Available online at www.sciencedirect.com
SCIENCE DIRECT®
Theoretical Computer Science 325 (2004) 111–140
Theoretical Computer Science
www.elsevier.com/locate/tcs

# Modelling biochemical pathways through enhanced $\pi$-calculus☆

M. Curti[a,*], P. Degano[a], C. Priami[b], C.T. Baldari[c]

[a] *Dipartimento di Informatica, Università di Pisa, Via Filippo Buonarroti, 2, I-56127 Pisa, Italy*
[b] *Dipartimento di Informatica e Telecomunicazioni, Università di Trento, Via Sommarive, 14, I-38050 Povo, TN, Italy*
[c] *Dipartimento di Biologia Evolutiva, Università di Siena, Via Mattioli, 4, I-53100 Siena, Italy*

**Abstract**

We use the $\pi$-calculus to model the evolution of biochemical systems, taking advantage of their similarities with global computation applications. First, we present a reduction semantics for the $\pi$-calculus from which causality and concurrency can be mechanically derived. We prove that our semantics agrees with the causal definitions presented in the literature. We also extend our semantics to model biological compartments. Then, we show the applicability of our proposal on a couple of biological examples.


*Keywords:* Concurrency; Causality; Reduction semantics; Systems biology

## 1. Introduction

A network of proteins can be seen as a computing machinery, made of processing agents that cooperate to achieve a common goal. Agents autonomously compute on their own and exchange information each other [51]. This informal description

☆ The first three authors have been partially supported by UE Project IST-32072-DEGAS within the FET initiative on Global Computing. The last two authors have been partially supported by the project FIRB "Metodi Formali per sistemi biochimici". Portions of this paper appeared in the extended abstract M. Curti, P. Degano, C.T. Baldari, *Causal $\pi$-Calculus* for Biochemical Modelling, Proc. CMSB '03, Lecture Notes in Computer Science, Vol. 2602, Springer, 2003.

* Corresponding author. Dipartimento di Informatica, Università di Pisa, Via Sommarive, 14, Povo, TN I-38050, Italy.

*E-mail addresses:* curtim@di.unipi.it, curti@di.unipi.it (M. Curti), degano@di.unipi.it (P. Degano), priami@science.unitn.it (C. Priami), baldari@unisi.it (C.T. Baldari).

0304-3975/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2004.03.066

applies as well to *concurrent system*, that are made of large number of *geographically dispersed*, possibly *mobile* and *communicating* computing agents. This paradigm is now-a-days called *global computing*. *Process calculi* are the most popular formalism to describe and study global computing applications. One of the most popular process calculi is the $\pi$-calculus [39]. Regev et al. [52] were the first to use the $\pi$-calculus as a model of biochemical processes, taking advantage of similar experiences in the Petri net field [17,18,20] and on the results from process calculi for mobility. They model reactants as $\pi$-calculus processes and biochemical reactions as communications. The authors claim that the $\pi$-calculus permits to better integrate dynamics, molecular and biochemical details. Further work lead to a more precise model [47], based on the *stochastic* $\pi$-calculus [45], an extension of the original calculus with probabilistic distributions that govern the race conditions. So, also quantitative aspects of reactions can be taken into account. Besides the study of quantitative aspects like the one mentioned above, the literature on concurrency has many proposals on the description of the causal relations between the activities that agents perform, as well as of the relation of independence between actions (*concurrency*). According to its supporters, causality permits more accurate representations of the behaviour of concurrent systems than classical interleaving representations; in particular, causality seems to play a relevant role in understanding complex biochemical pathways. Also, a single causal representation describes all the behaviour in which independent activities are temporally linearized, thus offering a more concise model. In this paper we offer such a causal extension of the $\pi$-calculus, and we apply it to a couple of biological examples. Technically, we exploit the so-called enhanced operational semantics (EOS for short) [9,10], a non-standard way of describing the behaviour of concurrent systems. In this approach, the transitions of a system from one state to another have rich labels that allows for retrieving many aspects of a computation, including qualitative ones, like causality, and quantitative ones, like those based on probabilistic distributions. Here, we re-phrase the EOS of the $\pi$-calculus proposed in [8] and we obtain a *reduction* semantics for the calculus, that also expresses causality—to the best of our knowledge, this is the first causal reduction semantics available. We also slightly refine our semantics to model biochemical compartments. Some biological interactions require that reactants belong to the same boundary region, and we model this constraint by requiring that processes modelling the reactants are "close" enough. Our first example models the activation of the transcription factor $NF\text{-}AT$, which plays a crucial role in the process of T-cell activations. This process has been widely studied in immunology and oncology and it is well-characterized. From our simulation, we obtain causal relations that reflect faithfully experiments in vitro; in particular, we describe the independence of the two pathways activated by the T-cell antigene receptor $TCR$ and the need of both for activating the transcription factors $AP$-1 and cytosolic $NF\text{-}AT$. Exploiting our extension concerning proximity of processes, we also model the situation in which the presence of Cyclosporine A inhibits the translocation of cytosolic $NF\text{-}AT$ in the cell nucleus. In the second example, we model (a simplified version of) the well-studied glycolysis pathway that degrades glucose. The causal relation that we extract from the computations of our concurrent systems has an intuitive graphical representation. Our diagrams, similar to those of Petri Nets (see e.g. [18]), seem to be superior to the pictures

commonly used by biologists to describe biochemical processes. For example, the diagrams used by KEGG [26] statically describe the net of possible reactions and provide no animation of it (the reader may compare our Fig. 6 for glycolysis pathway with the KEGG version at http://www.kegg.com); also, the semantics of KEGG diagrams is left implicit. Instead, our graphics explicitly describe pathway *evolutions*, stage by stage, originated from a formal model. Consequently, the attention is focused on the flow of reactions that occur in the process, and not only on their reactants. Additionally, the formal description of the pathways permits software simulations that offer cheap pre-views of tests before actually carrying them out (e.g., relying on the BIOSPI software [47,52]). This paper mainly concentrates on causality, but in the EOS approach, the qualitative and the quantitative aspects are orthogonal to each other; see [42] for a discussion on the flexibility of EOS. Note that the possibility of merging together description of the evolution of biological systems with their quantitative measures driving the dynamic behaviour within a single framework is a step towards the definition of a powerful tool for assisting biologists. The important aspect of our approach is that it is based on formal methods. So we offer fine grounds both for defining mathematical models of biological systems and for analysing their various facts, as well as for making stochastic simulations. Indeed, the very same semantics allows for deriving many different aspects that can have the same uniform graphical representation, and then can be combined together to yield increasingly more detailed and accurate models of biochemical processes. All the above makes us confident that our proposal can be used as a descriptive (biological) tool and supports our feeling that it may offer the basis for developing also prescriptive (biological) tools.

The paper is organized as follows. Section 2 briefly recalls the standard $\pi$-calculus, while our proved variant of it is in Section 3. This section contains the needed definitions and results, among which the theorem stating the correctness of our semantics. All the proofs and some auxiliary definitions are in Appendix A and may be safely skipped by an uninterested reader. Section 4 recalls from [8] the notion of causality. Our simple treatment of compartments is in Section 5. Section 6 works out in detail our two examples. The final sections briefly discuss related work and draw some conclusions.

## 2. The standard $\pi$-calculus

We briefly recall the $\pi$-calculus [39], a model of concurrent communicating agents, or processes, based on the notion of naming. Roughly, processes can perform actions in sequence or concurrently and can also choose among alternatives. The basic actions consist of the complementary activities of sending or receiving values (represented by names) along a channel (also represented by a name). When two sub-processes perform complementary activities on the same channel, a communication occurs, and the whole process performs a transition, possibly changing its state, thus altering its future behaviour. The *semantics* of the calculus specifies how processes perform their actions, giving rise to their *computations*. We first define formally the syntax of processes as follows.

**Definition 1** (Processes). Let $\mathcal{N}$ be a countable, infinite set of *names* ranged over by $a, b, \ldots, x, y$, and let $\tau$ be a distinguished element such that $\mathcal{N} \cap \{\tau\} = \emptyset$. *Processes* are denoted by $A, B, C, \ldots$, and are built from names according to the BNF-like syntax

$$A ::= \mathbf{0} | \pi.A | A + A | \ A \mid A \ | (\nu x)A | [x = y]A \mid X(y),$$

where $\pi$ is either $x(y)$ for *input*, or $\overline{x}\langle y\rangle$ for *output* or $\tau$ for *silent* moves and $X$ is an agent identifier (with a single parameter $y$, for simplicity).

We assume that processes are *guarded*, so summands are of the form $\pi.A$ or $\mathbf{0}$ (that will be omitted when in trailing position).

The inactive process $\mathbf{0}$ does nothing. The prefix $\pi$ denotes the first atomic action that the process $\pi.A$ can perform. The input prefix binds the name $y$ in the prefixed process. Intuitively, some name $y$ is received along the link named $x$. The output prefix does not bind the name $y$ which is sent along $x$. The silent prefix $\tau$ is needed to keep short the representation of biological behaviour (see Section 6.2). The operator $(\nu x)$ acts as a *static binder* for the name $x$ in the process $A$ that it prefixes. In other word, $x$ is a unique name in $A$ which is different from all external names. Matching $[x = y]A$ is an `if–then` operator: process $A$ is activated only if $x = y$. The process $A \mid B$ describes a system in which there are two *concurrent* processes, $A$ and $B$, that may evolve independently or may interact through a communication. The process $A + B$ behaves either as $A$ or as $B$. Finally, $X(y)$ represents the call to a constant, that has a unique definition on the form $X(z) = A$, where $z$ plays the role of formal parameter. The semantics of a process is defined by a transition system, i.e. a pair $\langle States, \rightarrow\rangle$. The set *States* contains all the $\pi$-processes, that represent the states that a concurrent system can pass through. A transition $A \rightarrow B$ describes the steps of a computation, i.e. the transition from the state $A$ to the state $B$. The standard way to define the transitions is by a set of inference rules, inducing on the syntax. A notion of *structural congruence* on processes is usually introduced. It reduces the number of inference rules needed, and reflects some intuition about the behaviour of processes, e.g. that $A \mid B$ behaves like $B \mid A$. We now define formally the structural congruence and the semantics of the $\pi$-calculus, given in a reduction style. The auxiliary notions of free names $fn(\mu)$ and bound names $bn(\mu)$ of a prefix $\pi$ is given by the following table; it can be extended to processes in a straightforward way assuming input prefix and restriction as binders:

| $\pi$ | Kind | $fn(\mu)$ | $bn(\mu)$ |
|---|---|---|---|
| $\tau$ | Silent | $\emptyset$ | $\emptyset$ |
| $x(y)$ | Input | $\{x, y\}$ | $\emptyset$ |
| $\overline{x}\langle y\rangle$ | Output | $\{x, y\}$ | $\emptyset$ |

The *structural congruence* $\equiv$ on processes is the least congruence satisfying the following clauses:

1. $A \equiv B$ if $A$ and $B$ are $\alpha$-equivalent;
2. $(\mathcal{A}/_{\equiv}, +, \mathbf{0}), (\mathcal{A}/_{\equiv}, |, \mathbf{0})$ are commutative monoids;
3. $[x = x]A \equiv A$;

Table 1
Reduction semantic

$$\textbf{Com}: (C + x(w).A)|(D + \bar{x}\langle y\rangle.B) \to A\{y/w\}|B$$

$$\textbf{Par}: \frac{A \to A'}{A|B \to A'|B}, \qquad \textbf{Res}: \frac{A \to A'}{(va)\ A \to (va)\ A'}$$

$$\textbf{Tau}: \tau.P \to P, \qquad \textbf{Struct}: \frac{B \equiv A \quad A \to A' \quad A' \equiv B'}{B \to B'}$$

$$\textbf{Const}: \frac{A\{y/x\} \to A'}{X(y) \to A'}, \quad X(x) \stackrel{\text{def}}{=} A$$

4. $(v\,x)(v\,y)A \equiv (v\,y)(v\,x)A$ (simply written as $(v\{x, y\})A$), $(v\,x)A \equiv A$ if $x \notin fn(A)$, $(v\,x)A \mid B \equiv A \mid (v\,x)B$ if $x \notin fn(A)$.

Roughly speaking, the first item says that processes that only differ in bound names have the same semantics. The second item says that the order of processes in a summation or in a parallel composition is immaterial; additionally, the inactive process can be discarded. The third item permits to simplify a condition when satisfied. Then, there are some clauses handling binders: their order is irrelevant and so is a binder on a name that does not occur; also, the scope of a binder can be enlarged under certain circumstances. The semantics for $\pi$-calculus is defined in a reduction-style [38] by the inference rules in Table 1. Note that we also have an axiom for an asynchronous $\tau$-move.

## 3. A proved reduction semantics for the $\pi$-calculus

The semantics of the $\pi$-calculus surveyed in the previous section does not consider concurrency as a first class notion. Indeed, when two (or more) transitions not mutually exclusive are possible for a process, either one comes first and the other later or vice versa (all their possible interleavings are considered); see the so-called expansion theorem in process algebras [36]. For this reason, the semantics above is sometimes referred to as *interleaving*. Also, there is no way to single out which are the subprocesses that actually communicate and the effects of this interaction. As mentioned in the Introduction, the relations between transitions and their effects, e.g. causality, may help in better describing process behaviour; also other aspects, e.g. the places where sub-processes reside may be important to know. If one is interested in these aspects, the interleaving semantics is not enough, and the literature has many proposals to overcome this problem (see e.g. the references in [9]). Our starting point here is the so-called Enhanced Operational Semantics [9] that has transitions carrying rich labels (see Appendix A for its formal definition). Although this semantics is interleaving in style, from the enhanced labels of its transitions one can mechanically derive several different aspects of computations, including qualitative ones (causality, locality, etc.) and quantitative ones (time, probabilities, etc.). Remarkably, the causal semantics

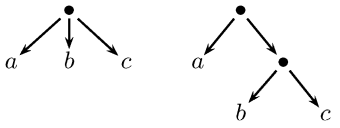


Fig. 1. Transforming a ternary tree into an equivalent binary one.

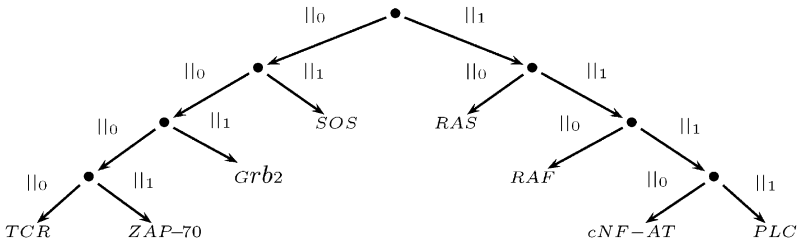


Fig. 2. The tree of (sequential) processes of *NF-AT* activation (see Section 6).

obtained in this way coincides with those presented in the literature [8]. Following this approach, we introduce a *reduction* semantics, hereafter called *proved reduction semantics*. We then show the correctness of our reduction semantics by proving (in Appendix A) that a transition exists in our proved reduction semantics if and only if the same transition, including its label, is in the EOS operational semantics. Our formal treatment starts with the basic notion of *address* of a sequential component of a system $R$ (roughly, a process with a prefix or a summation as top-level operator). The following intuitive explanation may help. Consider the abstract syntax tree of $R$, built using binary parallel composition as the main operator (Fig. 1). [1]

So parallel processes are the nodes of the tree and sequential processes are its leaves. Now, label the arc leading to the left (resp. right) son of a parallel composition with $\|_0$ (resp. $\|_1$). The (label of the) path from the root to a sequential process is its address. Fig. 2 shows the tree of processes of the network of proteins used in an our example. The process *ZAP*-70 has address $\|_0\|_0\|_0\|_1$. Once chosen a particular tree of processes, the addresses uniquely identify the sequential processes of the system (for more details, see [2]). Note that the address of a sub-process $P$ encodes a skeleton of the *context* in which $P$ occurs, as it represents the structure of the context as far as the parallel operator is concerned.

Below we introduce our variant of the $\pi$-calculus. The syntax of processes and their congruence rules are given, pointing up the differences from standard ones. We show then some simple property, and we conclude presenting the proved reduction semantics and the equivalence with the SOS semantics. Our reduction semantics carries additional information on transitions: we can reduce our semantics to the standard one trashing them. Above we show how to derive a causal relation and (simple)

[1] Note that our trees are binary because the $|$ operator is binary. Considering an $n$-adic operator is easy: it suffices to have $\|_0 \cdots \|_{n-1}$ different tags. Anyway, every $n$-adic tree can be represented as a binary one.

Table 2
Non-interleaving reduction semantic

$$\textbf{Com}: (R+\vartheta\|_i\vartheta_0 x(w).P)|(S+\vartheta\|_{1-i}\vartheta_1\overline{x}\langle y\rangle.Q)\xrightarrow{\vartheta\langle\|_i\vartheta_0 x(w),\|_{1-i}\vartheta_1\overline{x}\langle y\rangle\rangle}{}_R P\{y/w\}|Q$$

$$\textbf{Par}: \frac{P\xrightarrow{\theta}{}_R P'}{P|Q\xrightarrow{\theta}{}_R P'|Q} \qquad \textbf{Res}: \frac{P\xrightarrow{\theta}{}_R P'}{(va)\ P\xrightarrow{\theta}{}_R (va)\ P'}$$

$$\textbf{Tau}: \vartheta\tau.P\xrightarrow{\vartheta\tau}{}_R P \qquad \textbf{Struct}: \frac{Q\equiv P \quad P\xrightarrow{\theta}{}_R P' \quad P'\equiv Q'}{Q\xrightarrow{\theta}{}_R Q'}$$

$$\textbf{Const}: \frac{P\{y/x\}\xrightarrow{\vartheta'}{}_R P'}{\vartheta X(y)\xrightarrow{\vartheta\vartheta'}{}_R \vartheta\rhd P'} \qquad X(x)\stackrel{\text{def}}{=} P$$

stochastic information from our proved semantics: the comparison is always with the SOS semantics of [8].

*Syntax*: Differently than above, we change the syntax of processes so that each action also carries the address of the sequential component it belongs to. Recall that $A, B, \ldots$ are processes in the standard $\pi$-calculus; below we extend them with addresses, and we denote the "extended" processes by $P, Q, \ldots$ .

**Definition 2** (Extended processes). An *address* is $\vartheta\in\{\|_0,\|_1\}^*$, $\lambda$ is the empty one. Let $\mathcal{N}$ be a countable, infinite set of *names* ranged over by $a, b, \ldots, x, y$, and let $\tau$ a set of distinguished element such that $\mathcal{N}\cap\{\tau\}=\emptyset$. *Extended processes*, or simply *processes* when unambiguous, are denoted by $P, Q, R, \ldots$, and are built from $\mathcal{N}$ as in Definition 1, with $\vartheta\mu.P$ that substitute $\mu.P$.

It is convenient to introduce a mapping from standard to extended processes and vice versa. It helps in connecting the various semantics of the $\pi$-calculus we consider. Additionally, specification of systems results more readable, as they can be written according to the standard syntax and transformed to the extended one on demand. The function $\mathcal{T}$ maps standard processes into extended ones and it is defined below by inductively prefixing actions with the address of the sequential processes they prefix; note that $\mathcal{T}$ unwinds the syntactic structure of processes until reaching a **0** or a constant. We first need an auxiliary operator $\rhd$ that is invoked when $\mathcal{T}$ reach a parallel operator or when a constant is replaced with his definition (see the rule for the constant in Table 2). It distributes to a sub-process $P$ an address that encodes the skeleton of *context* of the process in which $P$ is supposed to be plugged in.

**Definition 3** (Distributing addresses).
- $\vartheta\rhd\mathbf{0}=\mathbf{0}$,
- $\vartheta\rhd P+Q=\vartheta\rhd P+\vartheta\rhd Q$,
- $\vartheta\rhd(va)P=(va)\vartheta\rhd P$,
- $\vartheta\rhd\vartheta' X(x)=\vartheta\vartheta' X(x)$.
- $\vartheta\rhd(\vartheta'\mu.P)=\vartheta\vartheta'\mu.(\vartheta\rhd P)$,
- $\vartheta\rhd P\,|\,Q=(\vartheta\rhd P)\,|\,(\vartheta\rhd Q)$,
- $\vartheta\rhd[x=y]P=[x=y]\vartheta\rhd P$,

We can now define $\mathcal{T}$.

**Definition 4.** Let $A, B$ be standard processes and let $\rhd$ the operator introduced in the next definition. The following is a bijection:

- $\mathcal{T}(\mathbf{0}) = \mathbf{0}$,
- $\mathcal{T}(A + B) = \mathcal{T}(A) + \mathcal{T}(B)$,
- $\mathcal{T}((va)A) = (va)\mathcal{T}(A)$,
- $\mathcal{T}(X(x)) = X(x)$.
- $\mathcal{T}(\mu.A) = \mu.\mathcal{T}(A)$,
- $\mathcal{T}(A|B) = \|_0 \rhd \mathcal{T}(A) \mid \|_1 \rhd \mathcal{T}(B)$,
- $\mathcal{T}([x = y]A) = [x = y]\mathcal{T}(A)$,

It is straightforward proving that $\mathcal{T}$ is a bijection between the two different presentation of processes, its inverse being the function that discards addresses (note that no congruence is assumed yet). Given a process on its standard form, this transformation operates in linear time with the number of prefixes. Also, $\mathcal{T}$ specifies a pre-processing step, that however has to be re-applied to a constant when invoked (see the definition of structural congruence below).

**Example 1.** Consider the standard process $E = (A \mid \mu.(B + C)) \mid D$, where $A, B, C$ and $D$ are constants. Below we compute its extended version (skipping some steps). Note that all the (standard) sequential processes get prefixed by their addresses:

$$\begin{aligned}
\mathcal{T}((A \mid \mu.(B + C)) \mid D) &= \|_0 \rhd \mathcal{T}(A \mid \mu.(B + C)) \mid \|_1 \rhd \mathcal{T}(D) \\
&= \|_0 \rhd (\|_0 \rhd \mathcal{T}(A) \mid \|_1 \rhd \mathcal{T}(\mu.B + C)) \mid \|_1 D \\
&= \|_0 \rhd (\|_0 A \mid \|_1 \rhd \mu.(B + C)) \mid \|_1 D \\
&= \|_0 \rhd (\|_0 A \mid \|_1 \mu.\|_1 \rhd (B + C)) \mid \|_1 D \\
&= \|_0 \rhd (\|_0 A \mid \|_1 \mu.(\|_1 B + \|_1 C)) \mid \|_1 D \\
&= \|_0 \|_0 A \mid \|_0 \|_1 \mu.\|_0 \rhd (\|_1 B + \|_1 C) \mid \|_1 D \\
&= \|_0 \|_0 A \mid \|_0 \|_1 \mu.(\|_0 \|_1 B + \|_0 \|_1 C) \mid \|_1 D.
\end{aligned}$$

For our subsequent treatment, it is convenient introducing two auxiliary operators. The first one selects the subprocess of a process reachable through a given address. Below, let $i \in \{0, 1\}$, and let $P_i, Q$ be extended processes.

**Definition 5** (Selector). The *selector* operator is defined as follows:

- $P@\lambda = P$,
- $(\|_0 \rhd P_0 \mid \|_1 \rhd P_1)@\|_i \vartheta = \|_i \rhd P_i@\vartheta$,
- $(va)P@\vartheta = P@\vartheta$,
- $[x = y]P@\vartheta = P@\vartheta$.

**Example 2.** Consider again the process $\mathcal{T}(E)$ in Example 1 and select its subprocess at $\|_0\|_1$:

$$((\|_0\|_0 A \mid \|_0\|_1 \mu.(\|_0\|_1 B + \|_0\|_1 C)) \mid \|_1 D)@\|_0\|_1.$$

By Definition 5, the above can be written as

$$\begin{aligned}
&(\|_0 \rhd (\|_0 A \mid \|_1 \mu.(\|_1 B + \|_1 C)) \mid \|_1 \rhd D)@\|_0\|_1 \\
&\quad = \|_0 \rhd [\|_0 A \mid \|_1 \mu.(\|_1 B + \|_1 C)]@\|_1
\end{aligned}$$

$$= \|_0 \rhd [\|_0 \rhd A \mid \|_1 \rhd [\mu.(B + C)]]@\|_1$$
$$= \|_0 \rhd \|_1 \rhd \mu.(B + C)@\lambda$$
$$= \|_0 \rhd \|_1 \rhd \mu.(B + C)$$
$$= \|_0\|_1\mu.(\|_0\|_1 B + \|_0\|_1 C).$$

In what follows, given a process $P$, we shall occasionally substitute a process $Q$ for $P@\vartheta$, whenever there is such a subprocess, written $P[\vartheta \mapsto Q]$. Note that $Q$ may replace a whole summation selected by $P@\vartheta$ because it is considered as sequential (see also the selector operator).

**Definition 6.** The localized substitution of $Q$ at $\vartheta$ within $P$ is defined as
- $P[\lambda \mapsto Q] = Q$,
- $(\|_0 \rhd P_0 \mid \|_1 \rhd P_1)[\|_0\vartheta \mapsto Q] = \|_0 \rhd P_0([\vartheta \mapsto Q]) \mid \|_1 \rhd P_1$,
- $(\|_0 \rhd P_0 \mid \|_1 \rhd P_1)[\|_1\vartheta \mapsto Q] = \|_0 \rhd P_0 \mid \|_1 \rhd P_1([\vartheta \mapsto Q])$,
- $((\nu a)P)[\vartheta \mapsto Q] = (\nu a)(P[\vartheta \mapsto Q])$,
- $([x = y]P)[\vartheta \mapsto Q] = [x = y](P[\vartheta \mapsto Q])$.

The two definitions above do not exploit the addresses of extended (sub)processes because they are driven by syntactic operators within terms. Indeed, the very same definitions can be applied to standard processes, as well. In what follows, we feel free to overload the symbols @ and $[\vartheta \mapsto Q]$.

*Congruence*: Below, we assume the *structural congruence* $\equiv$ of the $\pi$-calculus on extended processes, as given in Definition 2.

*Structure preserving manipulations*: We now state a couple of properties that relate the syntactic structure of standard and of extended processes. These facts facilitate the proof that the proved reduction semantics we are going to introduce below is just a different formalization of the behaviour of $\pi$-processes. In particular, we shall prove the equivalence between ours and the SOS semantics of [8]. Recall that $A, B$ are processes of the standard $\pi$-calculus, and hereafter assume that they are closed (i.e. they contain no free name). Our first property (items (a1), (a2)) shows that the function $\mathcal{T}$, while generating extended processes from standard ones, preserves structural congruence, provided that associativity and commutativity (as well as absorption of $\mathbf{0}$) of the parallel operator is not assumed on standard processes—a basic choice in the causal semantics of [8]. Our second property (b1), (b2) shows that the selector operator also preserves structural congruence in the same way.

**Proposition 1.** *Let $\equiv'$ be the least congruence on standard processes satisfying the clauses defining $\equiv$, except for $(\mathcal{P}/_{\equiv}, |, \mathbf{0})$ being a commutative monoid. Then*
(a1) *$A \equiv' B$ implies $\mathcal{T}(A) \equiv \mathcal{T}(B)$,*
(a2) *$\mathcal{T}(A) \equiv \mathcal{T}(B)$ implies $A \equiv B$*
*and*
(b1) *$A@\vartheta \equiv' \mu.B$ implies $T(A)@\vartheta \equiv \vartheta\mu.\vartheta \rhd \mathcal{T}(B)$,*
(b2) *$\mathcal{T}(A)@\vartheta \equiv \vartheta\mu.\vartheta \rhd \mathcal{T}(B)$ implies $A@\vartheta \equiv \mu.B$.*

From now onwards, we feel free to re-arrange processes in such a way that restrictions occur all in the outermost position. This is done by exploiting the congruence rules that enlarge restriction scope, possibly $\alpha$-converting names (recall Turner–Milner theorem [37]).

*Reduction semantics*: Table 2 shows our reduction semantics. Unconventionally, our semantics carries labels on the arrows to extract causality, just as it happens with the SOS causal semantics of [8]. The labels record the address $\vartheta$ where the action took place. For the special case of $\tau$-actions, the actual label can only be $\vartheta\tau$, but for technical reasons we allow the more general label $\vartheta\tau$ below (see beginning of Section 4). In case of a communication, also the addresses and the actions of the partners $P_{\text{in}}$ and $P_{\text{out}}$ are recorded in a pair. For $P_{\text{in}} \mid P_{\text{out}}$ to communicate, they must occur in some context reachable, say, through the address $\vartheta$. So the actions of $P_{\text{in}}$ and $P_{\text{out}}$ must be on the form $\vartheta\|_i\vartheta_0 x(w)$ and $\vartheta\|_{1-i}\vartheta_1\overline{x}\langle y\rangle$, respectively. The axiom **Com** originates then the label of the reduction by singling out the common prefix $\vartheta$ of the input and the output actions. The other rules are much standard, except for the rule **Const**, that distributes the actual address of the constant $X$ over the residual $P'$ of its body $P$.

**Definition 7.** A *label* is defined by the following syntax:

$$\theta ::= \vartheta\mu \mid \vartheta\langle \|_i\vartheta_0 x(w), \|_{1-i}\vartheta_1\overline{x}\langle y\rangle\rangle.$$

**Example 3.** We apply our new semantics to the process

$$P = \mathcal{T}(a(x).Q \mid (\overline{b}\langle a\rangle \mid b(y).\overline{y}\langle b\rangle.R)),$$

which is on the form $P = S \mid (T \mid U)$. We first derive that

$$T \mid U = \|_1\|_0\mathbf{0} + \|_1\|_0\overline{b}\langle a\rangle.\|_1\|_0\mathbf{0} \mid \|_1\|_1\mathbf{0} + \|_1\|_1 b(y).\|_1\|_1\overline{y}\langle b\rangle.\|_1\|_1 \rhd \mathcal{T}(R)$$
$$\xrightarrow{\langle \|_1\|_0\overline{b}\langle a\rangle.\|_1\|_1 b(y)\rangle}_R$$
$$\underbrace{\|_1\|_0\mathbf{0} \mid \|_1\|_1\overline{a}\langle b\rangle.\|_1\|_1 \rhd \mathcal{T}(R)\{a/y\}}_{(*)}$$

This transition becomes the premise from which we derive that

$$P \xrightarrow{\|_1\langle \|_1\|_0\overline{b}\langle a\rangle.\|_1\|_1 b(y)\rangle}_R \underbrace{\|_0 a(x).\|_0 \rhd \mathcal{T}(Q) \mid *}_{(**)}.$$

Note that the inactive process have been first added and then removed to match the **Com** rule. The next computation step is

$$(**) \xrightarrow{\langle \|_0 a(x).\|_1\|_1\overline{a}\langle b\rangle\rangle}_R \|_0 \rhd \mathcal{T}(Q)\{b/x\} \mid \|_1\|_1 \rhd \mathcal{T}(R)\{a/y\}.$$

It is straightforward recovering the standard semantics of the $\pi$-calculus in Table 1 from our proved reduction semantics in Table 2. It sufficed to discard the axiom **Tau** and the labels from the rules (or from computations). As a matter of fact, we shall use the extended labels to derive some aspects of computations, including causality.

To make sure that our development is correct, we compare the semantics given here to the one in [8] that is defined in an SOS style (see Appendix A). The following theorem says that the two semantics do coincide.

**Theorem 1.** *Let* $\xrightarrow{\theta}_S$ *the relation defined in the late SOS semantics of* [8], *where* $\theta$ *is as in Definition* 7. *Then*, $\forall A, B \in \mathcal{A}$, $A \xrightarrow{\theta}_S B$ *iff* $\mathcal{T}(A) \xrightarrow{\theta}_R \mathcal{T}(B)$, *where* $\theta$ *is as in Definition* 7.

Theorem 1 enables us to carry over the computations defined through the proved reduction semantics all the non-interleaving aspects defined for the SOS semantics in [8]. In the next section, we consider one aspect of those considered there, namely *causality*. It will be used to accurately describe qualitative aspects of biological evolutions. A different interpretation of transition labels permits to represent also quantitative aspects (for an idea of how to do this see Section 5, for a complete presentation see, e.g. the stochastic $\pi$-calculus in [45] or the relabelling functions of [9,42]). The two interpretations are orthogonal and can be combined together to enhance the expressivity of our descriptions [46].

## 4. Causality

Intuitively, a transition $t$ *causes* a transition $t'$ if $t$ occurs before $t'$ and $t$ is a necessary condition for $t'$, i.e. the prefixes that originated them are nested in a prefix chain. Such a nesting is reflected in the labels of the transitions: roughly speaking, the address in the label of $t$ is a prefix of the address in the other.

Following this intuition, we rephrase below the definition of causality given in [8] and called there enabling. In the actual definition, we make use of the following auxiliary function that "flattens" labels:

- $f(\vartheta\tau) = \{\vartheta\tau\}$,
- $f(\vartheta\langle\|_0\vartheta_0 x(w), \|_1\vartheta_1\overline{x}\langle y\rangle\rangle) = \{\vartheta\|_0\vartheta_0 x(w), \vartheta\|_1\vartheta_1\overline{x}\langle y\rangle\}$.

Note that below, we denote a transition $P \xrightarrow{\theta}_R Q$ by its label $\theta$, only; we shall use this shorthand hereafter, whenever unambiguous.

**Definition 8** (Causal relation). Let $P_0 \xrightarrow{\theta_0}_R P_1 \xrightarrow{\theta_1}_R \cdots \xrightarrow{\theta_n}_R P_{n+1}$ be a computation, and stipulate that $\theta_i \sqsubset \theta_j$ iff $i < j$ and $\vartheta\mu \in f(\theta_i)$ and $\vartheta\vartheta'\mu' \in f(\theta_j)$ (i.e. $\theta_i$ "contains" an address that is a prefix of an address of $\theta_j$). Then

$$\theta_i \sqsubseteq \theta_j \text{ iff } \theta_i \sqsubset^* \theta_j$$

(in words $\theta_i$ *causes* $\theta_j$), i.e. $\sqsubseteq$ is the reflexive and transitive closure of $\sqsubset$.

**Example 4.** Consider the system of Section 6.1, the computation of which is in Fig. 4. Its first fragment models the phosphorylation of the Z-chains of the transcription factor *NF-AT* (transition $t_0$) and the beginning of two independent pathways: the

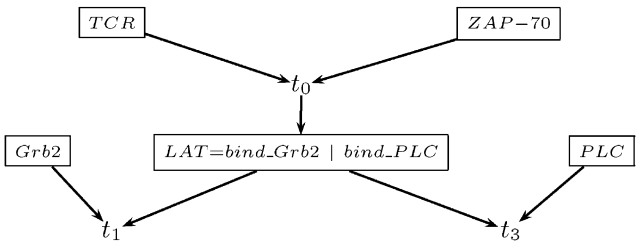


Fig. 3. A simple partial order: $t_0$ causes both $t_1$ and $t_3$ that are in turn concurrent.

calcium/calcineurin ($t_3$) and the Ras/MAP kinase ($t_1$). So, the transition $t_0$ precedes in time the transitions $t_1$ and $t_3$, that have the following labels $\theta_0, \theta_1, \theta_3$, respectively:

$$\theta_0 = \langle \|_0\|_0\|_0\|_0 \overline{bind\_z}\langle tcr\rangle, \|_0\|_0\|_0\|_1 bind\_z(tcr)\rangle,$$
$$\theta_1 = \langle \|_0\|_0\|_0\|_1\|_0 \overline{bind\_grb2}\langle lat\rangle, \|_0\|_0\|_1 bind\_grb2(lat)\rangle,$$
$$\theta_3 = \langle \|_0\|_0\|_0\|_1\|_1 \overline{bind\_PLC}\langle lat1\rangle, \|_1\|_1\|_1\|_1 bind\_PLC(lat1)\rangle.$$

Now, $t_0 \sqsubseteq t_1$ because $\|_0\|_0\|_0\|_1 bind\_z(tcr) \sqsubset \|_0\|_0\|_0\|_1\|_0 \overline{bind\_grb2}\langle lat\rangle$ and $t_0 \sqsubseteq t_3$ because $\|_0\|_0\|_0\|_1 bind\_z(tcr) \sqsubset \|_0\|_0\|_0\|_1\|_1 \overline{bind\_PLC}\langle lat1\rangle$. Instead, neither $t_1 \sqsubseteq t_3$ nor $t_3 \sqsubseteq t_1$.

The definition above yields a partial order $\langle E, \sqsubseteq\rangle$, where $E$ is the set of transitions. The three transitions of the example above give raise to the partial order that is graphically represented in Fig. 3. The elements of the partial order are the transitions themselves, and the presence of an arc between, e.g. $t_0$ and $t_3$, means that the first causes the second. In other words, $t_0$ must occur *before* and it is *necessary* for $t_3$. To enhance readability, we also represent the sub-processes that act in the transitions, e.g. the process *TCR* and *ZAP*-70 interact to become *LAT* in the transition $t_0$, and the causal link from $t_0$ to $t_3$ passes through *LAT*. So we model pictorially the relationships between reactants and reactions in the evolution of a biochemical process (cf. the diagrams describing the dynamics of Petri nets [18]).

In the example above, the two transitions $t_1$ and $t_3$ are not causally related, and we call them *concurrent*, as usual. In symbols

$$t \not\sqsubseteq t' \quad \text{if and only if} \quad t \smile t'.$$

Since $t_1$ and $t_3$ are concurrent, they may occur in *any temporal order* and even at the *same time*.

The relation of concurrency between transitions not only holds for the computation considered, but for *all* the computations in which the transitions considered occur. Indeed, Theorem 6.4 of [8] guarantees this fundamental fact and lifts the notion of concurrency to processes.

## 5. A simple compartment semantics

In the previous section, we derived from the proved semantics of the $\pi$-calculus a description of the causal relationship that holds between the transitions of a computation and of concurrency between the prefixes in a process. These relations can be represented pictorially, thus helping in modelling and in better understanding biochemical pathways. Here we are mainly interested in causality, but we wish to give evidence that notions like neighbourhood can be easily accommodated in our framework.

In what follows, we present a simple manner of constraining reactions between molecules, according to whether they are close enough to each other. In this way, we naively model the regions of a cell, e.g. the nucleus and the cytosol. It is only a matter of technicalities to make this description more sophisticated, and consider explicitly reactant quantities and probabilistic distributions on reactions, rather than the simple yes/no constraint we are going to describe. Nevertheless, the proposal below suffices in describing the behaviour of a typical inhibitor of the *NF-AT* activation (see Section 6.1).

We first introduce a notion of neighbourhood between sequential processes. Given a(n extended) process $P$, let $Addr(P) = \{\vartheta \mid P@\vartheta \text{ is defined}\}$. Then, one assumes as given a distance

$$\partial : Addr(P) \times Addr(P) \to \mathbb{R}^+,$$

i.e. a function such that

- $\partial(\vartheta, \vartheta') = 0$ iff $\vartheta = \vartheta'$,
- $\partial(\vartheta, \vartheta') = \partial(\vartheta', \vartheta)$,
- $\partial(\vartheta, \vartheta') + \partial(\vartheta', \vartheta'') \geqslant \partial(\vartheta, \vartheta'')$.

The function $\partial$ can now be used to forbid interaction between sub-processes whose distance is greater than a given threshold $\varepsilon$. To do that, it suffices putting a side condition on the rule for interaction of the semantics in Table 2 as follows:

$$(R + \vartheta\|_0\vartheta_0 x(w).P)|(S + \vartheta\|_1\vartheta_1\bar{x}\langle y\rangle.Q) \xrightarrow{\vartheta\langle\|_i\vartheta_0 x(w),\|_{1-i}\vartheta_1\bar{x}\langle y\rangle\rangle}_R P\{y/w\}|Q$$
$$\text{if } \partial(\vartheta\|_i\vartheta_0, \vartheta\|_{1-i}\vartheta_1) \leqslant \varepsilon.$$

## 6. Two examples

In this section, we apply our reduction semantic to two well-characterized biochemical process. The results offered by our modelizations reflect those of the experiments in vitro. This makes us confident that our tools and techniques could be extended not only to obtain readable descriptions of known biological evolutions, but also to extract new properties on them.

Following [52], we view molecules and their behaviour as concurrent processes in the $\pi$-calculus. Essentially, molecules are a set of *domains* denoted by *motifs*. Two molecules can interact if they have two *complementary* motifs. Such an interaction produces a result called the *residual* of the reaction. Any molecule has some private "information", called *backbone*, that determines its identity. The interaction between

two molecules can be seen then as sharing a backbone. The same happens with protein complexes or cellular compartments [47].

As done in [52], motifs are represented by global channels (i.e. free names in a system), and a reaction involving them is a communication on one of these channels. Residuals of a reaction are processes prefixed by an input/output action. Backbones are private names of processes declared by a restriction. So, interaction between molecules may result also in a scope enlargement. For simplicity, we left many residuals empty. Indeed, in our example residuals are molecules activated (or deactivated) by phosphorylations (or dephosphorylations) and we ignore these activities here.

### 6.1. *NF-AT* activation

We model here the activation of the transcription factor *NF-AT*, which plays a crucial role in the process of T-cell activations. *NF-AT* is composed of two subunits, a cytosolic subunit, which belongs to the family of *NF-AT* proteins, and a nuclear subunit, identified as *AP*-1. The activation of *NF-AT* requires translocation of the cytosolic subunit to the nucleus, where it can assemble with *AP*-1. This subunit in turn requires to be phosphorylated to become transcription-competent. Regulation of the subcellular localization of *NF-AT* is achieved by phosphorylation on specific serine residues. When phosphorylated, the nuclear localization signal of *NF-AT* is masked and *NF-AT* is therefore segregated to the cytosol. The engagement of a T-cell antigen receptor (*TCR*) triggers an increase in intracellular calcium ions, which induces the activation of the phosphatase calcineurin. Dephosphorylation of *NF-AT* by calcineurin results in exposure of the nuclear localization sequence and translocation of *NF-AT* to the nucleus. *AP*-1 is a heterodimer of nuclear oncoproteins Jun and Fos and it is activated by phosphorylation of Jun and Fos on serine residues. Activation of the MAP kinase pathway, which is also triggered by the *TCR*, is required for this process to occur [48]. Hence, *TCR* engagement results in the activation of both pathways, the Ras/MAP kinase and the calcium/calcineurin pathway, required for *NF-AT* activation [25].

To model the above, we consider the system *Sys* composed by the (active and inactive) proteins needed to activate *NF-AT*. To show how causality helps describing this biochemical process, it suffices to have a single instance of each reagent. More copies only affect the readability of the system and the representation of computations (see Fig. 4). Each protein is represented by a process, that runs in parallel with each other:

$$Sys = (((TCR \mid ZAP\text{-}70) \mid grb2) \mid SOS) \mid (RAS \mid (RAF \mid (cNF\text{-}AT \mid PLC))).$$

The part in common with the Ras/MAP kinase and the calcium/calcineurin pathway involves two reactants, *TCR* and *ZAP*-70, that perform a phosphorylation of the *Z*-chains of *TCR* and an interaction between the two. The residual of recruitment of *ZAP*-70 by *TCR* enables the phosphorylation of *LAT* (i.e. his activation), that concludes the initialization. As said above, phosphorylation and dephosphorylation are ignored, as well as the residuals they activate. So we group the three steps above in a single activity, represented by the communication of the backbone *tcr* of *TCR* to *ZAP*-70.

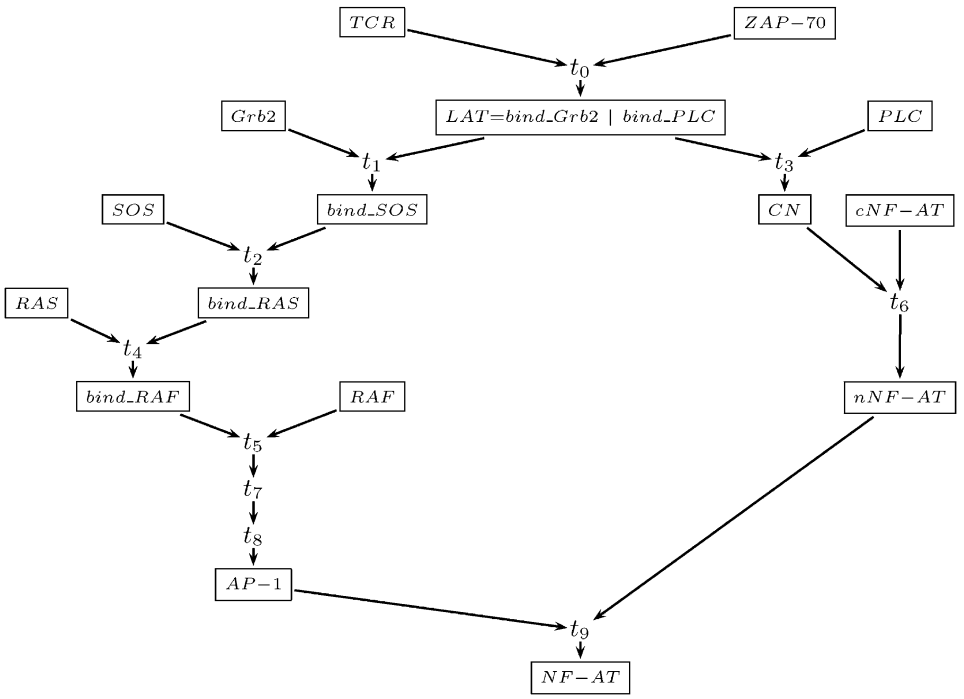


Fig. 4. A computation of *Sys*. For readability, the processes, enclosed in boxes, have no address. Causality (both on transitions and processes) is represented by the (Hasse diagram resulting from the) arrows; their absence makes it explicit concurrent activities.

Our specification of the proteins *TCR* and *ZAP*-70 is then

$$TCR = (\nu\, tcr)\overline{bind\_z}\langle tcr\rangle,$$

$$ZAP\text{-}70 = bind\_z(tcr).LAT.$$

Activated *LAT* has two specific phosphorylated tyrosine residuals responsible of interaction with *Grb*2 (*bind_grb*2) and *PLC* (*bind_PLC*). Also, it has a backbone that will be shared with the proteins which it will interact with, namely *Grb*2 and *PLC*. The two domains may interact in parallel along the two pathways. To model correctly these interactions, it is convenient to represent the backbone with two different names, each with its own scope. We call them $lat1$ and $lat2$.

Now we start modelling the first pathway, which leads to the activation of the nuclear subunit *AP*-1. The domain $(\nu\, lat1)\overline{bind\_grb2}\langle lat1\rangle$ of *LAT* has no residual and its motif *bind_grb*2 is complementary to that of the protein *Grb*2. So the two proteins can interact and activate the residual of *Grb*2. In turn, this interacts with *SOS* and activates *RAS*. A further interaction sequentially activates *RAF*, *MEK* and *ERK*. The pathway ends with the phosphorylation of *Jun* and *Fos* by *ERK* that stimulates the ability of the *AP*-1 subunit to interact with the *DNA* in the cell nucleus. We specify

the above in the standard $\pi$-calculus as follows, where the members of the map kinase cascade are represented by (indexed) internal actions $\tau$.

$$LAT = (\nu\, lat1)\overline{bind\_grb2}\langle lat1\rangle \mid (\nu\, lat2)\overline{bind\_PLC}\langle lat2\rangle,$$

$$Grb2 = bind\_grb2(lat1).\overline{bind\_SOS}\langle lat1\rangle,$$

$$SOS = bind\_SOS(lat1).\overline{bind\_RAS}\langle lat1\rangle,$$

$$RAS = bind\_RAS(lat1).\overline{bind\_RAF}\langle lat1\rangle,$$

$$RAF = bind\_RAF(lat1).\tau_{MEK}.\tau_{ERK}.AP-1.$$

We now model the second pathway that activates the cytosolic subunit, represented by the process $cNF-AT$. The domain $(\nu\, lat2)\overline{bind\_PLC}\langle lat2\rangle$ of $LAT$ interacts with phospholipase $C$. This interaction results in an increase of the free intracellular calcium and activation of calcineurin $CN$. This molecule is a phosphatase that can dephosphorylate $cNF\text{-}AT$, enabling it to translocate to the nucleus. We group the interaction of $CN$ with $cNF\text{-}AT$ and the dephosphorylation of the latter as a single communication between the two molecules:

$$PLC = bind\_PLC(lat2).CN,$$

$$CN = (\nu\, cn)\overline{bind\_sub}\langle cn\rangle,$$

$$cNF\text{-}AT = bind\_sub(cn).nNF\text{-}AT,$$

$$AP\text{-}1 = bind\_ap1(cn),$$

$$nNF\text{-}AT = \overline{bind\_ap1}\langle cn\rangle.$$

As mentioned above, the tuning of $NF\text{-}AT$ depends on both pathways. The first one, RAS/MAP kinases, activates the nuclear subunit $AP$-1. The second pathway, calcium/calcineurin, enables $AP$-1 to translocate within the nucleus of a cell, where it interacts with the $DNA$.

To show these causality relations, we first extend the specification above according to Definition 4. (Admittedly, this representation is heavy, but it is intended to be internal only, and is mechanically generated.) Only the whole system and the constant $LAT$ are affected and prefixed by suitable strings made of tags $\|_0$ and $\|_1$:

$$Sys = (((\|_0\|_0\|_0\|_0 TCR \mid \|_0\|_0\|_0\|_1 ZAP\text{-}70) \mid \|_0\|_0\|_1 Grb2) \mid \|_0\|_1 SOS) \mid (\|_1\|_0 RAS \mid (\|_1\|_1\|_0 RAF \mid (\|_1\|_1\|_1\|_0 cNF\text{-}AT \mid \|_1\|_1\|_1\|_1 PLC))),$$

$$LAT = (\nu\, lat1)\|_0\overline{bind\_grb2}\langle lat1\rangle \mid (\nu\, lat1)\|_1\overline{bind\_PLC}\langle lat1\rangle.$$

As mentioned in Section 4, the causal relation $\langle E, \sqsubseteq\rangle$ of a computation has an immediate graphical representation. We construct a directed (acyclic) graph whose nodes are the transitions themselves that represent the reaction (the elements of $E$), and there is an arc from a transition $t$ to $t'$ if and only if $t \sqsubseteq t'$, i.e. if the first reaction is a necessary condition for the second to occur. Recall also that we represent the causal

relation of a computation as a Hasse diagram, where only the immediate causes are depicted and absence of arrows means concurrency. The computation of *Sys* is in Fig. 4. The labels of its transitions are

$t_0$ — $\langle \|_0\|_0\|_0\|_0\overline{bind\_z}\langle tcr\rangle, \|_0\|_0\|_0\|_1 bind\_z(tcr)\rangle$,
$t_1$ — $\langle \|_0\|_0\|_0\|_1\|_0\overline{bind\_grb2}\langle lat\rangle, \|_0\|_0\|_1 bind\_grb2(lat)\rangle$,
$t_2$ — $\langle \|_0\|_0\|_1\overline{bind\_SOS}\langle lat\rangle, \|_0\|_1 bind\_SOS(lat)\rangle$,
$t_3$ — $\langle \|_0\|_0\|_0\|_1\|_1\overline{bind\_PLC}\langle lat1\rangle, \|_1\|_1\|_1\|_1 bind\_PLC(lat1\rangle$,
$t_4$ — $\langle \|_0\|_1\overline{bind\_RAS}\langle lat\rangle, \|_1\|_0 bind\_RAS(lat)\rangle$,
$t_5$ — $\langle \|_1\|_0\overline{bind\_RAF}\langle lat\rangle, \|_1\|_1\|_0 bind\_RAF(lat)\rangle$,
$t_6$ — $\langle \|_1\|_1\|_1\|_1\overline{bind\_sub}\langle cn\rangle, \|_1\|_1\|_1\|_0 bind\_sub(cn)\rangle$,
$t_7$ — $\|_1\|_1\|_0\tau_{MEK}$,
$t_8$ — $\|_1\|_1\|_0\tau_{ERK}$,
$t_9$ — $\langle \|_1\|_1\|_1\|_0\overline{bind\_ap1}\langle cn\rangle, \|_1\|_1\|_0 bind\_ap1(cn)\rangle$.

Example 4 shows that the initial interaction $t_0$ causes both $t_1$ and $t_3$, that instead are unrelated. It is also straightforward deducing that also $t_1 \sqsubseteq t_2 \sqsubseteq t_4 \sqsubseteq t_5 \sqsubseteq t_7 \sqsubseteq t_8$ but none of them depend on $t_3, t_6$, or vice versa. Finally, $t_5, t_6 \sqsubseteq t_9$, so reflecting that *NF-AT* needs both pathways.

Our modelization of *NF-AT* activation pathway is correct, and faithfully respect in vitro experiments. Unfortunately, as said in Section 5, our framework has a purely non-deterministic semantics. Instead, compounds react according to some probability distribution, determined by several quantitative factors, among which the most important is the concentration of reactants, considered in the stochastic semantics proposed by Priami et al. [47]. Also the distance between molecules has some effects that we can model by exploiting the simple compartment semantics of Section 5 to obtain more accurate descriptions.

We illustrate the above by adding to *Sys* an inhibitor of the interaction of calcineurin with cytosolic *NF-AT*. The cyclosporin A (*CsA*, for short) is an immunosuppressive medication that interacts with the calcineurin in *NF-AT* and produces cyclophilin. Roughly, the binding power of *CsA* is greater than that of *NF-AT*: this inhibitor has thus a probability of bind calcineurin close to 1, while cytosolic subunit of *NF-AT* has a low probability instead. To model this, we impose that the distance between *CsA* and *CN*, the calcineurin present in *NF-AT*, is small, while the distance of *CN* and *cN*F-*AT* is great.

Formally, we define *CsA* as

$$CsA = bind\_sub(x).Cyclophilin$$

and the whole process becomes

$$Sys \mid CsA.$$

Now consider a suitable value $T$ as threshold to be fixed according to many aspects, e.g. presence of the reacting compounds in the same compartment, their concentration, etc.

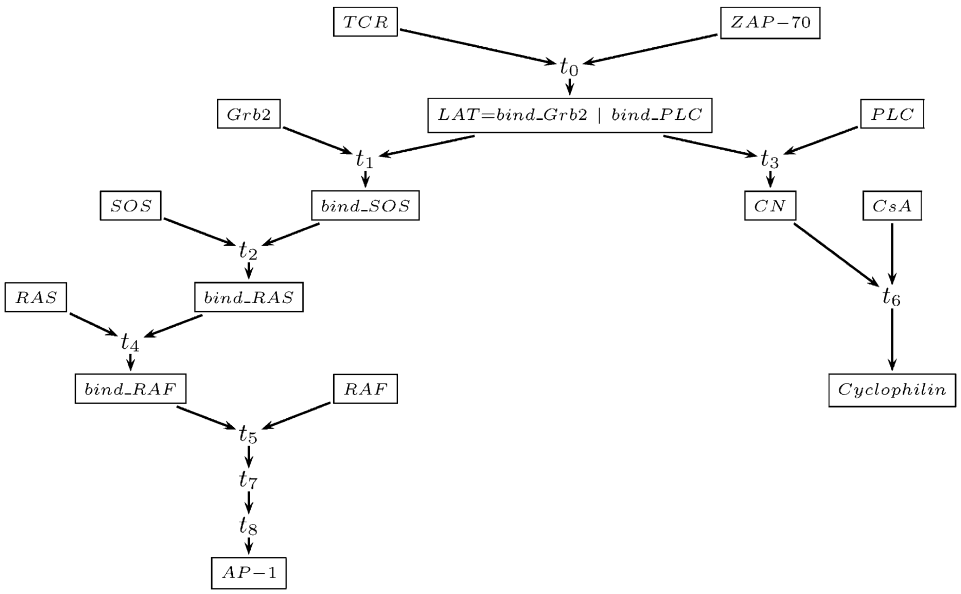


Fig. 5. The computation of $Sys \mid CsA$ when the distance between $CN$ and $CsA$ is below threshold. It shows that $CsA$ act as an inhibitor of $CN$ with $cNF\text{-}AT$.

Then define the distance $\partial$ on the addresses of processes as

$$\partial(\|_0\|_1\|_1\|_1\|_0, \|_0\|_1\|_1\|_1\|_1) > T,$$

$$\partial(\|_0\|_1\|_1\|_1\|_1, \|_1) \leqslant T,$$

$$\partial(\vartheta, \vartheta') \leqslant T \quad \text{elsewhere.}$$

As already seen the transition $t_3$ produces $CN$. According to the new rule for communication given in Section 5, $CN$ and $CsA$ can interact through *bind_sub* (see transition $t_{10}$ in Fig. 5) because their distance is below the threshold. Instead, $CN$ and $cNF\text{-}AT$ cannot, being too far from each other. So $CsA$ inhibits the transition $t_6$ present in Fig. 4 and the whole system gives raise to the partial order displayed in Fig. 5. Note that the Ras/MAP kinase pathway is still followed, because we assumed that all the reactants involved are close enough. Instead, the $NF\text{-}AT$ is not generated, because the calcium/calcineurin pathway has been interrupted.

The last part of our example shows how our approach could help testing in silico medicines, thus making one step towards predictive medicine.

### 6.2. Glycolysis pathway

The glycolysis pathway (GP) is a sequence of reactions that converts glucose into Pyruvate and Lactate with the production of a small amount of adenosine triphosphate. The GP is well studied in Systems Biology, e.g. there is an XML-based static,

quantitative and qualitative representation [22] and a qualitative static and dynamic representation in terms of a double stratified coloured Petri Net [18]. Particularly relevant with respect to our modeling is the last one, because it has a graphical representation similar to ours. As in the previous example, we put all the (active and inactive) proteins needed to produce Lactate. For brevity here actions and constants are prefixed by their addresses. We consider the following system:

$$GPSys = (((\|_0\|_0\|_0 Gluc \mid \|_0\|_0\|_1 ATP) \mid \|_0\|_1 ATP) \mid ((\|_1\|_0\|_0 G6Piso \mid \|_1\|_0\|_1 GAP - 2) \mid \|_1\|_1 D3P)).$$

$ATP$ is a molecule diffused in all living organism, and is involved in all biochemical processes in which an energy exchange occurs. As a matter of fact, $ATP$ was also involved in the previous example about the activation of $TCR$; there we decided to ignore it for keeping that description manageable. The $ATP$ molecule can release energy becoming $ADP$ (adenosine diphosphate); $ADP$ in turn can consume energy becoming $ATP$. Following this, we define $ATP$ and $ADP$ as

$$ATP = bind\_atp(x).ADP,$$
$$ADP = bind\_adp(x).ATP.$$

The first step of GP is represented by the phosphorylation of glucose ($Gluc$). Glucose interacts with $ATP$ trough channel $bind\_atp$ and behaves as glucose 6-phosphate ($G6P$):

$$Gluc = (v\, gluc)\overline{bind\_atp}\langle gluc\rangle.G6P.$$

The second step in glycolysis is the isomerization of glucose 6-phosphate to fructose 6-phosphate ($F6P$). We model this by an interaction of $G6P$ with a process $G6Piso$:

$$G6P = (v\, g6p)\overline{bind\_g6piso}\langle g6p\rangle.FSP,$$
$$G6Piso = bind\_g6piso(g6p).$$

A second phosphorylation involves $FSP$ trough an interaction with $ATP$. Its result is fructose 1,6-bisphosphate ($FBP$):

$$FSP = (v\, fsp)\overline{bind\_atp}\langle fsp\rangle.FBP.$$

The second stage of glycolysis involves the cleavage of the 6-carbon fructose 1,6-bisphosphate to 3-carbon sugars followed by isomerizations. This reaction generates dihydroxyacetone phosphate ($DHAP$) and glyceraldehyde 3-phosphate ($GAP$):

$$FBP = (v\, fbp)\overline{bind\_fbp}\langle fbp\rangle.(\|_0 DHAP \mid \|_1 GAP),$$
$$D3P = bind\_fbp(fbp).$$

The next reaction of glycolysis generates a high-potential phosphorylated compound, 1,3-bisphosphoglycerate ($BPS$). This compound is formed from glyceraldehyde

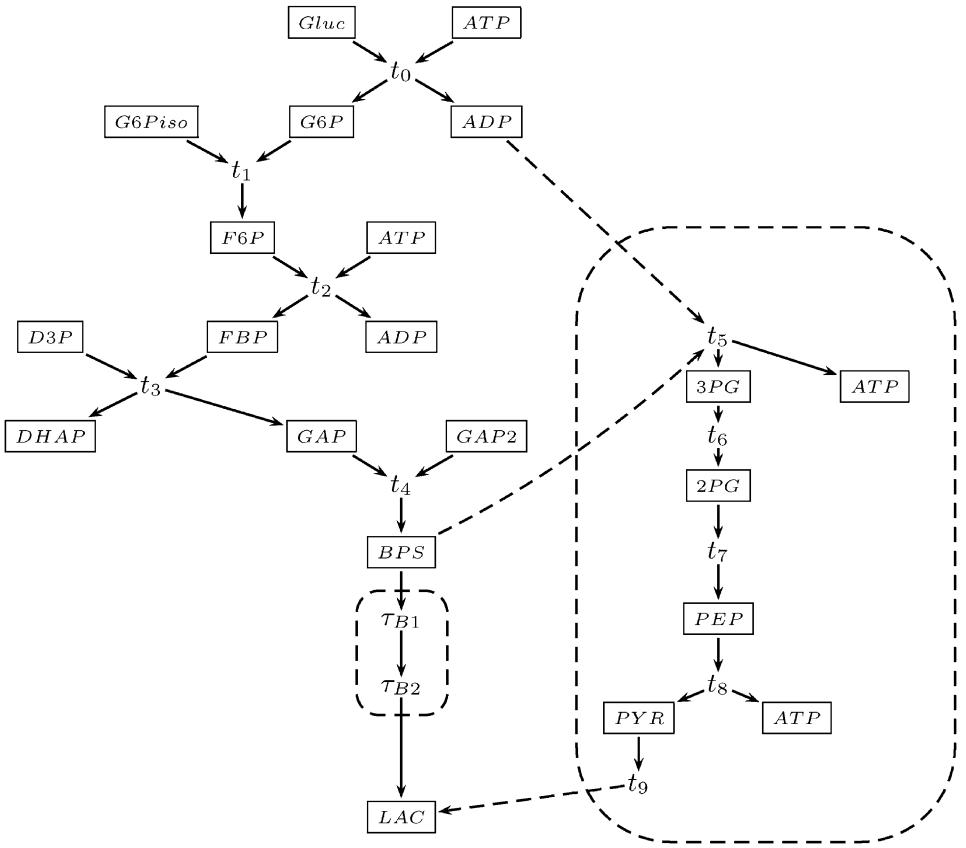


Fig. 6. A computation of glycolysis pathway. The $\tau$-transitions in the box give an abstract representation of a series of reactions. Further details are in the dashed rightmost part.

3-phosphate by the action of the enzyme glyceraldehyde 3-phosphate dehydrogenase ($GAP2$).

$$GAP = (\nu\, gap)\overline{bind\_gap}\langle gap\rangle.BPS,$$
$$GAP2 = bind\_gap(gap).$$

Finally, the definition of $BPS$ model the production of lactate ($Lac$):

$$BPS = \tau_{B1}.\tau_{B2}.Lac.$$

Here $\tau_{B1}$ and $\tau_{B2}$ are an abstraction of a 4-step reaction. The computation in Fig. 6 describes the evolution of $GPSys$ and models the GP pathway. As said above the last reactions of glycolysis produce also the formation of Pyruvate and more molecules of $ATP$. This is accomplished by a rearrangement of 3-phosphoglycerate ($3PG$) to form 2-phosphoglycerate ($2PG$) followed by a dehydration to form phosphoenolpyruvate ($PEP$). The final reactions are the formation of $ATP$ and Pyruvate catalyzed by the

enzyme Pyruvate kinase ($Pyr$), and their subsequent reduction to Lactate. Formally, the two transition $\tau_{B1}$ and $\tau_{B2}$ are detailed as follows:

$$
\begin{aligned}
BPS &= (\nu\, b\, ps)\overline{bind\_adp}\langle bps\rangle.3PG,\\
3PG &= \tau_{3PG}.2PG,\\
2PG &= \tau_{2PG}.PEP,\\
PEP &= (\nu\, pep)\overline{bind\_adp}\langle pep\rangle.PYR,\\
Pyr &= \tau_{Pyr}.Lac.
\end{aligned}
$$

The leftmost part shows the computation where $BPS$ has the abridged definition. The two $\tau$-actions within the box are to be replaced by the whole computation in the dashed box in the rightmost part of the figure. The tool under development, that displays computations, offers its users to choose between the abstract representation and the more detailed one, thus simplifying to zoom in and out in pieces of computations.

## 7. Related work

There are many proposal of biochemical modeling [4,7,11,17,41,49,52,55]. Here we concentrate our attention to those relying on a linguistic level and to those based on Petri Nets. Also, we briefly survey proposals that use differential equations to model reactions. A final paragraph discusses some proposals for repositories of biochemical pathways.

The pioneering work on modeling biochemical systems with a calculus is the work by Fontana and Buss [13] where a version of the $\lambda$-calculus is used. A better account of pathways descriptions is proposed by Regev et al. [52] via a calculus for mobility. As said in more detail at the beginning of Section 6, processes represent compounds and communications represent interactions. This is the approach we followed here. Then, Priami et al. [47] enrich this model with quantitative aspects. They exploit a stochastic variant of the $\pi$-calculus, originally proposed by Priami [45] to evaluate the performance of concurrent and mobile processes. Along the same line described above, we mention also the Bio-calculus proposed by Nagasaki et al. [41], a framework that include an equation editor, an interpreter and a graphical interface; its users describe pathways by using conventional biochemical equations. Recently, Regev [50] proposed *Bio-ambients*, a variant of Cardelli and Gordon's [3] Ambient Calculi in which compartments are described as a hierarchy of boundary ambients. This hierarchy can be modified by suitable operations that have an immediate biological interpretation. For example, the *enter n* primitive, that moves an ambient into a (sibling) ambient $n$, models a compartment entry. Ambients contain compounds that interact via communications. A communication is only possible if the involved processes obey to some constraints, e.g. either they are in the same compartment (*local* communications), or they belong to two parallel compartments (*sibling* communication), or they belong to two ambients one within the other (*parent–child* communication). The last

two (non-standard)interactions are particularly suitable for biochemical modeling: for example, parent–child communications model interactions between compounds that reside in the cytosol and in the nucleus, respectively. This approach is alternative to our simple proposal of Section 5 that requires reactants to be close enough for interacting. Danos and Laneve [6,7] and Chiaverini and Danos [5] recently proposed *Core Formal Molecular Biology*, a process algebra that builds up on the basic primitives of the $\pi$-calculus. As in the other language-based models mentioned above, processes represent compounds, set of processes represent solutions, and their behaviour is given by a set of rewriting rules, driven by suitable side conditions. The proposed rules are very related to the biological realm and mimic typical reactions that occur in biochemical networks, e.g. activation, synthesis, complexation, etc. The present version neither has quantitative descriptions (like the stochastic one of [47]), nor advanced qualitative ones (like causality studied here).

*Petri Nets* is a graphical formalism used to model concurrent systems [53]. Essentially, a Petri Net is an automaton whose states are sets of distributed components, called *places*. A *transition* may transform only some elements of a state, so more than one transition can occur at the same time. Variants of Petri Nets has been widely used in System Biology [17,18,20,31,44,49], also because of their intuitive graphical representation, quite similar to our causal representation of process evolutions. However, Petri Nets lack a satisfactory linguistic and algebraic level, so making it difficult, e.g. the composition of a system and a fortiori of their behaviour. Place/transition Petri Nets are used in [49] for a qualitative analysis of biochemical processes. As expected, places represent molecules and transitions represent reactions.

Recently, Heiner et al. [19] uses the semantics of the formalism to decide coherency of models. Self-modified Petri Nets are used to represent a quantitative model of biochemical networks [20] and Hybrid Petri Nets [33] model regulatory networks by taking into account concentrations of proteins and RNA. Stochastic activity networks (SANs) extend Petri Nets, and are used in [40] for representing biological pathways and simulating their kinetics. MetaNets [31] are a graph theoretical model of metabolic gene-expression networks. Nodes represent metabolites, enzymes and nucleic acid, while arcs represent relationship between pairs of node (e.g. substrate, inhibitor). The method can identify regulatory properties of metabolic networks. Among them, the GP pathway of Section 6.2 is modelled as a (double stratified) coloured Petri Nets and some properties have been deduced using invariant-based techniques. Recent approaches are based on statecharts that build a qualitative model of cellular systems with molecular components [21,23,24].

There is another approach to the simulation of biochemical pathways, mainly focused on determining the quantitative input/output behaviour of bio-systems, rather than on a detailed description of the stages in pathways. In most of these models, reactions are represented by stochastic (continuous or discrete) processes, and Monte-Carlo algorithms [14,15] drive the evolution of the system. Among these proposals, we only mention E-cell [55], Bio-calculus [41] and Gepasi [35] that have a graphical implementation, in which users can insert pathway models and run their simulation in a single framework. The output of these tools shows the variation of the concentration of the compounds involved while the pathway proceeds.

Finally, there is the problem of storing the large amount of data available on biochemical pathways [1,4,22,26–29,32,34,43,54]. These archives offer a large amount of static information about pathways, their relations and their evolution. The aim is to share as much as possible *formal* biochemical knowledge, instead of informal natural language descriptions (e.g. textbooks as [16]). Many of these repositories are usable through the Internet (e.g. [26,29,43]), making the data stored worldwide available in a standard presentation. A successful example of standardization is defined through XML [56], a meta-language that offers standard ways to define markup languages. Several groups use it for exchanging biochemical models (e.g. SBML [22] and CellML [4]). Tightly connected to the above, there is the definition of which kind of information is necessary to describe a pathway. First proposals are in [27,34] that define ontologies to identify biochemical concepts and formalize relationships between them.

## 8. Conclusions

We gave the $\pi$-calculus a non-standard reduction semantics, in which reductions have labels. A mechanical interpretation of them provides us with the causal relations between the transitions in a computation. Besides its interest in se, a causal reduction semantics receives motivation from its ability to faithfully describe biochemical processes. In fact, the dynamic analysis of this kind of specifications can predict "real" biological evolutions at a certain degree of accuracy [47,52]. We demonstrated that causality enhances the precision of such simulations, through two simple examples, that however suffices to make our point. The description of the networks of proteins we considered shows the potential interactions between them, stage by stage. The causal behaviour of the $\pi$-processes used coincide with the experiments in vitro. In our examples, there is a single computation; of course our abstract semantics produces all the possible computations, covering all the evolutions of the system. Another way to improve accuracy of modelling biological systems is by endowing the $\pi$-calculus with spatial information. In Section 5, we have shown a naive approach to a description of compartments. Transitions may only occur when the involved processes are "close" enough. A matter of fact quantitative aspects and causality are orthogonal, so they can be combined together. As shown in [9,42], our proved semantics provides us with the basis for a single framework permitting accurate qualitative and quantitative descriptions and analyses of biochemical process. The designer writes a single specification and, without changing it, s/he can derive these different descriptions of its evolution. We claim that our modelization makes biochemical descriptions simpler (cf. Fig. 4) than those relying on other models, e.g. [11,12,17,28,30], because here we establish a close link between each interaction of proteins and each step in the system evolution.

A (semi-)automatic tool is under development, that supports the definition of a biochemical system as a $\pi$-calculus process, its execution and the display of the resulting pathway(s) in a graphical form, similar to the one used for Petri Nets. Among the features of our tool, there should be the possibility of changing the grain of the graphical representation by grouping together families of reactants or of reactions. In this way, long or complex pathways can be represented at different levels of detail, so enhancing

the readability of the representation, even in presence of many interacting components (see Fig. 6). Also, different views of the same pathway should be displayed, e.g. with or without quantitative information. Finally, it could be hard representing in a single screen all the many different computations that may arise from the same biochemical process being modelled: our tool will offer its users a facility for interactively choosing among them.

## Acknowledgements

We wish to thank the referees for their precise and invaluable suggestions.

## Appendix A.

### A.1. SOS semantics

**Definition A.1** (Syntax). Let $\mathcal{N}$ be a countable infinite set of *names* ranged over by $a, b, \ldots, x, y, \ldots$ with $\mathcal{N} \cap \{\tau\} = \emptyset$. We also assume a set $\mathcal{X}$ of *agent identifiers* ranged over by $X, X_1, \ldots$ . *Processes* (denoted by $A, B, C, \ldots \in \mathcal{A}$) are built from names according to the syntax

$$A ::= \mathbf{0} \mid \pi.A \mid A + A \mid A|A \mid (\nu x)A \mid [x = y]A \mid X(y_1, \ldots, y_n),$$

where $\pi$ may be either $x(y)$ for *input*, or $\bar{x}y$ for *output* (where $x$ is the *subject* and $y$ the *object*) or $\tau$ for *silent* moves. Hereafter, the trailing $\mathbf{0}$ will be omitted.

Syntax is quite similar to that introduced in Section 2. $X(y_1, \ldots, y_n)$ is the definition of constants (hereafter, $\tilde{y}$ denotes $y_1, \ldots, y_n$). Each agent identifier $X$ has a unique defining equation of the form $X(y_1, \ldots, y_n) = A$, where the $y_i$ are distinct and $fn(A) \subseteq \{y_1, \ldots, y_n\}$ (see below for the definition of free names $fn$).

Congruence rules also are quite similar to those defined in Section 2 for standard $\pi$-calculus, except for the fact that parallel composition is *neither* associative *nor* commutative.

The late operational semantics for the $\pi$-calculus (see Table 3) is defined in the *SOS* style (the symmetric rules for $Com_0$ and $Close_0$ are omitted), and the labels of the transitions are $\tau$ for silent actions, $x(y)$ for input, $\bar{x}y$ for free output, and $\bar{x}(y)$ for bound output. We will use $\mu$ as a metavariable for the labels of transitions (it is distinct from $\pi$, the metavariable for prefixes, though it coincides in two cases).

The notions of free names $fn(\mu)$ is as in Section 2, and $fn(x(y)) = \{x, y\}$ and $bn(x(y)) = \emptyset$.

### A.2. Comparing SOS and reduction semantics

In showing that the reduction and the SOS semantics coincide on close processes, the following two lemmata help, one for each semantics, having exactly the same meaning

Table 3
Late proved SOS transition system for the $\pi$-calculus

$$Act : \mu.A \xrightarrow{\mu}_S A \qquad Ide : \frac{A\{\tilde{y}/\tilde{x}\} \xrightarrow{\theta}_S A'}{X(\tilde{y}) \xrightarrow{\theta}_S A'}, X(\tilde{x}) = A$$

$$Par_0 : \frac{A \xrightarrow{\theta}_S A'}{A|B \xrightarrow{\|_0\theta}_S A'|B}, bn(\ell(\theta)) \cap fn(B) = \emptyset \qquad Sum : \frac{A \xrightarrow{\theta}_S A'}{A + B \xrightarrow{\theta}_S A'}$$

$$Par_1 : \frac{A \xrightarrow{\theta}_S A'}{B|A \xrightarrow{\|_1\theta}_S B|A'}, bn(\ell(\theta)) \cap fn(B) = \emptyset \qquad Res : \frac{A \xrightarrow{\theta}_S A'}{(\nu x)A \xrightarrow{\theta}_S (\nu x)A'}, x \notin n(\ell(\theta))$$

$$Com_0 : \frac{A \xrightarrow{\vartheta\overline{x}y}_S A', B \xrightarrow{\vartheta' x(w)}_S B'}{A|B \xrightarrow{\langle \|_0\vartheta\overline{x}y, \|_1\vartheta' x(w)\rangle}_S A'|B'\{y/w\}} \qquad Open : \frac{A \xrightarrow{\vartheta\overline{x}y}_S A'}{(\nu y)A \xrightarrow{\vartheta\overline{x}(y)}_S A'}, x \neq y$$

$$Close_0 : \frac{A \xrightarrow{\vartheta\overline{x}(y)}_S A', B \xrightarrow{\vartheta' x(w)}_S B'}{A|B \xrightarrow{\langle \|_0\vartheta\overline{x}(y), \|_1\vartheta' x(w)\rangle}_S (\nu y)(A'|B'\{y/w\})}, \ y \notin fn(B)$$

and shape. They show that it is possible to single out the sub-processes active in a transition, by only inspecting its label; similarly for their residuals.

**Lemma A.1.** (a) $A \xrightarrow{\vartheta\mu}_S B$ *if and only if*

$$A@\vartheta \equiv \mu.A' + C \text{ and } B = A[\vartheta \mapsto A'],$$

(b)

$$A \xrightarrow{\vartheta\langle \|_i\vartheta_0 x(w), \|_{1-i}\vartheta_1\overline{x}\langle y\rangle\rangle}_S B$$

*if and only if*

$$A@\vartheta\|_i\vartheta_0 \equiv x(w).A_0 + C_0, \ A@\vartheta\|_{1-i}\vartheta_1 \equiv \overline{x}\langle y\rangle.A_1 + C_1,$$
$$B \equiv A[\vartheta\|_i\vartheta_0 \mapsto A_0\{y/w\}][\vartheta\|_{1-i}\vartheta_1 \mapsto A_1].$$

**Proof.** (a) In order to establish item (a) of the lemma, we prove the following more general statement. Let $A$ a process, $A \equiv (\nu I)A'$, with $A'$ free of restrictions. Then

$$\forall \tau, \hat{\vartheta}.\ A' \xrightarrow{\hat{\vartheta}\tau}_S B' \text{ iff } A'@\hat{\vartheta} \equiv \tau.A'' + C, B' \equiv A'[\hat{\vartheta} \mapsto A''] \qquad (*)$$

We proceed by induction on the length of $\hat{\vartheta}$.

(*Base*) $\hat{\vartheta} = \varepsilon$. $A'@\varepsilon = A' \xrightarrow{\tau}_S B'$ iff $A' \equiv \tau.A'' + C$ and $B' \equiv A''$. Immediate, because only the axiom **Act** can occur.

(*Induction step*) Suppose now that $\hat{\vartheta} = \|_0\hat{\vartheta}'$ (resp. $\|_1\hat{\vartheta}'$). We have that the transition $t = A' \xrightarrow{\|_0\hat{\vartheta}'\tau}_S B'$ was deduced using rule $\mathbf{Par_0}$ with premise $t' = \hat{A}' \xrightarrow{\hat{\vartheta}'\tau}_S Z$ for some $Z$,

and so $t$ is on the form $A' \equiv \hat{A}'|D \xrightarrow{\|_0\hat{\vartheta}'\tau}_S Z|D \equiv B'$ for some $D$. Now we can apply to $t'$ our inductive hypothesis and obtain

$$\hat{A}'@\hat{\vartheta}' \equiv \tau.A'' + C,$$
$$Z \equiv \hat{A}'[\hat{\vartheta}' \mapsto A''].$$

We can then conclude that

$$A'@\hat{\vartheta} = A'@\|_0\hat{\vartheta}' \equiv \hat{A}'@\hat{\vartheta}' \equiv \tau.A'' + C,$$
$$B' \equiv Z|D \equiv \hat{A}'[\hat{\vartheta}' \mapsto A'']|D \equiv A'[\|_0\hat{\vartheta}' \mapsto A''].$$

(b) ($\Leftarrow$) Because of item (a), we have that

(i) $A@\vartheta\|_i \xrightarrow{\vartheta_0 x(w)}_S (A@\vartheta\|_i)[\vartheta_0 \mapsto A_0]$ iff $A@\vartheta\|_i\vartheta_0 \equiv x(w).A_0 + C$,

(ii) $A@\vartheta\|_{1-i} \xrightarrow{\vartheta_1 \bar{x}\langle y\rangle}_S (A@\vartheta\|_{1-i})[\vartheta_1 \mapsto A_1]$ iff $A@\vartheta\|_{1-i}\vartheta_1 \equiv x(w).A_1 + D$.

So, from (i) and (ii) we can deduce

$$(A@\vartheta\|_i)|(A@\vartheta\|_{1-i}) \xrightarrow{\langle\vartheta_0 x(w),\vartheta_1\bar{x}\langle y\rangle\rangle}_S (A@\vartheta\|_i)[\vartheta_0 \mapsto A_0] \,|\, (A@\vartheta\|_{1-i})[\vartheta_1 \mapsto A_1\{y/w\}]$$

By definition $(A@\vartheta\|_i \,|\, A@\vartheta\|_{1-i}) = A@\vartheta$. We conclude that

$$A \xrightarrow[S]{\vartheta\langle\|_i\vartheta_0 x(w),\|_{1-i}\vartheta_1\bar{x}\langle y\rangle\rangle} (A@\vartheta\|_i)[\vartheta_0 \mapsto A_0] \,|\, (A@\vartheta\|_{1-i})[\vartheta_1 \mapsto A_1\{y/w\}].$$

($\Rightarrow$) Reversing the argument above. □

The following proposition says that for every process of our enriched $\pi$-calculus, there exists another one in which addresses respect the structure of the processes.

**Proposition A.1.** *$\forall P.\ \exists A$ process s.t. $P \equiv \hat{P} = \mathcal{T}(A)$.*

**Proof.** By induction on the shape of $P$:

- $\vartheta\mathbf{0}$ or $\vartheta X(y)$: then $A$ is on the form $\mathbf{0}$ or $X(y)$, respectively, and the proof is trivial;
- $P_0 + P_1$, $(\nu x)P_0$, $[x = y]P_0$: then by induction hypothesis there exist $\hat{P}_i \equiv P_i$, $i \in \{0,1\}$ and $A_i$ s.t. $\mathcal{T}(A_i) = \hat{P}_i$. Therefore $A$ is on the form $A_0 + A_1$, $(\nu x)A_0$, $[x = y]A$, respectively, and the induction hypothesis suffices;
- $\|_i \rhd P_0 \,|\, \|_{1-i} \rhd P_1$. If $i = 0$ then let $\hat{P} = P$; otherwise let $\hat{P} = \|_{1-i} \rhd P_1 \,|\, \|_i \rhd P_0$. In both cases the induction hypothesis suffices. □

The following lemma is analogous to Lemma A.1, applied for the reduction semantics.

**Lemma A.2.** (a) *$P \xrightarrow{\vartheta\tau}_R Q$ if and only if $P@\vartheta \equiv \tau.P' + R$ and $Q = P[\vartheta \mapsto P']$*

(b)

$$P \xrightarrow{\vartheta\langle\|_i\vartheta_0 x(w),\|_{1-i}\vartheta_1\bar{x}\langle y\rangle\rangle}_R Q$$

*if and only if*

$$P@\vartheta\|_i\vartheta_0 \equiv x(w).P_0 + R_0, P@\vartheta\|_{1-i}\vartheta_1 \equiv \bar{x}\langle y\rangle.P_1 + R_1,$$
$$Q \equiv P[\vartheta\|_i\vartheta_0 \mapsto P_0\{y/w\}][\vartheta\|_{1-i}\vartheta_1 \mapsto P_1].$$

**Proof.** (b) The transition

$$P \xrightarrow{\vartheta\langle\|_i\vartheta_0 x(w), \|_{1-i}\vartheta_1\bar{x}\langle y\rangle\rangle}_R Q$$

can be deduced starting from the rule **Com**, followed by application of other rules. By Proposition A.1 we have that $\exists\hat{P}$ such that $\hat{P}@\vartheta\|_i\vartheta_0 \equiv x(w).P_0 + R_0$ and $\hat{P}@\vartheta\|_{1-i}\vartheta_1 \equiv \bar{x}\langle y\rangle.P_1 + R_1$ (because $\exists A.\mathcal{T}(A) = \hat{P}$ and $\mathcal{T}(A)@\vartheta\|_i\vartheta_0 = x(w).P_0 + R_0$, symmetrically for $\hat{P}@\vartheta\|_{1-i}\vartheta_1$). Any other rule only adds context in the conclusions and never change the label. So the thesis follows immediately. The same arguments prove (a). □

The equivalence between the SOS and the reduction semantics follows from the lemmas above.

**Theorem A.1.** *Let* $\xrightarrow{\theta}_S$ *the relation defined in the SOS semantics of* [8]. *Then*, $A \xrightarrow{\theta}_S B$ *iff* $\mathcal{T}(A) \xrightarrow{\theta}_R \mathcal{T}(B)$.

**Proof.** ($\theta = \vartheta\tau$) ($\Rightarrow$) Lemma A.1 ensures that $A \xrightarrow{\vartheta\tau}_S B$ if and only if $A@\vartheta \equiv \tau.A' + C$ and $B = A[\vartheta \mapsto A']$. Proposition A.1 says that for $A, A_0, C$ and $B$ there exist "equivalent" $P, P_0, R$ and $Q$. Finally, Lemma A.2 establish that $P \xrightarrow{\vartheta\tau}_R Q$ if and only if $P@\vartheta \equiv \tau.P' + R$ and $Q = P[\vartheta \mapsto P']$

($\Leftarrow$) Reversing the argument above. ($\theta = \vartheta\langle\|_i\vartheta_0 x(w), \|_{1-i}\vartheta_1\bar{x}\langle y\rangle\rangle$): If the processes $A, B$ stipulate a communication with a label $\vartheta\langle\|_i\vartheta_0 x(w), \|_{1-i}\vartheta_1\bar{x}\langle y\rangle\rangle$, then, due to Lemma A.1, we can deduce

$$A@\vartheta\|_i\vartheta_0 \equiv x(w).A_0 + C_0,$$
$$A@\vartheta\|_{1-i}\vartheta_1 \equiv \bar{x}\langle y\rangle.A_1 + C_1,$$
$$B \equiv A[\vartheta\|_i\vartheta_0 \mapsto A_0\{y/w\}][\vartheta\|_{1-i}\vartheta_1 \mapsto A_1].$$

But due to Proposition A.1 and Lemma A.2, we have that the same happens to extended processes

$$\mathcal{T}(A)@\vartheta\|_i\vartheta_0 \equiv x(w).\mathcal{T}(A_0) + \mathcal{T}(C_0),$$
$$\mathcal{T}(A)@\vartheta\|_{1-i}\vartheta_1 \equiv \bar{x}\langle y\rangle.\mathcal{T}(A_1) + \mathcal{T}(C_1),$$
$$\mathcal{T}(B) \equiv \mathcal{T}(A)[\vartheta\|_i\vartheta_0 \mapsto \mathcal{T}(A_0)\{y/w\}][\vartheta\|_{1-i}\vartheta_1 \mapsto \mathcal{T}(A_1)].$$

The thesis follows. □

## References

[1] Altschul, et al. Issues in searching molecular sequence database, Nat. Gen. 6 (1994) 119–129.
[2] C. Bodei, P. Degano, C. Priami, Names of mobile agents handled locally, Theoret. Comput. Sci. 253 (2) (2000) 155–184.
[3] L. Cardelli, A.D. Gordon, Mobile ambients, in: Foundations of Software Science and Computation Structures: First Internat. Conf., FOSSACS '98, Springer, Berlin, Germany, 1998.
[4] CellML Home Page, http://www.cellml.org.
[5] M. Chiaverini, V. Danos, A core modeling language for the working molecular biologist, in: C. Priami (Ed.), Computational Methods in Systems Biology, First Internat. Workshop, CMSB 2003, Proc., Rovereto, Italy, February 24–26, 2003, Lecture Notes in Computer Science, Vol. 2602, Springer, Berlin, 2003.
[6] V. Danos, C. Laneve, Causal $\pi$-calculus for biochemical modelling, in: C. Priami (Ed.), Computational Methods in Systems Biology, First Internat. Workshop, CMSB 2003, Proc., Rovereto, Italy, February 24–26, 2003, Lecture Notes in Computer Science, Vol. 2602, Springer, Berlin, 2003.
[7] V. Danos, C. Laneve, Core formal molecular biology, in: P. Degano (Ed.), Programming Languages and Systems, 12th European Symp. on Programming, ESOP 2003, Warsaw, Poland, April 7–11, 2003, Lecture Notes in Computer Science, Vol. 2618, Springer, Berlin, 2003.
[8] P. Degano, C. Priami, Non-interleaving semantics for mobile processes, Theoret. Comput. Sci. 216 (1, 2) (1999) 237–270.
[9] P. Degano, C. Priami, Enhanced operational semantics: a tool for describing and analysing concurrent systems, ACM Comput. Surv. 33 (2) (2001) 135–176.
[10] P. Degano, C. Priami, Eos for system biology, in: C. Priami (Ed.), Computational Methods in Systems Biology, First Internat. Workshop, CMSB 2003, Proc., Rovereto, Italy, February 24–26, 2003, Lecture Notes in Computer Science, Vol. 2602, Springer, Berlin, 2003.
[11] S. Eker, M. Knapp, K. Laderoute, P. Lincoln, J. Meseguer, K. Sonmez, Pathway logic: symbolic analysis of biological signaling, in: Proc. Pacific Symp. on Biocomputing, 2002, pp. 400–412.
[12] D. Endy, R. Brent, Modelling cellular behaviour, Nature 409 (2001) 391–395.
[13] W. Fontana, L.W. Buss, The Barrier of Objects: From Dynamical System to Bounded Organizations, Addison-Wesley, Reading MA, 1996, pp. 56–116.
[14] D.T. Gillespie, A general method for numerically simulating the stochastic time evolution of coupled chemical reactions, J. Comput. Phys. 22 (1976) 403–434.
[15] D.T. Gillespie, Exact stochastic simulation of coupled chemical reactions, J. Phys. Chem. 81 (1977) 2340–2361.
[16] B.D. Gomperts, I.M. Kramer, P.E.R. Tatham, Signal Transduction, Academic Press, San Diego, CA, 2002.
[17] P.J.E. Goss, J. Peccoud, Quantitative modeling of stochastic systems in molecular biology by using stochastic Petri nets, in: Proc. of the National Academy of Sciences, USA, No. 12, 1998, pp. 6750–6754.
[18] M. Heiner, I. Koch, K. Voss, Analysis and simulation of steady states in metabolic pathways with petri nets, in: K. Jensen (Ed.), CPN'01-Third Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools, 2001, pp. 15–34.
[19] M. Heiner, I. Koch, J. Will, Model validation of biological pathway using petri nets—demonstrated for apoptosis, in: C. Priami (Ed.), Computational Methods in Systems Biology, First Internat. Workshop, CMSB 2003, Rovereto, Proc., Italy, February 24–26, 2003, Lecture Notes in Computer Science, Vol. 2602, Springer, Berlin, 2003.
[20] R. Hofestädt, S. Thelen, Quantitative modeling of biochemical networks, In. Silico Biol. 1 (1) (1998) 39–53.
[21] M. Holcombe, A. Bell, Computational Models of Immunological Pathways, Proceedings of IPCAT 97, Sheffield, UK, 1997, pp. 213–226.
[22] M. Hucka, et al., The System biology markup language (SBML): a medium for representation and exchange of biochemical network models, Bioinformatics 19 (2003) 524–531.
[23] N. Kam, I.R. Cohen, D. Harel, The immune system as a reactive system: modelling t all activation with statecharts, Tech. Rep MCS01–09, The Weizmann Institute of Science, Rehovot, Israel, June 2001.

[24] N. Kam, D. Harel, I.R. Cohen, Modelling biological nactivity: statecharts vs. boolean logic, Proc. 2nd Int. Conf. on Systems Biology, Pasadena, CA, USA, November 2001.
[25] L.P. Kane, J. Lin, A. Weiss, Signal transduction by the tcr for antigen, Curr. Opin. Immunol. 12 (2000) 242–249.
[26] M. Kanehisa, S. Goto, Kegg: Kyoto encyclopedia of genes and genomes, Nucleic Acids Res. 28 (1) (2000) 27–30.
[27] P.D. Karp, An ontology for biological function based on molecular interactions, Bioinformatics 16 (2000) 269–285.
[28] P.D. Karp, Pathway databases: a case study in computational symbolic theories, Science 293 (2001) 2040–2044.
[29] P.D. Karp, M. Riley, S.M. Paley, A. Pellegrini-Toole, M. Krummenacker, Eco cyc: encyclopedia of *Escherichia coli* genes and metabolism, Nucleic Acids Res. 27 (1) (1999) 55–58.
[30] K.W. Kohn, Molecular interaction map of the mammalian cell cycle control and dna repair systems, Molecular Biology of the Cell 10 (1999) 2703–2734.
[31] M.C. Kohn, D.R. Lemieux, Identification of regulatory properties of metabolic networks by graph theoretical modeling, J. Theoret. Biol. 150 (1991) 3–25.
[32] F.A. Kolpakov, E.A. Ananko, G.B. Kolesov, N.A. Kolchanov, Genenet: a gene network database and its automated visualization, Bioinformatics 14 (8) (1998) 529–537.
[33] H. Matsuno, A. Doi, M. Nagasaki, S. Miyano, Hybrid petri net representation of gene regulatory network, in: Pacific Symp. on Biocomputing, 2000.
[34] R. McEntire, P. Karp, N. Abernethy, D. Benton, G. Helt, M. DeJongh, R. Kent, A. Kosky, S. Lewis, D. Hodnett, E. Neumann, F. Olken, D. Pathak, P. Tarczy-Hornoch, L. Toldo, T. Topaloglou, An evaluation of ontology exchange languages for bioinformatics, in: R. Altman, L. Bailey, Timothy, P. Bourne, M. Gribskov, T. Lengauer, I.N. Shindyalov (Eds.), Proc. of the 8th Internat. Conf. on Intelligent Systems for Molecular (ISMB-00), AAAI Press, Menlo Park, CA, August 16–23, 2000, pp. 239–250.
[35] P. Mendes, Gepasi: a software package for modelling the dynamics, steady states and control of biochemical and other systems, Comput. Appl. Biosci. 9 (5) (1993) 563–571.
[36] R. Milner, Communication and Concurrency, Prentice-Hall, Englewood Cliffs, NJ, 1989.
[37] R. Milner, Functions as processes, Math. Struct. CS 2 (2) (1992) 119–141.
[38] R. Milner, Communicating and Mobile Systems: The $\pi$-Calculus, Cambridge University Press, 1999.
[39] R. Milner, J. Parrow, D. Walker, A calculus of mobile processes (i and ii), Inform. Comput. 100 (1) (1992) 1–72.
[40] W.M. Mounts, M.N. Liebman, Qualitative modeling of normal blood coagulation and its pathological states using stochastic activity networks, Internat. J. Biol. Macromol. 20 (1997) 265–281.
[41] M. Nagasaki, S. Onami, S. Miyano, H. Kitano, Bio-calculus: its concept and molecular interaction, Genome Informatics 10 (1999) 133–143.
[42] C. Nottegar, C. Priami, P. Degano, Performance evaluation of mobile processes via abstract machines, IEEE Trans. Software Engrg. 27 (10) (2001) 867–889.
[43] H. Ogata, S. Goto, K. Sato, W. Fujibuchi, H. Bono, M. Kanehisa, Kegg: Kyoto encyclopedia of genes and genomes, Nucleic Acid Res. 27 (1) (2000) 29–34.
[44] M. Peleg, I. Yeh, R. Altman, Modeling biological processes using workflow and petri net models, Bioinformatics 18 (2002) 825–837.
[45] C. Priami, Stochastic $\pi$-calculus, Comput. J. 38 (6) (1995) 578–589.
[46] C. Priami, Language-based performance prediction for distributed and mobile systems, Inform. and Comput. 175 (2002).
[47] C. Priami, A. Regev, W. Silverman, E. Shapiro, Application of a stochastic passing-name calculus to representation and simulation of molecular processes, Inform. Process. Lett. 80 (2001) 25–31.
[48] A. Rao, C. Luo, P.G. Hogan, Transcription factors of the nfat family: regulation and function, Ann. Rev. Immunol. 15 (1997) 707–747.
[49] V.N. Reddy, M.L. Mavrouvouniotis, M.N. Liebman, Qualitative analysis of biochemical reduction systems, Comput. Biol. Med. 26 (1) (1996) 9–24.
[50] A. Regev, Computational system biology: a calculus for biomolecular knowledge, PhD Thesis, Tel Aviv University, Israel, 2003.

[51] A. Regev, E. Shapiro, Cells as computations, Nature 419 (2002) 343.
[52] A. Regev, W. Silverman, E. Shapiro, Representation and simulation of biochemical processes using the π-calculus process algebra, in: Pacific Symp. of Biocomputing (PSB2001), 2001, pp. 459–470.
[53] W. Reisig, Petri Nets: An Introduction, Springer, Berlin, 1985.
[54] M. Hucka, A. Finney, H. Sauro, H. Bolouri, The ERATO systems biology workbench: architectural evolution, in: T.M. Yi, M. Hucka, H. Kitano (Eds.), Proc. 2nd International Conference on Systems Biology, 2001.
[55] M. Tomita, et al., E-CELL: software environment for whole-cell simulation, Bioinformatics 15 (1998) 72–84.
[56] Xml web site: http://www.w3.org/xml.