The Microsoft Research - University of Trento
Centre for Computational and Systems Biology

Technical Report CoSBi 01/2006

Causality and Concurrency in Beta-binders

Maria Luisa Guerriero
University of Trento
guerrier@dit.unitn.it

Corrado Priami
CoSBi
and
DISI, University of Trento
priami@cosbi.eu

Causality and Concurrency in *Beta-binders*

Maria Luisa Guerriero
University of Trento
guerrier@dit.unitn.it
and
Corrado Priami
The Microsoft Research - University of Trento
Centre for Computational and Systems Biology
priami@dit.unitn.it

Abstract

Causal relations allow us to understand the causes of single transitions/events in a computation and, consequently, to acquire information on the whole systems. In this paper a definition of a causal relation and of an enabling relation for *Beta-binders* is given, together with the description of some important properties of these relations; in particular we show that the concurrency relation is the complement of the union of causal and enabling relations for each possible computation. The application domains which we are mostly interested in are biology and medical sciences, thus the application of the defined relations to a model of the intensively studied ERK/MAPK pathway is described.

1 Introduction

When analyzing the interactions between different entities in a concurrent system, it is useful to take into account the *causal* and *temporal* relations between the events. This is becoming particularly relevant in the emergent field of dynamical modeling of biological systems.

Obviously, the knowledge of the dependencies between some bio-chemical events could help us in the study of the behavior of complex systems by fixing the order of some events, and hence by limiting the number of interleaving interactions to be considered. Moreover, to understand the causes of a given event (for example a disease) life scientists could take a limited number of events into account (the ones that exercise some dependence on it), while they could safely ignore the others.

In particular, causality seems to play a relevant role in understanding complex biochemical pathways, since it permits to sequence independent activities and, hence, to simplify the models.

In this paper we define the relations of *causality* and *enabling* on actions in *Beta-binders*, which is a formalism recently introduced [10] to specify the interaction of bio-chemical entities.

Intuitively, an activity A *causes* an activity B if A influences the execution of B, while an activity A *enables* an activity B if either A is a necessary condition for the execution of B or A cannot be executed after B.

The union of causality and enabling gives all the possible dependencies of an action on other actions and hence the *concurrency* relation is defined as the complement of the union of causality and enabling relations for each possible computation. Intuitively, two activities A and B are *concurrent* if they can be executed in parallel.

In the literature there are several works on causal and concurrent relations for other models, such as CCS [3, 5, 9], π-calculus [1, 6, 4, 7, 12] and Petri Nets [13]; instead, currently there is no related work for *Beta-binders*.

In this paper we adopt the transition system-based technique used in [6, 4] to define a causal and an enabling relation for the π-calculus. Since *Beta-binders* is an extension of the π-calculus, we basically extend these relations to address the new *Beta-binders* operators. Hence, an approach similar to the one used in [6, 4] is adopted for the operators present in both languages, while we had to reinterpret the notion of causality with respect to the operators which are specific to *Beta-binders*, in particular the splitting and joining of boxes. The relations we define are not directly comparable to the ones defined in [6, 4], since the languages they refer to are different. Nevertheless, the intuitive notions of causality and concurrency for the two languages are similar; one noteworthy difference is that according to our definition, contrary to [6], the causal relation is not included in the enabling one; indeed we show that in our case none of these relations include the other one.

Examples of use of causal relations in modeling bio-chemical systems are in [2].

In the next section we briefly describe the language of *Beta-binders*, while in Sect. 3 the labeled semantics of *Beta-binders* is introduced. In Sect. 4 we concentrate on the definition of the relations, and in Sect. 5 some important properties of the defined relations are studied. Some of the potential applications are described in Sect. 6 and, in particular, an example is shown in Sect. 6.1. Finally some concluding remarks are presented.

2 Beta binders and Bio-processes

In this section we briefly recall the syntax and the semantics of *Beta-binders* (see [10, 11] for details).

Basically, a *Beta-binders* process is a π-calculus process enclosed in a box (or a parallel composition of them) and the actions that such a process can execute are a superset of those of π-calculus.

2.1 Syntax

An *elementary beta binder* has the form $\beta(x : \Delta)$, where the name x is the *subject* of the beta binder and Δ is the *type* of x (it is a non-empty set of names such that $x \notin \Delta$).

Composite beta binders are generated by the following BNF-like grammar:

$$\mathbf{B} ::= \beta(x : \Delta) \;\Big|\; \beta^h(x : \Delta) \;\Big|\; \beta(x : \Delta)\,\mathbf{B} \;\Big|\; \beta^h(x : \Delta)\,\mathbf{B} \;.$$

Pi-processes, which are referred to with this name because of their similarity to π-calculus processes, are generated by the following BNF-like grammar:

$$P \;::=\; \mathsf{nil} \;\Big|\; \pi.\,P \;\Big|\; P|P \;\Big|\; \nu y\, P \;\Big|\; !\,P$$

where $\pi ::= c \,\big|\, b$, with $c ::= x(w) \,\big|\, \overline{x}y$ and $b ::= \mathsf{expose}(x,\,\Delta) \,\big|\, \mathsf{hide}(x) \,\big|\, \mathsf{unhide}(x)$.

Input $x(w)$, restriction νy and $\mathsf{expose}(x,\,\Delta)$ act as binders.

Bio-processes, which realize the encapsulation of pi-processes into boxes whose interfaces consist of composite beta binders, are generated by the following BNF-like grammar:

$$B \;::=\; \mathsf{Nil} \;\Big|\; \mathbf{B}[\;P\;] \;\Big|\; B \parallel B \;\;.$$

2.2 Semantics

The semantics of bio-processes is given in [10] in terms of a *reduction relation* ($\longrightarrow$), which uses a *structural congruence relation* ($\equiv$).

We postpone the formal definitions of these relations to the introduction of the labeling on processes in next section. For their standard definitions, see [10].

3 Labeled Semantics of *Beta-binders*

We define $\vartheta \in \{\|_0, \|_1\}^*$ and $\varphi \in \{|_0, |_1, !\}^*$, and we use them to label bio-processes and pi-processes respectively. Hence, in the syntactic definition of pi-processes and bio-processes, we replace each process of the form $\pi.P$ with a labeled process $\varphi\pi.P$ (where φ provides a linear encoding of the syntactical location of the sub-tree of $\pi.P$ in the syntax tree of the whole pi-process within a bio-process). Moreover we replace each bio-process $\mathbf{B}[\;P\;]$ with $\vartheta\mathbf{B}[\;P\;]$. Intuitively ϑ encodes the parallel structure of bio-processes, while φ encodes the parallel structure of pi-processes taking care of replications as well. For instance, the bio-process $\beta(x:\Gamma)\,\beta^h(y:\Delta)\,[\;P_0|P_1\;] \parallel \beta(z:\Delta)\,[\;Q_0|Q_1\;]$ is mapped to $\|_0\,\beta(x:\Gamma)\,\beta^h(y:\Delta)\,[\;|_0P_0||_1P_1\;] \parallel \|_1\,\beta(z:\Delta)\,[\;|_0Q_0||_1Q_1\;]$.

The set θ of the labels of the transitions is defined by the following BNF-like grammar:

$$\begin{aligned}\theta \;\;::=\;\; & \vartheta\varphi\mu \;\Big|\; \vartheta\rho \;\Big|\; \vartheta\varphi\langle\varphi_0 x(w), \varphi_1\overline{x}z\rangle \;\Big|\; \vartheta\langle\|_i\vartheta_0\varphi_0{}'x(w)`, \|_{1-i}\vartheta_1\varphi_1{}'\overline{y}z`\rangle \;\Big| \\ & \vartheta\,\langle\|_0\vartheta_0\mathsf{join}\,P_0, \|_1\vartheta_1\mathsf{join}\,P_1\rangle\end{aligned}$$

where $\mu ::= c \,\big|\, i \,\big|\, b$ with $i ::= {}'x(w)` \,\big|\, {}'\overline{x}y`$, and $\rho ::= \mathsf{split}\langle P_0, P_1\rangle \,\big|\, \mathsf{join}\,P$.

In the above, the first pair of labels is used to denote intra-communications (communications within one bio-process), while the second one is used to denote inter-communications (communications between different bio-processes) and the last one is used to denote *join* actions. Note that the definition of i allows us to distinguish between the input/output actions used in internal communications ($x(w)$ / $\overline{x}y$) and the ones used in inter-communications (${}'x(w)`$ / ${}'\overline{x}y`$).

According to the definition of binders, y is a bound name in $x(y)$, in $'x(y)`$ and in $\mathsf{expose}(y,\, \Delta)$.

We introduce a set of labels with metavariable δ that will be useful in the following:

$$\delta ::= \varphi\mu \;\big|\; \varphi\langle\varphi_0 x(w), \varphi_1 \overline{x} z\rangle \ .$$

Definition 1 Structural congruence *over pi-processes, denoted by* $\equiv$*, is the smallest relation which satisfies the laws in Table 1 (a).* Structural congruence *over bio-processes, denoted by* $\equiv$*, is the smallest relation which satisfies the laws in Table 1 (b) where* $\hat{\beta}$ *ranges over* $\{\beta, \beta^h\}$.

Table 1: Laws for structural congruence.

(a) Pi-processes	(b) Bio-processes
$P_1 \equiv P_2$ provided P_1 is an α-converse of P_2	$\mathbf{B}[\,P_1\,] \equiv \mathbf{B}[\,P_2\,]$ provided $P_1 \equiv P_2$
$P_1\vert(P_2\vert P_3) \equiv (P_1\vert P_2)\vert P_3$	$B_1 \parallel (B_2 \parallel B_3) \equiv (B_1 \parallel B_2) \parallel B_3$
$P_1\vert P_2 \equiv P_2\vert P_1$	$B_1 \parallel B_2 \equiv B_2 \parallel B_1$
$P\vert\,\mathsf{nil} \equiv P$	$B \parallel \mathsf{Nil} \equiv B$
$\nu z\, \nu w\, P \equiv \nu w\, \nu z\, P$	$\mathbf{B}_1\mathbf{B}_2[\,P\,] \equiv \mathbf{B}_2\mathbf{B}_1[\,P\,]$
$\nu z\ \mathsf{nil} \equiv \mathsf{nil}$	$\mathbf{B}^*\hat{\beta}(x:\Gamma)[\,P\,] \equiv \mathbf{B}^*\hat{\beta}(y:\Gamma)[\,P\{y/x\}\,]$
$\nu z\,(P_1\vert P_2) \equiv P_1\vert\nu z\, P_2$ provided $z \notin \mathsf{fn}(P_1)$	provided y fresh in the system

Definition 2 *The* reduction relation $\longrightarrow$ *is the smallest relation over bio-processes obtained by applying the axioms and rules in Table 2.*

4 Causality, Enabling and Concurrency

We first introduce two auxiliary functions, l and $subj$, to simplify the presentation of our subsequent treatment. For each action μ and ρ, the function l specifies its type, while the function $subj$ specifies the name it operates on (i.e. for input/output actions it is the name of the channel on which it receives/sends, while for *expose*/*hide*/*unhide* actions it is the subject of the beta binder).

$$\begin{array}{ll}
l(x(w)) = \mathsf{in} & l('x(w)`) = \mathsf{in_inter} \\
l(\overline{x}w) = \mathsf{out} & l('\overline{x}w`) = \mathsf{out_inter} \\
l(\mathsf{expose}(x,\, \Delta)) = \mathsf{expose} & l(\mathsf{join}\ P) = \mathsf{join} \\
l(\mathsf{hide}(x)) = \mathsf{hide} & l(\mathsf{split}\langle P_0, P_1\rangle) = \mathsf{split} \\
l(\mathsf{unhide}(x)) = \mathsf{unhide} &
\end{array}$$

$$\begin{array}{l}
subj(x(w)) = subj(\overline{x}w) = subj('x(w)`) = subj('\overline{x}w`) = \{x\} \\
subj(\mathsf{expose}(x,\, \Delta)) = subj(\mathsf{hide}(x)) = subj(\mathsf{unhide}(x)) = \{x\} \\
subj(\mathsf{join}\ P) = subj(\mathsf{split}\langle P_0, P_1\rangle) = \bot
\end{array}$$

Hereafter, we assume that all the bound names in the system are not used except for their bound occurrences. In general it is possible to modify any system to satisfy this constraint by applying α-conversion. We also

Table 2: Axioms and rules for the reduction relation.

(intra)
$$\frac{P \equiv \nu\tilde{u}\,(\varphi\varphi_0 x(w).\,P_0|\varphi\varphi_1\overline{x}z.\,P_1|P_2)}{\vartheta\mathbf{B}[\ P\ ] \xrightarrow{\vartheta\varphi\langle\varphi_0 x(w),\varphi_1\overline{x}z\rangle} \vartheta\mathbf{B}[\ \nu\tilde{u}\,(\varphi\varphi_0 P_0\{z/w\}|\varphi\varphi_1 P_1|P_2)\ ]}$$

(inter)
$$\frac{P \equiv \nu\tilde{u}\,(\varphi_0 x(w).\,P_1|P_2) \qquad Q \equiv \nu\tilde{v}\,(\varphi_1\overline{y}z.\,Q_1|Q_2)}{X \xrightarrow{\vartheta\langle\|_i\vartheta_0\varphi_0{}'x(w)`,\|_{1-i}\vartheta_1\varphi_1{}'\overline{y}z`} Y}$$

where $X = \vartheta\|_i\vartheta_0\,\beta(x:\Gamma)\,\mathbf{B}_0^*[\ P\ ]\ \|\ \vartheta\|_{1-i}\vartheta_1\,\beta(y:\Delta)\,\mathbf{B}_1^*[\ Q\ ]$,
$Y = \vartheta\|_i\vartheta_0\,\beta(x:\Gamma)\,\mathbf{B}_0^*[\ P'\ ]\ \|\ \vartheta\|_{1-i}\vartheta_1\,\beta(y:\Delta)\,\mathbf{B}_1^*[\ Q'\ ]$,
$P' = \nu\tilde{u}\,(\varphi_0 P_1\{z/w\}|P_2)$ and $Q' = \nu\tilde{v}\,(\varphi_1 Q_1|Q_2)$
provided $\Gamma\cap\Delta\neq\emptyset$ and $x,z\notin\tilde{u}$ and $y,z\notin\tilde{v}$

(expose)
$$\frac{P \equiv \nu\tilde{u}\,(\varphi\,\mathsf{expose}(x,\,\Gamma)\,.\,P_1|P_2)}{\vartheta\mathbf{B}[\ P\ ] \xrightarrow{\vartheta\varphi\,\mathsf{expose}(x,\Gamma)} \vartheta\mathbf{B}\,\beta(y:\Gamma)\,[\ \nu\tilde{u}\,(\varphi P_1\{y/x\}|P_2)\ ]}$$
provided y fresh in the system

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

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

(join)
$$\vartheta\|_0\vartheta_0\mathbf{B}_0[\ P_0\ ]\ \|\ \vartheta\|_1\vartheta_1\mathbf{B}_1[\ P_1\ ] \xrightarrow{\vartheta\,\langle\|_0\vartheta_0\mathsf{join}\,P_0,\|_1\vartheta_1\mathsf{join}\,P_1\rangle} Y$$
where $Y = \vartheta\|_0\vartheta_0\mathbf{B}[\ |_0P_0\sigma_0||_1P_1\sigma_1\ ]\ \|\ \vartheta\|_1\vartheta_1\,\mathsf{Nil}$
provided that f_{join} is defined in $(\mathbf{B}_0,\mathbf{B}_1,P_0,P_1)$
and with $f_{join}(\mathbf{B}_0,\mathbf{B}_1,P_0,P_1) = (\mathbf{B},\sigma_0,\sigma_1)$

(split)
$$\vartheta\mathbf{B}[\ P_0|P_1\ ] \xrightarrow{\vartheta\,\mathsf{split}\langle P_0,P_1\rangle} \vartheta\|_0\mathbf{B}_0[\ P_0\sigma_0\ ]\ \|\ \vartheta\|_1\mathbf{B}_1[\ P_1\sigma_1\ ]$$
provided that f_{split} is defined in $(\mathbf{B},P_0,P_1)$
and with $f_{split}(\mathbf{B},P_0,P_1) = (\mathbf{B}_0,\mathbf{B}_1,\sigma_0,\sigma_1)$

(bang)
$$\frac{\vartheta\mathbf{B}[\ P|Q\ ] \xrightarrow{\theta} \vartheta\mathbf{B}'[\ P'|Q'\ ]}{\vartheta\mathbf{B}[\ !P|Q\ ] \xrightarrow{!\theta} \vartheta\mathbf{B}'[\ !P|P'|Q'\ ]}$$

(redex)
$$\frac{B \xrightarrow{\theta} B'}{B\ \|\ B'' \xrightarrow{\theta} B'\ \|\ B''}$$

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

assume that all the names in the system are marked by an index and that at the beginning of the computation its value is 0. Moreover, the new name introduced by the *expose* operation is the same as the bound name in the primitive with the index incremented by one (e.g. $\mathbf{B}[\ \mathsf{expose}(x_n,\ \Delta)\,.P\ ] \longrightarrow \mathbf{B}\,\beta(x_{n+1} : \Delta)\ [\ P\{x_{n+1}/x_n\}\ ]$).

4.1 Causal Relation

Now we can define the causal relation between pairs of transitions in a computation. Recall that an activity A *causes* an activity B if A influences the execution of B. Our labels allow us to use them as unique names for the transitions as they are linearizations encoding the position of the prefixes and processes originating the transitions in the syntax tree.

Definition 3 (Immediate causal relation) *Given a computation* $B_0 \xrightarrow{\theta_0} B_1 \xrightarrow{\theta_1} \cdots \xrightarrow{\theta_n} B_{n+1}$*, we say that* θ_n *immediately depends on* θ_h *(or, symmetrically,* θ_h *immediately causes* θ_n*) if* $h < n$ *and* $\theta_h \lessdot \theta_n$ *can be derived by repeated applications of the following rules, where* $i, j \in \{0, 1\}$*.*

1. $\|_i\theta \lessdot \|_i\theta'$ *if* $\theta \lessdot \theta'$
2. $|_i\delta \lessdot |_i\delta'$ *if* $\delta \lessdot \delta'$
3. $!\delta \lessdot |_0\delta'$
4. $!\delta \lessdot |_1\delta'$ *if* $\delta \lessdot \delta'$
5. $\theta \lessdot \langle\theta'_0, \theta'_1\rangle$ *if* $\exists i.\theta \lessdot \theta'_i$
6. $\langle\theta_0, \theta_1\rangle \lessdot \theta'$ *if* $\exists i.\theta_i \lessdot \theta'$
7. $\langle\theta_0, \theta_1\rangle \lessdot \langle\theta'_0, \theta'_1\rangle$ *if* $\exists i, j.\theta_i \lessdot \theta'_j$
8. $\mu \lessdot \varphi\mu'$ *if* $(l(\mu) = \mathsf{in} \ \vee\ l(\mu) = \mathsf{in\_inter} \ \vee\ l(\mu) = \mathsf{expose})$
9. $\varphi\mu \lessdot \varphi'\mu'$ *if* $\quad (subj(\mu) = subj(\mu') \wedge ((l(\mu) = \mathsf{hide} \wedge l(\mu') = \mathsf{unhide}) \vee (l(\mu) = \mathsf{unhide} \wedge l(\mu') = \mathsf{hide}) \vee (l(\mu) = \mathsf{unhide} \wedge (l(\mu') = \mathsf{in\_inter} \vee l(\mu') = \mathsf{out\_inter}))))$
10. $\varphi\mu \lessdot \rho$
11. $\rho \lessdot \varphi\mu$
12. $\rho \lessdot \vartheta\rho$

The rules listed above are applied recursively to a pair of actions θ_h, θ_n in order to verify if there is a structural dependency between them.

The recursive step is implemented by removing the common prefixes of θ_h and θ_n through rules 1 and 2. First, rule 1 is applied since it concerns the labels of the bio-processes; then, if θ_h and θ_n refer to the same bio-process, rule 2 is applied. Note that for π-calculus only one such rule is defined [6], while for *Beta-binders* two levels of recursion are needed to take into account the parallel structure of both bio-processes and pi-processes.

Rules 3 and 4 take into account the replication operator and are analogous to the ones for the π-calculus, as described in [4].

Rules 5, 6 and 7 state that a coupled action (a communication or a *join*) is caused by/causes another action if one of the two partners of the coupled action is caused by/causes the other action.

Rules 8, 9, 10, 11 and 12 are applied at the end of the recursive steps and are peculiar to *Beta-binders*, since they are relative to the operators which are peculiar in *Beta-binders*.

Rule 8 describes the causal dependency imposed by the sequential structure of the pi-processes: an action $\theta_h = \vartheta\varphi\mu$ causes an action θ_n whose label has $\vartheta\varphi$ as a prefix if θ_h is a binder (i.e. if θ_h is an input action (either in an internal or in an inter-communication) or an *expose* operation). The idea which lead to the definition of this rule is that all binder operators cause a flow of information to their suffix, so they must cause them. The third case, the *expose* operator, is peculiar to *Beta-binders*: since it introduces a new beta binder on the interface of the process, the rest of the process is considered to be necessarily caused by it.

Rule 9 describes how the operations on the interfaces influence each other: an *hide* causes an *unhide* of the same beta binder; an *unhide* causes an *hide* of the same beta binder or an inter-communication on it; this causal dependency holds both if the address label of θ_h is a prefix of the one of θ_n and if they just refer to the same bio-process.

Rules 10, 11 and 12 are relative to the causal relation between actions happening within a bio-process and join/split functions involving that bio-process.

Rule 10 states that an operation that involves one of the bio-processes later merged by a *join* or divided by a *split* causes the execution of that function; for example the introduction of a new binder in the interface of a bio-process can be fundamental for the subsequent application of f_{join} and f_{split} functions.

Rule 11 and 12 state that *join* and *split* operations cause all the operations that involve the bio-processes obtained after their execution (communications, operations on the interfaces, *join* and *split* operations).

The definition of the causal relation between two transitions of a computation is obtained by taking into account the transitive closure of the immediate causal relation.

Definition 4 (Causal relation) *Let $< \triangleq (\lessdot)^*$ be the transitive closure of $\lessdot$. Then, given a computation $B_0 \xrightarrow{\theta_0} B_1 \xrightarrow{\theta_1} \cdots \xrightarrow{\theta_n} B_{n+1}$, we say that θ_n depends on θ_h (or, symmetrically, θ_h causes θ_n) if $\theta_h < \theta_n$.*

4.2 Enabling Relation

In this section we define the enabling relation between pairs of transitions in a computation. Recall that an activity A *enables* an activity B if either

A is a necessary condition for the execution of B or A cannot be executed after B.

Definition 5 (Immediate enabling relation) *Given a computation* $B_0 \xrightarrow{\theta_0} B_1 \xrightarrow{\theta_1} \cdots \xrightarrow{\theta_n} B_{n+1}$*, we say that* θ_n *is immediately enabled by* θ_h *(or, symmetrically,* θ_h *immediately enables* θ_n*) if* $h < n$ *and* $\theta_h \lll \theta_n$ *can be derived by repeated applications of the following rules, where* $i, j \in \{0, 1\}$*.*

1. $\|_i\theta \lll \|_i\theta'$ *if* $\theta \lll \theta'$
2. $|_i\delta \lll |_i\delta'$ *if* $\delta \lll \delta'$
3. $!\delta \lll |_0\delta'$
4. $!\delta \lll |_1\delta'$ *if* $\delta \lll \delta'$
5. $\theta \lll \langle\theta'_0, \theta'_1\rangle$ *if* $\exists i.\theta \lll \theta'_i$
6. $\langle\theta_0, \theta_1\rangle \lll \theta'$ *if* $\exists i.\theta_i \lll \theta'$
7. $\langle\theta_0, \theta_1\rangle \lll \langle\theta'_0, \theta'_1\rangle$ *if* $\exists i, j.\theta_i \lll \theta'_j$
8. $\mu \lll \varphi\mu'$
9. $\varphi i \lll \varphi' b$ *if* $(l(b) = \mathsf{hide} \wedge subj(i) = subj(b))$
10. $\varphi\mu \lll \rho$

Rules 1-7 are analogous to rules 1-7 in Def. 3 for causality.

Rule 8 describes the temporal dependency imposed by the sequential structure of the pi-processes: every action $\theta_h = \vartheta\varphi\mu$ causes an action θ_n whose label has $\vartheta\varphi$ as a prefix. This rule is a generalization of the respective one for causality, without the constraint on the type of the action μ.

Rules 9 and 10 are peculiar to *Beta-binders*.

Rule 9 describes the temporal dependency between an inter-communication and an *hide* operation on the same beta binder: in fact it is not possible to execute the inter-communication after the beta binder is hidden.

Rule 10 is analogous to rule 10 in Def. 3 for causality.

The definition of the enabling relation between two transitions of a computation is obtained by taking into account the transitive closure of the immediate enabling relation.

Definition 6 (Enabling relation) *Let* $\ll \triangleq (\lll)^*$ *be the transitive closure of* $\lll$*. Then, given a computation* $B_0 \xrightarrow{\theta_0} B_1 \xrightarrow{\theta_1} \cdots \xrightarrow{\theta_n} B_{n+1}$*, we say that* θ_n *is enabled by* θ_h *(or, symmetrically,* θ_h *enables* θ_n*) if* $\theta_h \ll \theta_n$*.*

We decided to keep the relation of causality and that of enabling distinct because they are logically different and, moreover, to be consistent with the analogous definitions for π-calculus proposed in the literature. In the following part, however, we will merge them in a single relation (their union) and we will only consider the latter.

The following propositions show that causality and enabling are distinct relations.

Proposition 1 $< \not\subseteq \ll$.

Proof. Consider $\theta_0 = \vartheta\varphi\,\mathsf{hide}(x)$ and $\theta_1 = \vartheta\varphi'\,\mathsf{unhide}(x)$. It is $\theta_0 < \theta_1$ but $\theta_0 \not\ll \theta_1$.

Proposition 2 $\ll \not\subseteq <$.

Proof. Consider $\theta_0 = \vartheta\varphi\,\mathsf{hide}(x)$ and $\theta_1 = \vartheta\varphi'\langle\varphi_0 y(w), \varphi_1\overline{y}z\rangle$, where $\varphi = \varphi'\varphi_0$. It is $\theta_0 \ll \theta_1$ but $\theta_0 \not< \theta_1$.

4.3 Concurrency Relation

Recall that two activities A and B are *concurrent* if they can be executed in parallel.

Definition 7 (Concurrency relation) *Let $\prec \triangleq (< \cup \ll)^*$. Then, given a system B, we say that θ_n and θ_h are concurrent (i.e. they can be executed simultaneously, written $\theta_n \smile \theta_h$) if $\forall$ computation $\xi = B \longrightarrow^* B'$ s.t. $\theta_n, \theta_h \in \xi$. $\theta_n \nprec \theta_h$ and $\theta_h \nprec \theta_n$.*

5 Properties of the Concurrency Relation

In this sections we state two lemmas and two corollaries and one theorem derived from them.

The first lemma states that if two consecutive transitions in a computation are concurrent, then they form a diamond in the proved transition system.

Lemma 1 *Given a computation $B_0 \xrightarrow{\theta_0} B_1 \xrightarrow{\theta_1} B_2$, if $\theta_0 \nprec \theta_1 \Rightarrow \exists B_1'.B_0 \xrightarrow{\theta_1} B_1' \xrightarrow{\theta_0} B_2$. In other words, the following diagram exists:*

$$\begin{tikzcd}
 & B_0 \arrow[dl, "\theta_0"'] \arrow[dr, "\theta_1"] & \\
B_1 \arrow[dr, "\theta_1"'] & & B_1' \arrow[dl, "\theta_0"] \\
 & B_2 &
\end{tikzcd}$$

Proof. We have by hypothesis that $B_0 \xrightarrow{\theta_0} B_1 \xrightarrow{\theta_1} B_2$ and $\theta_0 \nprec \theta_1$ (in particular, since θ_0 and θ_1 are consecutive transactions, we have that $\theta_0 \not< \theta_1$ and $\theta_0 \not\ll \theta_1$).

The proof is done by cases on the labels of the transitions.

1. $\theta_0 = \vartheta\varphi\,\mathsf{expose}(x, \Delta)$:

(a) $\theta_1 = \vartheta'\varphi'\,\mathsf{expose}(y,\Gamma)$: by Def. 3, $\theta_0 \lessdot \theta_1$ iff $\vartheta\varphi$ is a prefix of $\vartheta'\varphi'$ (rule 8); by Def. 5, $\theta_0 \lll \theta_1$ iff $\vartheta\varphi$ is a prefix of $\vartheta'\varphi'$ (rule 8); by hypothesis $\theta_0 \not\lessdot \theta_1$ and $\theta_0 \not\lll \theta_1$, hence $\vartheta\varphi$ is not a prefix of $\vartheta'\varphi'$; if $\vartheta \neq \vartheta'$, then θ_0 and θ_1 refer to different bio-processes, so it is possible to exchange their order without any consequence; if, otherwise, $\vartheta = \vartheta'$ and φ is not a prefix of φ', then θ_0 and θ_1 refer to the same bio-process but to different pi-processes, so they can be exchanged as well (recall that we have assumed that $x \neq y$, so the two *expose* operations do not influence each other);

(b) $\theta_1 = \vartheta'\varphi'\,\mathsf{hide}(y)$: by Def. 3, $\theta_0 \lessdot \theta_1$ iff $\vartheta\varphi$ is a prefix of $\vartheta'\varphi'$ (rule 8); by Def. 5, $\theta_0 \lll \theta_1$ iff $\vartheta\varphi$ is a prefix of $\vartheta'\varphi'$ (rule 8); hence $\vartheta\varphi$ is not a prefix of $\vartheta'\varphi'$, so θ_0 and θ_1 refer to different beta binders (recall that x is only available for actions with the same prefix as θ_0, so $x \neq y$); so they can be exchanged;

(c) $\theta_1 = \vartheta'\varphi'\,\mathsf{unhide}(y)$: by Def. 3, $\theta_0 \lessdot \theta_1$ iff $\vartheta\varphi$ is a prefix of $\vartheta'\varphi'$ (rule 8); by Def. 5, $\theta_0 \lll \theta_1$ iff $\vartheta\varphi$ is a prefix of $\vartheta'\varphi'$ (rule 8); hence $\vartheta\varphi$ is not a prefix of $\vartheta'\varphi'$, so θ_0 and θ_1 refer to different beta binders (again, x is only available for actions with the same prefix as θ_0, so $x \neq y$); so they can be exchanged;

(d) $\theta_1 = \vartheta'\varphi'\langle\varphi_0 y(t), \varphi_1\overline{y}z\rangle$: by Def. 3, $\theta_0 \lessdot \theta_1$ iff $\vartheta\varphi$ is a prefix of $\vartheta'\varphi'\varphi_0$ or of $\vartheta'\varphi'\varphi_1$ (rule 8); by Def. 5, $\theta_0 \lll \theta_1$ iff $\vartheta\varphi$ is a prefix of $\vartheta'\varphi'\varphi_0$ or of $\vartheta'\varphi'\varphi_1$ (rule 8); hence $\vartheta\varphi$ is a prefix of none of them, so θ_0 and θ_1 can be exchanged since *expose* operations and internal communications do not influence each other;

(e) $\theta_1 = \vartheta'\langle\|_i\vartheta_0\varphi_0 y(t), \|_{1-i}\vartheta_1\varphi_1\overline{w}z\rangle$: by Def. 3, $\theta_0 \lessdot \theta_1$ iff $\vartheta\varphi$ is a prefix of $\vartheta'\|_i\vartheta_0\varphi_0$ or of $\vartheta'\|_{1-i}\vartheta_1\varphi_1$ (rule 8); by Def. 5, $\theta_0 \lll \theta_1$ iff $\vartheta\varphi$ is a prefix of $\vartheta'\|_i\vartheta_0\varphi_0$ or of $\vartheta'\|_{1-i}\vartheta_1\varphi_1$ (rule 8); hence $\vartheta\varphi$ is a prefix of none of them, so θ_0 and θ_1 do not influence each other since they refer to different beta binders (again, x is only available for actions with the same prefix as θ_0, so $x \neq y$ and $x \neq w$); so they can be exchanged;

(f) $\theta_1 = \vartheta'\,\langle\|_0\vartheta_0\mathsf{join}\,Q_0, \|_1\vartheta_1\mathsf{join}\,Q_1\rangle$: by Def. 3, $\theta_0 \lessdot \theta_1$ iff $\vartheta = \vartheta'\|_0\vartheta_0$ or $\vartheta = \vartheta'\|_1\vartheta_1$ (rule 10); by Def. 5, $\theta_0 \lll \theta_1$ iff $\vartheta = \vartheta'\|_0\vartheta_0$ or $\vartheta = \vartheta'\|_1\vartheta_1$ (rule 10); hence θ_0 and θ_1 refer to different bio-processes, so they can be exchanged;

(g) $\theta_1 = \vartheta'\,\mathsf{split}\langle Q_0, Q_1\rangle$: by Def. 3, $\theta_0 \lessdot \theta_1$ iff $\vartheta = \vartheta'$ (rule 10); by Def. 5, $\theta_0 \lll \theta_1$ iff $\vartheta = \vartheta'$ (rule 10); hence θ_0 and θ_1 refer to different bio-processes, so they can be exchanged;

2. $\theta_0 = \vartheta\varphi\,\mathsf{hide}(x)$:

(a) $\theta_1 = \vartheta'\varphi'\,\mathsf{expose}(y,\Delta)$: by Def. 5, $\theta_0 \lll \theta_1$ iff $\vartheta\varphi$ is a prefix of $\vartheta'\varphi'$ (rule 8); hence $\vartheta\varphi$ is not a prefix of $\vartheta'\varphi'$, so θ_0 and θ_1 refer to different beta binders (again, y is only available for actions with the same prefix as θ_1, so $x \neq y$); so they can be exchanged;

(b) $\theta_1 = \vartheta'\varphi'\,\mathsf{hide}(y)$: by Def. 5, $\theta_0 \lll \theta_1$ iff $\vartheta\varphi$ is a prefix of $\vartheta'\varphi'$ (rule 8); hence $\vartheta\varphi$ is not a prefix of $\vartheta'\varphi'$; moreover, $x \neq y$ (since it is not possible to execute two *hide* operations on the same beta binder consecutively), so they can be exchanged;

(c) $\theta_1 = \vartheta'\varphi'\,\mathsf{unhide}(y)$: by Def. 3, $\theta_0 \lessdot \theta_1$ iff $\vartheta = \vartheta'$ and $x = y$ (rule 9); by Def. 5, $\theta_0 \lll \theta_1$ iff $\vartheta\varphi$ is a prefix of $\vartheta'\varphi'$ (rule 8); hence $\vartheta\varphi$ is not a prefix of $\vartheta'\varphi'$ and $x \neq y$, so they can be exchanged, since the two operations do not influence each other;

(d) $\theta_1 = \vartheta'\varphi'\langle\varphi'_0 y(t), \varphi'_1 \overline{y}u\rangle$: by Def. 5, $\theta_0 \lll \theta_1$ iff $\vartheta\varphi$ is a prefix of $\vartheta'\varphi'\varphi'_0$ or of $\vartheta'\varphi'\varphi'_1$ (rule 8); hence $\vartheta\varphi$ is a prefix of none of them, so θ_0 and θ_1 can be exchanged since *hide* operations and internal communications do not influence each other;

(e) $\theta_1 = \vartheta'\langle\|_i\vartheta_0\varphi'_0 y(t), \|_{1-i}\vartheta_1\varphi'_1\overline{v}u\rangle$: by Def. 5, $\theta_0 \lll \theta_1$ iff $\vartheta\varphi$ is a prefix of $\vartheta'\|_i\vartheta_0\varphi'_0$ or of $\vartheta'\|_{1-i}\vartheta_1\varphi'_1$ (rule 8); hence $\vartheta\varphi$ is a prefix of none of them; moreover, $x \neq y$ and $x \neq v$ (since it is not possible to execute an inter-communication immediately after an *hide* operation on the same beta binder), so they can be exchanged;

(f) $\theta_1 = \vartheta'\,\langle\|_0\vartheta_0\mathsf{join}\,Q_0, \|_1\vartheta_1\mathsf{join}\,Q_1\rangle$: by Def. 3, $\theta_0 \lessdot \theta_1$ iff $\vartheta = \vartheta'\|_0\vartheta_0$ or $\vartheta = \vartheta'\|_1\vartheta_1$ (rule 10); by Def. 5, $\theta_0 \lll \theta_1$ iff $\vartheta = \vartheta'\|_0\vartheta_0$ or $\vartheta = \vartheta'\|_1\vartheta_1$ (rule 10); hence θ_0 and θ_1 refer to different bio-processes, so they can be exchanged;

(g) $\theta_1 = \vartheta'\,\mathsf{split}\langle Q_0, Q_1\rangle$: by Def. 3, $\theta_0 \lessdot \theta_1$ iff $\vartheta = \vartheta'$ (rule 10); by Def. 5, $\theta_0 \lll \theta_1$ iff $\vartheta = \vartheta'$ (rule 10); hence θ_0 and θ_1 refer to different bio-processes, so they can be exchanged;

3. $\theta_0 = \vartheta\varphi\,\mathsf{unhide}(x)$:

(a) $\theta_1 = \vartheta'\varphi'\,\mathsf{expose}(y, \Delta)$: by Def. 5, $\theta_0 \lll \theta_1$ iff $\vartheta\varphi$ is a prefix of $\vartheta'\varphi'$ (rule 8); hence $\vartheta\varphi$ is not a prefix of $\vartheta'\varphi'$, so θ_0 and θ_1 refer to different beta binders (again, y is only available for actions with the same prefix as θ_1, so $x \neq y$); so they can be exchanged;

(b) $\theta_1 = \vartheta'\varphi'\,\mathsf{hide}(y)$: by Def. 3, $\theta_0 \lessdot \theta_1$ iff $\vartheta = \vartheta'$ and $x = y$ (rule 9); by Def. 5, $\theta_0 \lll \theta_1$ iff $\vartheta\varphi$ is a prefix of $\vartheta'\varphi'$ (rule 8); hence $\vartheta\varphi$ is not a prefix of $\vartheta'\varphi'$ and $x \neq y$, so they can be exchanged, since the two operations do not influence each other;

(c) $\theta_1 = \vartheta'\varphi'\,\mathsf{unhide}(y)$: by Def. 5, $\theta_0 \lll \theta_1$ iff $\vartheta\varphi$ is a prefix of $\vartheta'\varphi'$ (rule 8); hence $\vartheta\varphi$ is not a prefix of $\vartheta'\varphi'$; moreover, $x \neq y$ (since it is not possible to execute two *unhide* operations on the same beta binder consecutively), so they can be exchanged;

(d) $\theta_1 = \vartheta'\varphi'\langle\varphi'_0 y(t), \varphi'_1 \overline{y}u\rangle$: by Def. 5, $\theta_0 \lll \theta_1$ iff $\vartheta\varphi$ is a prefix of $\vartheta'\varphi'\varphi'_0$ or of $\vartheta'\varphi'\varphi'_1$ (rule 8); hence $\vartheta\varphi$ is a prefix of none of them, so θ_0 and θ_1 can be exchanged since *unhide* operations and internal communications do not influence each other;

(e) $\theta_1 = \vartheta'\langle\|_i\vartheta_0\varphi'_0 y(t), \|_{1-i}\vartheta_1\varphi'_1\overline{v}u\rangle$: by Def. 3, $\theta_0 \lessdot \theta_1$ iff ($\vartheta = \vartheta'\|_i\vartheta_0$ and $x = y$) or ($\vartheta = \vartheta'\|_{1-i}\vartheta_1$ and $x \neq v$) (rule 9); by Def. 5, $\theta_0 \lll \theta_1$ iff $\vartheta\varphi$ is a prefix of $\vartheta'\|_i\vartheta_0\varphi'_0$ or of $\vartheta'\|_{1-i}\vartheta_1\varphi'_1$ (rule 8); hence $\vartheta\varphi$ is a prefix of none of them and θ_0 and θ_1 refer to different beta binders, so they can be exchanged;

(f) $\theta_1 = \vartheta' \langle \|_0 \vartheta_0 \mathsf{join}\, Q_0, \|_1 \vartheta_1 \mathsf{join}\, Q_1 \rangle$: by Def. 3, $\theta_0 \lessdot \theta_1$ iff $\vartheta = \vartheta' \|_0 \vartheta_0$ or $\vartheta = \vartheta' \|_1 \vartheta_1$ (rule 10); by Def. 5, $\theta_0 \lll \theta_1$ iff $\vartheta = \vartheta' \|_0 \vartheta_0$ or $\vartheta = \vartheta' \|_1 \vartheta_1$ (rule 10); hence θ_0 and θ_1 refer to different bio-processes, so they can be exchanged;

(g) $\theta_1 = \vartheta' \mathsf{split} \langle Q_0, Q_1 \rangle$: by Def. 3, $\theta_0 \lessdot \theta_1$ iff $\vartheta = \vartheta'$ (rule 10); by Def. 5, $\theta_0 \lll \theta_1$ iff $\vartheta = \vartheta'$ (rule 10); hence θ_0 and θ_1 refer to different bio-processes, so they can be exchanged;

4. $\theta_0 = \vartheta \varphi \langle \varphi_0 x(w), \varphi_1 \overline{x} z \rangle$:

   (a) $\theta_1 = \vartheta' \varphi' \mathsf{expose}(y, \Delta)$: by Def. 3, $\theta_0 \lessdot \theta_1$ iff $\vartheta \varphi \varphi_0$ is a prefix of $\vartheta' \varphi'$ (rule 8); by Def. 5, $\theta_0 \lll \theta_1$ iff either $\vartheta \varphi \varphi_0$ or $\vartheta \varphi \varphi_1$ is a prefix of $\vartheta' \varphi'$ (rule 8); hence θ_0 and θ_1 can be exchanged, since the two operations do not influence each other;

   (b) $\theta_1 = \vartheta' \varphi' \mathsf{hide}(y)$: by Def. 3, $\theta_0 \lessdot \theta_1$ iff $\vartheta \varphi \varphi_0$ is a prefix of $\vartheta' \varphi'$ (rule 8); by Def. 5, $\theta_0 \lll \theta_1$ iff either $\vartheta \varphi \varphi_0$ or $\vartheta \varphi \varphi_1$ is a prefix of $\vartheta' \varphi'$ (rule 8); hence θ_0 and θ_1 can be exchanged;

   (c) $\theta_1 = \vartheta' \varphi' \mathsf{unhide}(y)$: by Def. 3, $\theta_0 \lessdot \theta_1$ iff $\vartheta \varphi \varphi_0$ is a prefix of $\vartheta' \varphi'$ (rule 8); by Def. 5, $\theta_0 \lll \theta_1$ iff either $\vartheta \varphi \varphi_0$ or $\vartheta \varphi \varphi_1$ is a prefix of $\vartheta' \varphi'$ (rule 8); hence θ_0 and θ_1 can be exchanged;

   (d) $\theta_1 = \vartheta' \varphi' \langle \varphi'_0 y(t), \varphi'_1 \overline{y} u \rangle$: by Def. 3, $\theta_0 \lessdot \theta_1$ iff $\vartheta \varphi \varphi_0$ is a prefix of $\vartheta' \varphi' \varphi'_0$ or of $\vartheta' \varphi' \varphi'_1$ (rule 8); by Def. 5, $\theta_0 \lll \theta_1$ iff either $\vartheta \varphi \varphi_0$ or $\vartheta \varphi \varphi_1$ is a prefix of $\vartheta' \varphi' \varphi'_0$ or of $\vartheta' \varphi' \varphi'_1$ (rule 8); hence θ_0 and θ_1 can be exchanged;

   (e) $\theta_1 = \vartheta' \langle \|_i \vartheta_0 \varphi'_0 y(t), \|_{1-i} \vartheta_1 \varphi'_1 \overline{v} u \rangle$: by Def. 3, $\theta_0 \lessdot \theta_1$ iff $\vartheta \varphi \varphi_0$ is a prefix of $\vartheta' \|_i \vartheta_0 \varphi'_0$ or of $\vartheta' \|_{1-i} \vartheta_1 \varphi'_1$ (rule 8); by Def. 5, $\theta_0 \lll \theta_1$ iff either $\vartheta \varphi \varphi_0$ or $\vartheta \varphi \varphi_1$ is a prefix of $\vartheta' \|_i \vartheta_0 \varphi'_0$ or of $\vartheta' \|_{1-i} \vartheta_1 \varphi'_1$ (rule 8); hence θ_0 and θ_1 can be exchanged;

   (f) $\theta_1 = \vartheta' \langle \|_0 \vartheta_0 \mathsf{join}\, Q_0, \|_1 \vartheta_1 \mathsf{join}\, Q_1 \rangle$: by Def. 3, $\theta_0 \lessdot \theta_1$ iff $\vartheta = \vartheta' \|_0 \vartheta_0$ or $\vartheta = \vartheta' \|_1 \vartheta_1$ (rule 10); by Def. 5, $\theta_0 \lll \theta_1$ iff $\vartheta = \vartheta' \|_0 \vartheta_0$ or $\vartheta = \vartheta' \|_1 \vartheta_1$ (rule 10); hence θ_0 and θ_1 refer to different bio-processes, so they can be exchanged;

   (g) $\theta_1 = \vartheta' \mathsf{split} \langle Q_0, Q_1 \rangle$: by Def. 3, $\theta_0 \lessdot \theta_1$ iff $\vartheta = \vartheta'$ (rule 10); by Def. 5, $\theta_0 \lll \theta_1$ iff $\vartheta = \vartheta'$ (rule 10); hence θ_0 and θ_1 refer to different bio-processes, so they can be exchanged;

5. $\theta_0 = \vartheta \langle \|_i \vartheta_0 \varphi_0 x(w), \|_{1-i} \vartheta_1 \varphi_1 \overline{y} z \rangle$:

   (a) $\theta_1 = \vartheta' \varphi' \mathsf{expose}(t, \Delta)$: by Def. 3, $\theta_0 \lessdot \theta_1$ iff $\vartheta \|_i \vartheta_0 \varphi_0$ is a prefix of $\vartheta' \varphi'$ (rule 8); by Def. 5, $\theta_0 \lll \theta_1$ iff either $\vartheta \|_i \vartheta_0 \varphi_0$ or $\vartheta \|_{1-i} \vartheta_1 \varphi_1$ is a prefix of $\vartheta' \varphi'$ (rule 8); hence θ_0 and θ_1 can be exchanged;

   (b) $\theta_1 = \vartheta' \varphi' \mathsf{hide}(t)$: by Def. 3, $\theta_0 \lessdot \theta_1$ iff $\vartheta \|_i \vartheta_0 \varphi_0$ is a prefix of $\vartheta' \varphi'$ (rule 8); by Def. 5, $\theta_0 \lll \theta_1$ iff (either $\vartheta \|_i \vartheta_0 \varphi_0$ or $\vartheta \|_{1-i} \vartheta_1 \varphi_1$ is a prefix of $\vartheta' \varphi'$) (rule 8) or ($\vartheta \|_i \vartheta_0 = \vartheta'$ and $x = t$) or ($\vartheta \|_{1-i} \vartheta_1 = \vartheta'$ and $y = t$) (rule 9); hence θ_0 and θ_1 refer to pi-processes and different binders, so they can be exchanged;

(c) $\theta_1 = \vartheta'\varphi'\,\mathsf{unhide}(t)$: by Def. 3, $\theta_0 \lessdot \theta_1$ iff $\vartheta\|_i\vartheta_0\varphi_0$ is a prefix of $\vartheta'\varphi'$ (rule 8); by Def. 5, $\theta_0 \lll \theta_1$ iff either $\vartheta\|_i\vartheta_0\varphi_0$ or $\vartheta\|_{1-i}\vartheta_1\varphi_1$ is a prefix of $\vartheta'\varphi'$ (rule 8); moreover, θ_0 and θ_1 refer to different binders (since it is not possible to execute an *unhide* operation immediately after an inter-communication on the same beta binder), so they can be exchanged;

(d) $\theta_1 = \vartheta'\varphi\langle\varphi_0'v(t), \varphi_1'\overline{v}u\rangle$: by Def. 3, $\theta_0 \lessdot \theta_1$ iff $\vartheta\|_i\vartheta_0\varphi_0$ is a prefix of $\vartheta'\varphi\varphi_0'$ or of $\vartheta'\varphi\varphi_1'$ (rule 8); by Def. 5, $\theta_0 \lll \theta_1$ iff either $\vartheta\|_i\vartheta_0\varphi_0$ or $\vartheta\|_{1-i}\vartheta_1\varphi_1$ is a prefix of $\vartheta'\varphi\varphi_0'$ or of $\vartheta'\varphi\varphi_1'$ (rule 8); hence θ_0 and θ_1 can be exchanged;

(e) $\theta_1 = \vartheta'\langle\|_i\vartheta_0'\varphi_0'v(t), \|_{1-i}\vartheta_1'\varphi_1'\overline{r}u\rangle$: by Def. 3, $\theta_0 \lessdot \theta_1$ iff $\vartheta\|_i\vartheta_0\varphi_0$ is a prefix of $\vartheta'\|_i\vartheta_0'\varphi_0'$ or of $\vartheta'\|_{1-i}\vartheta_1'\varphi_1'$ (rule 8); by Def. 5, $\theta_0 \lll\cdot\, \theta_1$ iff either $\vartheta\|_i\vartheta_0\varphi_0$ or $\vartheta\|_{1-i}\vartheta_1\varphi_1$ is a prefix of $\vartheta'\|_i\vartheta_0'\varphi_0'$ or of $\vartheta'\|_{1-i}\vartheta_1'\varphi_1'$ (rule 8); hence θ_0 and θ_1 can be exchanged;

(f) $\theta_1 = \vartheta'\,\langle\|_0\vartheta_0'\mathsf{join}\,Q_0, \|_1\vartheta_1'\mathsf{join}\,Q_1\rangle$: by Def. 3, $\theta_0 \lessdot \theta_1$ iff $\vartheta\|_i\vartheta_0$ or $\vartheta\|_1\vartheta_1$ are equal to $\vartheta'\|_0\vartheta_0'$ or to $\vartheta'\|_1\vartheta_1'$ (rule 10); by Def. 5, $\theta_0 \lll\cdot\, \theta_1$ iff $\vartheta\|_i\vartheta_0$ or $\vartheta\|_1\vartheta_1$ are equal to $\vartheta'\|_0\vartheta_0'$ or to $\vartheta'\|_1\vartheta_1'$ (rule 10); hence θ_0 and θ_1 refer to different bio-processes, so they can be exchanged;

(g) $\theta_1 = \vartheta'\,\mathsf{split}\langle Q_0, Q_1\rangle$: by Def. 3, $\theta_0 \lessdot \theta_1$ iff $\vartheta\|_i\vartheta_0 = \vartheta'$ or $\vartheta\|_{1-i}\vartheta_1 = \vartheta'$ (rule 10); by Def. 5, $\theta_0 \lll \theta_1$ iff $\vartheta\|_i\vartheta_0 = \vartheta'$ or $\vartheta\|_{1-i}\vartheta_1 = \vartheta'$ (rule 10); hence θ_0 and θ_1 refer to different bio-processes, so they can be exchanged;

6. $\theta_0 = \vartheta\,\langle\|_0\vartheta_0\mathsf{join}\,Q_0, \|_1\vartheta_1\mathsf{join}\,Q_1\rangle$:

(a) $\theta_1 = \vartheta'\varphi\,\mathsf{expose}(x,\,\Delta)$: by Def. 3, $\theta_0 \lessdot \theta_1$ iff $\vartheta\|_0\vartheta_0 = \vartheta'$ (rule 11); hence θ_0 and θ_1 refer to different bio-processes and so they can be exchanged;

(b) $\theta_1 = \vartheta'\varphi\,\mathsf{hide}(x)$: by Def. 3, $\theta_0 \lessdot \theta_1$ iff $\vartheta\|_0\vartheta_0 = \vartheta'$ (rule 11); hence θ_0 and θ_1 refer to different bio-processes and so they can be exchanged;

(c) $\theta_1 = \vartheta'\varphi\,\mathsf{unhide}(x)$: by Def. 3, $\theta_0 \lessdot \theta_1$ iff $\vartheta\|_0\vartheta_0 = \vartheta'$ (rule 11); hence θ_0 and θ_1 refer to different bio-processes and so they can be exchanged;

(d) $\theta_1 = \vartheta'\varphi\langle\varphi_0 y(w), \varphi_1\overline{y}z\rangle$: by Def. 3, $\theta_0 \lessdot \theta_1$ iff $\vartheta\|_0\vartheta_0 = \vartheta'$ (rule 11); hence θ_0 and θ_1 refer to different bio-processes and so they can be exchanged;

(e) $\theta_1 = \vartheta'\langle\|_i\vartheta_0'\varphi_0 y(t), \|_{1-i}\vartheta_1'\varphi_1\overline{w}z\rangle$: by Def. 3, $\theta_0 \lessdot \theta_1$ iff $\vartheta\|_0\vartheta_0 = \vartheta'\|_i\vartheta_0'$ or $\vartheta\|_0\vartheta_0 = \vartheta'\|_{1-i}\vartheta_1'$ (rule 11); hence θ_0 and θ_1 refer to different bio-processes and so they can be exchanged;

(f) $\theta_1 = \vartheta'\,\langle\|_0\vartheta_0'\mathsf{join}\,Q_0', \|_1\vartheta_1'\mathsf{join}\,Q_1'\rangle$: by Def. 3, $\theta_0 \lessdot \theta_1$ iff $\vartheta\|_0\vartheta_0 = \vartheta'\|_i\vartheta_0'$ or $\vartheta\|_0\vartheta_0 = \vartheta'\|_{1-i}\vartheta_1'$ (rule 12); hence θ_0 and θ_1 refer to different bio-processes and so they can be exchanged;

(g) $\theta_1 = \vartheta'\,\mathsf{split}\langle Q_0', Q_1'\rangle$: by Def. 3, $\theta_0 \lessdot \theta_1$ iff $\vartheta\|_0\vartheta_0 = \vartheta'$ (rule 12); hence θ_0 and θ_1 refer to different bio-processes and so they can be exchanged;

7. $\theta_0 = \vartheta\, \mathsf{split}\langle Q_0, Q_1\rangle$:

   (a) $\theta_1 = \vartheta'\varphi\, \mathsf{expose}(x,\, \Delta)$: by Def. 3, $\theta_0 \lessdot \theta_1$ iff $\vartheta\|_j = \vartheta'$ (rule 11); hence θ_0 and θ_1 refer to different bio-processes and so they can be exchanged;

   (b) $\theta_1 = \vartheta'\varphi\, \mathsf{hide}(x)$: by Def. 3, $\theta_0 \lessdot \theta_1$ iff $\vartheta\|_j = \vartheta'$ (rule 11); hence θ_0 and θ_1 refer to different bio-processes and so they can be exchanged;

   (c) $\theta_1 = \vartheta'\varphi\, \mathsf{unhide}(x)$: by Def. 3, $\theta_0 \lessdot \theta_1$ iff $\vartheta\|_j = \vartheta'$ (rule 11); hence θ_0 and θ_1 refer to different bio-processes and so they can be exchanged;

   (d) $\theta_1 = \vartheta'\varphi\langle\varphi_0 y(w), \varphi_1\overline{y}z\rangle$: by Def. 3, $\theta_0 \lessdot \theta_1$ iff $\vartheta\|_j = \vartheta'$ (rule 11); hence θ_0 and θ_1 refer to different bio-processes and so they can be exchanged;

   (e) $\theta_1 = \vartheta'\langle\|_i\vartheta_0\varphi_0 y(t), \|_{1-i}\vartheta_1\varphi_1\overline{w}z\rangle$: by Def. 3, $\theta_0 \lessdot \theta_1$ iff $\vartheta\|_j = \vartheta'\|_i\vartheta_0$ or $\vartheta\|_j = \vartheta'\|_{1-i}\vartheta_1$ (rule 11); hence θ_0 and θ_1 refer to different bio-processes and so they can be exchanged;

   (f) $\theta_1 = \vartheta'\,\langle\|_0\vartheta_0'\mathsf{join}\, Q_0', \|_1\vartheta_1'\mathsf{join}\, Q_1'\rangle$: by Def. 3, $\theta_0 \lessdot \theta_1$ iff $\vartheta\|_j = \vartheta'\|_0\vartheta_0'$ or $\vartheta\|_j = \vartheta'\|_1\vartheta_1'$ (rule 12); hence θ_0 and θ_1 refer to different bio-processes and so they can be exchanged;

   (g) $\theta_1 = \vartheta'\, \mathsf{split}\langle Q_0', Q_1'\rangle$: by Def. 3, $\theta_0 \lessdot \theta_1$ iff $\vartheta\|_j = \vartheta'$ (rule 12); hence θ_0 and θ_1 refer to different bio-processes and so they can be exchanged.

Corollary 1 *Given a computation* $B_0 \xrightarrow{\theta_0} B_1 \xrightarrow{\theta_1} B_2$*, if* $\theta_0 \smile \theta_1 \Rightarrow \exists B_1'.B_0 \xrightarrow{\theta_1} B_1' \xrightarrow{\theta_0} B_2$*.*

The second lemma, usually known as *permutation of transitions*, states that there always exists a computation in which two concurrent transitions occur consecutively.

Lemma 2 *Given a computation* $\xi = B_0 \xrightarrow{\theta_0} B_1 \longrightarrow \cdots \longrightarrow B_n \xrightarrow{\theta_n} B_{n+1}$*, if* $\theta_0 \not\prec \theta_n \Rightarrow \exists$ *a permutation* $\sigma : [0..n] \to [0..n]$ *and a computation* $B_0 \xrightarrow{\theta_0'} B_1' \xrightarrow{\theta_1'} \cdots B_n' \xrightarrow{\theta_n'} B_{n+1}$ *such that* $\exists i \in [0..n]\,.(\sigma(0) = i \wedge \sigma(n) = i+1 \wedge \sigma(j) = j-1$ *for* $0 < j \leq i \wedge \sigma(m) = m+1$ *for* $i+1 \leq m < n)$ *with* $\theta'_{\sigma(l)} = \theta_l$ *for each* $l \in [0..n]$*.*

Proof. The proof is by induction on the length n of the computation.

- *Induction basis*
  $n = 1$ (i.e. $\xi = B_0 \xrightarrow{\theta_0} B_1 \xrightarrow{\theta_1} B_2$ and $\theta_0 \not\prec \theta_1$ (hence $\theta_0 \not\lessdot \theta_1$ and $\theta_0 \not\lll \theta_1$)).
  The permutation is $\sigma : [0,1] \to [0,1]$ and the computation ξ satisfies the properties.

- *Inductive step*
  Let us assume that the lemma holds for $n = k$ (i.e. $k-1$ transitions between θ_0 and θ_n); now let us consider $n = k+1$ (i.e. k transitions between θ_0 and θ_n).
  Let h be the minimum index such that $\theta_0 \not\prec \theta_h$; hence $\theta_l \not\prec \theta_h$ holds for any $l < h$ (because if per absurdum $\theta_l \prec \theta_h$ (i.e. $\theta_l \not< \theta_h$ or $\theta_l \not\ll \theta_h$), then, for the transitive property of $<$ and $\ll$, $\theta_0 \prec \theta_h$). Hence, by Lemma 1 we can swap θ_h and θ_{h-1} in ξ, obtaining $\xi^1 = B_0 \xrightarrow{\theta_0} \cdots \longrightarrow B_{h-1} \xrightarrow{\theta_h} B'_h \xrightarrow{\theta_{h-1}} B_{h+1} \longrightarrow \cdots \xrightarrow{\theta_n} B_{n+1}$. We can then repeat this procedure h times, until we obtain $\xi^h = B_0 \xrightarrow{\theta_h} B'_0 \xrightarrow{\theta_0} \cdots \xrightarrow{\theta_n} B_{n+1}$, in which there are $k-1$ transitions between θ_0 and θ_n, so that it is possible to apply the inductive hypothesis.

Corollary 2 *Given a computation* $\xi = B_0 \xrightarrow{\theta_0} B_1 \longrightarrow \cdots \longrightarrow B_n \xrightarrow{\theta_n} B_{n+1}$*, if* $\theta_0 \smile \theta_n \Rightarrow \exists$ *a permutation* $\sigma : [0..n] \rightarrow [0..n]$ *and a computation* $B_0 \xrightarrow{\theta'_0} B'_1 \xrightarrow{\theta'_1} \cdots B'_n \xrightarrow{\theta'_n} B_{n+1}$ *such that* $\exists i \in [0..n]\,.(\sigma(0) = i \wedge \sigma(n) = i+1 \wedge \sigma(j) = j-1$ *for* $0 < j \leq i \wedge \sigma(m) = m+1$ *for* $i+1 \leq m < n)$ *with* $\theta'_{\sigma(l)} = \theta_l$ *for each* $l \in [0..n]$.

The following theorem derives from Corollary 1 and Corollary 2 and it states that if in a computation there are two concurrent transitions, then there exists another computation in which the two transitions occur in reverse order.

Theorem 1 *Given a computation* $B_0 \xrightarrow{\theta_0} \cdots \xrightarrow{\theta_n} B_{n+1}$*, if* $\theta_0 \smile \theta_n \Rightarrow \exists$ *a computation* $B_0 \longrightarrow \cdots \xrightarrow{\theta_n} \cdots \xrightarrow{\theta_0} \cdots \longrightarrow B_{n+1}$.

6 Application to Biology, Pharmacology and Medicine

There are many potential applications of this technology. One interesting application is in biology: by analysing these properties, life scientists could obtain interesting predictions on the dynamical behaviour of complex systems under investigation (for example the interaction of distinct entities in signalling pathways).

Another application is in pharmacology: causality and concurrency analysis of a system made up of an ill organism and a new drug to be tested can assist pharmacologists by identifying the effects (both positive and negative) of the new drug on the organism. Compared to the traditional method used in drug discovery, which is to test a new drug on animals and then on a small number of selected human beings, the method provided by software is faster and safer, so that "in vivo" tests can be done at a lower risk. Hence, computer simulation with causality and concurrency analysis can greatly help pharmacologists, who need to specify the system composed of the ill organism and the drug, to select a set of relevant tests to be performed on animals and human beings.

Finally, another application is in medicine: causality analysis can assist doctors in medical diagnosis by allowing them to consider only the events and the subpart of the organism which are relevant to the abnormal behaviour under investigation; this is very important in medicine since the organism under investigation, the human body, is a huge and complex system, which is definitely intractable altogether.

6.1 An Example: the ERK/MAPK Pathway

In this section we model and analyse the ERK pathway, which is an instance of the important and intensively studied MAPK pathway. The term 'MAPK pathway' refers to a module of three kinases activated by sequentially phosphorylating each other.

Figure 1 shows the Raf/MEK/ERK pathway (see [8] for details).

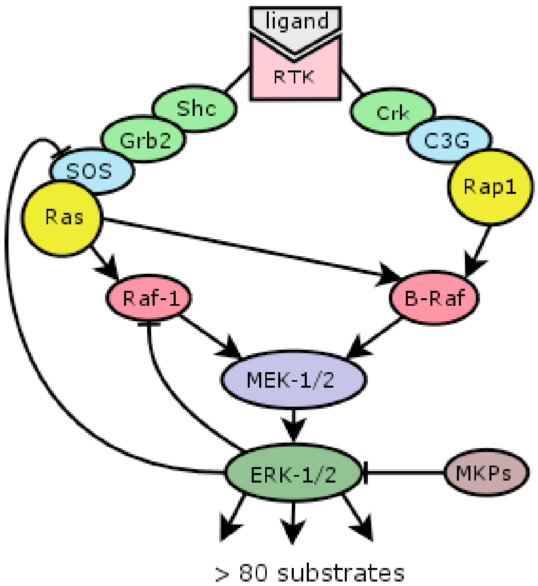

Figure 1: Structure of the ERK pathway.

The binding of a ligand to RTK (receptor tyrosine kinase) causes the autophosphorylation of RTK on tyrosine residues, which are docking sites for adaptor and signalling molecules.

By means of the adaptor proteins Shc and Grb, Ras recruits SOS (a GDP/GTP exchange factor), which allows Ras to be activated; symmetrically, by means of the adaptor protein Crk, Rap1 recruits C3G (a GDP/GTP exchange factor), which allows Rap1 to be activated.

Ras can activate two types of Raf proteins (Raf-1 and B-Raf), while Rap1 can only activate B-Raf.

Both types of Raf proteins can activate MEK-1/2 by phosphorylation on two serine residues.

MEK-1/2 can activate ERK-1/2 by phosphorylation on threonine and tyrosine residues.

Upon activation, ERK can phosphorylate over 80 substrates in the cytoplasm and the nucleus, and it can regulate gene expression by phosphorylating transcription factors such Ets, Elk and Myc.

Negative feedback loops include the induction of MKPs by ERK, and the inhibitory phosphorylation of Raf-1 and SOS.

6.1.1 *Beta-binders* Model of the ERK/MAPK Pathway

$$
\begin{array}{ll}
\mathrm{Sys} = & (((((\|_0\|_0\|_0\|_0\|_0\|_0\mathrm{RTK} \,\big\|\, \|_0\|_0\|_0\|_0\|_0\|_1\mathrm{Ligand}) \,\big\|\, \|_0\|_0\|_0\|_0\|_1\mathrm{SOS}) \,\big\|\\
& \|_0\|_0\|_0\|_1\mathrm{C3G}) \,\big\|\, \|_0\|_0\|_1\mathrm{GDP/GTP}) \,\big\|\, \|_0\|_1\mathrm{GDP/GTP}) \,\big\|\, (\|_1\|_0\mathrm{RAS} \,\big\|\\
& (\|_1\|_1\|_0\mathrm{RAP1} \,\big\|\, (\|_1\|_1\|_1\|_0\mathrm{RAF1} \,\big\|\, (\|_1\|_1\|_1\|_1\|_0\mathrm{BRAF} \,\big\|\, (\|_1\|_1\|_1\|_1\|_1\|_0\mathrm{MEK} \,\big\|\\
& (\|_1\|_1\|_1\|_1\|_1\|_1\|_0\mathrm{ERK} \,\big\|\, (\|_1\|_1\|_1\|_1\|_1\|_1\|_1\|_0\mathrm{MKP} \,\big\|\, (\|_1\|_1\|_1\|_1\|_1\|_1\|_1\|_1\|_0\mathrm{Ets} \,\big\|\\
& (\|_1\|_1\|_1\|_1\|_1\|_1\|_1\|_1\|_1\|_0\mathrm{Elk} \,\big\|\, \|_1\|_1\|_1\|_1\|_1\|_1\|_1\|_1\|_1\|_1\mathrm{MyC})))))))))
\end{array}
$$

where

$$
\begin{array}{ll}
\mathrm{RTK} & = \beta(rcpt : RTK)\,\beta^h(tyrosine : Shc, Grb, Crk)\\
& \big[\; rcpt(ligand).\,\mathsf{unhide}(tyrosine)\,.(|_0|_0 tyrosine(adpt1).\,\mathsf{nil} \,\big|\\
& \qquad |_0|_1 \overline{tyrosine}(adpt2).\,\mathsf{nil} \,\big||_1 tyrosine(adpt3).\,\mathsf{nil})\;\big]\\
\mathrm{Ligand} & = \beta(bind : RTK)\,\big[\; \overline{bind}ligand.\,\mathsf{nil}\;\big]\\
\mathrm{SOS} & = \beta^h(SOSact : SOS)\,\beta(adpt1 : Shc)\,\beta(adpt2 : Grb)\\
& \beta(exchange : GDP)\,\beta(act : SOS\_ERK)\\
& \big[\; |_0|_0\overline{adpt1}bind.\overline{adpt2}bind.\,\mathsf{unhide}(SOSact)\,.\,\mathsf{nil} \,\big|\\
& \;|_0|_1\overline{SOSact}exch.\overline{exchange}GTP.\,\mathsf{unhide}(Ras)\,.\,\mathsf{nil} \,\big|\\
& \;|_1!\,!\,act(x).\,\mathsf{hide}(SOSact)\,.\,\mathsf{nil}\;\big]\\
\mathrm{C3G} & = \beta^h(C3Gact : C3G)\,\beta(adpt : Grk)\,\beta(exchange : GDP)\\
& \big[\; |_0\overline{adpt}bind.\,\mathsf{unhide}(C3Gact)\,.\,\mathsf{nil} \,\big|\\
& \;|_1\overline{C3Gact}exch.\overline{exchange}GTP.\,\mathsf{unhide}(Rap1)\,.\,\mathsf{nil}\;\big]\\
\mathrm{GDP/GTP} & = \beta(GDP/GTP : GDP)\\
& \big[\; !\,!\,GDP/GTP(x).\,\mathsf{hide}(GDP/GTP)\,.\,\mathsf{expose}(GDP/GTP,\, x)\,.\,\mathsf{nil}\;\big]\\
\mathrm{Ras} & = \beta(exchfact : SOS\_GDP)\,\beta^h(Ras : Raf1, BRaf)\\
& \big[\; |_0|_0 exchfact(x).\,\mathsf{nil} \,\big||_0|_1\overline{Ras}phosphorylate.\,\mathsf{nil} \,\big||_1\overline{Ras}phosphorylate.\,\mathsf{nil}\;\big]\\
\mathrm{Rap1} & = \beta(exchfact : C3G\_GDP)\,\beta^h(Rap1 : BRaf)\\
& \big[\; |_0 exchfact(x).\,\mathsf{nil} \,\big||_1\overline{Rap1}phosphorylate.\,\mathsf{nil}\;\big]\\
\mathrm{Raf1} & = \beta(Raf : Raf1)\,\beta^h(Raf\_MEK : Raf1\_MEK)\,\beta(act : Raf1\_ERK)\\
& \big[\; |_0 Raf(x).\,\mathsf{unhide}(Raf\_MEK)\,.\overline{Raf\_MEK}phosphorylate.\,\mathsf{nil} \,\big|\\
& \;|_1!\,!\,act(x).\,\mathsf{hide}(Raf1\_MEK)\,.\,\mathsf{nil}\;\big]\\
\mathrm{BRaf} & = \beta(Raf : BRaf)\,\beta^h(Raf\_MEK : BRaf\_MEK)\\
& \big[\; Raf(x).\,\mathsf{unhide}(Raf\_MEK)\,.\overline{Raf\_MEK}phosphorylate.\,\mathsf{nil}\;\big]\\
\mathrm{MEK} & = \beta(serine : Raf1\_MEK, BRaf\_MEK)\,\beta(MEK : MEK1/2)\\
& \big[\; serine(x).(|_0\overline{MEK}phosphorylate.\,\mathsf{nil} \,\big||_1\overline{MEK}phosphorylate.\,\mathsf{nil})\;\big]\\
\mathrm{ERK} & = \beta(threonine : MEK1/2)\,\beta(tyrosine : MEK1/2)\\
& \beta(inhibitRaf1 : Raf1\_ERK)\,\beta(inhibitSOS : SOS\_ERK)\\
& \beta(act : ERK\_MKP)\,\beta^h(ERK : ERK1/2)\\
& \big[\; |_0|_0|_0|_0|_1 threonine(x).tyrosine(y).\,\mathsf{unhide}(ERK)\,.\,\mathsf{nil} \,\big|\\
& \;|_0|_0|_0|_0|_1\overline{ERK}phosphorylatetransfacts.\,\mathsf{nil} \,\big||_0|_0|_1!\,!\,act(x).\,\mathsf{hide}(ERK)\,.\,\mathsf{nil}\big|\\
& \;|_0|_1!\,!\,\overline{inhibitRaf1}inhibit.\,\mathsf{nil}\big|\big||_1!\,!\,\overline{inhibitSOS}inhibit.\,\mathsf{nil}\;\big]\\
\mathrm{MKP} & = \beta^h(act : MKP)\,\beta(inhibit : ERK\_MKP)\,\big[\; !\,!\,\overline{inhibit}inhibitERK.\,\mathsf{nil}\;\big]\\
\mathrm{Ets} & = \beta(Ets : ERK1/2)\,\big[\; Ets(x).\,\mathsf{nil}\;\big]\\
\mathrm{Elk} & = \beta(Elk : ERK1/2)\,\big[\; Elk(x).\,\mathsf{nil}\;\big]\\
\mathrm{MyC} & = \beta(MyC : ERK1/2)\,\big[\; MyC(x).\,\mathsf{nil}\;\big]
\end{array}
$$

The following join function is also defined:

$$
f_{join}(\mathbf{B}_0, \mathbf{B}_1, P_0, P_1) = \begin{array}{l} \text{if } (\beta(x : SOS) \in \mathbf{B}_0 \text{ and } \beta(y : SOS\_GDP) \in \mathbf{B}_1) \\ \text{or } (\beta(x : C3G) \in \mathbf{B}_0 \text{ and } \beta(y : C3G\_GDP) \in \mathbf{B}_1) \\ \text{then } (\mathbf{B}_0\mathbf{B}_1 \setminus \{y\}, P_0P_1\{x/y\}) \end{array}
$$

One of the possible computations of this system is the following (for the sake of simplicity we do not consider inhibitory activities):

$$
\begin{array}{rl}
t_1 = & \|_0\|_0\|_0\|_0\|_0\langle \|_0 rcpt(ligand), \|_1 \overline{bind}ligand\rangle \\
t_2 = & \|_0\|_0\|_0\|_0\|_0\|_0 \,\mathsf{unhide}(tyrosine) \\
t_3 = & \|_0\|_0\|_0\|_0\langle \|_0\|_0|_0|_0 tyrosine(adpt1), \|_1|_0|_0\overline{adpt1}bind\rangle \\
t_4 = & \|_0\|_0\|_0\langle \|_0\|_0\|_0|_1 tyrosine(adpt3), \|_1|_0\overline{adpt}bind\rangle \\
t_5 = & \|_0\|_0\|_0\|_1|_0 \,\mathsf{unhide}(C3Gact)\rangle \\
t_6 = & \langle \|_0\|_0\|_0\|_1\mathsf{join}\ |_0\ \mathsf{nil}\ \Big| |_1\overline{C3Gact}exch.\overline{exchange}GTP.\ \mathsf{unhide}(Rap1)\ .\ \mathsf{nil}, \\
 & \|_1\|_1\|_0\mathsf{join}\ |_0 exchfact(x).\ \mathsf{nil}\ \Big| |_1\overline{Rap}phosphorylate.\ \mathsf{nil}\rangle \\
t_7 = & \|_0\|_0\|_0\|_1\langle |_0|_1\overline{C3Gact}exch, |_1|_0 C3Gact(x)\rangle \\
t_8 = & \|_0\|_0\langle \|_0\|_1|_0|_1\overline{exchange}GTP, \|_1!GDP/GTP(x)\rangle \\
t_9 = & \|_0\|_0\|_0\|_1|_0|_1 \,\mathsf{unhide}(Rap1)\rangle \\
t_{10} = & \langle \|_1\|_1\|_1\|_1\|_0 Raf(x), \|_0\|_0\|_0\|_1|_1|_1\overline{Rap1}phosphorylate\rangle \\
t_{11} = & \|_0\|_0\|_0\|_0\langle \|_0\|_0|_0|_1 tyrosine(adpt2), \|_1|_0|_0\overline{adpt2}bind\rangle \\
t_{12} = & \|_0\|_0\|_0\|_0\|_1|_0|_0 \,\mathsf{unhide}(SOSact) \\
t_{13} = & \langle \|_0\|_0\|_0\|_0\|_1\mathsf{join}\ |_0|_0\ \mathsf{nil}\ \Big| |_0|_1\overline{SOSact}exch.\overline{exchange}GTP.\ \mathsf{unhide}(Ras)\ .\ \mathsf{nil}\ \Big| \\
 & \qquad\qquad |_1!\ !\,act(x).\ \mathsf{hide}(SOSact)\ .\ \mathsf{nil}, \\
 & \|_1\|_0\mathsf{join}\ |_0|_0 exchfact(x).\ \mathsf{nil}\ \Big| |_0|_1\overline{Ras}phosphorylate.\ \mathsf{nil}\ \Big| |_1\overline{Ras}phosphorylate.\ \mathsf{nil}\rangle \\
t_{14} = & \|_0\|_0\|_0\|_0\|_1\langle |_0|_0|_1\overline{SOSact}exch, |_1|_0|_0 SOSact(x)\rangle \\
t_{15} = & \|_0\langle \|_0\|_0\|_0\|_1|_0|_0|_1\overline{exchange}GTP, \|_1!GDP/GTP(x)\rangle \\
t_{16} = & \|_0\|_0\|_0\|_0\|_1|_0|_0|_1 \,\mathsf{unhide}(Ras) \\
t_{17} = & \langle \|_1\|_1\|_1\|_0|_0 Raf(x), \|_0\|_0\|_0\|_0\|_1|_1|_1\overline{Ras}phosphorylate\rangle \\
t_{18} = & \|_1\|_1\|_1\|_0|_0 \,\mathsf{unhide}(Raf\_MEK) \\
t_{19} = & \|_1\|_1\|_1\langle \|_0|_0\overline{Raf\_MEK}phosphorylate, \|_1\|_1\|_0 serine(x)\rangle \\
t_{20} = & \|_1\|_1\|_1\|_1\|_1\langle \|_1\|_0|_0|_0|_0|_0|_1 threonine(x), \|_0|_0\overline{MEK}phosphorylate\rangle \\
t_{21} = & \|_1\|_1\|_1\|_1\|_1\langle \|_1\|_0|_0|_0|_0|_0|_1 tyrosine(y), \|_0|_1\overline{MEK}phosphorylate\rangle \\
t_{22} = & \|_1\|_1\|_1\|_1\|_1\|_1\|_0|_0|_0|_0|_0|_1 \,\mathsf{unhide}(ERK) \\
t_{23} = & \|_1\|_1\|_1\|_1\|_1\|_1\langle \|_0|_0|_0|_0|_1\overline{ERK}phosphorylatetransfacts, \|_1\|_1\|_1\|_0 Elk(x)\rangle \ .
\end{array}
$$

t_1 represents the binding of a ligand to RTK, and t_2 is the following authophosphorylation of RTK.

t_3, t_4 and t_{11} are the bindings, respectively, of Shc, Crk and Grb.

The block t_4-t_{10} is the Rap1-subpathway, which is relative to the activation of C3G, its binding to Rap1 and the following activation of B-Raf.

The block t_3, t_{11}-t_{17} is the Ras-subpathway, which is relative to the activation of SOS, its binding to Ras and the following activation of Raf-1.

The block t_{18}-t_{22} is the activation of MEK-1/2, and hence of ERK-1/2, by means of Raf-1.

Finally, t_{23} refer to ERK gene expression.

6.1.2 Concurrency Analysis in ERK/MAPK Pathway Model

Applying the causality rules, we obtain that:

$t_1 \lessdot t_2, t_3, t_4, t_{11}$	$t_6 \lessdot t_7, t_8, t_9, t_{10}$	$t_{13} \lessdot t_{14}, t_{15}, t_{16}, t_{17}$	$t_{19} \lessdot t_{20}, t_{21}$
$t_3 \lessdot t_{13}$	$t_9 \lessdot t_{10}$	$t_{16} \lessdot t_{17}$	$t_{20} \lessdot t_{21}, t_{22}$
$t_4 \lessdot t_6$	$t_{11} \lessdot t_{13}$	$t_{17} \lessdot t_{18}, t_{19}$	$t_{21} \lessdot t_{22}$
$t_5 \lessdot t_6$	$t_{12} \lessdot t_{13}$	$t_{18} \lessdot t_{19}$	$t_{22} \lessdot t_{23}$

Applying the enabling rules, we obtain that the following relations also hold:

$t_1 \ll t_2, t_3, t_4, t_{11}$	$t_7 \ll t_8, t_9$	$t_{15} \ll t_{16}$	$t_{21} \ll t_{22}$
$t_2 \ll t_3, t_4, t_{11}$	$t_8 \ll t_9$	$t_{17} \ll t_{18}, t_{19}$	
$t_3 \ll t_{11}, t_{12}, t_{13}, t_{14}, t_{15}, t_{16}$	$t_{11} \ll t_{12}, t_{13}, t_{14}, t_{15}, t_{16}$	$t_{18} \ll t_{19}$	
$t_4 \ll t_5, t_6, t_7, t_8, t_9$	$t_{12} \ll t_{13}, t_{14}, t_{15}, t_{16}$	$t_{19} \ll t_{20}, t_{21}$	
$t_5 \ll t_6, t_7, t_8, t_9$	$t_{14} \ll t_{15}, t_{16}$	$t_{20} \ll t_{21}, t_{22}$	

By considering the transitive union $\prec$ of causality and enabling, we obtain that:

$t_1 \prec t_{2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23}$
$t_2 \prec t_{3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23}$
$t_3 \prec t_{11,12,13,14,15,16,17,18,19,20,21,22,23}$

$t_4 \prec t_{5,6,7,8,9,10}$	$t_{14} \prec t_{15,16,17,18,19,20,21,22,23}$
$t_5 \prec t_{6,7,8,9,10}$	$t_{15} \prec t_{16,17,18,19,20,21,22,23}$
$t_6 \prec t_{7,8,9,10}$	$t_{16} \prec t_{17,18,19,20,21,22,23}$
$t_7 \prec t_{8,9,10}$	$t_{17} \prec t_{18,19,20,21,22,23}$
$t_8 \prec t_{9,10}$	$t_{18} \prec t_{19,20,21,22,23}$
$t_9 \prec t_{10}$	$t_{19} \prec t_{20,21,22,23}$
$t_{11} \prec t_{12,13,14,15,16,17,18,19,20,21,22,23}$	$t_{20} \prec t_{21,22,23}$
$t_{12} \prec t_{13,14,15,16,17,18,19,20,21,22,23}$	$t_{21} \prec t_{22,23}$
$t_{13} \prec t_{14,15,16,17,18,19,20,21,22,23}$	$t_{22} \prec t_{23}$

We can observe that t_1 and t_2 cause all the rest of the computation: this reflects the reality, since the whole pathway is triggered by the ligand binding to RTK.

We can also notice that the blocks t_4-t_{10} and t_3, t_{11}-t_{17} are unrelated: in fact the RAS-subpathway and the Rap1-subpathway are independent on each other.

In the chosen computation Ras activates Raf-1, while Rap1 activates independently B-Raf. Activation of MEK-1/2 is eventually triggered by Raf-1, so the following part is not caused by the Rap1-subpathway, while it is caused by the Ras-subpathway.

In order to analyse concurrency, we need to analyse any possible computation. What results from this analysis, is that the Ras-pathway and the Rap1-pathway are concurrent: they are independent from each other, and thus can happen in any order. The following part of the pathway (t_{18}-t_{23}), instead, is concurrent with neither Ras-subpathway nor Rap1-subpathway, since there exists a computation in which each of them causes t_{18}-t_{23}: hence hate formal analysis reflects what we know about reality, that is that both subpathways can cause the activation of ERK-1/2.

7 Conclusions and Further Work

We defined some relations between the transitions of a computation originated by a *Beta-binders* process. We showed that the connection between causality, enabling and concurrency usually defined in process algebras and the permutation of transitions property are valid for *Beta-binders* as well. The bio-inspiration of *Beta-binders* makes particularly appealing a notion of causality. In fact, when studying complex biological systems, e.g. the interaction of a drug with a disease, we have huge systems and only in few parts the interaction is occurring. Causality can be used to single out the subsystem of interest, thus reducing the size of the problem to a manageable one.

Therefore, we plan to apply our definitions to real biological case studies and to implement a tool for concurrency analysis, which is supposed to be integrated in a simulator for *Beta-binders*, that can be used in medical and pharmacological research to study the interactions of entities in complex biological systems.

References

[1] M. Boreale and D. Sangiorgi. A fully abstract semantics for causality in the π-calculus. In *Proc. of Symposium on Theoretical Aspects of Computer Science*, volume 900 of *LNCS*, pages 243–254, 1995.

[2] M. Curti, P. Degano, C. Priami, and C. T. Baldari. Modelling biochemical pathways through enhanced π-calculus. *Theor. Comp. Sci.*, 325(1):111–140, 2004.

[3] P. Darondeau and P. Degano. Causal Trees. In *Proc. of Int. Colloquium on Automata Languages and Programming*, volume 372 of *LNCS*, pages 234–248, 1989.

[4] P. Degano, F. Gadducci, and C. Priami. Causality and Replication in Concurrent Processes. In *Proc. of Perspectives of System Informatics*, volume 2890 of *LNCS*, pages 307–318, 2003.

[5] P. Degano, R. De Nicola, and U. Montanari. A Partial Ordering Semantics for CCS. *Theor. Comp. Sci.*, 75(3):223–262, 1990.

[6] P. Degano and C. Priami. Non-interleaving semantics for mobile processes. *Theor. Comp. Sci.*, 216(1–2):237–270, 1999.

[7] U. Montanari and M. Pistore. Concurrent Semantics for the π-calculus. In *Proc. of Mathematical Foundations of Programming Semantics*, volume 1 of *ENTCS*, pages 411–429, 1995.

[8] R. J. Orton, O. E. Sturm, V. Vyshermirsky, M. Calder, D. R. Gilbert, and W. Kolch. Computational modelling of the receptor-tyrosine-kinase-activated MAPK pathway. *Biochem. J.*, 392:249–261, 2005.

[9] V. R. Pratt. Modelling Concurrency with Partial Orders. *Int. J. of Parallel Programming*, 15(1):33–71, 1986.

[10] C. Priami and P. Quaglia. Beta binders for biological interactions. In *Proc. of Computational Methods in Systems Biology*, volume 3082 of *LNCS*, pages 20–33, 2005.

[11] C. Priami and P. Quaglia. Operational patterns in Beta-binders. *Transactions on Computational Systems Biology*, 1:50–65, 2005.

[12] D. Sangiorgi. Locality and True-concurrency in Calculi for Mobile Processes. In *Theor. Aspects of Computer Software*, volume 789 of *LNCS*, pages 405–424, 1994.

[13] W. Vogler. *Modular Construction and Partial Order Semantics of Petri Nets*. Springer-Verlag, 1992.