from i and it will be used to specify system's properties as we explain later. For instance, a subexponential variable $l_x : 4^+$ can be instantiated with any time-unit (in the future) starting from 4. The linear subexponentials $0, 1, 2, \cdots$ (resp. $s_a.i$, $s_b.i, \cdots$) represent time-units (resp. the *space* domain s_x in the time-unit i). An alternative way of defining the spatial locations is to consider the use of *families* [8], roughly, functions from elements in the subexponential signature to subexponential indexes. Hence, given a family of the shape $\mathbf{s_x}$, $\mathbf{s_x}(t)$ would be the subexponential representing the space x in time-unit t.

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 as the formula

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

where $\mathscr{A}$ denotes the set of reactants and $\mathscr{S}$ the set of domain spaces. If there are no species of kind A_j in the space s_k, then $c_j = \mathbf{0}$.

We model the set of reactions of the system as

$$\mathtt{eqs} \stackrel{\mathtt{def}}{=} !^{t_\omega}\left[\Cap l_x : 0^+.\,[\mathtt{eq}_1(l_x)\&\cdots\&\mathtt{eq}_k(l_x)]\right]$$

The unbounded subexponential $!^{t_\omega}$ allows us to use the set of reactions as many times as needed. The universal quantification $\Cap l_x : 0^+$ says that, at any time-unit, the reactions are available (see Example 2). The connective & allows us to *choose* (non-deterministically) one of the reactions and discard the others.

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

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

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

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

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

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 one step in a focused derivation corresponds exactly to one step in the evolution of the system. Hence, proofs in SELL$^{\Cap}$ can be directly mapped to traces of the system. Before stating the result, we introduce some needed notation.

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

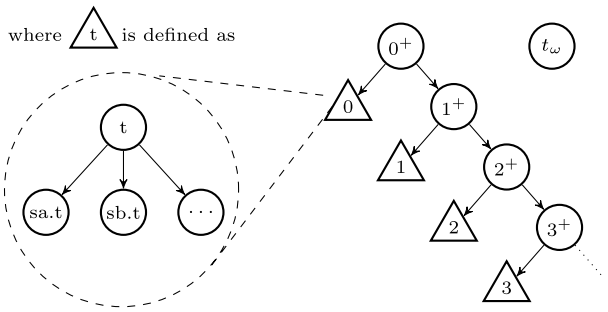

Fig. 3. Subexponential structure for spatial and temporal domains. $b \to a$ means $a \preceq b$. The only unbounded subexponentials are i^+ and t_ω.

Theorem 1 *(Adequacy). Let $\mathfrak{s}_1$ and $\mathfrak{s}_2$ be states, r a reaction and $t \geq 0$. 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 $\mathfrak{s}_2$.*

Proof. We show that the introduction of any formula, following the focused discipline, corresponds exactly to the occurrence of one reaction in the biochemical system. More precisely, if we focus on the left of the sequent $[\![\mathfrak{s}_1]\!]_t \longrightarrow G$, one flip of the polarity of the proof corresponds to the operational step $\mathfrak{s}_1 \xrightarrow{(r,k)} \mathfrak{s}_2$.

In $[\![\mathfrak{s}_1]\!]_t$ we have a conjunction ($\otimes$) of formulas of the shape $!^s F$. Then, in a negative phase, what we observe is the following:

$$\frac{[\mathscr{K} +_{t_\omega} (\mathtt{eqs}') + \sum_{s_i.t}\{\mathtt{ct}(A_1,c_1),\ldots,\mathtt{ct}(A_n,c_n)\} : \cdot] \longrightarrow G}{[\mathscr{K} : \cdot], [\![\mathfrak{s}_1]\!]_t \longrightarrow G}\ !^s_L, \otimes_L$$

i.e., the context t_ω stores the formula $\mathtt{eqs}' \stackrel{\text{def}}{=} \mathbin{⫛} l_x : 0^+.[\mathtt{eq}_1(l_x)\&\cdots\&\mathtt{eq}_k(l_x)]$ and each $s_i.t$ context stores the formulas of the shape $\mathtt{ct}(A_1,c_1),\ldots,\mathtt{ct}(A_n,c_n)$.

We note that the negative phase ends here (the multiset Δ is empty). At this point, we have three choices: focus on the right, focus on the formula stored in one of the contexts $s_i.t$ or focus on the formula stored in the context t_ω. Focus on the right ($[\![\mathfrak{s}_2]\!]_{t+k}$) will fail since the atoms in the subexponential $s_i.(t+k)$ are not already in the context. Since formulas stored in the context $s_i.t$ are atoms, focus will be lost immediately, thus leading to a useless detour. Hence, the only choice is to focus on $\mathtt{eqs}'$ and we get a derivation of the following shape:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Pi \quad \Psi}{\Gamma_1 -\!\!\bigotimes_{s\in\mathscr{S}}\bigotimes_{A_i\in\mathscr{A}} !^{s.t'}[\mathtt{ct}(A_i,N_i)]\to \quad \Gamma_2 \xrightarrow{\mathtt{produce}_i(t'+k)} G}\ -\!\circ_L}{[t_\omega \mapsto \mathtt{eqs}'; \{s_i.t \mapsto \{\mathtt{ct}(A_1,c_1),\ldots,\mathtt{ct}(A_n,c_n)\}\}_i : \cdot] \xrightarrow{\mathtt{consume}_i(t')\multimap\mathtt{produce}_i(t')} G}\ \forall_L}{[t_\omega \mapsto \mathtt{eqs}'; \{s_i.t \mapsto \{\mathtt{ct}(A_1,c_1),\ldots,\mathtt{ct}(A_n,c_n)\}\}_i : \cdot] \xrightarrow{\forall\vec{x}.\mathtt{consume}_i(t')\multimap\mathtt{produce}_i(t')} G}\ \&_L}{[t_\omega \mapsto \mathtt{eqs}'; \{s_i.t \mapsto \{\mathtt{ct}(A_1,c_1),\ldots,\mathtt{ct}(A_n,c_n)\}\}_i : \cdot] \xrightarrow{\mathtt{eq}_1(t')\&\ldots\&\mathtt{eq}_n(t')} G}\ \mathbin{⫛}_L}{[t_\omega \mapsto \mathtt{eqs}'; \{s_i.t \mapsto \{\mathtt{ct}(A_1,c_1),\ldots,\mathtt{ct}(A_n,c_n)\}\}_i : \cdot] \xrightarrow{\mathtt{eqs}'} G}\ D_{L1}}{[t_\omega \mapsto \mathtt{eqs}'; \{s_i.t \mapsto \{\mathtt{ct}(A_1,c_1),\ldots,\mathtt{ct}(A_n,c_n)\}\}_i : \cdot] \longrightarrow G}$$

Here Γ_1 and Γ_2 correspond to the splitting of the context due to the rule $-\!\circ_L$. Since $\otimes_R$ must be introduced in a positive phase, the focusing persists on $\mathtt{consume}_i(t')$ and the derivation Π must be of the shape:

$$\dfrac{\dfrac{\dfrac{\Pi_i}{\Upsilon_1^i \longrightarrow \mathtt{ct}(A_i,N_i)}\ !^s_R \quad \Pi'}{\Gamma_1^i -_{!^{s_i.t'}\,\mathtt{ct}(A_i,n_i)}\to \quad \Gamma_1' -_{C'}\to}\ \otimes_R}{\Gamma_1 -\!\!\bigotimes_{s\in\mathscr{S}}\bigotimes_{A_i\in\mathscr{A}} !^{s.t'}[\mathtt{ct}(A_i,N_i)]\to}$$

Since $s_i.t'$ is related neither to t_ω nor to any subexponential $s_j.t'$, the context Υ_1^i can only be of the shape $[s_i.t' \mapsto \{F\} : \cdot]$ and then, the derivation Π_i corresponds to the application of the initial rule I_R (see Example 2). A similar analysis can be done for Π' that corresponds to the proof of the other conjuncts in the formula $\bigotimes_{s\in\mathscr{S}} \bigotimes_{A_i\in\mathscr{A}} !^{s.t'}[\mathtt{ct}(A_i, N_i)]$.

Since $\mathtt{consume}_i(t')$ was defined to "consume" all the formulas of the shape $\mathtt{ct}(A_i, n_i)$ in all the spaces, it must be the case that $\Gamma_2 = [t_\omega \mapsto \mathtt{eqs}' : \cdot]$.

Now, let's analyze the formula $\mathtt{produce}(t' + k)$. The main connective of this formula is $\otimes$ and then, focus is lost in the derivation Ψ. In a negative phase, we have

$$\frac{[t_\omega \mapsto \mathtt{eqs}'; \{s_i.(t'+k) \mapsto \{\mathtt{ct}(A_1, c_1'), \cdots, \mathtt{ct}(A_n, c_n')\}\}_i : \cdot] \longrightarrow G}{[t_\omega \mapsto \mathtt{eqs}' : \cdot], \mathtt{produce}(t'+k) \longrightarrow G}\ !_L^{\mathscr{S}}, \otimes_L$$

where the resulting context encodes the state $\mathfrak{s}_2$.

It is worth noticing that the focusing discipline forces the rule $\Uparrow_L$ to choose $t' = t$, in other case, the formula $!^{s_i.t'}\,\mathtt{ct}(A_i, c_i)$ would not be provable in Π_i. Moreover, the focusing discipline also forces the rule $\&_L$ to choose the encoding of a reaction whose reactants are already in the context (see Example 1). Hence, what we observe in the derivation is exactly that some reaction is applied and the products are produced in the time-unit $t + k$, i.e., focusing on the left corresponds exactly to the system's evolution $\mathfrak{s}_1 \xrightarrow{(r,k)} \mathfrak{s}_2$. □

3.2. Properties of interest

In this section we present some applications of our framework. For the sake of readability, we shall introduce simple reaction schemes to show how to verify a given property. Then, we will relate such reaction schemes to our case study, namely, $\mathtt{system}(0) \longrightarrow \Uparrow t : 0^+.!^t\,\mathtt{stable}$.

The types of intercommunications occurring in a signaling pathway can be broadly classified into physical interactions, enzyme catalysis interactions, activation/inhibition interactions and transport interactions (translocations). These reactions, most of the time, take place in cellular domains such as the extracellular, the plasma membrane, the cytoplasm or the nucleus among others.

Let us start with a small biochemical system composed of two unimolecular reactions of the form:

$$\begin{aligned} r_1 &: [1.A]_x \longrightarrow^1 [1.B]_x \\ r_2 &: [1.B]_x \longrightarrow^2 [1.B]_y \end{aligned} \tag{2}$$

Reaction r_1 models the situation where one unit of A, *located* in the *space domain* x, is consumed in one time-unit to produce one unit of B in the same *space domain*. Reaction r_2 takes two time-units to translocate one unit of B from *space* x to *space* y.

Roughly speaking, signaling pathways are networks of biochemical reactions that allow cells to read environmental cues, translate them into intracellular commands, and react with an appropriate response. The small set of reactions in Equation (2) could be a good example to represent an interaction of enzyme catalysis ($\mathtt{cat}$) in a cellular domain followed by a transport interaction or translocation ($\mathtt{trans}$) between two arbitrary interaction domains. Such reactions can be related to (a fragment of) the TWEAK-Fn14 cell signaling pathway [19]. This pathway is expressed in several different tissue types and has implications with several diseases including autoimmune disorders, cancer and cardiovascular abnormalities. Generally, translocations and catalysis are common in this network. For instance, in the shaded box in Fig. 4, we highlight the reaction of phosphorylation of the transcription factor $\mathtt{RELA}$ (kappa light chain gene enhancer in B cells 3) that takes place in the interaction domain of the cytoplasm (CY) to produce $[\mathtt{RELA\text{-}P}]_{\mathtt{CY}}$. Next, the phosphorylated form $[\mathtt{RELA\text{-}P}]_{\mathtt{CY}}$ is *translocated* to the interaction domain of the nucleus (NU) $[\mathtt{RELA\text{-}P}]_{\mathtt{NU}}$. Consequently, due to this transport interaction, the molecule $[\mathtt{RELA\text{-}P}]_{\mathtt{NU}}$ can be again *catalyzed* to its original form and placed in its original location to form $[\mathtt{RELA}]_{\mathtt{CY}}$.

3.2.1. Reachability properties

We can verify reachability properties in our system by proving sequents of the shape $\mathtt{system}(0) \longrightarrow \Uparrow t : 0^+.\left[!^{a.t}\,\mathtt{ct}(A, n)\right] \otimes \top$. 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". Since we are testing the presence of a given component (and not all of them) in a specific domain, the $\top$ connective allows us to erase the unused formulas (see rule $\top_R$).

Due to Theorem 1, a focused proof of the above sequent can be directly traced to the moves the system has to perform to reach the state $\mathtt{ct}(A, n)$. As a matter of example, let us prove that the concentration of A in the space domain x eventually falls to 0 in the system described in Equation (2). Before that, let us introduce some useful notation:

$$\begin{aligned} \mathtt{state}(t : k, m, n) &\stackrel{\mathtt{def}}{=} !^{t.x}\,\mathtt{ct}(A, k) \otimes !^{t.x}\,\mathtt{ct}(B, m) \otimes !^{t.y}\,\mathtt{ct}(B, n) \\ \mathtt{state}(t : k_{>0}, m, n) &\stackrel{\mathtt{def}}{=} !^{t.x}\,\mathtt{ct}(A, \mathtt{suc}(k)) \otimes !^{t.x}\,\mathtt{ct}(B, m) \otimes !^{t.y}\,\mathtt{ct}(B, n) \end{aligned}$$

A similar notation is used for, e.g., $\mathtt{state}(t : k_{>0}, m, n_{>0})$. These formulas represent the concentration of A in the domain x and the concentration of B in the domains x and y. Moreover, we shall also use the shorthand

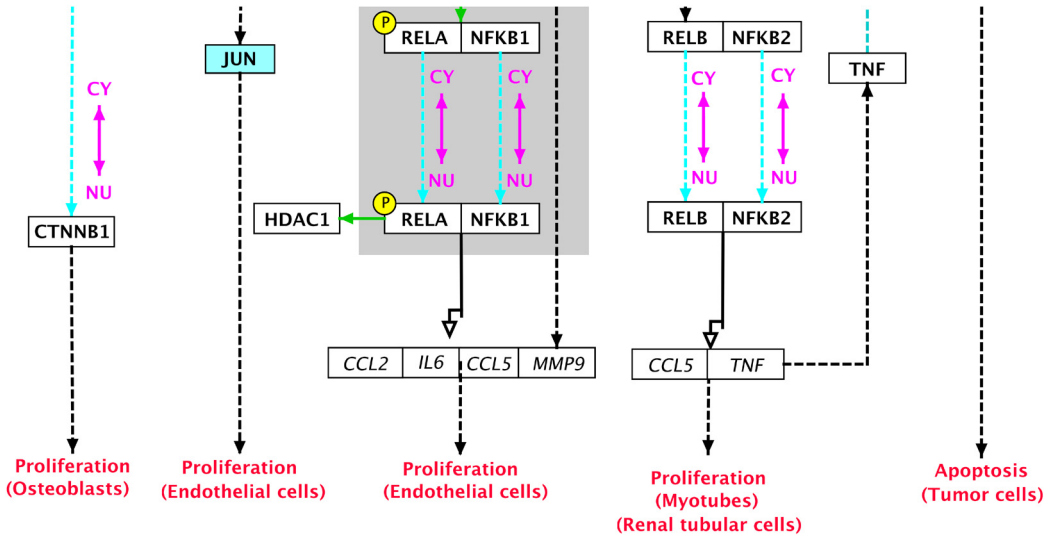

Fig. 4. Excerpt of the TWEAK-Fn14 cell signaling pathway from [19]. The shaded box corresponds to the reaction schemes $r_1 : [\mathrm{RELA}]_{\mathrm{CY}} \overset{\mathrm{cat}}{\longrightarrow} [\mathrm{RELA\text{-}P}]_{\mathrm{CY}}$, $r_2 : [\mathrm{RELA\text{-}P}]_{\mathrm{CY}} \overset{\mathrm{trans}}{\longrightarrow} [\mathrm{RELA\text{-}P}]_{\mathrm{NU}}$ and $r_3 : [\mathrm{RELA\text{-}P}]_{\mathrm{NU}} \overset{\mathrm{cat/trans}}{\longrightarrow} [\mathrm{RELA}]_{\mathrm{CY}}$.

$$\mathop{\&}_{k,m,n} \mathtt{state}(t:k,m,n)[F_{k,m,n}] \overset{\mathrm{def}}{=} (\mathtt{state}(t:0,0,0) \multimap F_{0,0,0})\& \\ (\mathtt{state}(t:k_{>0},0,0) \multimap F_{\mathtt{suc}(k),0,0})\& \cdots \& \\ (\mathtt{state}(t:k_{>0},m_{>0},n_{>0}) \multimap F_{\mathtt{suc}(k),\mathtt{suc}(m),\mathtt{suc}(n)})$$

This formula will be used to specify that the property F holds regardless the values of the concentrations.

The following sequent formalizes the reachability property to be proved:

$$[t_\omega \mapsto \mathtt{eqs} : \cdot] \longrightarrow \forall k,m,n. \mathop{\&}_{k,m,n} \mathtt{state}(t:k,m,n)[F]$$

where $F = \mathbb{U}l : t^+. \left(!^{x.l}\, \mathtt{ct}(A,0)\right) \otimes \top$.

After decomposing (in a negative phase) the connectives $\forall$, & and $\multimap$ on the right and then $\otimes$ and $!^s$ on the left we have several cases to be proved (see rule $\&_R$). The case $\mathtt{state}(t:0,0,0)$ is trivial since focusing on one of the reactions will fail. Hence, the only choice we have is to focus on the formula F on the right. Since $\mathtt{ct}(A,0)$ is already in the context, the proof is easy.

In the other cases, we can apply one of the reactions and we observe a (focused) derivation of the shape (see proof of Theorem 1):

$$\frac{[t_\omega \mapsto \mathtt{eqs}'; \mathtt{state}(t':k'_t,m'_t,n'_t) : \cdot] \longrightarrow F}{[t_\omega \mapsto \mathtt{eqs}'; \mathtt{state}(t:k_t,m_t,n_t) : \cdot] \longrightarrow F}$$

Here, for the sake of presentation, we abuse of the notation and we use the formula $\mathtt{state}(t:k,m,n)$ to denote also the context it represents. In the above derivation, k_t may be 0 or $\mathtt{suc}(k)$ depending on the considered case. If $k_t = \mathtt{suc}(k)$ and the derivation corresponds to focusing on reaction r_1, and then, it must be the case that $k'_t = k$. Similar observations can be done for m_t, n_t and rule r_2. Regardless the choice of the rule (r_1 or r_2), we can show that $\langle k'_t, m'_t\rangle \prec \langle k_t, m_t\rangle$ where "$\prec$" is the (well-founded) lexicographical order where the first element is the predominant component. Hence, by induction, we can show that there will be a state where $\mathtt{ct}(A,0)$ holds.

3.2.2. Stable states

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 $\mathtt{eq}_i$ formulas in $\mathtt{eq}_1\&\ldots\&\mathtt{eq}_n$ can be chosen. In this section, we show a possible way to circumvent this problem. For that, let us introduce the following reaction scheme:

$$\begin{aligned} r_1 &: [2.A]_x + [1.B]_x \longrightarrow^1 [1.C]_x \\ r_2 &: [1.C]_x \longrightarrow^1 [1.A]_x \end{aligned} \tag{3}$$

We encode the set of reactions as in Section 3 but with a slight difference:

$$\begin{aligned}
\mathtt{eqs}_d &\stackrel{\mathtt{def}}{=} !^{t_\omega}\left(\Uparrow l_x : 0^+.\,(\mathtt{eq}_1(l_x)\&\cdots\&\mathtt{eq}_k(l_x)\&\mathtt{eq}_d(l_x))\right)\\
\mathtt{eq}_d(t) &\stackrel{\mathtt{def}}{=} \left(\left(!^{x.t}\,\mathtt{ct}(A,\mathbf{0})\oplus !^{x.t}\,\mathtt{ct}(A,\mathtt{suc}(\mathbf{0}))\oplus !^{x.t}\,\mathtt{ct}(B,\mathbf{0})\right)\&\left(!^{x.t}\,\mathtt{ct}(C,\mathbf{0})\right)\right)\\
&\quad \multimap !^{t}\,\mathtt{stable}
\end{aligned}$$

Intuitively, $\mathtt{eq}_d$ checks whether the reactants are not sufficient to trigger any of the rules. In that case, the atom $\mathtt{stable}$ is produced. In the system described in Equation (3) this happens when: the concentration of A is below 2 or the concentration of B is 0 (r_1); and when the concentration of C is 0 (r_2).

Consider the sequent $[t_\omega \mapsto \mathtt{eqs}_d; \mathtt{state}(t:1,0,0):\cdot] \longrightarrow G$ where we assume that the concentrations are $A=1$, $B=0$, $C=0$. We already know that focusing on a formula representing one of the reactions in Equation (3) will fail immediately. Hence, the only available choice is to focus on the formula $\mathtt{eq}_d$. Focusing on $\mathtt{eq}_d$ implies proving the following sequent:

$$[t_\omega \mapsto \mathtt{eqs}_d; \mathtt{state}(t:1,0,0):\cdot] \multimap_{F_1 \& F_2} \rightarrow$$

with $F_1 = \left[!^{s.t}\,\mathtt{ct}(A,\mathbf{0})\oplus !^{s.t}\,\mathtt{ct}(A,\mathtt{suc}(\mathbf{0}))\oplus !^{s.t}\,\mathtt{ct}(B,\mathbf{0})\right]$ and $F2 = !^{s.t}\,\mathtt{ct}(C,\mathbf{0})$. Rule $\&_R$ belongs to the negative phase and then, the focus is lost. Hence, we cannot guarantee that the proof has to finish *immediately* proving from the context the atoms $\mathtt{ct}(\cdot,\cdot)$. However, since the subexponential $x.t$ is unrelated to all other subexponentials, we do guarantee that, for proving such atoms, the set of reactions cannot be used again (see rule $!^s_R$). We can thus prove that the system reaches a stable state iff the sequent $\mathtt{system}(0) \longrightarrow \Uparrow t : 0^+.!^t\,\mathtt{stable}$ is provable.

Now let us put in context the previous result in the setting of the TWEAK-Fn14 signaling pathway, and consider the following reaction scheme resembling that of Equation (3):

$$\begin{aligned}
r_1 &: [2.\mathtt{TNFSF12}]_{\mathtt{PM}} + [1.\mathtt{TNFRSF25}]_{\mathtt{PM}} \longrightarrow [1.\mathtt{TNFSF12\text{-}TNFRSF25}]_{\mathtt{PM}}\\
r_2 &: [1.\mathtt{TNFSF12\text{-}TNFRSF25}]_{\mathtt{PM}} \longrightarrow [1.\mathtt{TNFSF12}]_{\mathtt{PM}}
\end{aligned} \tag{4}$$

The TWEAK ligand, also known as $[\mathtt{TNFSF12}]_{\mathtt{PM}}$ (tumor necrosis factor ligand superfamily member 12), can bind in the plasma membrane (PM) not only to the receptor $[\mathtt{Fn14}]_{\mathtt{PM}}$ (fibroblast growth factor inducible immediate early response protein 14) but also to the receptor $[\mathtt{TNFRSF25}]_{\mathtt{PM}}$ (tumor necrosis factor receptor superfamily member 25) [19]. Hence, a stable state in this scenario could demonstrate that, once a receptor has achieved his maximum occupancy, no rule can be applied. In other words, an interaction of the type ligand–receptor is constrained by the availability of the receptor to bind a single ligand molecule with the option of unbinding it at any time. Moreover, this property also allows us to capture the notion of the limiting reactant (LR), a very common concept used in stoichiometry. Roughly, the LR of a chemical reaction is the reactant which is totally consumed or transformed when the system reaches the equilibrium. As a result, the reaction will stop when all the LR is consumed. Here, the receptor plays the role of the LR. Thus, to reach a chemical equilibrium in Equation 4, one molecule of ligand and one molecule of receptor are needed to form a ligand–receptor interaction.

3.2.3. Oscillations

Consider the following set of equations that extends that of Equation (2) with reaction r_3 below:

$$\begin{aligned}
r_1 &: [1.A]_x \longrightarrow^1 [1.B]_x\\
r_2 &: [1.B]_x \longrightarrow^2 [1.B]_y\\
r_3 &: [1.B]_y \longrightarrow^1 [1.A]_x
\end{aligned} \tag{5}$$

We can prove that, for all t, if $\mathtt{state}(t:k,m,n)$ holds then there exists $t' > t$ such that $\mathtt{state}(t':k,m,n)$ is reachable, i.e., the system can always go back to the same state. Using the notation introduced in Section 3.2.1, this can be formally stated as:

$$[t_\omega \mapsto \mathtt{eqs}'' : \cdot] \longrightarrow \forall k,m,n.\ \mathop{\&}_{k,m,n}\ \mathtt{state}(t:k,m,n) \multimap F_{k,m,n}$$

where $F_{k,m,n} = \Uparrow t' : (t+1)^+.\mathtt{state}(t':k,m,n) \otimes \top$ and $\mathtt{eqs}''$ is as $\mathtt{eqs}$ but adding the dummy reaction:

$$\begin{aligned}
\mathtt{eq}_d(t) &\stackrel{\mathtt{def}}{=} \left[!^{x.t}\,\mathtt{ct}(A,\mathbf{0})\otimes !^{x.t}\,\mathtt{ct}(B,\mathbf{0})\otimes !^{y.t}\,\mathtt{ct}(B,\mathbf{0})\right]\multimap\\
&\quad [!^{x.(t+1)}\,\mathtt{ct}(A,\mathbf{0})\otimes !^{x.(t+1)}\,\mathtt{ct}(B,\mathbf{0})\otimes !^{y.(t+1)}\,\mathtt{ct}(B,\mathbf{0})]
\end{aligned}$$

The formula $\mathtt{eq}_d(t)$ is similar to the formula we added in the previous section to detect stable states: one can focus on $\mathtt{eq}_d(t)$ only when the concentrations are zero and none of the reactions (r_1, r_2, r_3) can be fired. Then, the concentrations in the next time-unit remain the same.

The proof of the property proceeds as the one in Section 3.2.1. The & connective (on the right) generates several cases to be proved. The case $\mathtt{state}(t:0,0,0)$ is immediate by first focusing on the formula $\mathtt{eq}_d(t)$ and then, focusing on F. In the other cases, it amounts to use some of the reactions of the system. For instance, the proof of the case $\mathtt{state}(t: k_{>0}, m_{>0}, n_{>0})$ proceeds as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{[t_\omega \mapsto \mathtt{eqs}''; \mathtt{state}(t+4 : \mathtt{suc}(k), \mathtt{suc}(m), \mathtt{suc}(n)) : \cdot] \longrightarrow \mathtt{state}(t+4 : \mathtt{suc}(k), \mathtt{suc}(m), \mathtt{suc}(n)) \otimes \top}\ \otimes_R, !^s{}_R, I_R, \top_R}{[t_\omega \mapsto \mathtt{eqs}''; \mathtt{state}(t+4 : \mathtt{suc}(k), \mathtt{suc}(m), \mathtt{suc}(n)) : \cdot] \longrightarrow G}\ D_R, \Bbbu_R}{[t_\omega \mapsto \mathtt{eqs}''; \mathtt{state}(t+3 : k, \mathtt{suc}(m), \mathtt{suc}(\mathtt{suc}(n))) : \cdot] \longrightarrow G}\ R3}{[t_\omega \mapsto \mathtt{eqs}''; \mathtt{state}(t+1 : k, \mathtt{suc}(\mathtt{suc}(m)), \mathtt{suc}(n)) : \cdot] \longrightarrow G}\ R2}{[t_\omega \mapsto \mathtt{eqs}''; \mathtt{state}(t : \mathtt{suc}(k), \mathtt{suc}(m), \mathtt{suc}(n)) : \cdot] \longrightarrow G}\ R1$$

where $G = \Bbbu t' : (t+1)^+.\mathtt{state}(t' : \mathtt{suc}(k), \mathtt{suc}(m), \mathtt{suc}(n)) \otimes \top$ and R_i represents the logical steps resulting from focusing on the formula encoding the reaction r_i. Hence, by using the sequence of reactions $r_1 \longrightarrow r_2 \longrightarrow r_3$, we can discard this case. The other cases follow similarly.

In the context of the TWEAK system, the set of reactions in Equation (5) is akin to the dephosphorylation of the molecule $[\mathtt{RELA\text{-}P}]_{\mathtt{NU}}$ in the domain of the nucleus. As we mentioned before, this protein can be *translocated* between the cytoplasm and the nucleus. We note that the reaction of dephosphorylation takes place only when the phosphorylated form $[\mathtt{RELA\text{-}P}]_{\mathtt{NU}}$ is already in the nucleus for subsequent catalysis and transportation to the cytoplasm to produce $[\mathtt{RELA}]_{\mathtt{CY}}$. Accordingly, the system oscillates between the phosphorylation and the dephosphorylation of that molecule ($[\mathtt{RELA}]_{\mathtt{CY}} \to [\mathtt{RELA\text{-}P}]_{\mathtt{CY/NU}} \to [\mathtt{RELA}]_{\mathtt{CY}}$) between the spaces CY and NU (see Fig. 4).

4. Logical view of P systems

In this section, in order to give a more general picture of our developments, we show how P systems [10,11] can be also characterized as SELL$^{\Bbbu}$ formulas.

P systems are a model 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. Here we shall focus on P systems with boundary rules [11,22], a variant of P systems where the rewriting rules are not internal to a region but, rather, they are able to see also the external environment (i.e., the boundary). More precisely,

Definition 1 *(P system with boundary rules).* A P system with boundary rules 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 two kinds:

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

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 *moves* objects through membranes: the multiset x' (resp. y') is moved inside (resp. outside) the membrane i.

P systems are synchronous systems in the sense that a global clock is assumed and such clock holds for all regions of the system. In each time-unit, a configuration μ_n moves to μ_{n+1} by applying, possibly several times, the rules in R. The evolution of the system is usually defined in a non-deterministic and maximally parallel manner [10]. Roughly, occurrences of objects in the system are assigned to rules in R until no further assignment is possible (*micro* steps). Then, all the chosen rules are applied exhaustively to produce the new configurations (a *macro* step).

In the following, we shall encode P systems by using a restricted version of the minimal parallelism semantics defined in [12]. In Section 4.1, we generalize the encoding in order to capture different degrees of parallelism in the semantics.

Notation 2 *(Reduction relation).* Given a set of rules R and 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 $\mathfrak{s}_2$ by applying once (if possible) each rule in R.

The above reduction relation is akin to *minimal* parallelism in [12] since rules are not exhaustively applied to all the objects of the system, as in the case of maximal parallelism [10,11]. The non-determinism of $\Longrightarrow$ comes from the fact that rules can compete for the same resource in the system. We note that, compared to the semantics in [12], $\Longrightarrow$ applies each rule at most once (and not several times).

Predicate symbols. We start defining the predicate symbols used in the encoding.

- $\mathtt{p}(a_1, \ldots, a_n)$. Assuming a set of n different components, the current state of the system, at time-unit t in the membrane s_i, is defined as the formula $!^{s_i.t}\,\mathtt{p}(a_1, \ldots, a_n)$ (see Fig. 3 for the ordering of the subexponentials).
- $\mathtt{f}(a_1, \ldots, a_n)$. Rules manipulate the state of the system by consuming elements in the current time-unit and then, producing new ones in the next time-unit. Hence, we shall use the formula $!^{s_i.t}\,\mathtt{f}(a_1, \ldots, a_n)$ to specify that, in the next time-unit, there will be a_i additional units of the component A_i in the membrane s_i.

- $\mathtt{ok}_i$, $\mathtt{tk}$. As explained below, we shall use these atomic propositions, respectively, to detect whether one rule was already applied ($\mathtt{ok}_i$) and to enable the set of reactions in a given time-unit ($\mathtt{tk}$).

Encoding of rules. In the encodings below, we assume a set of reactants $\mathscr{A} = \{A_1, \ldots, A_n\}$ and we use the following notation: $\vec{x} = x_1, \ldots, x_n$ (similarly for $\vec{y}, \vec{z}$ and $\vec{w}$). We write $\mathtt{f}(\vec{x})$ to denote $\mathtt{f}(x_1, \ldots, x_n)$ and $\mathtt{f}(\vec{x} + \vec{b})$ to denote $\mathtt{f}(x_1 + b_1, \ldots, x_n + b_n)$ (similarly for $\mathtt{p}(\cdot)$).

- Transformation rule:

$$[\![r_w : [_i S \longrightarrow [_i S']\!]_t = \forall \vec{x}\, \vec{y}.[!^{s_i.t}(\mathtt{f}(\vec{x}) \otimes \mathtt{p}(\vec{y})) \multimap [(\bigotimes y_i \geq a_i \multimap (!^{s_i.t}\, \mathtt{f}(\vec{x}+\vec{b}) \otimes !^{s_i.t}\, \mathtt{p}(\vec{y}-\vec{a}) \otimes !^{t}\, \mathtt{ok}_w)) \\ \&(\bigoplus y_i < a_i \multimap (!^{s_i.t}\, \mathtt{f}(\vec{x}) \otimes !^{s_i.t}\, \mathtt{p}(\vec{y}) \otimes !^{t}\, \mathtt{ok}_w))]]$$

  Here, from the initial set of components S, a_k units of A_k are consumed to produce b_l units of A_l in the space domain s_i, leading to the multiset S'.
- Communication rule:

$$[\![r_w : [_j T [_i S \longrightarrow [_j T' [_i S']\!]_t \\ = \forall \vec{x}\, \vec{y}\, \vec{z}\, \vec{w}.[(!^{s_i.t}(\mathtt{f}(\vec{x}) \otimes \mathtt{p}(\vec{y})) \otimes !^{s_j.t}(\mathtt{f}(\vec{z}) \otimes \mathtt{p}(\vec{w}))) \multimap \\ [((\bigotimes y_i \geq a_i \otimes \bigotimes w_i \geq c_i) \multimap (!^{s_i.t}\, \mathtt{f}(\vec{x}+\vec{b}) \otimes !^{s_i.t}\, \mathtt{p}(\vec{y}-\vec{a}) \otimes !^{s_j.t}\, \mathtt{f}(\vec{z}+\vec{d}) \otimes !^{s_j.t}\, \mathtt{p}(\vec{w}-\vec{c}) \otimes !^{t}\, \mathtt{ok}_w)) \\ \&((\bigoplus y_1 < a_i \oplus \bigoplus w_i < c_i) \multimap (!^{s_i.t}\, \mathtt{f}(\vec{x}) \otimes !^{s_i.t}\, \mathtt{p}(\vec{y}) \otimes !^{s_j.t}\, \mathtt{f}(\vec{z}) \otimes !^{s_j.t}\, \mathtt{p}(\vec{w}) \otimes !^{t}\, \mathtt{ok}_w))]]$$

  Here, T (resp. S) is the initial multiset of components in the space domain s_j (resp. s_i). The rule consumes a_k units of A_k in s_i and c_k units of A_k in s_j in order to produce b_l units of A_l in s_i (resp. d_l units of A_l in s_j).

The first implication in each rule, that we shall call *available rule*, is similar to the encodings studied in Section 3. We note that the elements are consumed in the current time-unit but the products are "stored" in the predicate $\mathtt{f}(\cdot)$ since they must be available only in the next-time unit. The new part is the second implication that we call the *absence rule*. In this implication we check whether there are not enough resources to fire the rule. Hence, the concentrations remain the same. We note that either the *available* rule or the *absence* rule are fired but not both due to the & connective. Furthermore, the choice is determined entirely by the current concentration of the components (i.e., the predicates $\mathtt{p}(\cdot)$ in each space). Finally, note that the *available* and *absence* rules add the formula $\mathtt{ok}$ which is needed as we explain in brief.

Remark 1. In the above encodings we use the relational symbols $\geq$ and $<$. Those symbols require a set of axioms to define their meaning. Adding such theory in the logical context would imply that, in the following results, we have to analyze the cases when an axiom of such theory is focused on. We note that this is not necessary since we can rewrite such formulas. Consider for instance a transformation rule that consumes two tokens per time-unit from membrane i. Instead of encoding such a rule as

$$\forall x, y.[!^{s_i.t}(\mathtt{f}(x) \otimes \mathtt{p}(y)) \multimap (y \geq 2 \multimap (!^{s_i.t}\, \mathtt{f}(x) \otimes !^{s_i.t}\, \mathtt{p}(y-2) \otimes !^{t}\, \mathtt{ok}_w))] \\ \&(y < 2 \multimap (!^{s_i.t}\, \mathtt{f}(x) \otimes !^{s_i.t}\, \mathtt{p}(y) \otimes !^{t}\, \mathtt{ok}_w))]$$

we can encode it as

$$[\forall x, y.\, !^{s_i.t}(\mathtt{f}(x) \otimes \mathtt{p}(\mathtt{suc}^2(y))) \multimap (!^{s_i.t}\, \mathtt{f}(x) \otimes !^{s_i.t}\, \mathtt{p}(y) \otimes !^{t}\, \mathtt{ok}_w)] \\ \&[\forall x.\, !^{s_i.t}(\mathtt{f}(x) \otimes \mathtt{p}(\mathbf{0})) \multimap (!^{s_i.t}\, \mathtt{f}(x) \otimes !^{s_i.t}\, \mathtt{p}(\mathbf{0}) \otimes !^{t}\, \mathtt{ok}_w)] \\ \&[\forall x.\, !^{s_i.t}(\mathtt{f}(x) \otimes \mathtt{p}(\mathtt{suc}(\mathbf{0}))) \multimap (!^{s_i.t}\, \mathtt{f}(x) \otimes !^{s_i.t}\, \mathtt{p}(\mathtt{suc}(\mathbf{0})) \otimes !^{t}\, \mathtt{ok}_w)]$$

In the first encoding, in a positive phase, the connectives $\forall$ and $\multimap$ (on the left) are introduced. Then, the focusing persists on & choosing one of the branches ($y \geq 2$ or $y < 2$). Finally, in a negative phase, it adds the resulting formulas into the context. Similarly, in the second encoding, the & connective chooses one of the branches representing, respectively, $y \geq 2$, $y = 0$ and $y = 1$. Then, focusing persists on $\forall$ and $\multimap$. Finally, in a negative phase, the predicates are stored into the context.

Auxiliary Formulas. Our encoding also requires the following formulas to control the execution of the rules:

$$\mathtt{next}(t) \quad \stackrel{\mathtt{def}}{=} \quad \forall \vec{x}\, \vec{y}.[!^{t}(\mathtt{ok}_1 \otimes \cdots \otimes \mathtt{ok}_m) \otimes \bigotimes_{i \in O} (!^{s_i.t}\, \mathtt{p}(\vec{x}_i) \otimes !^{s_i.t}\, \mathtt{f}(\vec{y}_i)) \\ \multimap !^{t+1}\, \mathtt{tk} \otimes \bigotimes_{i \in O} (!^{s_i.t}\, \mathtt{p}(\vec{x}_i + \vec{y}_i) \otimes !^{s_i.t}\, \mathtt{f}(\mathbf{0}, \ldots, \mathbf{0}))]$$

$$\mathtt{system}(t) \quad \stackrel{\mathtt{def}}{=} \quad !^{t_\omega} \parr t_x : 0^+.\, !^{t_x}\, \mathtt{tk} \multimap (\mathtt{next}(t_x) \otimes \bigotimes_{r_j \in \mathscr{R}} [\![r_j]\!]_{t_x})$$

$$\mathtt{state}(t, \vec{a_1}, \ldots, \vec{a_n}) \quad \stackrel{\mathtt{def}}{=} \quad !^{t}\, \mathtt{tk} \otimes \bigotimes_{i \in O} [!^{s_i.t}\, \mathtt{p}(\vec{a}_i) \otimes !^{s_i.t}\, \mathtt{f}(\mathbf{0}, \ldots, \mathbf{0})]$$

where $\vec{a}$ is the current (initial) concentration of the components. 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. 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, at most once, 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 $t+1$.

In the encoding presented here, we cannot prove that one flip of the polarity in the proof corresponds exactly to a macro step ($\Longrightarrow$) of the P system. The reason is simple: when a rule is fired (i.e., a micro step is performed), we have a flip in the polarity of the proof. Hence, applying the k rules of the system at the time-unit t requires flipping $k+2$ times the polarity of the proof. The "+2" is due to the extra phases needed to consume the token $\mathtt{tk}$ and decompose the formula $\mathtt{next}$ as shown in the proof of the following theorem.

Theorem 2 *(Adequacy). Let $\mathfrak{s}_1$ and $\mathfrak{s}_2$ be states, $[\![\mathfrak{s}]\!]_t = \mathtt{state}(t, \vec{a}_1, \ldots, \vec{a}_n)$ and $t \geq 0$. Then, $\mathfrak{s}_1 \Longrightarrow \mathfrak{s}_2$ iff the sequent $\mathtt{system}(t), [\![\mathfrak{s}_1]\!]_t \longrightarrow [\![\mathfrak{s}_2]\!]_{t+1}$ is provable.*

Proof. Consider the sequent

$$[\cdot : \cdot], \mathtt{system}(t), [\![\mathfrak{s}_1]\!]_t \longrightarrow [\![\mathfrak{s}_2]\!]_{t+1}$$

We shall show that a proof of such sequent corresponds to the operational step $\mathfrak{s}_1 \Longrightarrow \mathfrak{s}_2$.

We start with a negative phase by decomposing the $!^s$ and $\otimes$ connectives on the left:

$$\frac{[t_\omega \mapsto F; t \mapsto \mathtt{tk}; \{s_i \mapsto \{\mathtt{p}(\vec{a}_i), \mathtt{f}(\mathbf{0}, \ldots, \mathbf{0})\}\}_i : \cdot] \longrightarrow [\![\mathfrak{s}_2]\!]_{t+1}}{[\cdot : \cdot], \mathtt{system}(t), [\![\mathfrak{s}_1]\!]_t \longrightarrow [\![\mathfrak{s}_2]\!]_{t+1}} \otimes_L, !^s_L$$

where $F = \Bbbwedge t_x : 1^+ . !^{t_x}\, \mathtt{tk} \multimap (\mathtt{next}(t_x) \otimes \bigotimes_{r_j \in \mathscr{R}} [\![r_j]\!]_{t_x})$. We note that the negative phase ends here and, for each space i, the context s_i stores the corresponding concentration of the components $(\vec{a}_i)$. To move up in the derivation, the only choice we have on the left is to focus on the formula stored in the context t_ω (i.e., F) and we observe the following:

$$\dfrac{\dfrac{\dfrac{}{[t \mapsto \mathtt{tk} : \cdot] \longrightarrow \mathtt{tk}}\, D_R, I_R}{[t_\omega \mapsto F; t \mapsto \mathtt{tk} : \cdot] \xrightarrow{!^t\, \mathtt{tk}}}\, !^s_R \qquad \dfrac{\Psi}{\dfrac{[t_\omega \mapsto F \ldots : \mathtt{next}(t), [\![r_1]\!]_t, \cdots, [\![r_m]\!]_t] \longrightarrow [\![\mathfrak{s}_2]\!]_{t+2}}{\mathtt{next}(t) \otimes \bigotimes_{r_j \in \mathscr{R}} [\![r_j]\!]_t}\, R_L, \otimes_L, St_L \qquad [t_\omega \mapsto F : \cdot] \longrightarrow [\![\mathfrak{s}_2]\!]_{t+1}}{!^t\, \mathtt{tk} \multimap (\mathtt{next}(t) \otimes \bigotimes_{r_j \in \mathscr{R}} [\![r_j]\!]_t)}\, \multimap_L}}{[t_\omega \mapsto F \ldots : \cdot] \longrightarrow [\![\mathfrak{s}_2]\!]_{t+1}}\, \Bbbwedge_R$$

$$\frac{[t_\omega \mapsto F \ldots : \cdot] \xrightarrow{F} [\![\mathfrak{s}_2]\!]_{t+1}}{[t_\omega \mapsto F \ldots : \cdot] \longrightarrow [\![\mathfrak{s}_2]\!]_{t+1}}\, D_{L1}$$

where $\mathtt{tk}$ is proved (in the subexponential t) and the formula $\mathtt{next}(t)$ and the encoding of the reactions are stored in the context.

In Ψ, we can continue by focusing again on the formula F (stored in t_ω). However, in that case, we cannot finish the proof since $\mathtt{tk}$ is not in the context t and it cannot be produced by focusing on F (nor by focusing on $\mathtt{next}(t)$). Hence, the only choice is to focus on one of the formulas of the shape $[\![r_j]\!]_t$. Such action will be similar to the derivations of $\mathtt{consume}$ and $\mathtt{produce}$ in the proof of Theorem 1. We also note that we can only focus on $\mathtt{next}(t)$ when all the formulas of the shape $[\![r_j]\!]_{t_x}$ were used, thus adding the tokens $\mathtt{ok}_i$ into the context.

Hence, due to the focusing discipline, we can guarantee that: the set of reactions are copied to the location t only if the predicate $\mathtt{tk}$ is in that location; then, the set of reactions are executed (each one in a change of the polarity of the proof); when all the reactions are executed, one can focus on the formula $\mathtt{next}(t)$ to allocate the resources (and the formula $\mathtt{tk}$) in the next time-unit. That is, after $k+2$ flips of the polarity we observe a macro step where the logical context encodes the state $\mathfrak{s}_2$. □

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 $\mathtt{system}(0), [\![\mathfrak{s}_1]\!]_0 \longrightarrow [\![\mathfrak{s}_1]\!]_{n+1}$ is provable. More generally, we can find such 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 $\mathtt{system}(0), [\![\mathfrak{s}_1]\!]_0 \longrightarrow \Bbbvee l : 1^+ . [\![\mathfrak{s}_1]\!]_l$.

4.1. Maximal and minimal parallelism semantics

In this section we show how the encodings studied in the previous section can be adapted in order to deal with other notions of parallelism in P systems. It turns out that we only need to control the way the token $\mathtt{ok}$ is added into the context.

In the following, we consider a predicate $\mathtt{rule}(i)$ that is added into the context when the rule i can be executed. The transformation rule $[_i S \longrightarrow [_i S'$ is encoded as

$$
\begin{array}{l}
[\![[_i S \longrightarrow [_i S']\!]_t = \\
!^t\,\mathtt{rule}(w) \multimap [\\
\forall \vec{x}\ \vec{y}.[(!^{s_i.t}(\mathtt{f}(\vec{x}) \otimes \mathtt{p}(\vec{y}))) \multimap [(\bigotimes y_i \geq a_i \multimap (!^{s_i.t}\,\mathtt{f}(\vec{x}+\vec{b}) \otimes\, !^{s_i.t}\,\mathtt{p}(\vec{y}-\vec{a}) \otimes\, !^t\,\mathtt{ok}_w)) \\
\qquad\qquad\qquad\qquad \&(\bigotimes y_i \geq a_i \multimap (!^{s_i.t}\,\mathtt{f}(\vec{x}+\vec{b}) \otimes\, !^{s_i.t}\,\mathtt{p}(\vec{y}-\vec{a}) \otimes\, !^t\,\mathtt{rule}(w))) \\
\qquad\qquad\qquad\qquad \&(\bigoplus y_i < a_i \multimap (!^{s_i.t}\,\mathtt{f}(\vec{x}) \otimes\, !^{s_i.t}\,\mathtt{p}(\vec{y}) \otimes\, !^t\,\mathtt{ok}_w))]]
\end{array}
$$

If there are enough resources (lines 2 and 3 above), one can decide to either add $\mathtt{ok}_w$ or $\mathtt{rule}(w)$ to the context after updating the state of the system. In the first case, the rule is applied only once in the current time-unit. In the second case, there is a chance of applying it again. The formula in the last line, as in the previous encodings, is chosen when the rule cannot be applied in the current state. The encoding of communication rules can be adapted similarly.

Since rules can be applied several times during the same time-unit, we need to adapt also the definition of system as follows:

$$
\begin{array}{rl}
\mathtt{system}(t) \stackrel{\mathtt{def}}{=} & !^{t_\omega} \Cap t_x : 0^+.\, !^{t_x}\,\mathtt{tk} \multimap (\mathtt{next}(t_x) \otimes \bigotimes_{r_i \in \mathscr{R}} !^{t_x}\,\mathtt{rule}(r_i)) \\
& \otimes\, !^{t_\omega} \Cap t_x : 0^+.[\![r_j]\!]_{t_x}
\end{array}
$$

Note that the formula $\mathtt{system}$ stores the encoding of the rules in the unbounded subexponential t_ω. Moreover, when $\mathtt{tk}$ can be deduced in time-unit t_x, the formula $\mathtt{next}(t_x)$, as well as the formulas $\mathtt{rule}(r_i)$, are added into the context.

The above encoding allows us to use the same rule zero or several times and such choice is non-deterministic (due to the & connective). Hence, similarly to Theorem 2, we can show that in each focused step, we observe the execution of one of the rules in a given time-unit (micro steps). However, we cannot bound the number of flipping of polarities needed to move to the next time-unit (macro step).

The maximal parallelism semantics (where all the rules must be exhaustively applied in all the possible objects of the system) can be characterized in SELL$^{\Cap}$ by encoding the rules of the system as follows:

$$
\begin{array}{l}
!^t\,\mathtt{rule}(w) \multimap [\\
\forall \vec{x}\ \vec{y}.[(!^{s_i.t}(\mathtt{f}(\vec{x}) \otimes \mathtt{p}(\vec{y}))) \multimap [(\bigotimes y_i \geq a_i \multimap (!^{s_i.t}\,\mathtt{f}(\vec{x}+\vec{b}) \otimes\, !^{s_i.t}\,\mathtt{p}(\vec{y}-\vec{a}) \otimes\, !^t\,\mathtt{rule}(w))) \\
\qquad\qquad\qquad\qquad \&(\bigoplus y_i < a_i \multimap (!^{s_i.t}\,\mathtt{f}(\vec{x}) \otimes\, !^{s_i.t}\,\mathtt{p}(\vec{y}) \otimes\, !^t\,\mathtt{ok}_w))]]
\end{array}
$$

In this case, the token $\mathtt{ok}_w$ is added only when there are not enough resources to apply the rule again.

Finally, if we were to consider sequential P systems, where only one rule is used in each step of a computation [23], the resulting encoding would be closer to the one in Section 3. More precisely, the encodings of the rules must be glued with the connective & in order to choose one of the available rules. Moreover, the predicate $\mathtt{ok}$ and the formula $\mathtt{next}$ would not be necessary to control the change of the time-unit. In this case, one flip of the polarity of the proof corresponds exactly to one step in the computation.

5. Logical frameworks and the verification problem

In the previous sections we have shown that SELL is able to specify and verify biochemical systems where temporal and spatial modalities can be combined. Besides the applications shown in Sections 3 and 4, it is important to further discuss the practical and theoretical implications of our results.

The main goal of this paper is to use theorem proving techniques for the verification of biochemical systems. For that, we followed two design principles: (1) the foundations must be settled on a logical framework with good proof theoretical properties. Moreover, the (meta) theory of the framework should help us drawing new conclusions of the studied systems; (2) the language of properties must be expressive enough to declaratively specify properties constrained by temporal and spatial modalities. Additionally, proofs in the logical framework must accurately characterize (operational) steps in the system. Let us elaborate on how we achieved (1) and (2) and compare our developments with other proposals in the literature.

5.1. Cut elimination and focusing

Linear logic [6] is a very expressive and elegant logical framework. The distinction between multiplicative and additive connectives offers a very precise control on the resources (formulas) during a proof. Such control is even more expressive in the case of SELL [7] where one can split the logical context into different parts, each representing a given modality (e.g., spatial locations or time-units). Our encodings rely extensively on such strict use of resources, and specially, on the promotion rule ($!^s_R$) that limits the use of resources to the ones related to the goal we are proving.

Linear logic and SELL have good proof theoretical properties: the $\mathtt{cut}$ rule below is admissible [6] (cut elimination) and it admits a sound and complete focused [13] proof system.

The cut rule in linear logic is as follows:

$$\frac{\Gamma \longrightarrow G \quad \Delta, G \longrightarrow F}{\Gamma, \Delta \longrightarrow F}\ \texttt{cut}$$

Intuitively, we prove the intermediate lemma G (using Γ) and then, using G, we prove the desired result F. The cut-elimination procedure shows that any proof with `cut` can be transformed into a (possibly larger) proof without `cut`. This result is fundamental in proof theory (see e.g., [20]). From it, we can show that the system is consistent (since we cannot derive falsity from the empty set of premises). Moreover, the system has the subformula property (only subformulas of the root sequent can appear in a proof).

The cut elimination procedure entails also an important result from the theoretical and practical points of view: it allows us to use intermediate lemmas to finish a proof. As a direct consequence, we have compositionality of properties: if the subsystem represented by Γ exhibits some behavior G and, assuming G we know that Δ exhibits F, then the whole system exhibits F. We can thus build libraries of (formally proved) theorems about biochemical systems, as it already happens in Mathematics (see e.g., the list of math theorems formalized in different theorems provers at http://www.cs.ru.nl/~freek/100/).

From the computational point of view, the `cut` rule is difficult to handle in an automatic procedure: the computer should guess the intermediate lemma G to continue the proof. The automatization of part of the verification technique comes from the focused system. As we already showed, focusing allows us to reduce the non-determinism during proof search (i.e., search for a cut-free proof). Therefore, it is possible to have semi-automatic procedures where the expert can compose previously proved lemmas and the computer handles the automatic proof search procedure. Such integration of automatic techniques and user-guided proofs are already available for some frameworks in the context of biological systems [24].

5.2. Theory of SELL

The meta theory of SELL brings important benefits for the verification of biochemical systems. In the following, we highlight some of these features.

As hinted above, the focused system allows us to control the shape of the proofs and reduce the non-determinism during the proof search procedure. This result was fundamental to show tight adequacy results relating proofs and steps in the specified system. Controlling the shape of the derivations opens also the possibility of using other reasoning techniques at the meta level. Take for instance the example in Section 3.2.1 where we used induction to verify a reachability property. In this case, we could use induction only because focusing guaranteed the exact shape of the formulas (and the context) in a step of the derivation.

Modeling spatial property propagation in Biological Systems. The subexponential structure may also play an important role for verification at the meta level. Consider for instance that we have a proof of the sequent $\Gamma \longrightarrow !^a F$ showing that the system represented by Γ exhibits certain behavior on the space domain a. We know that in SELL provability is preserved *downwards*, i.e., from a proof of $\Gamma \longrightarrow !^a F$ we can also conclude $\Gamma \longrightarrow !^b F$ for any $b \preceq a$. This corresponds, intuitively, to show that F holds in every subordinated (w.r.t. $\preceq$) location in the system. This precise control on the hierarchy (defined by $\preceq$) can be used, for instance, to specify systems where stimuli/actions are propagated into the internal spatial structure. Take for instance the process of oncosis/necrosis [25], a pre-lethal pathway leading to cell death accompanied by cellular swelling, organelle swelling, blebbing, and increased membrane permeability. This process is caused by physical disruption on cellular structure and function through injury, bacterial toxins, or nutritional deprivation.

The use of subexponentials allows us to divide the logical context in order to neatly distinguish different subsystems. We can thus derive *local* information from each subsystem. In order to draw more general conclusions of the whole system, we can combine local properties using the rule `cut`. From the point of view of automatic procedures, the promotion rule has also an important consequence in practice: non-related subexponentials must be dropped (weakened) from the context. This allows us to "simplify" the context of a proof by (safely) discarding some part of it that cannot "interact" with the formula we are proving.

Another interesting meta-theoretic result that is entailed from our approach is that, whenever we prove an existential property from a set of (encoded) rules, the property is satisfied also for larger systems (due to weakening and the use of subexponentials). This indeed is quite interesting since it gives some compositionality for verification which is not possible, in general, in Model Checking.

Some other (practical and theoretical tools) in proof theory may also be useful for our verification task. For instance, in theorem provers, it is possible to greatly simplify proofs of properties with an existential quantifier on the right (or universal quantifiers on the left) by using unification (see, e.g., [26]). Roughly, the term to be instantiated does not need to be determined immediately but delayed until the application of the initial rule. This meta-theoretic result may simplify the proof of periodicity properties (as the one before Section 4.1) in our framework.

5.3. SELL and other logics

There exist other logics such as CTL (Computation tree logic), temporal logic [27], spatial logics [28], among several others, that we may have used as foundations for our framework. We preferred SELL mainly due to its proof theoretic

elegance, in particular, its proof system with good properties. The above mentioned logics, as far as we know, do not have cut-free nor focused proof systems. Then, devising verification procedures for those logics implies building specific structures/procedures to use model checking (see e.g., [29]). Moreover, compositionality is not guaranteed as in the case of SELL.

On the other side, the above mentioned logics aim only to specify properties of the system but not the system itself. In that sense, those logics are not a logical framework as in the case of SELL: inside SELL, we can specify, verify and also compare the behavior of different systems inside the same logical framework. We believe that this is an interesting feature of our approach since properties (goals in the logic) can be expressed using the same language, and reachability corresponds directly to logical entailment. The language of properties is quite rich and it includes spatial and temporal modalities (the main feature of our framework). For instance, we can express spatial and temporal patterns like "once reaction r happens, reaction r' can be fired in the space domain a". Some other properties, e.g., oscillations, can be naturally expressed as shown in Section 3.2.3.

The representation of the system inside SELL, as shown by Theorems 1 and Theorem 2 (adequacy) is not ad-hoc. We showed that the maximal level of adequacy can be achieved and then, a derivation in the logic corresponds exactly to a possible trajectory/configuration of the system. By looking at such logical representation of the system, we have been able to characterize, in a modular way, different concurrent semantics for P systems. We note that, in other frameworks, usually different structures/procedures are needed to deal with different semantics [30].

5.4. Model checking, theorem proving and concurrency theory

In the literature, there are several tools and techniques for the verification of biochemical systems (see also the related work section below). In particular, there exists an extensive work on using model checkers for the verification of P systems. In the following, we highlight some of the differences between Model Checking [31] (MC) techniques and our Theorem Proving approach (TP).

We first note that TP and MC are complementary (and rather different) techniques. MC is an automatic technique that usually requires a finite state model of the system. TP is a semi-automatic technique where some proofs need to be "assisted" by the user.

P systems, in general, may generate an infinite number of states. In fact, reachability is undecidable [30] since simple fragments of P systems are Turing complete. Then, only an abstraction of the systems can be verified by MC techniques. TP can deal with infinite state systems. This, of course, comes with a price: the technique cannot be fully mechanizable and an expert is needed to find the right invariants, e.g., in proofs by induction or using `cut`.

MC is a useful companion for a theorem prover. A good strategy for understanding the behavior of a system is to use a model checker to verify a (finite/abstract) representation of the system. Once the property is verified on that abstraction of the system, we may attempt to formally prove the property in the whole system. Then, TP offers the possibility to combine already proved results to conclude more facts.

MC explicitly requires a (symbolic) representation of the transition system. The need for symbolic/abstract representations of the system comes from the inherent state-explosion problem of this technique. In TP, we encode the system and the desired property as a SELL formula. Hence, proving a property requires the application of logical rules and it is not required to explicitly build the transition system.

It is worth noticing that the proof theory of SELL was also useful to understand different concurrent behaviors in P Systems. As shown in Section 4.1, our encodings can modularly capture different degrees of parallelism in P systems in the same logical framework. In MC techniques (see, e.g., [30]), the structures/procedures must be adjusted in order to verify P systems with different parallel semantics.

We believe that the connections between logic (proof theory), concurrency theory and biochemical system may have much to offer. To be more precise, we have shown in [9] that Concurrent Constraint (CCP) languages have a strong connection with SELL and then, such languages can be seen as a runnable specification of (fragments of) SELL. In [32,33] we have shown that CCP can indeed be used as a declarative language to implement simulation tools for biochemical systems. Such tools can be improved by the recent findings in [34], where optimizations for CCP were derived from the proof theory of linear logic. Similarly, we hope that the results in this paper and the body of knowledge in proof theory will allow us to build more automatic (and efficient) theorem provers to verify biochemical systems.

6. Concluding remarks

We presented a formal method to specify and verify computational biological systems grounded on proof theory. We have dealt with the problem of representing both spatial and time-dependent information. Our proposal relies on linear logic [6] with subexponentials (SELL$^{\Cap}$) [7–9]. We have shown examples of properties of biochemical systems that can be proven in our framework. Moreover, we have proved that our logical characterization has a strong level of adequacy: derivations in the logical system follow exactly the rules (reactions) of the modeled system. We have then shown that our framework is general enough to give a logical characterization to P Systems. In other terms, the embeddings presented show 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 linear logic 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 to use an assisted theorem prover (e.g., Coq) to have a semi-automatic process of verification (see e.g., [24]). Providing skeletons/templates to ease the specification of properties may be also useful.

Related work. Many works in the literature (see e.g., [14–17,8]) have shown that linear logic is general enough to give a proof-theoretic account of many concurrent systems and formalisms. In all these works, computation amounts to proofs, thus bringing new tools and frameworks to study those systems. In the context of biochemical systems, a work closely related to ours is [24] where the authors show that temporal properties of biochemical systems can be expressed in HyLL (hybrid linear logic). 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$^{\Uparrow}$. In this work, we show that time and spatial modalities can be examined in the same framework. Moreover, the focusing discipline allowed us to show stronger adequacy results.

In [32], we proposed a tailored Concurrent Constraint Programming-based [35] formalism for dealing with spatial modalities. However, in [32] we could not deal with temporal information, and we did not study thoroughly how to specify and prove properties of our systems.

The notion of probabilities and preferences have been introduced in HyLL and SELL in [24] and [36] respectively. This may open the possibility to enhance our encodings to consider set of rules that can be chosen according to priorities or probabilities. In fact, we can encode P systems with priorities (as defined in [10]) straightforwardly. In this semantics, if r_1 has higher priority than r_2, then r_2 can be applied only if there are not more resources to apply r_1. This would correspond to guard the encoding of r_2 with a token added by the encoding of r_1 in its *absence* rule.

There are also other formalisms for specifying time modalities. For instance, in Pathway Logic (PL) [4], each rule (biochemical reaction) is associated with a scalar value called affinity. Such value can be bounded by a time-dependent interpretation either by using exponential random or deterministic amortized variables. In [37] a timed-π-calculus is used to deal with time-stamps and clocks. These are handled as other names and transmitted through channels.

For dealing with spatial information, formalisms such as Bio-Pepa [2], BioNetGen [38], BioAmbients [39], and Brane Calculi [40] embed a tree representation of the hierarchical structure of cellular compartments. Alternatively, in Biocham [41], PL [4], and Beta-Binders [42], cellular compartments can be abstracted as symbolic locations by assigning *labels* to molecular compounds. In the $\pi@$-calculus [43], restricted names are exploited to model compartments.

Other frameworks allow reasoning about biological properties by using different types of logics and techniques (usually the formalisms for modeling and that for proving properties are different). Properties of Biocham models [1], for instance, can be formalized within the boolean, differential and stochastic semantics by using (probabilistic) temporal logics. Bio-Pepa's models [2] can be translated into PRISM [44], a probabilistic model checker. Processes in PL [4] can be analyzed by using the Maude system. Bounded Linear Temporal Logic [45] 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 [46] or modal logics to express spatial and temporal modalities [47]. Similarly, modal logic can be used to express spatial and temporal properties over membranes and systems, which is known as Brane Logic [48]. In the case of Beta-Binders models, causality properties [42] as well as flow control analyses [49] can be performed.

Acknowledgements

We thank the anonymous reviewers for their detailed comments and suggestions which helped us to improve our paper. We also thank Joëlle Despeyroux for her helpful suggestions, specially for her contributions on Section 5. The work of Olarte was funded by CNPq (Brazil).

References

[1] F. Fages, S. Soliman, Formal cell biology in Biocham, in: M. Bernardo, P. Degano, G. Zavattaro (Eds.), SFM 2008, in: LNCS, vol. 5016, Springer, 2008, pp. 54–80.
[2] F. Ciocchetta, M.L. Guerriero, Modelling biological compartments in Bio-PEPA, Electron. Notes Theor. Comput. Sci. 227 (2009) 77–95.
[3] L. Dematté, C. Priami, A. Romanel, The BlenX language: a tutorial, in: M. Bernardo, P. Degano, G. Zavattaro (Eds.), SFM 2008, in: LNCS, vol. 5016, Springer, 2008, pp. 313–365.
[4] A. Abate, Y. Bai, N. Sznajder, C.L. Talcott, A. Tiwari, Quantitative and probabilistic modeling in pathway logic, in: BIBE 2007, IEEE CS, 2007, pp. 922–929.
[5] J. Fisher, T.A. Henzinger, Executable biology, in: L.F. Perrone, B. Lawson, J. Liu, F.P. Wieland (Eds.), WSC, 2006, pp. 1675–1682.
[6] J. Girard, Linear logic, Theoret. Comput. Sci. 50 (1987) 1–102.
[7] V. Danos, J. Joinet, H. Schellinx, The structure of exponentials: uncovering the dynamics of linear logic proofs, in: G. Gottlob, A. Leitsch, D. Mundici (Eds.), KGC'93, in: LNCS, vol. 713, Springer, 1993, pp. 159–171.
[8] V. Nigam, C. Olarte, E. Pimentel, A general proof system for modalities in concurrent constraint programming, in: P.R. D'Argenio, H.C. Melgratti (Eds.), CONCUR 2013, in: LNCS, vol. 8052, Springer, 2013, pp. 410–424.
[9] C. Olarte, E. Pimentel, V. Nigam, Subexponential concurrent constraint programming, Theoret. Comput. Sci. 606 (2015) 98–120.
[10] G. Păun, G. Rozenberg, A guide to membrane computing, Theoret. Comput. Sci. 287 (1) (2002) 73–100.
[11] F. Bernardini, V. Manca, Dynamical aspects of P systems, Biosystems 70 (2) (2003) 85–93.
[12] G. Ciobanu, L. Pan, G. Păun, M.J. Pérez-Jiménez, P systems with minimal parallelism, Theoret. Comput. Sci. 378 (1) (2007) 117–130.
[13] J.-M. Andreoli, Logic programming with focusing proofs in linear logic, J. Logic Comput. 2 (3) (1992) 297–347.
[14] F. Fages, P. Ruet, S. Soliman, Linear concurrent constraint programming: operational and phase semantics, Inform. and Comput. 165 (1) (2001) 14–41.
[15] K. Watkins, I. Cervesato, F. Pfenning, D. Walker, A concurrent logical framework: the propositional fragment, in: S. Berardi, M. Coppo, F. Damiani (Eds.), TYPES 2003, in: LNCS, vol. 3085, Springer, 2003, pp. 355–377.

[16] A.F. Tiu, D. Miller, A proof search specification of the pi-calculus, Electron. Notes Theor. Comput. Sci. 138 (1) (2005) 79–101.
[17] I. Cervesato, A. Scedrov, Relating state-based and process-based concurrency through linear logic (full-version), Inform. and Comput. 207 (10) (2009) 1044–1077.
[18] D. Chiarugi, M. Falaschi, D. Hermith, C. Olarte, Verification of spatial and temporal modalities in biochemical systems, Electron. Notes Theor. Comput. Sci. 316 (2015) 29–44.
[19] M. Bhattacharjee, R. Raju, A. Radhakrishnan, et al., A bioinformatics resource for TWEAK-Fn14 signaling pathway, J. Signal Transduct. (2012).
[20] S. Negri, J. von Plato, Structural Proof Theory, Cambridge University Press, 2001.
[21] D. Miller, A. Saurin, From proofs to focused proofs: a modular proof of focalization in linear logic, in: J. Duparc, T.A. Henzinger (Eds.), CSL 2007, in: LNCS, vol. 4646, Springer, 2007, pp. 405–419.
[22] F. Bernardini, M. Gheorghe, V. Manca, On P systems and almost periodicity, Fund. Inform. 64 (1–4) (2005) 29–42.
[23] R. Freund, Asynchronous P systems and P systems working in the sequential mode, in: G. Mauri, G. Paun, M.J. Pérez-Jiménez, G. Rozenberg, A. Salomaa (Eds.), WMC 2004, in: LNCS, vol. 3365, Springer, 2004, pp. 36–62.
[24] E.D. Maria, J. Despeyroux, A.P. Felty, A logical framework for systems biology, in: Formal Methods in Macro-Biology – First International Conference, FMMB 2014, 2014, pp. 136–155.
[25] S.L. Fink, B.T. Cookson, Apoptosis, pyroptosis, and necrosis: mechanistic description of dead and dying eukaryotic cells, Infect. Immun. 84 (4) (2005) 1907–1916.
[26] D. Miller, G. Nadathur, F. Pfenning, A. Scedrov, Uniform proofs as a foundation for logic programming, Ann. Pure Appl. Logic 51 (1–2) (1991) 125–157.
[27] Z. Manna, A. Pnueli, The Temporal Logic of Reactive and Concurrent Systems – Specification, Springer, 1992.
[28] L. Caires, L. Cardelli, A spatial logic for concurrency – II, Theoret. Comput. Sci. 322 (3) (2004) 517–565.
[29] R. Mardare, C. Priami, P. Quaglia, O. Vagin, Model checking biological systems described using ambient calculus, in: V. Danos, V. Schächter (Eds.), CMSB 2004, in: LNCS, vol. 3082, Springer, 2005, pp. 85–103.
[30] Z. Dang, O.H. Ibarra, C. Li, G. Xie, On the decidability of model-checking for P systems, J. Autom. Lang. Comb. 11 (3) (2006) 279–298.
[31] E.M. Clarke, O. Grumberg, D.A. Peled, Model Checking, MIT Press, 2001.
[32] D. Chiarugi, M. Falaschi, D. Hermith, C. Olarte, A framework for modelling spatially dependent interactions of biological systems in CCP, in: I. Rojas, F.M.O. Guzman (Eds.), IWBBIO 2014, Copicentro Editorial, 2014, pp. 912–923.
[33] D. Chiarugi, M. Falaschi, C. Olarte, C. Palamidessi, A declarative view of signaling pathways, in: C. Bodei, G.L. Ferrari, C. Priami (Eds.), Programming Languages with Applications to Biology and Security, in: Lecture Notes in Computer Science, vol. 9465, Springer, 2015, pp. 183–201.
[34] C. Olarte, E. Pimentel, Proving concurrent constraint programming correct, revisited, Electron. Notes Theor. Comput. Sci. 312 (2015) 179–195.
[35] V.A. Saraswat, M.C. Rinard, P. Panangaden, Semantic foundations of concurrent constraint programming, in: D.S. Wise (Ed.), POPL 1991, ACM Press, 1991, pp. 333–352.
[36] E. Pimentel, C. Olarte, V. Nigam, A proof theoretic study of soft concurrent constraint programming, Theory Pract. Log. Program. 14 (4–5) (2014) 649–663.
[37] N. Saeedloei, G. Gupta, Timed π-calculus, in: M. Abadi, A. Lluch-Lafuente (Eds.), TGC 2013, in: LNCS, vol. 8358, Springer, 2013, pp. 119–135.
[38] L.A. Harris, J.S. Hogg, J.R. Faeder, Compartmental rule-based modeling of biochemical systems, in: A. Dunkin, R.G. Ingalls, E. Yücesan, M.D. Rossetti, R. Hill, B. Johansson (Eds.), WSC, 2009, pp. 908–919.
[39] A. Regev, E.M. Panina, W. Silverman, L. Cardelli, E.Y. Shapiro, Bioambients: an abstraction for biological compartments, Theoret. Comput. Sci. 325 (1) (2004) 141–167.
[40] L. Cardelli, Brane calculi, in: V. Danos, V. Schächter (Eds.), CMSB 2004, in: LNCS, vol. 3082, Springer, 2005, pp. 257–278.
[41] N. Chabrier-Rivier, F. Fages, S. Soliman, The biochemical abstract machine BIOCHAM, in: V. Danos, V. Schächter (Eds.), CMSB 2004, in: LNCS, vol. 3082, Springer, 2005, pp. 172–191.
[42] M.L. Guerriero, C. Priami, A. Romanel, Modeling static biological compartments with beta-binders, in: H. Anai, K. Horimoto, T. Kutsia (Eds.), AB 2007, in: LNCS, vol. 4545, Springer, 2007, pp. 247–261.
[43] C. Versari, R. Gorrieri, pi @: a pi-based process calculus for the implementation of compartmentalised bio-inspired calculi, in: M. Bernardo, P. Degano, G. Zavattaro (Eds.), SFM 2008, in: LNCS, vol. 5016, Springer, 2008, pp. 449–506.
[44] A. Hinton, M.Z. Kwiatkowska, G. Norman, D. Parker, PRISM: a tool for automatic verification of probabilistic systems, in: H. Hermanns, J. Palsberg (Eds.), TACAS 2006, in: LNCS, vol. 3920, Springer, 2006, pp. 441–444.
[45] H. Gong, P. Zuliani, A. Komuravelli, J.R. Faeder, E.M. Clarke, Computational modeling and verification of signaling pathways in cancer, in: K. Horimoto, M. Nakatsui, N. Popov (Eds.), ANB 2010, in: LNCS, vol. 6479, Springer, 2010, pp. 117–135.
[46] R. Gori, F. Levi, An analysis for proving temporal properties of biological systems, in: N. Kobayashi (Ed.), APLAS 2006, in: LNCS, vol. 4279, Springer, 2006, pp. 234–252.
[47] L. Caires, L. Cardelli, A spatial logic for concurrency (part I), Inform. and Comput. 186 (2) (2003) 194–235.
[48] M. Miculan, G. Bacci, Modal logics for brane calculus, in: C. Priami (Ed.), CMSB 2006, in: LNCS, vol. 4210, Springer, 2006, pp. 1–16.
[49] C. Bodei, A control flow analysis for beta-binders with and without static compartments, Theoret. Comput. Sci. 410 (33–34) (2009) 3110–3127.