Technical Report CoSBi 10/2006

Beta-binders with Biological Transactions

Federica Ciocchetta
University of Trento
fedecioc@dit.unitn.it

Corrado Priami
The Microsoft Research - University of Trento Centre for Computational and Systems Biology
priami@cosbi.eu

Beta-binders with Biological Transactions

Federica Ciocchetta
University of Trento
fedecioc@dit.unitn.it

Corrado Priami
The Microsoft Research - University of Trento
Centre for Computational and Systems Biology
priami@cosbi.eu

Abstract

In this work we propose an extension of Beta-binders with *biological transactions*, called *TBeta-binders*, in order to model a sequence of elementary actions atomically. This extension is useful when we need to specify multi-reactant multi-product reactions or when we use a sequence of actions to represent a single biological interaction. Some properties of these transactions are reported. Finally, some simple but explicative examples are described to validate our extension.

1 Introduction

In the last years process algebras have been used to produce biological models [23, 21, 10, 8]. These techniques have been originally defined in computer science for the analysis of complex concurrent systems and they seem to be appropriate to represent formally and to analyze biological systems as well. One example is the π-calculus with its stochastic version [14, 18, 17]. Recently, there have been some efforts to define specific calculi for biology [22, 20, 19]. Among them, Beta-binders [20] are an extension of the π-calculus inspired to biological phenomena. This calculus is based on the definition of *bio-process*, a box with some sites (*beta-binders*) to express the interaction capabilities of the element, in which π-calculus-like processes (*pi-process*) are encapsulated. Beta-binders enrich the standard π-calculus with some interesting features, as the join between two bio-processes, the split of one bio-process into two ones, the change of the bio-process interface by hiding, unhiding and exposing a site. Moreover, it supports the promiscuity of communication between two bio-processes, with the interaction compatibility expressed by the types of the interaction sites.

A critical task in the translation of biological models into Beta-binders (and generally into other process algebras) is the specification of multi-reactant multi-product reactions. These reactions are rare in nature but they are quite frequent in biological models as abstractions of complex situations whose details are unknown or not of interest. Since actions in Beta-binders involve at most two processes,

a possible translation of multiple-reactant multiple-product reactions is to decompose them into a sequence of one-reactant or two-reactant reactions. This approach has some drawbacks. Firstly, the system may block at intermediate steps leading to a deadlock. Secondly, we have to select an order of elementary steps, among the $\frac{n!}{2}$ available (where n is the number of reactants). Finally, there is the problem to find appropriate rates for each step in which the reaction is decomposed.

In this paper we introduce *transactions* to model sequences of actions atomically and to overcome the drawbacks highlighted above. Transactions are mechanisms originally used in web services and databases to execute distributed computations as they were a single atomic action. Recently there have been different attempts to model web-service transactions by using process algebras [2, 5, 15, 16, 3, 6, 7]. In [2] πt-calculus is presented: it is an extended version of asynchronous π-calculus to deal with long time transactions and offers failure handlers when interruptions are met. Another extension of asynchronous π-calculus with long-time transactions, called *web-*π, is introduced in [15]. In this case the main aspects are the interruptible processes, the failure handlers and the concept of time. A web-π transaction may terminate successfully or may fail, either as an error occurs or the time deadline is reached. CSP is the process algebras adopted in [7] to model long-running transactions with traces. The authors of [6] introduced a new calculus, *StAC* (structural Activity Compensation), inspired to both CCS and CSP in order to model Long-running Business Transactions. It gives a precise interpretation of compensation, including the combination of compensation with parallel execution, hierarchy and exceptions. Finally, in [5] a formal study of the serializability of transactions in JavaSpaces is proposed. At this purpose the authors abstract from away from the concrete language and embed the primitives in a process calculus.

The π-calculus has been enriched with *biological transactions* to model complex reactions [9]. Biological transactions are simple transactions that have to satisfy some basic properties suitable for modeling biology. These properties are formalized by using the concepts of *atomicity* and *serialisability (isolation)*. Atomicity is summarized as ”all or nothing”: either a transaction is executed and finally commits or it does nothing. Serializability expresses the fact that different activities have the same effect whether they are executed in sequence or in parallel.

In this work we focus on Beta-binders [20] and we enrich the standard version of the calculus with *biological transactions*. We call it *TBeta-binders*. Differently from other calculi enriched with transactions, we do not consider neither compensation and rollback mechanisms nor nested transactions nor time. Indeed these features are not necessary to describe biological reactions. The approach followed in similar to [9], but in this case transactions are extended to bio-processes too.

The paper is organized in the following way. In section 2 an introduction to Beta-binders and to the stochastic version is reported. The section 3 presents the problems related to complex reactions. In section 4 the extended calculus is described in detail. Some properties are introduced in the following section. In section 6 the stochastic version of TBeta-binders is presented. Some examples are shown in section 7. Finally, the last section reports discussion and some final remarks.

2 Beta-binders

This section presents the syntax and the semantics of Beta-binders, as reported in [20]. The definition of Beta-binders is based on the concept of *bio-processes*. A bio-process is a box with interacting capabilities through apposite sites (*beta-binders*) and where inside π-calculus processes (*pi-processes*) are encapsulated. The syntax of the π-calculus is enriched to manipulate boxes in the following way:

$$\begin{array}{lcl} P & ::= & \mathsf{nil} \mid \pi.P \mid P \mid P \mid \nu y\, P \mid !\,\pi.P \\ \pi & ::= & x(y) \mid \overline{x}\langle z\rangle \mid \mathsf{expose}(x,\Gamma) \mid \mathsf{hide}(x) \mid \mathsf{unhide}(x) \end{array}$$

where we assume a countable infinite set of names $\mathcal{N}$(ranged over lower-case letters). The processes generated by the grammar above differ from the π-calculus processes in expose, hide, and unhide prefixes. These are intended for changing the external interfaces of boxes by adding a new site, hiding a site, and unhidding a site which has been previously hidden, respectively. All other processes have the same meaning as in the π-calculus.

The π-calculus definitions of *name substitution* and of *free* and *bound names* (denoted by fn(-) and bn(-), respectively) are extended to the processes generated by the above syntax in the obvious way, assuming expose(x, Γ) as a binder for x.

An *elementary beta binder* has either the form $\beta(x : \Gamma)$ or the form $\beta^h(x : \Gamma)$. The name x is the subject of the beta binder, and Γ is the type of x, a non-empty set of names such that $x \notin \Gamma$. *Composite beta binders* are generated by the following grammar:

$$\mathbf{B} \quad ::= \quad \beta(x:\Gamma) \mid \beta^h(x:\Gamma) \mid \beta(x:\Gamma)\,\mathbf{B} \mid \beta^h(x:\Gamma)\,\mathbf{B}$$

A composite beta binder is said to be well-formed when the subjects of its elementary components are all distinct. We let well-formed beta binders be ranged over by $\mathbf{B}, \mathbf{B}_1, \mathbf{B}_2, \ldots, \mathbf{B}', \ldots$. The set of the subjects of all the elementary beta binders in $\mathbf{B}$ is denoted by sub($\mathbf{B}$). Finally, the metavariables $\mathbf{B}^*, \mathbf{B}_1^*, \mathbf{B}_2^*, \ldots$ stay for either a well-formed beta binder or the empty string.

Formally, *Bio-processes* (ranged over by $B, B_1, \ldots, B', \ldots$) are generated by the following grammar:

$$B \quad ::= \quad \mathsf{Nil} \mid \mathbf{B}[P] \mid B \parallel B$$

The system may be either empty (Nil) or composed by parallel composition of bio-processes, written $B \parallel B$.

Bio-processes are given an operational reduction semantics that makes use of both a structural congruence over bio-processes and a structural congruence over pi-processes. We overload the same symbol $\equiv$, to denote both congruences, and let the context disambiguate the intended relation. The structural congruence is the smallest relation satisfying the rules in Table 1. The *reduction transition system* is $TSR(\mathcal{B}, \longrightarrow)$, where $\mathcal{B}$ is the set of states (equivalence classes of bio-processes w.r.t. $\equiv$) and the reduction relation $\longrightarrow$ is the smallest relation over bio-processes obtained by applying the axioms and rules in Table 2. The rule intra lifts to the level of bio-processes the internal pi-processes interactions without changing the interface. The rule inter models interactions between boxes through complementary actions (input/output) over complementary sites (sites with non-disjoint types).

$P_1 \equiv P_2$ if P_1 and P_2 are α-equivalent	$\mathbf{B}[P_1] \equiv \mathbf{B}[P_2]$ if $P_1 \equiv P_2$
$(\mathcal{P}/_{\equiv}, \mid, \mathsf{nil})$ is a commutative monoid	$(\mathcal{B}/_{\equiv}, \|, \mathsf{Nil})$ is a commutative monoid
$\nu z\, \nu w\, P \equiv \nu w\, \nu z\, P$	$\mathbf{B}_1\mathbf{B}_2[P] \equiv \mathbf{B}_2\mathbf{B}_1[P]$
$\nu z\ \mathsf{nil} \equiv \mathsf{nil}$	$\mathbf{B}^*\hat{\beta}(x:\Gamma)[P] \equiv \mathbf{B}^*\hat{\beta}(y:\Gamma)[P\{y/x\}]$
$\nu y\,(P_1 \mid P_2) \equiv P_1 \mid \nu y\, P_2$ if $y \notin \mathsf{fn}(P_1)$	if y fresh in P and $y \notin \mathsf{sub}(\mathbf{B}^*)$
$!\,\pi.\,P \equiv \pi.\,(P \mid !\,\pi.\,P)$	

Table 1: Laws for structural congruence for Beta-binders.

The rule expose causes the addition of an extra site with the declared type, while the rules hide and unhide force the specified site to become hidden and unhidden, respectively. The axiom join models the merge of boxes. It can model different interactions depending on the definition of the function f_{join}. In a similar way, the axiom split is used to model a split of a bio-process into two bio-processes and it is described by the function f_{split}. The rules redex and struct are typical rules of reduction semantics. They are meant, respectively, to interpret the reduction of a subcomponent as a reduction of the global system, and to infer a reduction after a proper structural shuffling of the process at hand.

2.1 Stochastic Beta-binders

An extension of Beta-binders has been defined to cope with quantitative description of biological data [11]. The main difference with respect to the original definition of Beta-binders is the addition of information on the rates and number of molecules. The reference simulation algorithm is Gillespie [12].

The only difference with respect to the terms described in the previous section is that the prefix $\pi.P$ is replaced by $(\pi, r).P$. The rate r is either a positive real number or ∞. In the former case the rate corresponds to the parameter of an exponential distribution that drives the stochastic behaviour of the action π. In this last case the associated action is immediate. In order to define the semantics, we introduce the following definitions.

- The transitions are enriched with a label $\theta \in \Theta$ collecting the quantitative information. The label θ is a 4-tuple $(k; B; c; (n_1, n_2))$, where k describes the kind of the reaction, B collects the bio-processes involved, c stores the reaction rates and finally the pair (n_1, n_2) records the number of reactants involved. The values of k vary in the set:

  $$\mathcal{L} = \mathcal{L}' \cup \mathcal{L}''$$

  $\mathcal{L}'$ is $\{i, h, u, e, I, J, S\}$, where i stays for intra-communication, h for hide, u for unhidden, e for expose, I and J are inter-communication and join between two different (up to $\equiv$) bio-processes, respectively and S is the split. The set $\mathcal{L}'' = \{I_h, J_h\}$ contains the labels for the intercommunication and the join between two equal (up to $\equiv$) bio-processes. The label θ says that a reaction of kind k involving the bio-process B is taking place in isolation with rate $c \times n_1 \times n_2$. All these information together with the number of reactants that can take part in it are necessary to calculate the actual rate of the reaction.

(intra) $$\frac{P \equiv \nu\tilde{u}\,(x(w).\,P_1 \mid \overline{x}\langle z\rangle.\,P_2 \mid P_3)}{\mathbf{B}[P] \longrightarrow \mathbf{B}[\nu\tilde{u}\,(P_1\{z/w\} \mid P_2 \mid P_3)]}$$

(inter) $$\frac{P \equiv \nu\tilde{u}\,(x(w).\,P_1 \mid P_2) \qquad Q \equiv \nu\tilde{v}\,(\overline{y}\langle z\rangle.\,Q_1 \mid Q_2)}{\beta(x:\Gamma)\,\mathbf{B}_1^*[P] \parallel \beta(y:\Delta)\,\mathbf{B}_2^*[Q] \longrightarrow \beta(x:\Gamma)\,\mathbf{B}_1^*[\nu\tilde{u}\,(P_1\{z/w\} \mid P_2)] \parallel \beta(y:\Delta)\,\mathbf{B}_2^*[\nu\tilde{v}\,(Q_1 \mid Q_2)]}$$

provided $\Gamma \cap \Delta \neq \emptyset$ and $x, z \notin \tilde{u}$ and $y, z \notin \tilde{v}$

(expose) $$\frac{P \equiv \nu\tilde{u}\,(\mathsf{expose}(x,\,\Gamma)\,.\,P_1 \mid P_2)}{\mathbf{B}[P] \longrightarrow \mathbf{B}\,\beta(y:\Gamma)\,[\nu\tilde{u}\,(P_1\{y/x\} \mid P_2)]}$$ provided $y \notin \tilde{u}$, $y \notin \mathsf{sub}(\mathbf{B})$ and $y \notin \Gamma$

(hide) $$\frac{P \equiv \nu\tilde{u}\,(\mathsf{hide}(x)\,.\,P_1 \mid P_2)}{\beta(x:\Gamma)\,\mathbf{B}^*[P] \longrightarrow \beta^h(x:\Gamma)\,\mathbf{B}^*[\nu\tilde{u}\,(P_1 \mid P_2)]}$$ provided $x \notin \tilde{u}$

(unhide) $$\frac{P \equiv \nu\tilde{u}\,(\mathsf{unhide}(x)\,.\,P_1 \mid P_2)}{\beta^h(x:\Gamma)\,\mathbf{B}^*[P] \longrightarrow \beta(x:\Gamma)\,\mathbf{B}^*[\nu\tilde{u}\,(P_1 \mid P_2)]}$$ provided $x \notin \tilde{u}$

(join) $$\mathbf{B}_1[P_1] \parallel \mathbf{B}_2[P_2] \longrightarrow \mathbf{B}[P_1\sigma_1 \mid P_2\sigma_2]$$

provided that $f_{join}(\mathbf{B}_1, \mathbf{B}_2, P_1, P_2) = (\mathbf{B}, \sigma_1, \sigma_2)$

(split) $$\mathbf{B}[P_1 \mid P_2] \longrightarrow \mathbf{B}_1[P_1\sigma_1] \parallel \mathbf{B}_2[P_2\sigma_2]$$

provided that $f_{split}(\mathbf{B}, P_1, P_2) = (\mathbf{B}_1, \mathbf{B}_2, \sigma_1, \sigma_2)$

(redex) $$\frac{B \longrightarrow B'}{B \parallel B'' \longrightarrow B' \parallel B''}$$ (struct) $$\frac{B_1 \equiv B_1' \qquad B_1' \longrightarrow B_2}{B_1 \longrightarrow B_2}$$

Table 2: Axioms and rules for the reduction relation for Beta-binders.

$\mathsf{In}_x(\mathsf{nil}) = 0 \quad \mathsf{In}_x(P_1 \vert P_2) = \mathsf{In}_x(P_1) + \mathsf{In}_x(P_2) \quad \mathsf{In}_x(!\,\pi.P) = \mathsf{In}_x(\pi.P)$ $\mathsf{In}_x((\pi, r).P) = 1$ if $ch(\pi) = x$, 0 otherwise $\quad \mathsf{In}_x(\nu y.P) = \mathsf{In}_x(P)$ if $x \neq y$, 0 otherwise
$\mathsf{Out}_x(\mathsf{nil}) = 0 \quad \mathsf{Out}_x(P_1 \vert P_2) = \mathsf{Out}_x(P_1) + \mathsf{Out}_x(P_2) \quad \mathsf{Out}_x(!\,\pi.P) = \mathsf{Out}_x(\pi.P)$ $\mathsf{Out}_x((\pi, r).P) = 1$ if $ch(\pi) = \overline{x}$, 0 otherwise $\quad \mathsf{Out}_x(\nu y.P) = \mathsf{Out}_x(P)$ if $x \neq y$, 0 otherwise
$\mathsf{G}_x(\mathsf{nil}) = 0 \quad \mathsf{G}_x(P_1 \vert P_2) = \mathsf{G}_x(P_1) + \mathsf{G}_x(P_2) \quad \mathsf{G}_x(!\,\pi.P) = \mathsf{G}_x(\pi.P)$ $\mathsf{G}_x((\pi, r).P) = 1$ if $\pi = \mathsf{g}(x)$, 0 otherwise $\quad \mathsf{G}_x(\nu y.P) = \mathsf{G}_x(P)$ if $x \neq y$, 0 otherwise where $(\mathsf{G}, \mathsf{g}) \in \{(\mathsf{Hide}, \mathsf{hide}), (\mathsf{Unhide}, \mathsf{unhide})\}$
$\mathsf{Num}(\mathbf{B}[P], \mathsf{Nil}) = 0$ $\mathsf{Num}(\mathbf{B}[P], \mathbf{B}'[P'] \Vert B) = 1 + \mathsf{Num}(\mathbf{B}[P] \Vert B)$ if $\mathbf{B}[P] \equiv \mathbf{B}'[P']$ $\mathsf{Num}(\mathbf{B}[P], \mathbf{B}'[P'] \Vert B) = \mathsf{Num}(\mathbf{B}[P] \Vert B)$ if $\mathbf{B}[P] \not\equiv \mathbf{B}'[P']$
$\mathsf{Count}_t(\mathsf{B}, \mathsf{B}') =$ case t in $\{i, h, u, e, S\}$: $(\mathsf{Num}(B, B'), 0)$ $\{J, I\}$: if $B \equiv \mathbf{B}_1[P_1] \Vert \mathbf{B}_2[P_2]$ then $(\mathsf{Num}(\mathbf{B}_1[P_1], B'), \mathsf{Num}(\mathbf{B}_2[P_2], B'))$ $\{J_h, I_h\}$: if $B \equiv \mathbf{B}[P] \Vert \mathbf{B}[P]$ then $(\mathsf{Num}(\mathbf{B}[P], B'), \mathsf{Num}(\mathbf{B}[P], B'))$

Table 3: Auxiliary functions used in the stochastic reduction relation.

- The labelled transition system for the stochastic extension of Beta-binders is $(\mathcal{B}, \Theta, \longrightarrow_s)$, where $\rightarrow_s \subseteq \mathcal{B} \times \Theta \times \mathcal{B}$ is the transition relation defined in Tab. 4. It makes use of the auxiliary functions defined in 3. The symbol $\not\equiv$ stays for not-congruent.

A brief description of each axiom/rule in Table 4 follows. The rule intra is a reduction that changes the internal structure of the box, i.e. the pi-processes inside. The label $(i; B; r_x \times n_I \times n_O; (1, 1))$ says that an intra-communication is happening inside B. The rate is given by the product of the basal rate r_x for the n_I, the number of firable inputs on x in P, and n_O, the number of firable outputs on x in P. The values n_I and n_O are expressed in terms of the functions $\mathsf{In}_x(P)$ and $\mathsf{Out}_x(P)$ defined in Table 3. The couple $(1, 1)$ needs to calculate the possible combination of reactants and the actual rate.

The rules hide, unhide, expose are similar to intra, but in this case there is the change of the interface too. The function Hide_x and Unhide_x are defined in table 3 and are used to count the number of firable hide and unhide prefixes on the binder x in P. In the case of the expose, the new binder has a new name and represents a new reaction. So there is no function to count the possible expose of a binder. The rules inter and $\mathsf{inter}_\mathsf{h}$. These two axioms represent a communication between two boxes that are different (*bimolecular reaction*) or equal up to $\equiv$ (*homodimerization reaction*), respectively. The communication is possible over *compatible* sites, expressed on the *affinity* of the types of the sites involved. The function $\alpha(\Gamma, \Delta)$ maps the two types into a real value expressing the strength of the interaction. The value of c is computed considering the involved interaction sites. The side conditions (C_1) and (C_2) are:

(C1) $\mathbf{B}_1 = \mathbf{B}_1^* \beta(x : \Gamma)$ and $\mathbf{B}_2 = \mathbf{B}_2^* \beta(y : \Delta)$ and $x, z \notin \tilde{u}$ and $y, z \notin \tilde{v}$ and $P' \equiv \nu\tilde{u}(P_1\{z/w\} \vert P_2)$ and $Q' \equiv \nu\tilde{v}(Q_1 \vert Q_2)$ and $\mathbf{B}_1[P_1] \not\equiv \mathbf{B}_2[P_2]$ and $\alpha(\Gamma, \Delta) > 0$

(C2) $\alpha(\Gamma, \Delta) > 0$ and $(\mathbf{B} = \mathbf{B}^* \beta(x : \Gamma)\beta(y : \Delta))$ and $Q_1 \equiv \nu\tilde{u}(P_1\{z/w\} \vert (\overline{y}\langle z \rangle, r_y).P_2 \vert P_3)$ and $Q_2 \equiv \nu\tilde{u}((x(w), r_x).P_1\{z/w\} \vert P_2 \vert P_3)$ and $x, z \notin \tilde{u}$ and $y, z \notin \tilde{v}$

(intra) $$\frac{P \equiv \nu\tilde{u}\,((x(w), r_x).\, P_1 \mid (\overline{x}\langle z\rangle, r_x).\, P_2 \mid P_3)}{B \equiv \mathbf{B}[P] \xrightarrow{i;B;r_x\times n_I\times n_O;(1,1)}_s \mathbf{B}[\nu\tilde{u}\,(P_1\{z/w\} \mid P_2 \mid P_3)]}$$
where $n_I = 1 + \mathsf{In}_x(P_3)$ and $n_0 = 1 + \mathsf{Out}_x(P_3)$

(hide) $$\frac{P \equiv \nu\tilde{u}\,((\mathsf{hide}(x), r_x).\, P_1 \mid P_2)}{B \equiv \beta(x:\Gamma)\,\mathbf{B}^*[P] \xrightarrow{h;B;r_x\times n_H;(1,1)}_s \beta^h(x:\Gamma)\,\mathbf{B}^*[\nu\tilde{u}\,(P_1 \mid P_2)]}$$
provided that $x \notin \tilde{u}$ and $n_H = 1 + \mathsf{Hide}_x(P_2)$

(unhide) $$\frac{P \equiv \nu\tilde{u}\,((\mathsf{unhide}(x), r_x).\, P_1 \mid P_2)}{B \equiv \beta^h(x:\Gamma)\,\mathbf{B}^*[P] \xrightarrow{u;B;r_x\times n_U;(1,1)}_s \beta(x:\Gamma)\,\mathbf{B}^*[\nu\tilde{u}\,(P_1 \mid P_2)]}$$
provided $x \notin \tilde{u}$ and $n_U = 1 + \mathsf{Unhide}_x(P_2)$

(expose) $$\frac{P \equiv \nu\tilde{u}\,((\mathsf{expose}(x,\,\Gamma), r_x).\, P_1 \mid P_2)}{B \equiv \mathbf{B}[P] \xrightarrow{(e;B;r_x;(1,1))} \mathbf{B}\,\beta(y:\Gamma)\,[\nu\tilde{u}\,(P_1\{y/x\} \mid P_2)]}$$
provided that $y \notin \tilde{u}$, $y \notin \mathsf{sub}(\mathbf{B})$ and $y \notin \Gamma$ and $x \notin fn(P_2)$

(inter) $$\frac{P \equiv \nu\tilde{u}\,((x(w)r_x).\, P_1 \mid P_2) \qquad Q \equiv \nu\tilde{v}\,((\overline{y}\langle z\rangle, r_y).\, Q_1 \mid Q_2)}{B \equiv \mathbf{B}_1[P] \parallel \mathbf{B}_2[Q] \xrightarrow{I;B;\alpha(\Gamma,\Delta);(1,1)}_s \mathbf{B}_1[P'] \parallel \mathbf{B}_2[Q']}$$
provided that C_1 holds

($\mathsf{inter_h}$) $$\frac{P \equiv \nu\tilde{u}\,((x(w), r_x).\, P_1 \mid (y(z), r_y).\, P_2 \mid P_3)}{B \equiv \mathbf{B}[P] \parallel \mathbf{B}[P] \xrightarrow{I_h;B;\alpha(\Gamma,\Delta)/2;(2,1)}_s \mathbf{B}[Q_1] \parallel \mathbf{B}[Q_2]}$$ provided that C_2 holds

(join) $$B \equiv \mathbf{B}_1[P_1] \parallel \mathbf{B}_2[P_2] \xrightarrow{J;B;c_j;(1,1)} \mathbf{B}[P_1\sigma_1 \mid P_2\sigma_2]$$
provided that $f_{join}(\mathbf{B}_1, \mathbf{B}_2, P_1, P_2) = (\mathbf{B}, \sigma_1, \sigma_2, c_j)$ and $\mathbf{B}_1[P_1] \not\equiv \mathbf{B}_2[P_2]$

($\mathsf{join_h}$) $$B \equiv \mathbf{B}[P] \parallel \mathbf{B}[P] \xrightarrow{J_h;B;c_j/2;(2,1)}_s \mathbf{B}[P\sigma_1 \mid P\sigma_2]$$
provided that $f_{join}(\mathbf{B}_1, \mathbf{B}_2, P_1, P_2) = (\mathbf{B}, \sigma_1, \sigma_2, c_j)$

(split) $$B \equiv \mathbf{B}[P_1 \mid P_2] \xrightarrow{S;B;c_s;(1,1)}_s \mathbf{B}_1[P_1\sigma_1] \parallel \mathbf{B}_2[P_2\sigma_2]$$
provided that $f_{split}(\mathbf{B}, P_1, P_2) = (\mathbf{B}_1, \mathbf{B}_2, \sigma_1, \sigma_2, c_s)$

(redex) $$\frac{B_1 \xrightarrow{t;B_1;c;(n_1,n_2)}_s B_2}{B_1 \parallel B' \xrightarrow{t;B_1;c;(n_1+n_1',n_2+n_2')}_s B_2 \parallel B'}$$
where $(n_1', n_2') = \mathsf{Count}_t(B, B')$

(struct) $$\frac{B_1 \equiv B_1' \qquad B_1' \xrightarrow{\theta}_s B_2}{B_1 \xrightarrow{\theta}_s B_2}$$

Table 4: Axioms and rules for the reduction relation for stochastic Beta-binders.

The join and join_h represent the merge of two bio-processes in the case of two different or two identical bio-processes (up to $\equiv$). The rate c_j is given in the definition of the f_{join}. The axiom split is for the splitting of the bo-process and the rate c_s is given in the definition of the associated split. Finally, the rule redex adds the context B' to the reduction specified by the label $(t; B; r; (n_1, n_2)$ and updates the values n_1 and n_2 searching in B' if there are other bio-processes that could participate in the reaction. This is made by means of the function Count defined in table 3. The rule struct is as before.

3 The problem

A critical task in the translation of biological models into Beta-binders and generally into other process algebras is the specification of multi-reactant multi-product reactions. These reactions, rare in nature, are quite frequent in biological models. Since actions in Beta-binders involve at most two processes, a possible way to translate these reactions is to decompose them into a sequence of one-reactant or two-reactant reactions.

Consider the following reaction:

$$R_1 + R_2 + R_3 \xrightarrow{r} P$$

It has three reactants R_1, R_2, and R_3 and one product P and is described by a global rate r. If we have no biological information about the reaction we can try to decompose it in the following two reactions with rates r_1 and r_2 respectively:

$$\begin{aligned} R_1 + R_2 &\xrightarrow{r_1} complex(R_1, R_2) \\ complex(R_1, R_2) + R_3 &\xrightarrow{r_2} P \end{aligned}$$

where $complex(R_1, R_2)$ is the intermediate complex of the first two reactants. If this approach is adopted, some problems arise, as described below.

- There are different ways to decompose a reaction, depending on the possible ways to combine the reactants. We may consider one of the possibility or all ones. In the case above we select one order, but other two are possible. Generally, given n the number of reactants, there are $\frac{n!}{2}$ possible ways to decompose a complex reaction.

- After starting, a reaction may block at intermediate steps leading to a deadlock. This may happen for instance if the reactant R_3 misses or it is consumed in a further reaction. If we put the first reaction reversible, it is possible to come back to the original situation, but also in this case we may obtain undesired behaviours as a sequence of complexation and decomplexation.

- In the case of quantitative models, we have the problem to assign the two rates r_1 and r_2 and see what is the relation between them and the rate r.

Generally, we cope with the same problems whenever one biological phenomenon is modeled by using a sequence of elementary actions.

4 Beta-binders with biological transactions

In this section the syntax and the semantics of the extension of Beta-binders with biological transactions are reported. We call it TBeta-binders. We refer to the bio-processes and the pi-processes defined in [20] as *standard* to distinguish them from the *extended* ones defined here.

4.1 Biological transactions

Transactions are the basic mechanisms for modeling database transactions and for composing web-services in orchestration and choreography languages. Different properties and features must be considered according to the field of application. In modeling biological phenomena, the transactions need to satisfy some simple properties. Transactions should not stop at intermediate steps (for the lack of opportune processes) and they work as atomic actions (*atomicity*). Furthermore, reaction results should be visible only after transactions have ended (*serializability*). Finally, neither compensation processes nor nested transactions nor timeout mechanisms are used because not necessary to model biological reactions.

We refer to the transactions described here as *"biological transactions"*, to distinguish them from database and web service ones. These transactions are considered in [9], where both the π-calculus and the biochemical stochastic π-calculus are enriched with biological transactions. Here we focus on Beta-binders [20] and we work on to extend transactions to bio-processes.

4.2 General ideas

The transaction names t, t', t'', ... are introduced to identify transactions. These names are added to the syntax of Beta-binders to indicate in which transactions the bio-processes and the pi-processes are involved. Given a transaction t, a bio-process can be *blocked* (it is part of) or *unblocked* (it is not part of) with respect to t. The standard reduction relation for Beta-binders is extended in order to consider these transaction names. Two new axioms start and end are added, both described by suitable functions, called f_{start} and f_{end} respectively. The former axiom describes the start of a transaction t and the consequent block of the bio-processes involved in it. The axiom end describes the end of the transaction and the unblock of the final bio-process.

The general idea about this extended calculus is that an action characterized by a set of names T may be executed only if the respective bio-process is blocked by $t \in T$. If a bio-process is unblocked, only the actions with T equal to the empty set may be executed.

4.3 Syntax and Semantics

Syntax. The syntax of Beta-binders is extended in the following way. The definition of *elementary beta binder* and of *beta binder* are as usual.

The *extended pi-processes* (ranged over by P, P', ..., Q, Q', ...) are defined as:

$$P ::= \mathsf{nil} \mid \pi^{T^*}.P \mid P|P \mid \nu y P \mid !\,P \mid T(P)$$

$T \succ \mathsf{nil} = \mathsf{nil}$ $T \succ P\|Q = (T \succ P)\|(T \succ Q)$ $T \succ !P = !(T \succ P)$	$T \succ \pi^{T'}.P = \pi^{T \cup T'}.(T \succ P)$ $T \succ \nu y\, P = \nu y\, (T \succ P)$ $T \succ T'(P) = (T \cup T') \succ P$

Table 5: The operator $\succ$.

$$\pi = \overline{x}\langle w \rangle \mid x(z) \mid \mathsf{expose}(x, \Gamma) \mid \mathsf{hide}(x) \mid \mathsf{unhide}(x)$$

where we assume a countable infinite set of names $\mathcal{N}$ (ranged over lower-case letters *x, y, z, ...*) and a countable set of transaction names $\mathcal{T}$ (ranged over lower-case letters *t, t', t''*, ...), with $\mathcal{N} \cap \mathcal{T} = \emptyset$. We use t^* to denote a transaction name in $\mathcal{T}$ or the null string (denoted by ϵ). Furthermore, T denotes a non-empty subset composed of transaction names in $\mathcal{T}$. The set T^* may be either T or the empty set. If $T^* = \emptyset$ we can omit it and we have the usual actions. Note that the name t is important and distinguishes one transaction from the other ones.

Compared to the standard definitions of Beta-binders, we have a new term $T(P)$ and the set T^* added to the prefixes. The process $T(P)$ means that the process P may be executed only in the transactions in T. The term T^* in the prefixes represents the transactions in which the associated prefix/action may be involved. The operator $\succ$ is introduced to move from $T(P)$ to the representation where P has all the prefixes, with T added. The definition of the operator $\succ$ is reported in table 5. The symbol $\cup$ is the usual union of (name) sets.

The *extended bio-processes* (ranged over $B, B_1, B', \ldots$) are generated by the following grammar:

$$B \quad ::= \quad \mathsf{Nil} \;\big|\; (\mathbf{B}[P])^{t^*} \;\big|\; (B \parallel B)$$

The bio-processes defined above are the standard ones, except for the addition of a name t^* to the bio-process. If $t^* = \epsilon$, we have the usual bio-process (*unblocked*), otherwise we have a bio-process *blocked* by t.

We use the following notation:

- $(\mathbf{B}_1[P_1] \parallel \mathbf{B}_2[P_2] ... \parallel \mathbf{B}_n[P_n])^{t^*}$ stays for $(\mathbf{B}_1[P_1])^{t^*} \parallel (\mathbf{B}_2[P_2])^{t^*} ... \parallel (\mathbf{B}_n[P_n])^{t^*}$
- $(B)^t$ denotes a bio-process whose sub-terms are all blocked by t (and by no other transactions).

Some observations about the choice of these new elements are necessary. First of all the transactions are associated to the prefixes, not to the channel names. As a consequence, a channel name could be used in different actions involved in different transactions. In order to simplify the notation, a prefix may be involved either in a set of transactions, or in no transactions at all. Finally, a set of transaction names is added to the prefixes, instead of a single name. This allows us to use a given action for more than one transaction.

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

$P_1 \equiv P_2$ if P_1 and P_2 are α-equivalent	$\mathbf{B}[P_1] \equiv \mathbf{B}[P_2]$ if $P_1 \equiv P_2$
$(\mathcal{P}/_{\equiv}, \mid, \mathsf{nil})$ is a commutative monoid	$(\mathcal{B}/_{\equiv}, \|\|, \mathsf{Nil})$ is a commutative monoid
$\nu z\, \nu w\, P \equiv \nu w\, \nu z\, P$	$\mathbf{B}_1\mathbf{B}_2[P] \equiv \mathbf{B}_2\mathbf{B}_1[P]$
$\nu z\ \mathsf{nil} \equiv \mathsf{nil}$	$\mathbf{B}^*\hat{\beta}(x:\Gamma)[P] \equiv \mathbf{B}^*\hat{\beta}(y:\Gamma)[P\{y/x\}]$ if y fresh in P and $y \notin \mathsf{sub}(\mathbf{B}^*)$
$\nu y\,(P_1 \mid P_2) \equiv P_1 \mid \nu y\, P_2$ if $y \notin \mathsf{fn}(P_1)$	$(B)^{\epsilon} \equiv B$
$!\,P \equiv P \mid !\,P$	$(\mathbf{B}_1[P_1])^t \equiv (\mathbf{B}_2[P_2])^t$ provided that $\mathbf{B}_1[P_1] \equiv \mathbf{B}_2[P_2]$
$\emptyset(P) \equiv P$ and $\pi^{\emptyset}.P \equiv \pi.P$	

Table 6: Laws for structural congruence.

beta-binders uses both structural congruence over pi-processes and structural congruence over bio-processes. These are defined as the smallest relations satisfying the laws in table 6, where we overload the symbol ≡ when unambiguous.

They are the standard laws for Beta-binders, except for the last two lines, which contain the laws about transactions. The last laws reported on the left claims that the pi-processes $\emptyset(P)$ and $\pi^{\emptyset}.P$ are congruent to the pi-processes P and $\pi.P$, respectively. The law $(B)^{\epsilon} \equiv B$ says what happens if the empty string is considered in the bio-processes. The last law claims that two blocked bio-processes are congruent if they are congruent when unblocked and the transaction that blocks them is the same one.

We need to define the auxiliary functions Act and act in order to find the active transactions in bio-processes and in pi-processes, respectively. The function Act is described by the set of transaction name t that blocks a given bio-process B. Similarly, the function act returns the set of transaction names that are in the prefixes of the sub-terms of a given process P. The definition of both functions is reported below (the symbol ∪ stays for the usual union of sets).

Definition 1. *The function* Act *from bio-processes to* $2^{\mathcal{T}}$ *is defined as:*

- $\mathsf{Act}(Nil) = \emptyset$
- $\mathsf{Act}(B_1 \parallel B_2) = \mathsf{Act}(B_1) \cup \mathsf{Act}(B_2)$
- $\mathsf{Act}((\mathbf{B}[P])^{t^*}) = \emptyset$ *if* $t^* = \epsilon, \{t^*\}$ *otherwise*

The function act *from pi-processes to* $2^{\mathcal{T}}$ *n tris defined by:*

- $\mathsf{act}(nil) = \emptyset$
- $\mathsf{act}((\pi^{T^*}.P) = \emptyset$ *if* $T^* = \emptyset, T^*$ *otherwise*
- $\mathsf{act}(P_1|P_2) = \mathsf{act}(P_1) \cup \mathsf{act}(P_2)$
- $\mathsf{act}(\nu y P) = \mathsf{act}(P)$
- $\mathsf{act}(!\,P) = \mathsf{act}(P)$

(intra_t) $$\frac{P \equiv \nu\tilde{u}\,(x^{T^*_1}(w).\,P_1 \mid \overline{x}^{T^*_2}z.\,P_2 \mid P_3)}{(\mathbf{B}[P])^{t^*} \rightarrow_t (\mathbf{B}[\nu\tilde{u}\,(P_1\{z/w\} \mid P_2 \mid P_3)])^{t^*}}$$
provided that $t^* \in T^*_1 \cap T^*_2$ or $t^* = \epsilon$ and $T^*_1 = T^*_2 = \emptyset$

(inter_t) $$\frac{B_1 \equiv (\beta(x:\Gamma)\,\mathbf{B}^*_1[\nu\tilde{u}\,(x^{T^*_1}(w).\,P_1 \mid P_2)])^{t^*} \quad B_2 \equiv (\beta(y:\Gamma)\,\mathbf{B}^*_1[\nu\tilde{v}\,(\overline{y}^{T^*_2}z.\,Q_1 \mid Q_2)])^{t^*}}{(B_1)^{t^*} \parallel (B_2)^{t^*} \rightarrow_t (\beta(x:\Gamma)\,\mathbf{B}^*_1[\nu\tilde{u}\,(P_1\{z/w\} \mid P_2)])^{t^*} \parallel (\beta(y:\Delta)\,\mathbf{B}^*_2[\nu\tilde{v}\,(Q_1 \mid Q_2)])^{t^*}}$$
provided $t^* \in T^*_1 \cap T^*_2$ or $t^* = \epsilon$ and $T^*_1 = T^*_2 = \emptyset$, $\Gamma \cap \Delta \neq \emptyset$ and $x, z \notin \tilde{u}$ and $y, z \notin \tilde{v}$

(expose_t) $$\frac{P \equiv \nu\tilde{u}\,(\mathsf{expose}^{T^*}(x,\,\Gamma)\,.\,P_1 \mid P_2)}{(\mathbf{B}[P])^{t^*} \rightarrow_t (\mathbf{B}\beta(y:\Gamma)\,[\nu\tilde{u}\,(P_1\{y/x\} \mid P_2)])^{t^*}}$$
provided that $t^* \in T^*$ or $t^* = \epsilon$ and $T = \emptyset$, $y \notin \tilde{u}$, $y \notin \mathsf{sub}(\mathbf{B})$ and $y \notin \Gamma$, $y \notin \mathsf{fn}(\mathsf{P}_2)$

(hide_t) $$\frac{P \equiv \nu\tilde{u}\ \mathsf{hide}^{T^*}(x)\,.\,P_1 \mid P_2)}{(\beta(x:\Gamma)\,\mathbf{B}^*[P])^{t^*} \rightarrow_t (\beta^h(x:\Gamma)\,\mathbf{B}^*[\nu\tilde{u}\,(P_1 \mid P_2)])^{t^*}}$$
provided that $t^* \in T^*$ or or $t^* = \epsilon$ and $T^* = \emptyset$, $x \notin \tilde{u}$

(unhide_t) $$\frac{P \equiv \nu\tilde{u}\,(\mathsf{unhide}^{T^*}(x)\,.\,P_1 \mid P_2)}{(\beta^h(x:\Gamma)\,\mathbf{B}^*[P])^{t^*} \rightarrow_t (\beta(x:\Gamma)\,\mathbf{B}^*[\nu\tilde{u}\,(P_1 \mid P_2)])^{t^*}}$$
provided $t^* \in T$ or or $t^* = \epsilon$ and $T = \emptyset$, $x \notin \tilde{u}$

(join_t) $(\mathbf{B}_1[P_1])^{t^*} \parallel (\mathbf{B}_2[P_2])^{t^*} \rightarrow_t (\mathbf{B}[P_1\sigma_1 \mid P_2\sigma_2])^{t^*}$
provided that $f_{join}(\mathbf{B}_1, \mathbf{B}_2, P_1, P_2, T^*) = (\mathbf{B}, \sigma_1, \sigma_2, T^*)$
and $t^* \in T^*$ or $t^* = \epsilon$ and $T^* = \emptyset$

(split_t) $(\mathbf{B}[P_1 \mid P_2])^{t^*} \rightarrow_t (\mathbf{B}_1[P_1\sigma_1])^{t^*} \parallel (\mathbf{B}_2[P_2\sigma_2])^{t^*}$
provided that $f_{split}(\mathbf{B}, P_1, P_2, T^*) = (\mathbf{B}_1, \mathbf{B}_2, \sigma_1, \sigma_2, T^*)$
and $t^* \in T^*$ or $t^* = \epsilon$ and $T^* = \emptyset$

Table 7: Axioms and rules for the reduction relation for *TBeta-binders* (part 1).

(start_t) $B \rightarrow_t (B)^t$
provided that $\mathsf{Act}(B) = \emptyset$ and $f_{start}(B, t) = (B)^t$

(end_t) $(B)^t \rightarrow_t (B)$ with $f_{end}((B)^t) = (B)$

(redex_t) $$\frac{B \rightarrow_t B'}{B \parallel B'' \rightarrow_t B' \parallel B''}$$ provided that $\mathsf{Act}(B'') = \emptyset$

(struct_t) $$\frac{B_1 \equiv B_1' \qquad B_1' \rightarrow_t B_2}{B_1 \rightarrow_t B_2}$$

Table 8: Axioms and rules for the reduction relation for *TBeta-binders* (part 2).

The *reduction transition system* is $TSR = (\mathcal{B}, \rightarrow_t)$, where $\mathcal{B}$ is the set of states (equivalence classes of bio-processes w.r.t. $\equiv$) and *the reduction relation* $\rightarrow_t$ is the smallest relation over bio-processes obtained by applying the axioms and rules in Tables 7 and 8.

The main differences with the reduction rules of the standard beta-binders are the addition of the names t, the name set T, the axioms start_t and end_t.

The rules intra_t and inter_t represent the communication inside a bio-process or between two bio-processes. With respect to the standard rules, there is the condition that a communication along a channel x is possible only if the respective bio-process/bio-processes is/are blocked by $t^* \in T_1^* \cap T_2^*$ or $t^* = \epsilon$ and $T_1^* = T_2^* = \emptyset$, where T_1^* and T_2^* are the set of transaction names in which the input and the output prefixes may be involved. In the latter case the standard rules are obtained. As for the inter rule, a communication is possible only if the two bio-processes are blocked by the same t. The following three rules are about the hide, unhide and the expose of a site. The set T^* is associated to all the three prefixes and the actions are possible if the respective bio-processes are blocked by a transaction $t^* \in T^*$ or $t^* = \epsilon$ and $T^* = \emptyset$.

The axiom join_t models the merge of two bio-processes. It is described by an extended f_{join}, defined in $(\mathbf{B}_1, \mathbf{B}_2, P_1, P_2, T^*)$ and with result in $(\mathbf{B}, \sigma_1, \sigma_2, T^*)$. The function σ_1, σ_2 are the substitution functions to apply to the pi-process P_1 and P_2 in the resulting bio-process and T^* is the set of transactions in which the join may be used. The join may be applied only if both bio-processes are blocked by the same t. The result bio-process is blocked by the same transaction. The axiom split is used to model a split of one bio-process into two ones and is described by a suitable function f_{split}. Similarly to f_{join}, it is possible to extend the f_{split} to a function from $(\mathbf{B}, P_1, P_2, T^*)$ into $(\mathbf{B}_1, \mathbf{B}_2, \sigma_1, \sigma_2, T^*)$, where T^* has the same meaning as above. If $t^* = \epsilon$ and $T^* = \emptyset$ the standard axioms and the standard functions are obtained.

The following two axioms/rules are about the start and the end of transactions. The axiom start_t describes the start of a transaction in terms of the bio-processes

that it blocks. When the transaction t is selected the bio-processes involved in the transaction are blocked. The start of the transaction is described by the function f_{start}, defined in (B, t), where B is an unblocked bio-process or a parallel composition of unblocked bio-processes and $t \in \mathcal{T}$ is the name of the transaction. The function returns $(B)^t$, i.e. the bio-process B blocked by t. It is worth noting that this function main contain some conditions on the bio-processes.

The axiom end_t is used to define what happens when a transaction t ends successfully: the bio-processes blocked by t are unblocked. This is described by the function f_{end}, defined in $(B)^t$, where $(B)^t$ is a blocked bio-process or a parallel composition of blocked bio-processes and $t \in \mathcal{T}$ is the transaction that blocks the elements. The function returns the unblocked bio-process B.

The rule redex_t interprets the reduction of a subcomponent as a reduction of the global system. The condition $\mathsf{Act}(B'') = \emptyset$ is added. This means that a reduction is possible only if the other bio-processes in the system are unblocked and so there are no transactions to be completed. This guarantees that the actions involving transactions have the precedence. Under this condition, the transaction results atomic and serialized (see section 5).

Finally, the rule struct_t infers a reduction after a proper structural shuffling of the process at hand.

We focus on simple transactions and suppose that after starting they end with success and errors never occur. This allows us to avoid the definition of abort actions and of an abort axiom, furthermore no compensation mechanisms are necessary. The stop of a transaction at intermediate steps happens when it is not possible to execute the actions that lead to the final bio-processes. As a consequence, the associate f_{end} cannot be applied. This may be due to the fact that either the bio-processes or the pi-processes involved the transactions may not be reduced. We use the term *well-defined* to indicate transactions that when they start, then they end successfully. Well defined transactions are guaranteed by the following facts:

- the definition of f_{start} must take in account all the bio-processes necessary in the transactions;
- the definition of the pi-processes and of the functions f_{join} and f_{split} use in the translation must be appropriate.

In the following, we consider only well-defined transactions.

A concluding remark concerns the relation between standard Beta-binders and TBeta-binders. Firstly, it is worth noting that if $T^* = \emptyset$ and $t^* = \epsilon$, TBeta-binders reduce to the standard Beta-binders proposed in [20]. Indeed, concerning the syntax, the Tbeta-binders bio-processes and pi-processes with $T^* = \emptyset$ and $t^* = \epsilon$ are structural congruent to standard bio-processes and pi-processes (this follows from the structural congruence in TBeta-binders). With regards to the reduction relation, any Tbeta-binders rule/axiom (with the exception of start_t and end_t not present in Beta-binders) when $T^* = \emptyset$ and $t^* = \epsilon$, describes the same behaviour than the respective rule/axiom in the standard calculus. In particular, the condition $\mathsf{Act}(B'')$ in the rule redex_t may be neglected, as it is always satisfy. Finally, the axioms start_t and end_t are not considered, as they are applied only when a transaction t is introduced. As a consequence of this fact, Tbeta-binders without transactions may be replaced by standard Beta-binders. This is generally false in the case in

which a system is modeled by using Tbeta-binders with transactions. Indeed, if we consider a transition sequence and we blocked it by using a transaction (Tbeta-binders), we may obtain different results in terms of possible transitions from the case we use the same sequence but unblocked (standard Beta-binders). If we use transactions, the transitions following the start action are only internal and have the precedence over all the other external actions. They are executed one after the other and may not be interleaved with other actions. On the other hand, in the case of standard Beta-binders, the transitions of the sequence may be interleaved with other ones. As a result, it may happen that the sequence starts but does not complete.

5 Properties

In this section we report some properties of TBeta-binders. In particular we focus on the *atomicity* and *serializability* properties, as they are generally valid criteria for proving the correctness of the transactions.

First we need to introduce some auxiliary definitions.

- We consider the following set of labels (ranged over by $\tilde{l}_1, \tilde{l}_2, ...$):

$$\widetilde{\mathcal{L}} = \mathcal{L} \cup \{t : start, t : end\} \cup \{t : \alpha | \alpha \in \mathcal{L}\}$$

  where $\mathcal{L} = \mathcal{L}' \cup \mathcal{L}''$, $\mathcal{L}'$ is $\{i, h, u, e, I, J, S\}$ and $\mathcal{L}'' = \{I_h, J_h\}$. The label $t : \beta$ (where β is *end*, *start* or α) indicates that the transition of kind β regards bio-processes blocked by t.

- We may distinguish two kinds of action with respect to a transaction t:

  1. *internal transitions w.r.t. t*, whose labels are of the form $t : \alpha$ or $t : end$;
  2. *external transitions w.r.t. t*, whose labels are $t : start$ or α.

- A function f_t defined over labels $\widetilde{\mathcal{L}}$, is introduced to return the transaction in which a transition is involved. It is defined as:

$$f_t(\tilde{l}) = \begin{cases} \{t\} & \text{if } \tilde{l} = t : \beta \\ \emptyset & \text{otherwise} \end{cases}$$

- $B_0 \xrightarrow{\gamma}_t B_n$ denotes the transition sequence $B_0 \xrightarrow{\tilde{l}_1}_t B_1 \xrightarrow{\tilde{l}_2}_t B_2 ... \xrightarrow{\tilde{l}_n}_t B_n$, with $\gamma = \tilde{l}_1 \tilde{l}_2, ... \tilde{l}_n$. In the particular case the transition sequence represents a transactions, i.e. $\tilde{l}_1 = t : start$, $\tilde{l}_n = t : end$ and $f_t(\tilde{l}_i) = \{t\}$ for all i, we may use the notation $B_0 \xrightarrow{t}_t B_n$. In this notation we consider only the initial and the final bio-process, all the internal states are not considered. The length of the transition sequence characterized by γ is defined as:

$$length(\gamma) = \begin{cases} 1 & \text{if } \gamma = \tilde{l} \\ n & \text{if } \gamma = \tilde{l}_1 \tilde{l}_2 .... \tilde{l}_n \end{cases}$$

- The definitions of *serialized and serializable transition sequences* proposed in [5] are modified in order to consider *bio-processes*:

Definition 2. *The transition sequence $B \xrightarrow{\gamma}_t B'$, with $\mathsf{Act}(B) = \mathsf{Act}(B') = \emptyset$, is serialized iff $\tilde{l}_i = t : \alpha$ or $\tilde{l}_i = (t : start)$ implies $\tilde{l}_{i+1} = t : \alpha$ or $\tilde{l}_{i+1} = (t : end)$ for $i = 1, ...(n-1)$.*

The transition sequence $B \xrightarrow{\gamma}_t B'$ is serializable, if there exists a permutation γ' of γ such that $B \xrightarrow{\gamma'}_t B'$ is serialized.

- A final definition concerns the *finite derivative* of a bio-process.

  Definition 3. *Given a bio-process B, a finite derivative of B is either B itself or any bio-process B' obtained by a finite transition sequence $B \xrightarrow{\gamma}_t B'$.*

Now some properties are reported. Firstly, two simple facts about transition sequences in TBeta-binders are described in the following lemma:

Proposition 1. *Given the bio-process B with $\mathsf{Act}(B) = \emptyset$, for any finite derivative B' of B,*

(i) there is at most one transaction active in B';

(ii) there are no nested transactions in B'.

Proof The proposition is proved in the two items below.

(i) We show that $\mathsf{Act}(B') = \emptyset$ or $\mathsf{Act}(B') = \{t\}$ for a given $t \in \mathcal{T}$ (i.e. the cardinality of $\mathsf{Act}(B')$ is less than or equal to 1). From the hypotheses B' is a finite derivative of B and then by definition $B' = B$ or there exists γ s.t. $B \xrightarrow{\gamma}_t B'$. If $B' = B$ by hypotheses we have $\mathsf{Act}(B') = \mathsf{Act}(B) = \emptyset$. If $B' \neq B$ we prove it by induction on the length of the derivative transition sequence. Let γ be $\tilde{l}_1\tilde{l}_2, ...\tilde{l}_n$ and $B \xrightarrow{\tilde{l}_1}_t B_1 ... B_{(n-1)} \xrightarrow{\tilde{l}_n}_t B'$. The length of the sequence is n. Consider the two cases:

   - $n = 1$. In this case $B \xrightarrow{\tilde{l}_1}_t B'$ and $\tilde{l}_1$ is either $t : start$, for a given t, or α. In the former case $\mathsf{Act}(B') = \{t\}$ and in the latter $\mathsf{Act}(B') = \emptyset$.
   - $n > 1$. By inductive hypothesis on the sequence $\gamma = \tilde{l}_1\tilde{l}_2, ...\tilde{l}_{(n-1)}$, we have that $\mathsf{Act}(B_{(n-1)}) = \emptyset$ or $\mathsf{Act}(B_{(n-1)}) = \{t\}$. In the former case $\tilde{l}_{(n)}$ may be either $t : start$ or α, in the latter case $\tilde{l}_{(n)}$ may be either $t : \alpha$ or $t : end$. In all the four cases we have that $\mathsf{Act}(B') = \emptyset$ or $\mathsf{Act}(B') = \{t\}$.

(ii) We show that there are not bio-processes that have sub-terms of the form $(B)^t$ with $\mathsf{Act}(B) \neq \emptyset$. In order to prove this it is sufficient to prove that it is not possible to block an already blocked bio-process. This follows directly from the axiom t_start, that may be applied only if the bio-process is unblocked.

□

Other results useful for the proof of the following theorems are reported in the Proposition 2.

Proposition 2. *The following facts hold.*

(i) Consider $B \xrightarrow{\tilde{l}}_{t2} B'$ *with* $\mathsf{Act}(B) = \{t\}$. *If* $\tilde{l} = t : end$ *then* $\mathsf{Act}(B') = \emptyset$ *otherwise if* $\tilde{l} = t : \alpha$ *then* $\mathsf{Act}(B') = \{t\}$.

(ii) If $B \xrightarrow{\tilde{l}}_{t2} B'$ *with* $\mathsf{Act}(B) = \emptyset$ *and* $\tilde{l} = t : start$ *then* $\mathsf{Act}(B') = \{t\}$.

(iii) If $B \xrightarrow{\tilde{l}}_{t2} B'$ *with* $\mathsf{Act}(B) = \emptyset$ *and* $f_t(\tilde{l}) = \emptyset$ *then* $\mathsf{Act}(B') = \emptyset$.

(iv) Given a bio-process B *such that* $\mathsf{Act}(B) = \emptyset$ *and given the transaction sequence* $B \xrightarrow{\tilde{l}_1}_t B_1 \xrightarrow{\tilde{l}_2}_t B_2 ... \xrightarrow{l_n}_t B_n$ *with* $\tilde{l}_1$=*t:start and* $\tilde{l}_n$=*t:end then* $\mathsf{Act}_t(B_i) = \{t\}$ *for* $i = 1, ...(n-1)$ *and* $\tilde{l}_i = t : \alpha$ *for* $i = 2, ...(n-1)$ *and* $\mathsf{Act}(B_n) = \emptyset$.

Proof The proof of each point follows.

(i) If $\tilde{l} = t : end$ the resulting bio-process is unblocked and therefore $\mathsf{Act}(B') = \emptyset$. In the case $\tilde{l} = t : \alpha$, the action is internal w.r.t. to t and $\mathsf{Act}(B') = \{l\}$.

(ii) If $\tilde{l} = t : start$ the resulting bio-process is blocked and therefore $\mathsf{Act}(B') = \{t\}$.

(iii) In this case the action is external w.r.t. to each transaction and therefore $\mathsf{Act}(B') = \emptyset$.

(iv) From the rule redex_t, the actions involving transactions have the precedence over the other ones. Therefore $\tilde{l}_i = t : \alpha$ for $i = 2, ...(n-1)$. From the first point, $\mathsf{Act}_t(B_i) = \{t\}$ for $i = 1, ...(n-1)$ and $\mathsf{Act}(B_n) = \emptyset$ follow.

□

The next two theorems concern the atomicity and the serializability of biological transactions.

Theorem 1. ***(Atomicity)*** *Consider the bio-process* $(B_1)^t$, *with* t *well-defined and* $\mathsf{Act}(B_1) = \emptyset$. *Let* B_n *be the bio-process obtained by the finite transition sequence* $(B_1)^t \xrightarrow{\gamma}_t B_n$ *with* $f_t(\tilde{l}_i) = \{t\}$ *for each* $\tilde{l}_i \in \gamma$, *and* $\mathsf{Act}(B_n) = \emptyset$. *Then for each* S *with* $\mathsf{Act}(S) = \emptyset$ $(B_1)^t \parallel S \xrightarrow{\gamma}_t B_n \parallel S$.

Proof Let consider $(B_1)^t \parallel S$. Since by hypothesis $\mathsf{Act}(S) = \emptyset$ and $\mathsf{Act}((B_1)^t) = \{t\}$ then, from rule redex_t, the actions involving $(B_1)^t$ have the precedence over all the other actions. In particular, as all the actions in γ are involved in the transaction t and S is unblocked, the actions in γ have the precedence over the ones involving S. As a consequence, from $(B_1)^t \xrightarrow{\gamma}_t B_n$ we have that $(B_1)^t \parallel S \xrightarrow{\gamma}_t B_n \parallel S$. □

Theorem 2. ***(Serializability)*** *Given* $B \xrightarrow{\gamma}_t B'$ *with* $\mathsf{Act}(B) = \mathsf{Act}(B') = \emptyset$, *it is serialized.*

Proof The proof is by induction on the length of the transition sequence $B \xrightarrow{\gamma}_t B'$. Let n be the length of the sequence, we have the following cases.

1. If $n = 1$ then $B \xrightarrow{\tilde{l}}_t B'$ and the property vacuously satisfied.

2. If $n > 1$ then $B \xrightarrow{\tilde{l}_1}_t B_1 \xrightarrow{\tilde{l}_2}_t B_2 ... B_{(n-1)} \xrightarrow{\tilde{l}_n}_t B'$. If $f_t(\tilde{l}_i) = \emptyset$ for $i = 1, ..., n$ then the condition is vacuously satisfied. Otherwise, let be k the first index for which $f_t(\tilde{l}_k) \neq \emptyset$. If $k > 1$ then by inductive hypothesis, the transition sequence $B_{(k-1)} \xrightarrow{\gamma'}_t B'$, with $\gamma' = \tilde{l}_k \tilde{l}_{(k+1)} ... \tilde{l}_n$, is serialized. Also the transition sequence $B \xrightarrow{\gamma''}_t B_{(k-1)}$ with $\gamma'' = \tilde{l}_1 \tilde{l}_{(2)} ... \tilde{l}_{(k-1)}$ is serialized. Therefore $B \xrightarrow{\gamma}_t B'$ is serialized. Finally, if $k = 1$ then $\tilde{l}_1 = t : start$, $\tilde{l}_n = t : end$ and from the third point of Proposition 2, $\tilde{l}_i = t : \alpha$ for $i = 2, ...(n-1)$ and therefore $B \xrightarrow{\gamma}_t B'$ is serialized.

□

Note that the condition $\mathsf{Act}(B'') = \emptyset$ in the rule redex_t is a sufficient condition to guarantee the properties above. If this condition is not considered, weaker properties are satisfied. In particular serializability is still valid, but atomicity is not still guaranteed.

Specifically, consider *the reduction relation* $\rightarrow_{t2}$ where the rule redex_t in the reduction relation $\rightarrow_t$ is replaced by the following rule:

$$\text{(redex\_t2)} \qquad \frac{B \rightarrow_{t2} B'}{B \parallel B'' \rightarrow_{t2} B' \parallel B''}$$

The new *reduction transition system* is $TSR2 = (\mathcal{B}, \rightarrow_{t2})$.

In this case two transactions involving any two actions (not in the same transaction) are permutable:

Lemma 1. *If* $B \xrightarrow{\tilde{l}_1}_{t2} B' \xrightarrow{\tilde{l}_2}_{t2} B''$, *with* $f_t(\tilde{l}_1) = \{t\}$, $f_t(\tilde{l}_1) \neq f_t(\tilde{l}_2)$ *and* $\tilde{l}_1 \neq t : end$ *then there exists* B''' *such that* $B \xrightarrow{\tilde{l}_2}_{t2} B''' \xrightarrow{\tilde{l}_1}_{t2} B''$.

Proof By hypothesis, $\tilde{l}_1$ is either $t : start$ or $t : \alpha$. We consider the latter case, the former is similar. As $\tilde{l}_1 = t : \alpha$, than there exists a sub-term $(B_1)^{t_1}$ of B and a sub-term $(B'_1)^{t_1}$ of B' s.t. $(B_1)^{t_1} \xrightarrow{\tilde{l}_1} (B'_1)^{t_1}$. Secondly, since $f_t(\tilde{l}_1) \neq f_t(\tilde{l}_2)$ the label $\tilde{l}_2$ refers either to a transaction different from t_1 or to a transition not involving transactions. Furthermore, there exists a process B_2, sub-term of B', and a process B'_2, sub-term of B'' such that $B_2 \xrightarrow{\tilde{l}_2} B'_2$.

From the previous observations, it is inferred that $B \equiv (B_1)^{t_1} \parallel B_2 \parallel B_3$, $B' \equiv (B'_1)^{t_1} \parallel B_2 \parallel B_3$ and $B'' \equiv (B'_1)^{t_1} \parallel B'_2 \parallel B_3$, for some bio-process B_3. If the rule (redex_t2) is first applied considering $B_2 \xrightarrow{\tilde{l}_2} B'_2$, we obtain $B \equiv B_2 \parallel (B_1)^{t^*} \parallel B_3 \xrightarrow{\tilde{l}_2} B'_2 \parallel (B_1)^{t^*} \parallel B_3$ and finally we derive the following transition sequence: $B \xrightarrow{\tilde{l}_2} B'_2 \parallel (B_1)^{t^*} \parallel B_3 \xrightarrow{\tilde{l}_1} B'_2 \parallel (B'_1)^{t^*} \parallel B'_2 \equiv B''$. The process $B'_2 \parallel (B_1)^{t^*} \parallel B_3$ is the process B''' which we are looking for. □

Now we can give the following result for the atomicity and the serializability.

Theorem 3. *Let $B \xrightarrow{\gamma_1}_{t2} B'$ a transition sequence such that* $\mathsf{Act}(B) = \emptyset$.

(i) If $\mathsf{Act}(B') = \emptyset$ *then there exists a permutation γ_2 of γ_1 s.t. $B \xrightarrow{\gamma_2}_{t2} B'$ is serialized.*

(ii) If $\mathsf{Act}(B') = \{t\}$ *then there exist γ_2 and γ_3 s.t. for each $\tilde{l} \in \gamma_3$ then $f_t(\tilde{l}) = \{t\}$, $\gamma_2\gamma_3$ is a permutation of γ_1 and $B \xrightarrow{\gamma_2}_{t2} B'' \xrightarrow{\gamma_3}_{t2} B'$ where* $\mathsf{Act}(B'') = \emptyset$.

(iv) Atomicity is not guaranteed.

Proof The proof of each item follows.

(i) Let be $\gamma_1 = \tilde{l}_1\tilde{l}_2....\tilde{l}_n$ with $n \geq 1$ and $B \xrightarrow{\tilde{l}_1}_{t2} B_1 \xrightarrow{\tilde{l}_2}_{t2} ... \xrightarrow{\tilde{l}_n}_{t2} B'$. We have to show that there exists a permutation $\gamma_2 = \tilde{l}'_1\tilde{l}'_2\tilde{l}'_3....\tilde{l}'_n$ of γ_1 such that $B \xrightarrow{\gamma_2}_{t2} B'$ is serialized. The proof is by induction on the length of the transition sequence. Let n be the length of the transition sequence, we consider the two cases:

1. If $n = 1$, the only possibility is that $\gamma_1 = \tilde{l}_1$ and $f_t(\tilde{l}_1) = \emptyset$ and it is obviously serialized.
2. If $n > 1$, we have two possibilities:
   - If $f_t(\tilde{l}_1) = \emptyset$, from the third point in Proposition 2 we have that $\mathsf{Act}(B_1) = \emptyset$. By applying the inductive hypothesis we obtain that $B_1 \xrightarrow{\gamma'_1}_{t2} B'$ is serialized, with γ'_1 a permutation of $\tilde{l}_2\tilde{l}_3...\tilde{l}_n$. Therefore $B_1 \xrightarrow{\tilde{l}_2}_{t2} B_2 \xrightarrow{\tilde{l}_3}_{t2} ... \xrightarrow{\tilde{l}_n}_{t2} B'$ is serialized and $\gamma_2 = \tilde{l}_1\gamma'_1$ is the permutation of γ_1 we are looking for.
   - If $f_t(\tilde{l}_1) \neq \emptyset$ we have two cases.
     * The first possibility is $f_t(\tilde{l}_i) \neq \emptyset$ for each $\tilde{l}_i \in \gamma_1$. since by hypothesis $\mathsf{Act}(B) = \mathsf{Act}(B') = \emptyset$ we have that $\tilde{l}_1 = t : \mathit{start}$, $\tilde{l}_n = t : \mathit{end}$ and the other labels are $\tilde{l}_1 = t : \alpha$, for some α. Therefore $B \xrightarrow{\gamma_1}_{t2} B'$ is serialized and $\gamma_2 = \gamma_1$.
     * The second possibility is when there exists at least one label not involved in a transaction. Let k be the first index for which the condition $f_t(\tilde{l}_k) = \emptyset$ is satisfied. By applying Lemma 1 more times we may move the transition with label $\tilde{l}_k$ to the first position and so we may apply the inductive hypothesis on the rest of the transition sequence, as seen above.

(ii) This proposition is proved by induction on the length of the transition sequence. The complete proof is not reported as similar to the one of the proposition above. It is worth noting that in the case $n = 1$ we have that $B''' = B$ and γ_2 reduced to the empty sequence. This is also the case when all the transitions are involved in a transaction. The general case is obtained by applying Lemma 1 to move all the transitions not involved in transactions to the first positions.

(iii) From the rule (redex_t2) the actions involved in transactions have no precedence over the other ones and they may interleave the standard ones. As a

consequence, it may happen that a transaction starts but does not complete. This is for instance the case in which there is a reversible reaction modeled by a join (for the forward direction) and a split (for the inverse direction). In this case a transaction can start, but after that we may have a (infinite) sequence of join and split describing the reversible reaction.

□

6 Stochastic Beta-binders with biological transactions

In this section we present the stochastic extension of TBeta-binders. As in [11], we consider Gillespie as reference algorithm, but other ones could be considered as well. Since transactions represent reactions with more than two reactants, an extended version of Gillespie approach is necessary [24].

With respect to TBeta-binders, the change in the syntax consists in the replacement of the prefixes π^{T^*} with (π^{T^*}, r), where r is ∞ (i.e. the associated action is immediate) or a positive real number that corresponds to the parameter of an exponential distribution that drives the dynamic behaviour of the system. In the semantics, we need to do some changes in the definition of labels, in the auxiliary functions and in the axioms/rules proposed in [11] and add new elements. To better represent the start axiom, a derived operator $\diamond$ is introduced. The *multi bio-process* $n \diamond B$, where $n \in \mathbb{N}$ is the *multiplicity* of the bio-process B, represents the parallel composition of n bio-processes congruent to B. A definition of $\diamond$-standard form follows (hereafter, we write $\prod_{i=1}^{n} B_i$ for the parallel composition of n bio-processes).

Definition 4. *A bio-process B is in $\diamond$-standard form if it is* Nil *or* $B \equiv \prod_{i=1}^{n} n_i \diamond B_i$, *with* $n \geq 1$ *where*

- $B_i \not\equiv B_j \; \forall i \neq j$
- $B_i \equiv (\mathbf{B}_i[P_i])^{t^*}$ *and* $n_i \geq 1$, $\forall i = 1, ..., n$.

$\prod_{i=1}^{n} B_i$ is the parallel composition of n bio-processes. We have the following result:

Proposition 3. *Every bio-process B is structural congruent to a $\diamond$-standard form.*

Proof The proof is by structural induction over bio-processes. Let's consider the three possible cases.

- Case $B =$ Nil. B is in a $\diamond$-standard form by definition.
- Case $B = (\mathbf{B}[P])^{t^*}$. In this case $B \equiv 1 \diamond (\mathbf{B}[P])^{t^*}$, that is a $\diamond$-standard form.
- Case $B = B_1 \parallel B_2$. By inductive hypothesis on B_1 and B_2 we have that $B_1 \equiv \prod_{i=1}^{n_1} k_i \diamond B_{1i}$ and $B_2 \equiv \prod_{j=1}^{n_2} l_j \diamond B_{2j}$. In order to obtain a $\diamond$-standard form congruent to B we apply the following procedure:

$$
\begin{array}{l}
B_1 \triangleright B_2 ::= \\
\quad (\mathbf{B}[P])^{t^*} \parallel B_1 \triangleright \mathsf{Nil} = B_1 \triangleright 1 \diamond (\mathbf{B}[P])^{t^*} \\
\quad (\mathbf{B}[P])^{t^*} \parallel B_1 \triangleright (\prod_{i=1}^{n} n_i \diamond B_i) = \\
\qquad B_1 \triangleright (\prod_{i=1}^{n} n'_i \diamond B_i) \qquad \text{with } n'_k = n_k + 1, \quad n'_i = n_i \quad \forall i \neq k \\
\qquad \text{if } \exists k \in [1, ..., n] \text{ with } (\mathbf{B}[P])^{t^*} \equiv B_k \\
\qquad B_1 \triangleright (\prod_{i=1}^{n+1} n_i \diamond B_i) \quad \text{with } B_{n+1} = (\mathbf{B}[P])^{t^*} \text{ and } n_{n+1} = 1 \qquad \text{otherwise} \\
\quad \mathsf{Nil} \triangleright B_2 = B_2
\end{array}
$$

Table 9: The definition of the operator $\triangleright$.

- Let n'_1 be the number of bio-processes B_{1i} not structural congruent to any B_{2j}. For each of these bio-processes, consider the term $B_{12h} = k_i \diamond B_{1i}$, for appropriate i, $h = 1, .., n'_1$.
- Similarly, let n'_2 be the number of bio-processes B_{2j} not structural congruent to any B_{1i}. For each of these bio-processes consider the term $B_{12h} = l_j \diamond B_{2j}$, for appropriate j and $h = (n'_1 + 1), ..., (n'_1 + n'_2)$.
- Let n'_3 be the number of bio-processes present in both the standard forms such that $B_{1i} \equiv B_{2j}$, for some i and j. For each bio-process of this kind, define $m_h = k_i + l_j$ and consider the term $B_{12h} = m_h \diamond B_{1i}$ for appropriate i and $h = (n'_1 + n'_2 + 1), ..., (n'_1 + n'_2 + +n'_3)$

Then we have that $\prod_{i=1}^{n} B_{12h}$, with $n = n'_1 + n'_2 + n'_3$, is in a $\diamond$-standard form and $B \equiv \prod_{i=1}^{n} B_{12h}$ by construction. □

An operator $\triangleright$ may be introduced to put a bio-process B into a $\diamond$-standard form B' s.t. $B \equiv B'$. It is defined in Table 9. Starting from a bio-process B the operator reduces it to a $\diamond$-standard form. The final result is obtained when the process on the left is Nil (there is nothing left to reduce).

The description of auxiliary definitions is reported below.

- The labels in TBeta-binders are replaced by $\phi = (k'; B; c; nl)$, with $\phi \in \Phi$. B and c are as before. The first component denotes the kind of the reaction but here more possibilities are given. Indeed k' belongs to the set

$$\widetilde{\mathcal{L}} = \mathcal{L} \cup \{t : start, t : end\} \cup \{t : \alpha | \alpha \in \mathcal{L}\}$$

  The element nl is a list, whose elements are couples (n_i, κ_i). The component κ_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

$\mathsf{In}_{x,t^*}(\mathsf{nil}) = 0 \quad \mathsf{In}_{x,t^*}(P_1 \vert P_2) = \mathsf{In}_{x,t^*}(P_1) + \mathsf{In}_{x,t^*}(P_2)$ $\mathsf{In}_{x,t^*}(!(\pi^{T^*}, r).P) = \mathsf{In}_{x,t^*}((\pi^{T^*}, r).P) \quad \mathsf{In}_{x,t^*}(\nu y.P) = \mathsf{In}_{x,t^*}(P)$ if $x \neq y$, 0 otherwise $\mathsf{In}_{x,t^*}((\pi^{T^*}, r).P) = 1$ if $ch(\pi) = x$ and $(t^* \in \mathsf{act}((\pi^{T^*}, r).P)$ or $(T^* = \emptyset$ and $t^* = \epsilon))$, 0 otherwise
$\mathsf{Out}_x t^*(\mathsf{nil}) = 0 \quad \mathsf{Out}_x t^*(P_1 \vert P_2) = \mathsf{Out}_x t^*(P_1) + \mathsf{Out}_x t^*(P_2)$ $\mathsf{Out}_x t^*(!(\pi^{T^*}, r).P) = \mathsf{Out}_x t^*((\pi^{T^*}, r).P) \quad \mathsf{Out}_x t^*(\nu y.P) = \mathsf{Out}_x t^*(P)$ if $x \neq y$, 0 otherwise $\mathsf{Out}_x t^*((\pi^{T^*}, r).P) = 1$ if $ch(\pi) = \overline{x}$ and $(t^* \in \mathsf{act}((\pi^{T^*}, r).P)$ or $(T^* = \emptyset$ and $t^* = \epsilon))$, 0 otherwise
$\mathsf{G}_{x,t^*}(\mathsf{nil}) = 0 \quad \mathsf{G}_{x,t^*}(P_1 \vert P_2) = \mathsf{G}_{x,t^*}(P_1) + \mathsf{G}_{x,t^*}(P_2)$ $\mathsf{G}_{x,t^*}(!(\pi^{T^*}, r).P) = \mathsf{G}_{x,t^*}((\pi^{T^*}, r).P) \quad \mathsf{G}_x(\nu y.P) = G_x(P)$ if $x \neq y$, 0 otherwise $\mathsf{G}_{x,t^*}((\pi^{T^*}, r).P) = 1$ if $\pi = g(x)$ and $t^* \in \mathsf{act}_T((\pi^{T^*}, r).P)$, 0 otherwise where $(\mathsf{G}, g) \in \{(\mathsf{Hide}, \mathsf{hide}), (\mathsf{Unhide}, \mathsf{unhide})\}$

Table 10: Modified auxiliary functions (first part).

ϕ 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 $h(nl)$, 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 and $\prod_{i=1}^{n}$ denotes the usual product (of numbers). The actual rate a_r is calculated as $a_r = c \times h(nl)$. In addition to the labels of kind ϕ we consider here the labels $\psi \in \Psi$. These labels are of the form $= (k', a_r)$ and contain the actual rate of the action, as calculated above.

- The functions In_x, Out_x, Hide_x and Unhide_x must be modified to take into account if the action is associated to a transaction or not. In the extended calculus not all the actions are enabled, it depends in which transaction they are involved or if they are enabled outside transactions. The new functions are called In_{x,t^*}, Out_{x,t^*}, Hide_{x,t^*} and Unhide_{x,t^*} in order to show the dependency not only from x, but also from t^*. They are described in Tab. 10.

- With relation to the functions to count the bio-processes, the modified functions are reported in Tab. 11.

The semantics is given in terms of *structural congruence* and *reduction relation*. The structural congruence is as defined previously, with the only difference that in this case the prefixes are enriched with the rates and they must be equal to have congruence over pi-processes. The *reduction relation* is the smallest relation over processes satisfying the rules and axioms given in Tab. 13 and Tab. 14. The labeled transition system for the stochastic Beta-binders with transactions is defined by $\mathcal{LTST} = (\mathcal{B}, \Psi, \mapsto)$ where the auxiliary transition relation $\rightarrow_{ts} \subseteq \mathcal{B} \times \Phi \times \mathcal{B}$ is used to define the transition relation $\mapsto \subseteq \mathcal{B} \times \Psi \times \mathcal{B}$.

In the following we give a brief description of each rule. Consider Table 12, in which all the actions describing unimolecular reactions (i.e. only one reactants) are considered. The rule intra describes the communication inside the box. The two auxiliary functions In_{x,t^*} and Out_{x,t^*} are used to count the number of firable

$\mathsf{Num}((\mathbf{B}[P])^{t^*}, \mathsf{Nil}) = 0$

$\mathsf{Num}((\mathbf{B}[P])^{t_1^*}, (\mathbf{B}'[P'])^{t_2^*} \| B) = 1 + \mathsf{Num}((\mathbf{B}[P])^{t_1^*} \| B)$ if $(\mathbf{B}[P])^{t_1^*} \equiv (\mathbf{B}'[P'])^{t_2^*}$

$\mathsf{Num}((\mathbf{B}[P])^{t_1^*}, (\mathbf{B}'[P'])^{t_2^*} \| B) = \mathsf{Num}((\mathbf{B}[P])^{tr_1} \| B)$ if $(\mathbf{B}[P])^{t_1^*} \not\equiv (\mathbf{B}'[P'])^{t_2^*}$

$\mathsf{Count_k}(B, B') =$ case k in

$\mathcal{U}$: $(\mathsf{Num}(B, B'))$

$\mathcal{B}$: if $B \equiv (\mathbf{B}_1[P_1])^{t_1^*} \| \mathbf{B}_2[P_2]^{t_2^*}$ then $(\mathsf{Num}((\mathbf{B}_1[P_1])^{t_1^*}, B'), \mathsf{Num}((\mathbf{B}_2[P_2])^{t_2^*}, B'))$

$\mathcal{H}$: if $B \equiv (\mathbf{B}[P])^{t^*} \| (\mathbf{B}[P])^{t^*}$ then $(\mathsf{Num}((\mathbf{B}[P])^{t^*}, B'))$

$\mathcal{M}$: if $B \equiv \prod_{i=1}^{n} \kappa_i \diamond (\mathbf{B}_i[P_i]))^{t_i^*}$ then $(\mathsf{Num}((\mathbf{B}_1[P_1])^{t_1^*}, B'), \mathsf{Num}((\mathbf{B}_2[P_2])^{t_2^*}, B'), ..., \mathsf{Num}((\mathbf{B}_n[P_n])^{t_n^*}, B'))$

where $\mathcal{U} = \{i, h, u, e, S, t : i, t : h, t : u, t : e, t : S\}$

$\mathcal{B} = \{J, I, t : J, t : I\}$

$\mathcal{H} = \{J_h, I_h, t : J_h, t : I_h\}$

$\mathcal{M} = \{t : start, t : end\}$

Table 11: Modified auxiliary functions (second part).

inputs and outputs on the channel x characterized by t^*. The rate r_x is the same for both input and output prefix as in the standard stochastic Beta-binders. The rule expose, hide, unhide are as in the qualitative case. The functions Hide_{x,t^*} and Unhide_{x,t^*} are used to count the number of hide and unhide binders on x involving the transaction t^*. The rule split is as usual, with the rate added to the f_{split} and with the appropriate conditions on the transactions. In the Table 13 the actions that describe bimolecular (two distinct reactants) and homodimeration (two identical reactants) reactions are reported. In the case of inter and $\mathsf{inter_h}$ the rates are expressed by the affinity function between the types of the sites involved in the communication. In the case of join and $\mathsf{join_h}$ the rate is associated to the relative f_{join}. Table 14 reports the axioms start_ts, end_ts and the rule redex_ts, struct_ts and fin_ts. The axiom start_ts describes the start of a transaction t. The process B is represented in the $\diamond$-standard form. The parameter κ_i is the stoichiometry of the reactant of kind i and the bio-process $\mathbf{B}_i[P]$ represents the process describing that reactant. A transaction describes a reaction with $N = \sum_i^n \kappa_i$ reactants, among which n are distinct. Concerning the quantitative information contained in the labels, c_s 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 $c_s \times h(nl)$. The axiom tend_ts describes the end of a transaction. The rate is reported in the relative f_{end}. In the rule redex_ts $\mathsf{Act}(B'') = \emptyset$ is added to guarantee that the actions inside the transaction have priority over the external ones. Four cases are considered, according to the kind of the action k'. For each of them is said how to update the list nl' by considering the bio-processes in the system. The set $\mathcal{U}$ is equal to $\{i, h, u, e, S, t : i, t : h, t : u, t : e, t : S\}$, all actions involving only one bio-process. The sets $\mathcal{B} = \{J, I, t : J, t : I\}$ and $\mathcal{H} = \{J_h, I_h, t : J_h, t : I_h\}$ represent the actions involving two bio-processes, different or equal (up to $\equiv$), respectively. Finally the set $\mathcal{M} = \{t : start, t : end\}$ contains the start and end actions, that generally involve more elements. The rule struct_ts is the standard rule with the possibility to deal with transaction. The last rule fin_ts describes how the system finally reduces at each step. The premise collects the quantitative information in its label $\phi = (l, r_b, nl)$. The conclusion describes the same relation, but its label gives

(intra_ts) $$\frac{P \equiv \nu\tilde{u}\,((x^{T^*{}_1}(w), r_x).\, P_1 \mid (\overline{x}^{T^*{}_2}z, r_x).\, P_2 \mid P_3)}{B \equiv (\mathbf{B}[P])^{t^*} \xrightarrow{k';B;r_x\times n_I\times n_O;[(1,1)]}_{ts} (\mathbf{B}[\nu\tilde{u}\,(P_1\{z/w\} \mid P_2 \mid P_3)])^{t^*}}$$

provided that $n_I = 1 + \mathsf{In}_{x,t^*}(P_3), n_O = 1 + \mathsf{Out}_{x,t^*}(P_3)$ and
either ($t^* = t \in T^*{}_1 \cap T^*{}_2$ and $k' = t : i$) or ($t^* = \epsilon$ and $T^*{}_1 = T^*{}_2 = \emptyset$ and $k' = i$) holds

(expose_ts) $$\frac{P \equiv \nu\tilde{u}\,((\mathsf{expose}^{T^*}(x, \Gamma), r_x).\, P_1 \mid P_2)}{B \equiv (\mathbf{B}[P])^{t^*} \xrightarrow{k';B;r_x;[(1,1)]}_{ts} (\mathbf{B}\beta(y : \Gamma)\,[\nu\tilde{u}\,(P_1\{y/x\} \mid P_2)])^{t^*}}$$

$y \notin \tilde{u}$, $y \notin \mathsf{sub}(\mathbf{B})$ and $y \notin \Gamma$ and $x \notin fn(P_2)$ and
provided that either ($t^* = t \in T$ and $k' = t : e$) or ($t^* = \epsilon$ and $T = \emptyset$ and $k' = e$)

(hide_ts) $$\frac{P \equiv \nu\tilde{u}\,((\mathsf{hide}^{T^*}(x), r_x).\, P_1 \mid P_2)}{B \equiv (\beta(x : \Gamma)\,\mathbf{B}^*[P])^{t^*} \xrightarrow{k';B;r_x\times n_H;[(1,1)]}_{ts} (\beta^h(x : \Gamma)\,\mathbf{B}^*[\nu\tilde{u}\,(P_1 \mid P_2)])^{t^*}}$$

provided that either ($t^* = t \in T^*$ and $k' = t : h$) or ($t^* = \epsilon$ and $T = \emptyset$ and $k' = h$) and
$x \notin \tilde{u}$ and $n_H = 1 + \mathsf{Hide}_{x,t^*}(P_2)$

(unhide_ts) $$\frac{P \equiv \nu\tilde{u}\,((\mathsf{unhide}^{T^*}(x), r_x).\, P_1 \mid P_2)}{B \equiv (\beta^h(x : \Gamma)\,\mathbf{B}^*[P])^{t^*} \xrightarrow{k';B;r_x\times n_U;[(1,1)]}_{ts} (\beta(x : \Gamma)\,\mathbf{B}^*[\nu\tilde{u}\,(P_1 \mid P_2)])^{t^*}}$$

provided that either ($t^* = t \in T$ and $k' = t : u$) or ($t^* = \epsilon$ and $T = \emptyset$ and $k' = u$) and
$x \notin \tilde{u}$ and $n_U = 1 + \mathsf{Unhide}_{x,t^*}(P_2)$

(split_ts) $$B \equiv (\mathbf{B}[P_1 \mid P_2])^{t^*} \xrightarrow{k';B;c_s;[(1,1)]}_{ts} (\mathbf{B}_1[P_1\sigma_1] \parallel \mathbf{B}_2[P_2\sigma_2])^{t^*}$$

provided that $f_{split}(\mathbf{B}, P_1, P_2, T^*) = (\mathbf{B}_1, \mathbf{B}_2, \sigma_1, \sigma_2, T^*, c_s)$
and either($t^* = t \in T^*$ and $k' = t : S$) or ($t^* = \epsilon$, $T^* = \emptyset$, $k' = S$) holds

Table 12: Stochastic TBeta-binders. Axioms and rules for the reduction relation (part 1).

(inter_ts) $$\frac{P = \nu\tilde{u}\,((x^{T^*_1}(w), r_x).\, P_1 \mid P_2) \quad Q = \nu\tilde{v}\,(\overline{y}^{T^*_2}z, r_y).\, Q_1 \mid Q_2)}{B \equiv (\mathbf{B}_1[P] \parallel \mathbf{B}_2[Q])^{t^*} \xrightarrow{k';B;\alpha(\Gamma,\Delta);[(1,1),(1,1)]}_{ts} (\mathbf{B}_1[P'] \parallel \mathbf{B}_2[Q'])^{t^*}}$$
provided that C_1 holds and
either ($t^* = t \in T_1 \cap T_2$ and $k' = t : I$) or ($t^* = \epsilon$ and $T^*_1 = T^*_2 = \emptyset$ and $k' = I$)

(inter$_h$_ts) $$\frac{P = \nu\tilde{u}\,((x^{T^*_1}(w), r_x).\, P_1 \mid (\overline{y}^{T^*_2}z, r_y).\, P_2 \mid P_3)}{B \equiv (\mathbf{B}[P] \parallel \mathbf{B}[P])^{t^*} \xrightarrow{k';B;\alpha(\Gamma,\Delta);[(2,2)]}_{ts} (\mathbf{B}[Q_1] \parallel \mathbf{B}[Q_2])^{t^*}}$$
provided that C_2 holds and
either ($t^* = t \in T_1 \cap T_2$ and $k' = t : I_h$) or ($t^* = \epsilon$ and $T^*_1 = T^*_2 = \emptyset$ and $k' = I_h$)

(join_ts) $B \equiv (\mathbf{B}_1[P_1] \parallel \mathbf{B}_2[P_2])^{t^*} \xrightarrow{k';B;c_j;[(1,1),(1,1)]}_{ts} (\mathbf{B}[P_1\sigma_1 \mid P_2\sigma_2])^{t^*}$
provided that $f_{join}(\mathbf{B}_1, \mathbf{B}_2, P_1, P_2, T^*) = (\mathbf{B}, \sigma_1, \sigma_2, T^*, c_j)$ and $\mathbf{B}_1[P_1] \not\equiv \mathbf{B}_2[P_2]$ and
and either ($t^* = t \in T^*$ and $k' = t : J$) or ($t^* = \epsilon$ and $T^* = \emptyset$ and $k' = J$)

(join$_h$_ts) $B \equiv (\mathbf{B}[P] \parallel \mathbf{B}[P])^{t^*} \xrightarrow{k';B;c_j;[(2,2)]}_{ts} (\mathbf{B}'[P_1\sigma_1 \mid P_2\sigma_2])^{t^*}$
provided that $f_{join}(\mathbf{B}, \mathbf{B}, P_1, P_2, T^*) = (\mathbf{B}', \sigma_1, \sigma_2, T^*, c_j)$ and
and either ($t^* = t \in T^*$ and $k' = t : J_h$) or ($t^* = \epsilon$ and $T^* = \emptyset$ and $k' = J_h$)

Table 13: Stochastic TBeta-binders. Axioms and rules for the reduction relation (part 2).

(start_ts) $B \xrightarrow{t:start;B_1;c_s;[(\kappa_1,\kappa_1),(\kappa_2,\kappa_2),...(\kappa_n,\kappa_n)]}_{ts} (B)^t$

provided that $f_{start}(B,t) = ((B)^t, c_s)$

and $B \equiv B_1 = \prod_{i=1}^{n} \kappa_i \diamond B_i$ and $\mathsf{Act}(B) = \emptyset$

(end_ts) $(B)^t \xrightarrow{t:end;B;c_e;[(\kappa_1,\kappa_1),(\kappa_2,\kappa_2),...]}_{ts} (B)$

with $f_{end}((B)^t) = (B, c_e)$

and $B \equiv \prod_{i=1}^{n} \kappa_i \diamond B_i$

(redex_ts) $$\frac{B \xrightarrow{k';B_1;c;nl}_{ts} B'}{B \parallel B'' \xrightarrow{k';B_1;c;nl'}_{ts} B' \parallel B''}$$

provided that $\mathsf{Act}(B'') = \emptyset$ and

Case $k' \in \mathcal{U}$: if $nl = [(n_1, 1)]$ and $\mathsf{count}_{k'}(B_1, B'') = (n'_1)$ then $nl' = [(n_1 + n'_1, 1)]$

Case $k' \in \mathcal{B}$: if $nl = [(n_1, 1), (n_2, 1)]$ and $\mathsf{count}_{k'}(B_1, B'') = (n'_1, n'_2)$

then $nl' = [(n_1 + n'_1, 1), (n_1 + n'_1, 1)]$

Case $k' \in \mathcal{H}$: if $nl = [(n_1, 2)]$ and $\mathsf{count}_{k'}(B_1, B'') = (n'_1)$ then $nl' = [(n_1 + n'_1, 2)]$

Case $k' \in \mathcal{M}$: if $nl = [(n_1, \kappa_1), (n_1, \kappa_2), ..., (n_n, \kappa_n)]$ and $\mathsf{count}_{k'}(B_1, B'') = (n'_1, n'_2, ..., n'_n)$ then $nl' = [(n_1 + n'_1, \kappa_1), (n_2 + n'_2, \kappa_2), ..., (n_n + n'_n, \kappa_n)]$

(struct_ts) $$\frac{B_1 \equiv B'_1 \qquad B'_1 \xrightarrow{\phi}_{ts} B_2}{B_1 \xrightarrow{\phi}_{ts} B_2}$$

(fin_ts) $$\frac{B_1 \xrightarrow{(k';B_1;c;nl)}_{ts} B_2}{B_1 \xmapsto{(k';c\times h(nl))} B_2}$$

Table 14: Stochastic TBeta-binders. Axioms and rules for the reduction relation (part 3).

explicitly the speed of the transition. This is calculated by the information given in the label of the premise.

6.1 Rates

A point that remains to investigate is how to associate the rates with each single step that is used to model the reactions. Since in this work we refer to Gillespie, the idea is to define the rates in such a way that the actual rate associated to the start action represents the actual rate of the reaction. Therefore, the global rate of the reaction is associated to the start prefix of the transaction and ∞ is assigned to all the other prefixes in the processes related to the transaction. In this way the the start action describes the start of the reactions and all the other actions follow as immediate.

It is worth noting that the simulation algorithm for Beta-binders must be extended in order to consider transactions. Since internal actions happens in a defined sequence (at least in the application we considered) and are immediate, it is possible to neglect them and simplify the algorithm. These aspects are not considered here as out of the purposes of this work.

7 Examples

In this section some simple examples are reported to see how *TBeta binders* are used to model reactions.

7.1 Translation of multiple-reactant multiple-product reactions

Consider a generic reaction R composed of n_r reactants, n_p products and n_m modifiers, with $n_r + n_m > 2$ and/or $n_p + n_m > 2$, described by a rate r:

$$A_1 + ... + A_{n_r} + M_1 + ... + M_{n_m} \xrightarrow{r} B_1 + ... + B_{n_p} + M_1 + ... + M_{n_m}$$

The following approach must be followed.

- If there are further biological information on the reaction, we may decompose the reactions into elementary steps as it happens in reality.
- If further information are not available and steps are unknown, we can translate the reaction into TBeta-binders in the following way:
  - 1 start action to block the bio-processes involved. It is described by a f_{start} and the associated basal rate is c, obtained from r by some simple relations.
  - $(n_r + n_m - 1)$ joins to merge the reactants and $(n_p + n_m - 1)$ splits in order to get the products. Each of them is associated to a rate ∞;
  - 1 end action to unblock the product-bio-processes, with rate ∞.

Globally we need $(n_r + 2n_m + n_p)$ definitions of instance of functions.

It is worth noting that we may applied TBeta-binders also in the former case, if we want the reaction does not stop at intermediate steps.

As an example, let consider the case of the three-reactant one-product reaction $R_1 + R_2 + R_3 \rightarrow P$. It may be modeled in TBeta-binders by two joins with the addition of one start and one end for the transaction. The start and the end actions are described by the functions f_{start} and f_{end}:

$$f_{start}(B_1 \parallel B_2 \parallel B_3, t_R) = ((B_1 \parallel B_2 \parallel B_3)^{t_R})$$

$$f_{end}((B_P)^{t_R}) = (B_P)$$

where $B_1 = \beta(x_1 : \Gamma_1)\,\mathbf{B}_1[P_1]$, $B_2 = \beta(x_2 : \Gamma_2)\,\mathbf{B}_2[P_2]$, $B_2 = \beta(x_3 : \Gamma_3)\,\mathbf{B}_2[P_2]$ and $B_P = \mathbf{B}_P[P_P]$ are the bio-processes representing the three reactants and the product. The elements $\beta(x_i : \Gamma_i)$ for $i = 1, 2, 3$ are the beta-binders of the three reactants involved in the reaction. A possible definition of f_{start} is:

$$\begin{array}{lll} \lambda B_1 B_2 B_3 t_r. & \text{if} & (B_1 = \beta(x_1 : \Gamma_1)\,\mathbf{B}_1[P_1] \text{ and } B_2 = \beta(x_2 : \Gamma_2)\,\mathbf{B}_2[P_2] \text{ and} \\ & & B_3 = \beta(x_3 : \Gamma_3)\,\mathbf{B}_3[P_3]) \\ & \text{then} & ((B_1 \parallel B_2 \parallel B_3)^{t_r}) \\ & \text{else} & \perp \end{array}$$

The conditions are only on the beta-binders, as no pi-processes are involved in the transaction. Two joins have to be defined to model the reaction and they both refer to elements blocked by the transaction t_R. The first join, described by f_{join_1}, represents the formation of the element $complex(R_1, R_2)$ composed of the first two reactants R_1 and R_2. The complex is represented by the bio-process $\mathbf{B}_{complex(R_1,R_2)}[P_1\sigma_1|P_2\sigma_2]$, where $\mathbf{B}_{complex(R_1,R_2)} = \beta(x_{complex(R_1,R_2)} : \Gamma_{complex(R_1,R_2)})$. The second join, described by f_{join_2}, represents the merge between the the third reactant R_3 and the $complex(R_1, R_2)$. The result is the bio-process for the product of the reaction. The two functions f_{join_1} and f_{join_2} are respectively:

$$f_{join_1}(\mathbf{B}_{R_1}, \mathbf{B}_{R_2}, P_1, P_2, \{t_R\}) = (\mathbf{B}_{complex(R_1,R_2)}, \sigma_1, \sigma_2, \{t_R\});$$

$$f_{join_2}(\mathbf{B}_{complex(R_1,R_2)}, \mathbf{B}_{R_3}, P_{12}, P_3, \{t_R\}) = (\mathbf{B}_P, \sigma_3, \sigma_4, \{t_R\}).$$

The functions $\sigma_1, \sigma_2\ \sigma_3, \sigma_4$ are substitution functions, $\{t_R\}$ is the set of transactions in which the join may applied, $P_{12} = P_1\sigma_1|P_2\sigma_2$.

A reduction of the system $S = B_1 \parallel B_2 \parallel B_3 \parallel S'$, with $Act(S') = \emptyset$, is:

$$\begin{array}{lll} B_1 \parallel B_2 \parallel B_3 \parallel S' & \xrightarrow{t_R:start} & (B_1 \parallel B_2 \parallel B_3)^{t_R} \parallel S' \\ & \xrightarrow{t_R:J} & (complex(R_1, R_2) \parallel R_3)^{t_R} \parallel S' \\ & \xrightarrow{t_R:J} & (B_P)^{t_R} \parallel S' \\ & \xrightarrow{t_R:end} & B_P \parallel S' \end{array}$$

The quantitative case is translated similarly, the only difference is the addition of rates.

7.2 The citric acid cycle

This model is from *KEGG* (Kyoto Encyclopedia of Genes and Genomes) metabolic pathway database [13, 1]. It regards the *citric acid cycle*, also known as the *Krebs*

cycle or *tricarboxylic acid cycle*. This cycle is a fundamental metabolic pathway involving enzymes essential for energy production through aerobic respiration and is also an important source of biosynthetic building blocks used in other processes as for instance the amino acid and the fatty acid biosyntheses. Only qualitative data are given and eventual further quantitative information could be searched in the literature or databases.

The model consists of a series of chemical reactions of central importance in all living cells that involves a lot of proteins, molecules and enzymes. The citric acid cycle takes place in mitochondria where it oxidizes acetyl-CoA, derived not only from glycolysis but also from the oxidation of fatty acids. An acetyl-CoA molecule enters the cycle interacting with oxaloacetate to create citrate, for which the subsequent cycle of reactions is named. Acetyl-CoA is oxidized gradually by a chain of reactions. Citrate serves as a substrate for a series of distinct enzyme-catalyzed reactions that occur in sequence and proceed with the formation of intermediate compounds, including Succinate, Fumarate, and (S)-Malate. Malate is converted to Oxaloacetate, which in turn reacts with yet another molecule of acetyl-CoA, thus producing citric acid and the cycle begins again.

As the cycle proceeds the intermediates are oxidized, transferring their energy to create high-energy electrons in the form of NADH (reduced nicotinamide adenine dinucleotide) or FADH2 (reduced flavin adenine dinucleotide) and one molecule of ribonucleotide GTP (guanosine triphosphate). The former ones are coenzymes (molecules that enable or enhance enzymes) that store energy and are passed to a membrane-bound electron-transport chain to produce H_20, the latter produces ATP. The oxidation of the metabolic intermediates of the pathway also releases two carbon dioxide molecules for each acetyl-CoA that enters the cycle, leaving the net carbons the same with each turn of the cycle. This carbon dioxide is the one of the sources of CO_2 released into the atmosphere when you breathe.

A schematic representation of the citric acid cycle is reported in figure 1. Here only the main reactions and the main species involved are reported. The species inside the square brackets are the enzymes involved in the reactions, the other ones are the reactants and products of the reactions of the cycle.

7.2.1 The translation of the model into Beta-binders

Initial System. Each *species* is represented by the respective bio-process with one beta-binders representing the interaction capabilities of the element. For instance, the element citrate is represented by the following bio-process:

$$B_{Citrate} = \beta(x_{Citrate} : \{r_{Citrate}\})[\mathsf{nil}]$$

The only pi-process used is nil. Indeed the model gives a high level of abstraction and for representing the reactions of the model we use only joins and splits. The other species are translated similarly. The initial system may be given by the bio-processes representing the enzymes, Acetyl-CoA and Oxaloacetate:

$$S = B_{Oxaloacetate} \parallel B_{Acetyl-Coa} \parallel B_{CS} \parallel B_{ID} \parallel B_A \parallel B_{SD} \parallel B_{2HD} \parallel B_{SCS} \parallel B_F$$

The enzymes are denoted by the initial of the names in capital letter. So, for instance, the bio-process B_{CS} represents the enzyme *Citrate Synthase*.

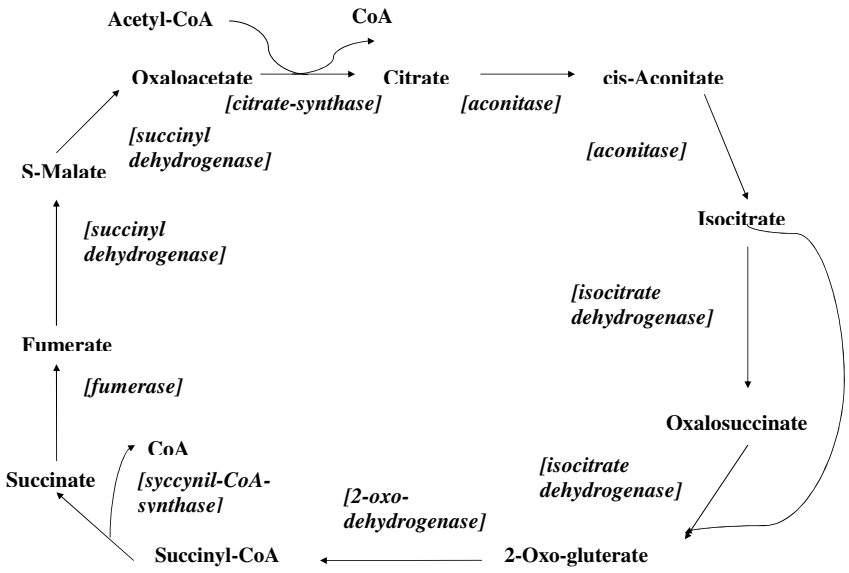

Figure 1: Schematic view of the citric acid cycle

The bio-processes $B_{Oxaloacetate}$ and $B_{Acetyl-Coa}$ represent Oxaloacetate and Acetyl-Coa, respectively. For simplicity we consider only an element for each species.

Reactions. Each *reaction* is rendered by a set of suitable joins and splits. We consider here biological transactions to represent each reaction atomically. The translation of the main reactions follows.

- The first kind of reaction is the enzymatic reaction with one reactant, one modifier and one product. It is translated by using a start action to block the bio-process involved, followed by a join and a split. Finally, an end action unblocks the bio-processes returning the final products. All the reactions with the exception of two cases are of this kind. We show how to translate one, the others are dealt in the same way. We consider the reaction:

$$Fumerate + FH \rightarrow (S) - Malate + FH$$

where FH stays for Fumerate Hydratase. Fumarate undergoes a hydration catalyzed by Fumarate Hydratase to produce S-Malate. Let be B_F, B_{FH} and B_{SM} the bio-processes representing Fumerate, Fumerate Hydratase and S-Malate respectively. The start of the reaction blocks the element B_F and B_{FH} and it is described by a function $f_{start}(B_F \parallel B_{FH}, t_1) = ((B_F \parallel B_{FH})^{t_1})$. The name t_1 indicates the transaction. One join is used to represent the formation of the intermediate element $complex(fumerate, fumerate\_hydratase)$ described by the bio-

process $B_{c(F,FH)}$. The respective f_{join} is:

$$f_{join}(\mathbf{B}_F, \mathbf{B}_{FH}, \mathsf{nil}, \mathsf{nil}, \{t_F\}) = (\mathbf{B}_{c(F,FH)}\mathsf{id}, \mathsf{id}, \{t_1\})$$

where $\mathbf{B}_F = \beta(x_F : \{r_F\})$, $\mathbf{B}_{FH} = \beta(x_{FH} : \{r_{FH}\})$ and $\mathbf{B}_{c(F,FH)} = \beta x_{c(F,FH)} : \{r_{c(F,FH)}\}$ are the binders of the bio-processes involved and id represents the identity function. A split is used to get the products and it is described by:

$$f_{split}(\mathbf{B}_{c(F,FH)}, \mathsf{nil}, \mathsf{nil}, \{t_F\}) = (\mathbf{B}_{SM}, \mathbf{B}_{FS}, \mathsf{id}, \mathsf{id}, \{t_1\})$$

where $\mathbf{B}_{SM} = \beta(x_{SM} : \{r_{SM}\})$ is the binder of the product $((S)-Malate)$ Finally the f_{end} to unblock the bio-process is defined as $f_{end}((B_{SM} \parallel B_{FH})^t) = (B_{SM} \parallel B_{FH})$.

- The second kind of reaction is described by two reactants, two products and one modifier and in the model it is represented by the reaction:

  $$acetyl - CoA + Oxoloacitate + CS \rightarrow CoA + Citrate + CS$$

  where CS stays for *Citrate Synthase*. *Acetyl-CoA* interacts with *Oxaloacetate* to form *Citrate* and *CoA*. The reaction is translated by using one start to block the bio-processes representing the reactants and the modifier, two joins to form the intermediate complex, two splits to get the products and one end to unblock the elements. The definitions of the functions are not reported, as similar to the ones above.

- The last kind of reaction is represented by:

  $$Oxolossucinate + ID \rightarrow 2 - Oxo - glutarate + CO_2 + ID$$

  where ID stays for *Isocitrate Dehydrogenase*. This enzyme catalyzes the reaction from *Oxalosuccinate* to *CO2* and *2-Oxo-glutarate*. In this case it is necessary to define a start, followed by a join and two splits and finally one end. The definitions are not reported as similar to the ones above.

In order to represent the whole model it is necessary to define 11 f_{start}, 11 f_{end}, 12 f_{join} and 13 f_{split}. Moreover, we need to give the definition of the initial system specifying the bio-processes presented initially.

8 Discussion and conclusion

In this paper an extension of Beta-binders with biological transactions is presented. The aim of this work is to model a sequence of actions representing complex reactions, as it were atomic and therefore to represent complex reactions with more than two reactants or more than two products in a suitable way with binary interactions. Indeed some problems arise when we model these reactions by using standard Beta-binders (see section 3). Specifically, the main difficulties are the decomposition of the reaction into elementary reaction, the fact that a reaction can stop at intermediate steps leading to a deadlock when a reactant misses and the problem of how to associate the rates at each step. The extended version presented here is useful to face these problems, as described below.

- Concerning the decomposition of the reaction, also in the case of TBeta-binders the reaction is decomposed into elementary steps. In this case the order is not important, as the interactions involving the elementary reactions are internal to the transactions. The relevant actions are the start and the end of the transactions, the other ones are only auxiliary.

- From the definition of biological transactions given in this work, it follows that when the transaction starts it completes. Therefore it is not possible that a reaction stops at the intermediate steps.

- In the case of rates, the global rate of the reaction is associated to the start action. The other actions follow as immediate and it is not necessary to find a rate for each elementary step. It is worth noting that the actual rate of the reactions depends on all the reactants involved.

From the observations above it is clear that TBeta-binders are useful to deal with complex reactions whose details are unknown. Some observations are due in the case of quantitative models. First of all, in this work we use the Gillespie method as the reference stochastic algorithm. We consider the extended version that consider reactions with more than two reactants, widely used in the simulator tools. Obviously other stochastic algorithms could be considered.

Last but not least, it is important to observe that the kinetic laws associated to complex reaction may be different from mass action. Although the interpretation of models with mass action kinetics for Gillespie algorithm is straightforward [12], the application of Gillespie (and the use of stochastic versions of Beta-binders and TBeta-binders) to reactions with kinetic laws different from mass action may give wrong results [4]. The authors of [4] show that when Gillespie is applied to Hill kinetics the magnitude of fluctuations is overstimated. In this case the application of complex kinetic laws in the stochastic context has leaded to errors in noise levels. In these cases the best solution is to decompose the reaction into the elementary reactions as it happens in reality. The possible decomposition depends on the kind of kinetic law.

References

[1] Kegg home page. Available at http://sbml.org/kegg2sbml.html.

[2] L. Bocchi, C. Laneve, and G. Zavattaro. A calculus for long-running transactions. In *Proc. 6th IFIP International Conference on Formal Methods for Open Object-based Distribuited Systems (FMOODS'03)*, volume 2884 of *Lecture Notes in Computer Science*. Springer-Verlag, 2003.

[3] R. Bruni, C. Laneve, and U. Montanari. Orchestrating transactions in join calculus. In *Proc. of CONCUR'02*, volume 2421 of *Lecture Notes in Computer Science*. Springer-Verlag, 2002.

[4] R. Bundschuh, F. Hayot, and C. Jayaprakash. Fluctuations and slow variables in genetic networks. *Biophys. J.*, 84:1606–1615, 2003.

[5] N. Busi and G. Zavattaro. On the Serializability of Transactions in JavaSpaces. In *Proc. of CONCOORD'01*, volume 54 of *Lecture Notes in Computer Science*. Springer-Verlag, 2001.

[6] M. Butler and C. Ferreira. An operational semantics for StAC, a language for modelling long-time business transactions. In *Proc. of COORDINATION'04*, volume 2949 of *Lecture Notes in Computer Science*. Springer-Verlag, 2002.

[7] M. Butler, T. Hoare, and C. Ferreira. A trace semantics for long-running transactions. In *Proc. of 25 year of CSP*, volume 3525 of *Lecture Notes in Computer Science*. Springer-Verlag, 2005.

[8] M. Calder, S. Gilmore, and J. Hillston. Modeling the influence of RKIP on the ERK signalling pathway using the stochastic process algebra PEPA. In *Computational Methods in Systems Biology '04 (CMSB04)*, Electronic Notes in Theoretical Computer Science, 2005.

[9] F. Ciocchetta and C. Priami. Biological transactions for quantitative models. In *Proc. of MeCBIC2006*, volume 171 of *ENTCS*, pages 55–67, 2007.

[10] V. Danos and J. Krivine. Formal molecular biology done in CCS-R. In *BioConcur '03, Workshop on Concurrent Models in MolecularBiology*, 2003.

[11] P. Degano, D. Prandi, C. Priami, and P. Quaglia. Beta binders for biological quantitative experiments. In *4th Int. Workshop on Quantitative Aspects of Programming Languages (QAPL 06)*, 2006. To appear.

[12] D.T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem.*, 81(25):2340–2361, 1977.

[13] M. Kanehisa and S. Goto. KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, 28(1):27–30, Jan 2000.

[14] C. Kuttler, J. Niehren, and R. Blossey. Gene regulation in the π-calculus: simulating cooperativity at the lambda switch. In *Second international workshop on concurrent models in molecular biology (BioConcur 2004)*, 2004.

[15] C. Laneve and G. Zavattaro. Foundations of web transactions. In *Proc. of Foundations of Software Science and Computation Structure(FOSSACS 2005)*, volume 3441 of *Lecture Notes in Computer Science*. Springer-Verlag, 2005.

[16] C. Laneve and G. Zavattaro. Web-π at work. In *Proc. of International Symposium on Thustworthy Global Computing*, volume 3705 of *Lecture Notes in Computer Science*. Springer-Verlag, 2005.

[17] P. Lecca and C. Priami. Cell Cycle control in Eukaryotes: a BioSpi model. In *Bioconcur 03*, 2003.

[18] P. Lecca, C. Priami, P. Quaglia, B. Rossi, C. Laudanna, and G. Costantin. Language modeling and simulation of autoreactive lymphocytes recruitment in inflamed brain vessels. *SIMULATION: Transactions of The Society for Modeling and Simulation International.*, 80(4):273–288, 2004.

[19] M. Nagasaki, S. Onami, S. Miyano, and H. Kitano. Bio-calculus: Its concept and molecular interaction. In K. Asai, S. Miyano, and T. Takagi, editors, *Genome Informatics 1999*, volume 10, pages 133–143. Universal Academy Press, 1999.

[20] C. Priami and P. Quaglia. Beta binders for biological interactions. In *Computational Methods in Systems Biology '04 (CMSB04)*, 2004.

[21] C. Priami, 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(1):25–31, 2001.

[22] A. Regev, E. M. Panina, W. Silverman, L. Cardelli, and E. Shapiro. BioAmbients: An Abstraction for Biological Compartments. *Theoretical Computer Science*, 325(1):141–167, September 2004.

[23] A. Regev, W. Silverman, and E. Shapiro. Representation and simulation of biochemical processes using the π-calculus process algebra. In *Proceedings of the Pacific Symposium of Biocomputing 2001*, volume 6, pages 459–470, 2001.

[24] O. Wolkenhauer, M. Ullah, W. Kolch, and K. H. Cho. Modelling and Simulation of IntraCellular Dynamics: Choosing an Appropriate Framework. *IEEE Transactions on NanoBioScience*, 3(3):200–207, September 2004.