

# Biological Transactions for Quantitative Models

Federica Ciocchetta<sup>a,1</sup> and Corrado Priami<sup>b,2</sup>

<sup>a</sup> *Dipartimento di Informatica e Telecomunicazioni  
University of Trento  
Trento, Italy*

<sup>b</sup> *Dipartimento di Informatica e Telecomunicazioni  
University of Trento  
the Microsoft Research-University of Trento Centre for Computational and Systems Biology  
Trento, Italy*

---

## Abstract

In this work an extension of stochastic  $\pi$ -calculus with *biological transactions* is presented. This permits to model multi-reactant multi-product reactions as atomic actions when quantitative information are given. First, the syntax and the semantics are defined, then some transaction properties are discussed. Finally, some examples are described.

*Keywords:* Systems Biology, stochastic  $\pi$ -calculus, biological transactions

---

## 1 Introduction

*Systems Biology* is a recent research field that aims at a system-level understanding of biological systems [14]. In the last years, as a response to the need of modeling the dynamics of complex biological systems, there have been some successful applications of process calculi in this research area [17,7,11,18]. One of the most used process algebras in this field is the  $\pi$ -calculus [16], a formal language originally developed for specifying concurrent computational systems. The biochemical stochastic  $\pi$ -calculus described in [17] has been applied to represent and analyze a great variety of biological models [10,3,19]. A critical point in the specification of biological systems is the translation of reactions with more than two reactants, as a reaction is modeled in  $\pi$ -calculus as a pairwise communication between two processes. Reactions with more than two reactants are rare in nature, but are often used

---

<sup>1</sup> fedecioc@dit.unitn.it

<sup>2</sup> priami@dit.unitn.it

in biological models as abstractions of complex phenomena. A common way to deal with these reactions is to decompose them into a sequence of two-reactant reactions. Following this approach some problems arise. First of all, different combinations of reactants are possible and there are  $\frac{n!}{2}$  ways to decompose a reaction (where  $n$  is the number of reactants). Moreover, it can happen that after the formation of an intermediate complex, the further needed reactant misses, leading to a deadlock. A further critical aspect regarding quantitative models is that only one rate is associated to the reaction and from it the rates for each elementary reactions should be derived. Here a different approach is considered. Following the idea in [9], *biological transactions* are introduced to model complex reactions. Generally speaking, a transaction is a sequence of computational steps of a distributed system which must be executed as if it were a single atomic action. Recently, there have been some attempts to model transactions formally by using process algebras [5,15,4]. In [5]  $\pi t$ -calculus is presented: it is an extended version of asynchronous  $\pi$ -calculus to deal with long time transactions and offers failure handlers when process interruptions are met. Another extension of asynchronous  $\pi$ -calculus with long-time transactions, called *web- $\pi$* , is introduced in [15]. CSP is the process algebras adopted in [4] to model long-running transactions with traces. In all these works only qualitative aspects are considered.

The aim of this work is to extend the approach presented in [17] to define a variant of stochastic  $\pi$ -calculus with biological transactions. The biological transactions proposed in this work satisfy some simple properties that are suitable for modeling complex reactions. For instance, after a transaction has started, it executes in isolation and ends successfully. Besides, the result is visible only when the transaction ends. In the standard calculus, only simple reactions with at most two reactants are considered, as proposed in [13]. Since transactions involve more than two reactants, an extended version of Gillespie's algorithm is necessary [8].

Other formalisms permit to represent multiple-reactant multiple-product reactions [12,1]. In [12] a core modeling language, called  $\kappa$ -calculus, is introduced. The complexation rule permits to represent reactions involving several reactants. Furthermore, an encoding of this calculus in the asynchronous  $\pi$ -calculus is given. However, this calculus offers only a qualitative description of the model. In [1], a stochastic version of Multiset Rewriting (SMSR) is used to model metabolic pathways. Also in this case it is possible to encode the calculus in a given process algebra, at least in the qualitative case [2]. In the proposed work we decide to enrich the standard stochastic  $\pi$ -calculus directly, without considering other intermediate languages.

The rest of the paper is organized as follows. The next section introduces the calculus. After that, some properties are discussed in section 3 and then two examples are described in 4. The last section reports some final remarks.

## 2 The stochastic $\pi$ -calculus with biological transactions

In this section we report the syntax and the semantics of the  $\pi$ -calculus with biological transactions.

### 2.1 General ideas

We follow the approach proposed in [17], founded on Gillespie [13]. Accordingly to Gillespie, the actual rate or speed of a reaction  $R$  is the product of the *basal rate*  $r_b$  (i.e. an empirical constant depending on the reaction) times the number of possible combinations of the reactants available in a given state. The dynamic evolution of the system is based on a *race condition*: all the different actions that are enabled in a state compete and the fastest one succeeds. The “standard” biochemical stochastic  $\pi$ -calculus considers only reactions with at most two reactants as a reaction is represented by a pairwise communication between two processes. Since transactions represent reactions with more than two reactants, an extended version of Gillespie approach is necessary (as proposed in [8]). Moreover, as the speed of a complex reaction is described by a unique rate, that rate is associated to the start prefix of the transaction and  $\infty$  (i.e. the associate action is immediate) is assigned to all the other prefixes in the processes related to the transaction. The actual rate of the reaction depends on the rate in the start prefix and on the possible combinations of all the other processes used in the transaction.

### 2.2 Syntax

The action prefixes  $\pi$  and the processes  $P$  are defined in the stochastic  $\pi$ -calculus with biological transactions by the following syntax:

$$\pi ::= (x(z), r) \mid (\bar{x}\langle z \rangle, r) \mid (start(t), r)[P] \mid (end(t), r)$$

$$P ::= \sum_{i=1}^m \pi_i.P_i \mid \nu y P \mid P \mid P \mid A(\tilde{y}) \mid t[P]$$

where we suppose a countable infinite set of channel names  $\mathcal{N}$  (ranged over by lower-case letters  $x, y, z, \dots$ ) and a countable set of transaction names  $\mathcal{T}$  (ranged over by lower-case letters  $t, t', t'', \dots$ ), with  $\mathcal{T} \cap \mathcal{N} = \emptyset$ . The processes generated by the grammar above differ from the ones of the stochastic  $\pi$ -calculus in the prefixes  $(start(t), r)[P]$ ,  $(end(t), r)$  and in the process  $t[P]$ . All other processes have the same meaning as in standard  $\pi$ -calculus, [16,20]. As for the prefixes, following [17] each prefix is characterized by a rate  $r$ , indicating the single parameter of an exponential distribution that characterizes the stochastic behavior of the activity corresponding to the prefix  $\pi$ . The rate  $r$  is either a non-negative real number or  $\infty$ . The prefix  $start(t)[P]$  is used to indicate the start of a transaction of name  $t$ . The process  $P$  inside the square brackets indicates an external process up to structural congruence necessary for the reaction to start. In the same way, the prefix  $end(t)$  indicates that the transaction  $t$  ends successfully. As for processes, note that the constant

$P_1 \equiv P_2$ provided $P_1$ is an $\alpha$ -converse of $P_2$ $A(\tilde{y}) \equiv P\{\tilde{y}/\tilde{x}\}$ if $A(\tilde{x}) ::= P$ $(P/\equiv,  , \text{nil})$ and $(P/\equiv, +, \text{nil})$ are commutative monoids $\nu z \nu w P \equiv \nu w \nu z P$ , $\nu z \text{nil} \equiv \text{nil}$ , $\nu y (P_1   P_2) \equiv P_1   \nu y P_2$ provided $y \notin \text{fn}(P_1)$ $(\text{start}(t), r)[P_0].P_1 \equiv (\text{start}(t'), r)[P'_0].P'_1$ provided that $P_0 \equiv P'_0$ and $P_1 \equiv P'_1\{t'/t\}$ $t[P_1] \equiv t'[P_2]$ provided that $P_1 \equiv P_2\{t'/t\}$
--

Table 1  
Laws for structural congruence.

definition  $A(\tilde{y})$  (where  $\tilde{y}$  denotes  $y_1, y_2, \dots, y_n$ ) is considered instead of replication, as in [17]. The constant term is equipped with a unique defining equation of the form  $A(\tilde{y}) ::= P$ , where the free names of  $P$  are a subset of  $\tilde{y}$ . The nil process is obtained putting  $m = 0$  in the summation. Summations are supposed to be in *head normal form*<sup>3</sup>.

Note that the standard definition of head normal processes must be extended to all the possible prefixes, since other prefixes beside the input and the output are present. The choice of head normal form is motivated by biological considerations, as each channel in the summation represents a different motif/site in the molecule. It is worth noting that differently from [17] here it is not considered a subset of names to describe homodimerization reactions (i.e. reactions between two reactants of the same kind). This point will be explained better when the reduction relation is given.

The process  $t[P]$  indicates the transaction of name  $t$  described by the *internal process*  $P$ . The other possible processes outside the transaction are called *external processes*.

### 2.3 Semantics

An operational reduction semantics that makes use of both a *structural congruence* and a *reduction relation* is given.

The *structural congruence* (indicated by  $\equiv$ ) is the smallest relations which satisfy the laws reported in table 1. In addition to the standard laws, there are the congruence laws for transactions and for the start action.

To better represent the semantics rules, a derived operator  $\diamond$  is introduced. The *multi process*  $n \diamond P$ , where  $n \in \mathbb{N}$  is the *multiplicity* of the process  $P$ , represents the parallel composition of  $n$  processes congruent to  $P$ . A definition of  $\diamond$ -standard form follows:

**Definition 2.1** A process  $P$  is in  $\diamond$ -standard form if  $P \equiv \nu \tilde{x} (\prod_{i=1}^n n_i \diamond P_i)$  where

- $P_i \text{not} \equiv P_j \ \forall i \neq j$

<sup>3</sup> A process  $P$  is in *extended head-normal form* if either it is nil or  $P \equiv \sum_{i=1}^m \pi_i.P_i$  and the following conditions hold,  $\forall i \neq j$ : (1) if  $\pi_i$  and  $\pi_j$  are input or output, then  $\text{sbj}(\pi_i) \neq \text{sbj}(\pi_j)$ , (2) if  $\pi_i = (\text{start}(t_i), r)[P_i]$  and  $\pi_j = (\text{start}(t_j), r)[P_j]$ , then  $P_i \text{not} \equiv P_j$  and  $t_i \neq t_j$ , (3) if  $\pi_i = \text{end}(t_i)$  and  $\pi_j = \text{end}(t_j)$ , then  $t_i \neq t_j$ .  $\text{sbj}(\pi_i)$  denotes the name of the channel and  $\text{not} \equiv$  means “not structural congruent”.

- $P_i \equiv t[Q]$  or  $P_i \equiv \sum_{j=1}^m \pi_j.Q_j$  with  $m \geq 1, \forall i = 1, \dots, n$ .

In the above definition,  $\nu\tilde{x}$  stays for  $\nu x_1 \nu x_2 \dots \nu x_n$ ,  $\prod_{i=1}^n Q_i$  is the parallel composition of  $n$  processes and  $\text{not} \equiv$  means “not structural congruent”. In particular, the process inside the square brackets in the start prefix is supposed to be in  $\diamond$ -standard form. We claim the following result that is proved by structural induction over processes:

**Proposition 2.2** *Every process  $P$  is structural congruent to a  $\diamond$ -standard form.*

An operator  $\triangleright^4$  may be introduced to put a process  $P$  into a  $\diamond$ -standard form  $Q$  s.t.  $P \equiv Q$ . In the following, some auxiliary functions and definitions are given.

**Labels** A set of labels  $\Theta$  must be added to the rules in order to collect the quantitative information. The label  $\theta$  is a 2-tuple of the form  $(l, a_r)$ . The first component (ranged over by  $l, l', \dots$ ) indicates the type of the action and belongs to the following set:

$$L = \{comm(x), t : start, t : comm(x), t : end\}$$

where  $x \in \mathcal{N}$  and  $t \in \mathcal{T}$ . The first label represents a communication between two processes outside a transaction along the channel  $x$ , while the others refer to actions involving a transaction of name  $t$ . They are the communication between two processes along  $x$  inside the transaction  $t$ , the start and the end of  $t$ . The second component  $a_r$  is a real number representing the actual rate of the associate transition. This kind of label gives explicitly the speed of the transition. In addition to these labels, another kind of label  $\vartheta \in \Theta'$  are used in the reduction rules. They are defined as a 3-tuple of the form  $(l, r, nl)$ . The first component is as defined above. The second element is the basal rate associated to the action. Finally, the element  $nl$  is a list, whose elements are couples  $(n_i, \kappa_i)$ . The component  $\kappa_i$  collects the multiplicity of a given reactant (*stoichiometry coefficient*) in a reaction and  $n_i$  is the number of such reactants in the system. The label  $\vartheta$  contains all the information to calculate the actual rate. Indeed, given  $nl$ , the possible combinations of all the reactants are calculated by using the function

---

<sup>4</sup>  $P \triangleright Q$  is defined as: (1)  $\nu\tilde{x} P_1 \mid P_2 \triangleright \text{nil} = P_2 \triangleright \nu\tilde{x} (1 \diamond P_1)$ , where  $P_1 = t[P_0]$  or  $P_1 = \sum_{i=1}^s \pi'_i.Q_s$ , (2)

$\nu\tilde{x} P_1 \mid P_2 \triangleright \nu\tilde{y} \prod_{i=1}^n n_i \diamond Q_i = \text{if } \exists j \text{ s.t. } \nu\tilde{x} P_1 \equiv \nu\tilde{x} Q_j \text{ then } P_2 \triangleright \nu\tilde{z} \prod_{i=1}^n \tilde{n}_i \diamond Q_i \text{ with } \tilde{n}_j = (n_j + 1) \text{ and}$

$\tilde{n}_i = n_i \forall i \neq j \text{ else } P_2 \triangleright \nu\tilde{z} (\prod_{i=1}^n n_i \diamond Q_i \mid 1 \diamond P'_1) \text{ where } P_1 \equiv \sum_{i=1}^s \pi'_i.P_i \text{ or } P_1 \equiv t[P_0] \text{ and } P'_1 \equiv P_1$  (3)

$\text{nil} \triangleright Q = Q$  It is supposed that  $P$  does not contain  $\diamond$ . The names  $\tilde{z}$  includes  $\tilde{y}$  and  $\tilde{x}'$ , where  $\tilde{x}'$  is obtained from  $\tilde{x}$  after opportune conversions over processes.

$h(nl)$ , from the list  $nl$  to  $\mathbb{N}$ :

$$h(nl) = \prod_{i=1}^n \binom{n_i}{\kappa_i}$$

where  $n \geq 1$  is the length of the list. The actual rate  $a_r$  is calculated as  $a_r = r_b \times h(nl)$ , where  $r_b$  is the basal rate of the reaction. We write  $\theta[i]$  to select the component  $i$ , with  $i \in [1, 2]$ , from the label  $\theta$ . Similarly for  $\vartheta$ .

**Sequence of transitions** The notation  $P_0 \xrightarrow{\sigma} P_n$  indicates a finite sequence of transitions  $P_0 \xrightarrow{\theta_1} P_1 \xrightarrow{\theta_2} P_2 \dots \xrightarrow{\theta_n} P_n$ , where  $\sigma = \theta_1 \theta_2 \dots \theta_n$ .

**Extension of  $In_x$  and  $Out_x$  functions** These two auxiliary functions count respectively the number of inputs and outputs on the channel  $x$  enabled in a process. The definition of  $In_x$  in [17] is extended with:

$$\begin{aligned} In_x(t[P]) &= In_x(P) & In_x(n \diamond P) &= n \times In_x(P) \\ In_x(start(t)[P_0].(P_1)) &= In_x(end(t).P_1) = 0 \end{aligned}$$

The definition of  $Out_x$  is obtained by replacing  $In_x$  with  $Out_x$ .

**Function to count processes ( $count_T$ ).** The auxiliary function  $count_T$  is defined in  $\mathcal{P} \times \mathcal{P} \rightarrow \mathbb{N}$  and it is introduced to count the number of instances of a summation  $P$  in a process  $Q$  up to structural congruence.

Given a process  $P \equiv \nu \tilde{x} \sum_{k=1}^m \pi_k.P_k$  the function  $count_T$  is defined as:

$$\begin{aligned} count_T(P, \text{nil}) &= count_T(P, \nu \tilde{y}(t[R])) = 0 \\ count_T(P, \nu \tilde{y}(\sum_{i=1}^s \pi'_i.Q_i)) &= \min_k \left\{ \sum_{i=1}^s f(\nu \tilde{x}(\pi_k.P_k), \nu \tilde{y}(\pi'_i.Q_i)) \right\} \\ count_T(P, \nu \tilde{y}(Q_1|Q_2)) &= count_T(P, \nu \tilde{y}(Q_1)) + count_T(P, \nu \tilde{y}(Q_2)) \end{aligned}$$

where  $f(R_1, R_2) = 1$  if  $R_1 \equiv R_2$ , 0 otherwise. For simplicity, the operator  $\diamond$  is not considered in the second process. Indeed, it may be replaced by using its definition in terms of the operator  $|$ .

**Function  $count_{end(t)}$**  It is used to count the number of  $end(t)$  prefixes in a given process. Its definition is defined inductively on the processes similarly to  $In_x$  and  $Out_x$ .

**Function  $act_T$**  This function gives the set of transaction names that are active in a given process. The definition is as in [6]:

$$act_T(P) = \{t[t[Q] \text{ is a subterm of } P]\}$$

In this work simple transactions are considered. It is supposed that after starting, transactions end with success and errors never occur. This allows us to avoid the definition of abort actions, of an abort axiom and of compensation mechanisms. To formalize these concepts, the definition of *well-defined* internal processes is introduced:

**Definition 2.3** An internal process  $P_0$  is *well-defined* if  $t[P_0]$  always reduces to the final process  $P_{fin}$  and contains only one end prefix that may end the transaction.

The hypothesis of only one end prefix that ends the transaction is added to simplify the semantics and has biological sense.

The *reduction relation* is the smallest relation over processes satisfying the rules and axioms given in table 2. The labeled transition system for the stochastic  $\pi$ -calculus with transactions is defined as  $\mathcal{LTS} = (\mathcal{P}, \Theta, \mapsto)$  where the auxiliary transition relation  $\rightarrow \subseteq \mathcal{P} \times \Theta' \times \mathcal{P}$  is used to define the transition relation  $\mapsto \subseteq \mathcal{P} \times \Theta \times \mathcal{P}$ .

The rules **comm** is as in [17]. It describes bimolecular reactions (i.e. two distinct molecules interact). The same rate is used in the input and output. Differently from [17], here it is not necessary to consider a specific axiom to describe the homodimerization reactions. The purpose of this further axiom is to describe the reaction involving two identical molecules by using a communication between two processes, not in head normal form, and to count the number of processes in the desired way. Here, we specify homodimerization reactions by using transactions. For instance, a reaction of kind  $2A \rightarrow B$  may be translated as:

$$\begin{aligned} T_R &= (start(t), r_b)[2 \diamond ((x_1(), \infty).nil + (x_2(), \infty).nil)].P'_1 \\ P'_1 &= (\overline{x_1}(), \infty).(\overline{x_2}(), \infty).end(t).(P_B|T_R) \\ P_A &= (x_1(), \infty).nil + (x_2(), \infty).nil + Q \end{aligned}$$

The axiom **tstart** describes the start of a transaction  $t$ . The internal process  $P_1$  is represented in the  $\diamond$ -standard form and the summation are in head normal form. The parameter  $\kappa_i$  is the stoichiometry of the reactant of kind  $i$  and the process  $P_{ij}$  represents the process describing that reactant. A transaction describes a reaction with  $N = \sum_i^n \kappa_i$  reactants, among which  $n$  are distinct. As for the quantitative information contained in the labels,  $r_b$  is the basal rate of the reaction and  $nl$  contains the number of processes of a given kind. The actual rate may be calculated by using the label information as  $r_b \times h(nl)$ . The transaction starts only if a process  $P_2$  representing the reactants is present in the system. At this point a new transaction is defined, with a fresh name and an internal process described in terms of  $P_0$  and of the processes  $P_{ij}$ . It is worth noting that the process describing the start is a sort of dummy process necessary to block the processes involved.

The axiom **tend** describes the end of a transaction. There is only one end prefix enabled inside the transaction with rate in  $\infty$ .

The rule **par** considers four different kinds of action. If  $l = comm(x)$  and  $l = (t : start)$ , the hypothesis  $Act_T(Q) = 0$  is added to ensure that no transactions are present in the process  $Q$ . This condition guarantees that the actions inside the transaction have priority over the external ones and so that some desired properties are satisfied (see section 3). Each of the subcases is described as follows.

- (i) If  $l = comm(x)$ , the communication among two different processes is described and we proceed as in the standard stochastic  $\pi$ -calculus.
- (ii) If  $l = t : comm(x)$  a communication among two processes inside a transaction  $t$

(comm)	$(x(w), r_b). P_1 \mid (\bar{x}\langle z \rangle, r_b). P_2 \dots \xrightarrow{(comm(x), r_b, [(1,1), (1,1)])} P_1 \{z/w\} \mid P_2 \dots$		
(tstart)	$P_2 \mid (start(t)[P_0], r_b). P_1 + P_3 \xrightarrow{(t': start, r_b, [(\kappa_1, \kappa_1), \dots, (\kappa_n, \kappa_n)])} t'[P_1 \{t'/t\} \mid P'_0]$ where $P_0 \equiv \nu \tilde{x} \prod_{i=1}^n (\kappa_i \diamond \sum_{j=1}^{\kappa_i} \pi_{ij}. P_{ij})$ is in $\diamond$ standard form, $P_2 \equiv \nu \tilde{y} \prod_{k=1}^N (P'_k + Q_k)$ , with $P_0 \equiv \nu \tilde{y} \prod_{k=1}^N P'_k$ , where $N = \sum_{i=1}^n \kappa_i$ and $P'_0 \equiv \nu \tilde{x} \prod_{i=1}^n \prod_{j=1}^{\kappa_i} P_{ij}$ , $t'$ is fresh		
(tend)	$t[(end(t), r_b). P_1 \mid P_2] \xrightarrow{(t: end, \infty, [(1,1)])} P_1$ provided that $count_{end(t)}(P_2) = 0$		
(par)	$\frac{P \xrightarrow{(l, r_b, nl)} P'}{P \mid Q \xrightarrow{(l, r_b, nl')} P' \mid Q}$ <b>CASE <math>l = \text{comm}(x)</math>:</b> $nl = [(n_1, 1), (n_2, 1)]$ , $nl' = [(n_1 + In_x(Q), 1), (n_2 + Out_x(Q), 1)]$ , provided that $act_T(Q) = \emptyset$ <b>CASE <math>l = (t: \text{comm}(x))</math>:</b> $nl' = nl$ <b>CASE <math>l = (t: \text{start})</math>:</b> $nl = [(n_1, \kappa_1), \dots, (n_n, \kappa_n)]$ and $nl' = [(n'_1, \kappa_1), \dots, (n'_n, \kappa_n)]$ with $n'_i = n_i + count_t(R_i, Q)$ , where $R_i \equiv \nu \tilde{x} \sum_{j=1}^{\kappa_i} \pi_{ij}. P_{ij}$ , provided that $act_T(Q) = \emptyset$ <b>CASE <math>l = (t: \text{end}(t))</math>:</b> $nl' = nl$		
(tred)	$\frac{P \xrightarrow{(comm(x), r_b, nl)} P'}{t[P] \xrightarrow{(t: comm(x), r_b, nl)} t[P']}$ provided that $r_b = \infty$		
(res)	$\frac{P \xrightarrow{(l, r_b, nl)} P'}{\nu w P \xrightarrow{(l, r_b, nl)} \nu w P'}$	(struct)	$\frac{P_1 \equiv P'_1 \quad P'_1 \xrightarrow{(l, r_b, (n_1, n_2))} P'_2 \quad P_2 \equiv P'_2}{P_1 \xrightarrow{(l, r_b, (n_1, n_2))} P_2}$
(fin)	$\frac{P \xrightarrow{(l, r_b, nl)} P'}{P \xrightarrow{(l, r_b \times h(nl))} P'}$		

Table 2  
Axioms and rules for the reduction relation for the stochastic  $\pi$ -calculus with biological transactions.

is given. The input/output prefixes outside the transaction are not considered, as the external actions are not enabled when a transaction is present. The rate is supposed to be  $\infty$ .

- (iii) If  $l = (t : start)$  the start of a transaction is described. It is necessary to count the processes congruent to the ones involved in the reaction and the function  $count_T$  is used at this purpose. The rate associate to the prefix coincides with the one of the whole reaction.



- (iv) if  $l = (t : \text{end})$ , the end of a transaction is given. Only one end action is supposed to be inside a transaction and the external ones are not considered, so  $n_1 = n_2 = 1$ . Besides, the rate is always  $\infty$ .

The rule **tred** say that if a communication is possible between two processes along the channel  $x$ , the same communication is possible if the two processes are inside the transaction. The rate is supposed to be  $\infty$ . The rule **res** and **struct** extend the standard rule with the possibility to deal with transaction. A generic kind of action is considered in the label.

The last rule **fin** describes how the system finally reduces at each step. The premise collects the quantitative information in its label  $\theta = (l, r_b, nl)$ . The conclusion describes the same relation, but its label gives explicitly the speed of the transition. This is calculated by the information given in the label of the premise.

### 3 Some properties

*Biological transactions* are simple transactions that are introduced to model complex reactions as atomic actions. They must satisfy some properties suitable for modeling biological reactions. For instance, after having started they end successfully in a finite number of steps and return the product process. Besides, it would be desirable that the transaction executes in isolation from the other processes. These concepts may be formalized in terms of the properties of *atomicity* and *serialiability*. The former may be expressed by “all or nothing”: a transaction ends successfully or does nothing. Serializability says that it is the same if transactions are executed in parallel or one after the other. Indeed, these properties are general criteria for proving the correctness of the transactions. Some results about these properties are reported. First of all, following [6], a *serialized transition processes* is defined as:

**Definition 3.1** The transaction sequence  $P \xrightarrow{\sigma} P'$ , with  $\text{act}T(P) = \text{act}T(P') = \emptyset$ , is serialized iff  $l_i = t : \text{comm}(x)$  or  $l_i = (t : \text{start})$  implies  $l_{i+1} = t : \text{comm}(x)$  or  $l_{i+1} = (t : \text{end})$  for  $i = 1, \dots, (n - 1)$ , where  $l_i = \theta_i[1]$ .

A property about transition sequences is described in the following proposition:

**Proposition 3.2** Given a process  $P$  such that  $\text{act}_t(P) = \emptyset$  and given the transaction sequence  $P \xrightarrow{\theta_1} P_1 \xrightarrow{\theta_2} P_2 \dots \xrightarrow{\theta_n} P_n$  with  $\theta_1[1] = \text{start}(t)$  and  $\theta_n[1] = \text{end}(t)$  then  $\text{act}_t(P_i) = \{t\}$  for  $i = 1, \dots, (n - 1)$  and  $\text{act}_t(P_n) = \emptyset$

This result follows directly from the semantics, in particular by the definition of the rule **par**. The next two propositions claim the atomicity and the serializability of the transactions:

**Proposition 3.3** Given a process  $P_0 = t[P]$  with  $P$  well-defined then there always exists a finite transaction sequence  $\sigma$  such that  $t[P_0] \parallel S \xrightarrow{\sigma} P_n \parallel S$ , where  $S$  is the rest of the system and  $P_n$  is the final process resulting from transaction. In the case  $P$  is univocally well defined there exists and it is unique the sequence  $\sigma$ .

**Proposition 3.4** *Given  $P \xrightarrow{\sigma} P'$  with  $\text{act}T(P) = \text{act}T(P') = \emptyset$  it is serialized.*

When a transaction starts, only the internal actions are enabled and it is executed in a serialized way. The condition  $\text{Act}_T(Q) = \emptyset$  in the rule **par** is fundamental to guarantee the properties above. If a different transaction system is defined as the previous one but with the rule **par** without the condition  $\text{Act}_T(Q) = \emptyset$ , the atomicity is not always guaranteed. Indeed, in this case an action among two standard processes is possible even after a transaction has started and so external actions may interleave the transaction ones. As a consequence, it may happen that a transaction starts but does not end in a finite number of steps.

## 4 Examples about biological reactions

In this section, two simple examples about the application of transactions to model reactions are presented.

### 4.1 The three-reactant reaction $R_1 + R_2 + R_3 \rightarrow P$

A generic reaction with three distinct reactants and one product is described in the stochastic  $\pi$ -calculus with biological transactions by the system  $S = P_T \mid P_1 \mid P_2 \mid P_3$ , where:

$$\begin{aligned} P_T &= (\text{start}(t), r_b)[1 \diamond P_1 \mid 1 \diamond P_2 \mid 1 \diamond P_3].P_0 \\ P_0 &= (\overline{x_1}\langle \rangle, \infty).(\overline{x_{12}}\langle \rangle, \infty).(\overline{x_{123}}\langle \rangle, \infty).\text{end}(t).(P_{Pr} \mid P_T) \\ P_1 &= (x_1(), \infty).\text{nil} \quad P_2 = (x_{12}(), \infty).\text{nil} \quad P_3 = (x_{123}(), \infty).\text{nil} \end{aligned}$$

The process  $P_T$  describes the transaction,  $P_i$  is the process representing the reactant  $R_i$  for  $i = 1, 2, 3$  and  $P_{Pr}$  the product  $P$ . In this case  $n = N = 3$  and  $\kappa_1 = \kappa_2 = \kappa_3 = 1$ . The rate  $r_b$  represents the basal rate of the reaction and is associated to the start of the transaction. A possible reduction of the system  $S$  is:

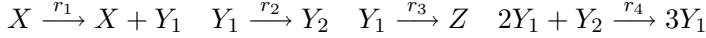
$$\begin{aligned} S &\xrightarrow{(t':\text{start}, r_b \times h(nl'))} t'[(\overline{x_1}\langle \rangle, \infty).P'_0 \mid (x_1(), \infty).\text{nil} \mid P_2 \mid P_3] \equiv S_1 \\ &\xrightarrow{(t':\text{comm}(x_1), \infty \times 1)} t'[(\overline{x_{12}}\langle \rangle, \infty).P''_0 \mid \text{nil} \mid (x_{12}(), \infty).\text{nil} \mid P_3] \equiv S_2 \\ &\xrightarrow{(t':\text{comm}(x_{12}), \infty \times 1)} t'[(\overline{x_{123}}\langle \rangle, \infty).\text{end}(t').(P_{Pr} \mid P_T) \mid \text{nil} \mid \text{nil} \mid P_3] \equiv S_3 \\ &\xrightarrow{(t':\text{comm}(x_{123}), \infty \times 1)} t'[\text{end}(t').(P_{Pr} \mid P_T) \mid \text{nil} \mid \text{nil} \mid \text{nil}] \equiv S_4 \\ &\xrightarrow{(t':\text{end}, \infty \times 1)} P_{Pr} \mid P_T \mid S' \mid \text{nil} \mid \text{nil} \mid \text{nil} \equiv P_{Pr} \mid P_T \mid S' \equiv S_5 \end{aligned}$$

where  $P'_0$  and  $P''_0$  stays for the remaining part of  $P_0$ . In the first label,  $nl'$  stays for  $[(1 + \text{count}_T(P_1, Q), 1), (1 + \text{count}_T(P_2, Q), 1), (1 + \text{count}_T(P_3, Q), 1)]$ , where  $Q = P_1 \mid P_2 \mid P_3$ . The premises of the rule **fin** applied as last steps of the derivations are respectively:

$$\begin{aligned} S &\xrightarrow{(t':\text{start}, r_b, nl')} S_1, S_1 \xrightarrow{(t':\text{comm}(x_1), \infty, [(1,1), (1,1)])} S_2, S_2 \xrightarrow{(t':\text{comm}(x_{12}), \infty, [(1,1), (1,1)])} S_3, \\ &\quad S_3 \xrightarrow{(t':\text{comm}(x_{123}), \infty, [(1,1), (1,1)])} S_4, S_4 \xrightarrow{(t':\text{end}, \infty, [(1,1)])} S_5 \end{aligned}$$

## 4.2 The Brussellator model

The *Brussellator model* is a well-known model of chemical reactions with oscillations [21]. Stochastic simulations of the model are presented in [13]. The reactions that compose the model are the following:



where  $X$ ,  $Z$ ,  $Y_1$  and  $Y_2$  represent the species involved and  $r_1$ ,  $r_2$ ,  $r_3$ ,  $r_4$  are the basal rates of the corresponding reactions. This model is represented by the system  $S = P_T | P_{Y_1} | P_{Y_1} | P_X | P_{Y_2} | P_E$ , where we have:

$$\begin{aligned} P_T &= (start(t), r_4)[2 \diamond (P_1 + P_2)][1 \diamond P_{Y_2}].P_0 \\ P_0 &= (\overline{x_6}\langle, \infty).(\overline{x_4}\langle, \infty).(\overline{x_5}\langle, \infty).(end(t), \infty).(P_{Y_1} | P_{Y_1} | P_{Y_1} | P_T) \\ P_X &= (\overline{x_1}\langle, r_1).(P_X + P_{Y_1}) \\ P_E &= (x_1(), r_1).P_E + (x_2(), r_2).P_E + (x_3(), r_3).P_E \\ P_{Y_1} &= (\overline{x_2}\langle, r_2).P_{Y_2} + (\overline{x_3}\langle, r_3).P_Z + P_1 + P_2 \\ P_1 &= (x_4(), \infty).nil \quad P_2 = (x_5(), \infty).nil \quad P_{Y_2} = (x_6(), \infty).nil \end{aligned}$$

The process  $P_T$  describes the transaction,  $P_E$  is a dummy process introduced to model unimolecular reactions,  $P_X$ ,  $P_{Y_1}$ ,  $P_{Y_2}$  and  $P_Z$  represent the processes involved in it in the transaction. The transaction is characterized by  $N = 3$ ,  $n = 2$ ,  $\kappa_1 = 2$  and  $\kappa_2 = 1$ . A possible reduction of  $S$  is given by:

$$\begin{aligned} S &\xrightarrow{(t':start, r_4 \times h(nl'))} t'[(\overline{x_6}\langle, \infty).P'_0 \mid (x_6(), \infty).nil \mid P_1 \mid P_2] \equiv S_1 \\ &\xrightarrow{(t':comm(x_6), \infty \times 1)} t'[(\overline{x_4}\langle, \infty).P''_0 \mid nil \mid (x_4(), \infty).nil \mid P_2] \equiv S_2 \\ &\xrightarrow{(t':comm(x_4), \infty \times 1)} t'[(\overline{x_5}\langle, \infty).end(t').(P_{Y_1} | P_{Y_1} | P_{Y_1} | P_T) \mid nil \mid (x_5(), \infty).nil \mid \\ &\quad nil] \equiv S_3 \\ &\xrightarrow{(t':comm(x_5), \infty \times 1)} t'[end(t').(P_{Y_1} | P_{Y_1} | P_{Y_1} | P_T) \mid nil \mid nil \mid nil] \equiv S_4 \\ &\xrightarrow{(t':end, \infty \times 1)} (P_{Y_1} | P_{Y_1} | P_{Y_1} | P_T) \equiv S_5 \end{aligned}$$

with  $nl' = [(n'_1, 2), (n'_2, 1)]$ ,  $n'_1 = count_T(P_1 + P_2, Q) + 1$ ,  $n'_2 = count_T(P_{Y_2}, Q) + 1$  and  $Q = P_X | P_{Y_1} | P_{Y_1} | P_{Y_2} | P_E$ . The premises of the rule *fin* applied as last steps of the derivations are respectively:

$$\begin{aligned} S &\xrightarrow{(t':start, r_4, nl')} S_1, S_1 \xrightarrow{(t':comm(x_6), \infty, [(1,1), (1,1)])} S_2, S_2 \xrightarrow{(t':comm(x_4), \infty, [(1,1), (1,1)])} S_3 \\ S_3 &\xrightarrow{(t':comm(x_5), \infty, [(1,1), (1,1)])} S_4, S_4 \xrightarrow{(t':end, \infty, (1,1))} S_5 \end{aligned}$$

## 5 Conclusion

We enrich the stochastic  $\pi$ -calculus with biological transactions. This allows us to model multi-reactant multi-product reactions as atomic actions avoiding some problems that we meet if the stochastic  $\pi$ -calculus with binary communication is used. Other formalisms [12,1] permit the representation of these complex reactions

and there exist encoding of these languages into  $\pi$ -calculus or other process algebras. In this work we extend the stochastic version of  $\pi$ -calculus directly by adding transactions to represent n-way synchronization interactions. A crucial point in the definition of the semantics is the extension of the standard Gillespie approach to consider complex reactions with more than two reactants. The key axiom is the one describing the start of the transaction. The actual rate of the start action is given by the basal rate times the number of possible combinations of all the reactants involved in the reaction. The basal rate assigned to the start action is the rate of the reaction. All the other actions involved in the transaction follow as immediate actions. A function  $count_T$  has been defined in order to count the number of processes congruent to the the ones describing reactants. Finally, it must be observed that the simulation algorithm of the standard stochastic  $\pi$ -calculus has to be modified in order to consider transactions.

## References

- [1] Bistarelli, S., I. Cervesato, G. Lenzini, R. Marangoni and F. Martinelli, *On Representing Biological Systems through Multiset Rewriting*, Proc. of Workshop on Computational Methods in Biomathematics (CMB'03), LNCS, Springer-Verlag, **2809** (2003).
- [2] Bistarelli, S., I. Cervesato, G. Lenzini and F. Martinelli, *Relating Multiset Rewriting and Process Algebras for Security Protocol Analysis*, Journal of Computer Security, **13** (2005).
- [3] Blosssey, R., C. Kuttler and J. Niehren, *Gene regulation in the  $\pi$ -calculus: simulating cooperativity at the lambda switch*, Proc. of BioConcur 2004, 2004.
- [4] Butler, M., T. Hoare and C. Ferreira, *A trace semantics for long-running transactions*, Proc. of 25 year of CSP, LNCS, Springer-Verlag, **3525** (2005), 133–150.
- [5] Bocchi, L., C. Laneve and G. Zavattaro, *A Calculus for Long-running Transactions*, Proc. of the 6th IFIP International Conference on Formal Methods for Open Object-based Distributed Systems (FMOODS'03), LNCS, Springer-Verlag, **2884** (2003).
- [6] Busi, N. and G. Zavattaro, *On the Serializability of Transactions in JavaSpaces*, Proc. of CONCOORD'01, LNCS, Springer-Verlag, **54** (2001).
- [7] Cardelli, L., E. M. Panina, A. Regev, E. Shapiro and W. Silverman, *BioAmbients: An Abstraction for Biological Compartments*, Theoretical Computer Science, Elsevier, **235** (2004), 141–167.
- [8] Cho, K.H., W. Kolch, M. Ullah and O. Wolkenhauer, *Modelling and Simulation of IntraCellular Dynamics: Choosing an Appropriate Framework*, IEEE Transactions on NanoBioScience, **3** (2004), 200–207.
- [9] Ciocchetta, F. and C. Priami, “The stochastic  $\pi$ -calculus with biological transactions”, Technical Report **TR-02-2006**, The Microsoft Research-University of Trento Centre for Computational and Systems Biology, 2006.
- [10] Costantin, G., C. Laudanna, P. Lecca, C. Priami, P. Quaglia and B. Rossi, *Language modeling and simulation of autoreactive lymphocytes recruitment in inflamed brain vessels*. SIMULATION: Transactions of The Society for Modeling and Simulation International, **80** (2003), 273–288.
- [11] Danos, V. and J. Krivine, *Formal molecular biology done in CCS-R*, Proc. of Workshop on Concurrent Models in MolecularBiology (BioConcur '03), 2003.
- [12] Danos, V. and C. Laneve, *Core Formal Molecular Biology*, Proc. of ESOP'03, LNCS, Springer-Verlag, **2618** (2003), 302–318.
- [13] Gillespie, D.T. *Exact stochastic simulation of coupled chemical reactions*, Journal of Physical Chemistry **81** (1977), 2340–2361.
- [14] Kitano, H., *Systems Biology: A Brief Overview*, Science, **295** (2002), 1662–1664.

- [15] Laneve, C. and G. Zavattaro, *Foundations of web transactions*, Proc. of Foundations of Software Science and Computation Structure (FOSSACS 2005), LNCS, Springer-Verlag, **3441** (2005), 282–298.
- [16] Milner, R., “Communicating and mobile systems: the  $\pi$ -calculus”, Cambridge University Press, 1999.
- [17] Priami, C., A. Regev, W. Silverman and E. Shapiro, *Application of a stochastic name-passing calculus to representation and simulation of molecular processes*, Information Processing Letters, **80** (2001), 25–31.
- [18] Priami, C. and P. Quaglia, *Beta-binders for biological interactions*, Proc. of Computational Methods in Systems Biology '04 (CMSB04), 2004.
- [19] Regev, A., *Representation and simulation of molecular pathways in the stochastic  $\pi$ -calculus*, Proc. of the 2nd workshop on Computation of Biochemical Pathways and Genetic Networks, 2001.
- [20] Sangiorgi, D. and D. Walker, “The  $\pi$ -calculus: a Theory of Mobile Processes”, Cambridge University Press, 2001.
- [21] Tyson, J. *Some Further Studies of Nonlinear Oscillators in Chemical Systems*, J. Chem. Phys., **58** (1973), 3919–3930.