# Towards a Causal Semantics for Brane Calculi

Nadia Busi

Dipartimento di Scienze dell'Informazione, Università di Bologna,
Mura A. Zamboni 7, I-40127 Bologna, Italy.
busi@cs.unibo.it

**Summary.** Brane Calculi are a family of biologically inspired process calculi, proposed in [6] to model the interactions of dynamically nested membranes. We propose a semantics that describes the causal dependencies occurring between the reactions of a system described in Brane Calculi. We investigate the basic properties that are satisfied by such a semantics. The notion of causality turns out to be quite relevant for biological systems, as it permits to point out which events occurring in a biological pathway are necessary for another event to happen.

## 1 Introduction

Brane calculi [6] are a family of process calculi proposed for modeling the behavior of biological membranes.

The formal investigation of biological membranes has been initiated by G. Păun [20], in the field of automata and formal language theory, with the definition of P systems. In a process algebraic setting, the notions of membranes and compartments are explicitly represented in BioAmbients [23], a variant of Mobile Ambients [8] based on a set of biologically inspired primitives of interaction.

Brane calculi represent an evolution of BioAmbients: the main difference w.r.t. previous approaches consists in the fact that the active entities reside on membranes, and not inside membranes. In [6] two basic instances of brane calculi have been proposed: the Phago/Exo/Pino (PEP) and the Mate/Bud/Drip (MBD) calculi.

In this paper we concentrate on the MBD calculus. The primitives of MBD are inspired by membrane fusion (mate) and fission (mito). Because membrane fission is an uncontrollable process that can split a membrane at an arbitrary place, it is replaced by two simpler operations: *budding*, that is splitting off one internal membrane, and *dripping*, that consists in splitting off zero internal membranes.

The aim of this work is to start an investigation of the causal dependencies arising in Brane Calculi, and more precisely in the MBD calculus. The main motivation for this work comes from system biology, as the understanding of the causal relations occurring between the events of a complex biological pathway could be of precious help, e.g., for limiting the search space in the case some unpredicted event occurs.

The study of a causal semantics for process algebras dates back to the early nineties for CCS [17] (see, e.g., [10, 9, 15]), and to the mid nineties for the $\pi$-calculus [18] (see, e.g., [1, 3, 11, 12]).

To the best of our knowledge, the only other work that deals with causality in bio-inspired calculi is [14], where a causal semantics for Beta Binders [21, 22] – based on the $\pi$-calculus semantics and on the enhanced operational semantics approach of [12] – is defined. One of the main differences between Beta Binders and Brane Calculi is that the membrane structure in Beta Binders is flat, whereas in Brane Calculi the membranes are nested to form a hierarchical structure. As we will see, this difference has a deep impact on the complexity of the causal relation. The other differences between the two approaches will be discussed throughout the paper.

The paper is organized as follows. Section 2 introduces the syntax and the interleaving semantics for the MBD fragment of the Brane Calculus. Sections 3 and 4 are devoted to the definition of the causal semantics. Section 3 provides an informal description of the features of the causal semantics we are defining, and illustrates the problems that have arised through a list of examples. The formal definition of the causal semantics is in Section 4, followed by a discussion concerning the properties that are (not) satisfied by such a semantics. Finally, Section 5 reports some conclusive remarks.

## 2 MBD Calculus: Syntax and Semantics

In this section we recall the syntax and the standard, interleaving semantics of Brane Calculi, and specialize it to MBD [6].

### 2.1 Syntax and structural congruence of Brane Calculi

A system consists of nested membranes, and a process is associated to each membrane.

**Definition 1.** *The set of systems is defined by the following grammar:*

$$P, Q \quad ::= \quad \diamond \mid P \circ Q \mid !P \mid \sigma(\!| P |\!)$$

*The set of membrane processes is defined by the following grammar:*

$$\sigma, \tau \quad ::= \quad 0 \mid \sigma|\tau \mid !\sigma \mid a.\sigma$$

*Variables $a, b$ range over actions, that will be detailed later.*

The term $\diamond$ represents the empty system; the parallel composition operator on systems is $\circ$. The replication operator ! denotes the parallel composition of an unbounded number of instances of a system. The term $\sigma(\!| P |\!)$ denotes the membrane that performs process $\sigma$ and contains system $P$.

The term 0 denotes the empty process, whereas $|$ is the parallel composition of processes; with $!\sigma$ we denote the parallel composition of an unbounded number of instances of process $\sigma$. Term $a.\sigma$ is a guarded process: after performing the action $a$, the process behaves as $\sigma$.

We adopt the following abbreviations: with $a$ we denote $a.0$, with $(\!| P |\!)$ we denote $0(\!| P |\!)$, and with $\sigma(\!| \, |\!)$ we denote $\sigma(\!| \diamond |\!)$.

The structural congruence relations on systems and processes is defined as follows:[1]

**Definition 2.** *The structural congruence $\equiv$ is the least congruence relation satisfying the following axioms:*

$$\begin{array}{ll}
P \circ Q \equiv Q \circ P & \sigma \mid \tau \equiv \tau \mid \sigma \\
P \circ (Q \circ R) \equiv (P \circ Q) \circ R & \sigma \mid (\tau \mid \rho) \equiv (\sigma \mid \tau) \mid \rho \\
P \circ \diamond \equiv P & \sigma \mid 0 \equiv \sigma \\
\\
!\diamond \equiv \diamond & !0 \equiv 0 \\
!(P \circ Q) \equiv !P \circ !Q & !(\sigma \mid \tau) \equiv !\sigma \mid !\tau \\
!!P \equiv !P & !!\sigma \equiv !\sigma \\
P \circ !P \equiv !P & \sigma \mid !\sigma \equiv !\sigma \\
\\
0(\!| \diamond |\!) \equiv \diamond &
\end{array}$$

## 2.2 Interleaving semantics of Brane Calculi

We recall the standard, interleaving semantics. At each computational step, a single reaction is chosen and executed. The next definition provides the set of generic reaction rules that are valid for all brane calculi, while the reaction axioms are specific for each brane calculus; the reaction axioms for MBD will be provided in Definition 5.

**Definition 3.** *The basic reaction rules are the following:*

$$(\texttt{par}) \quad \frac{P \rightarrow Q}{P \circ R \rightarrow Q \circ R} \qquad\qquad (\texttt{brane}) \quad \frac{P \rightarrow Q}{\sigma(\!| P |\!) \rightarrow \sigma(\!| Q |\!)}$$

$$(\texttt{strucong}) \quad \frac{P' \equiv P \qquad P \rightarrow Q \qquad Q \equiv Q'}{P' \rightarrow Q'}$$

[1] With abuse of notation we use $\equiv$ to denote both structural congruence on systems and structural congruence on processes.

Rules (par) and (brane) are the contextual rules that respectively permit to a system to execute also if it is in parallel with another process or if it is inside a membrane, respectively. Rule (strucong) ensures that two structurally congruent systems have the same reactions.

With $\rightarrow^*$ we denote the reflexive and transitive closure of a relation $\rightarrow$. Given a reduction relation $\rightarrow$, we say that the system $P'$ is a *derivative* of the system $P$ if $P \rightarrow^* P'$; the set of *derivatives* of a system $P$ is denoted by $Deriv(P)$.

We say that a system $P$ *has a divergent computation* (or infinite computation) if there exist an infinite sequence of systems $P_0, P_1, \ldots, P_i, \ldots$ such that $P = P_0$ and $\forall i \geq 0 : P_i \rightarrow P_{i+1}$. We say that a system $P$ *has a terminating computation* if there exists $Q \in Deriv(P)$ such that $Q \not\rightarrow$. We say that *all computations of a system $P$ terminate* if $P$ has no divergent computations.

We use $\prod$ (resp. $\bigcirc$) to denote the parallel composition of a set of processes (resp. systems), i.e., $\prod_{i\in\{1,\ldots,n\}} \sigma_i = \sigma_1 \mid \ldots \mid \sigma_n$ and $\bigcirc_{i\in\{1,\ldots,n\}} P_i = P_1 \circ \ldots \circ P_n$. Moreover, $\prod_{i\in\emptyset} \sigma_i = 0$ and $\bigcirc_{i\in\emptyset} P_i = \diamond$. Finally, $\prod_n \sigma$ (resp. $\bigcirc_n P$) denotes the parallel composition of $n$ copies of process $\sigma$ (resp. system $P$).

### 2.3 Syntax and interleaving semantics of MBD

The actions of the MBD calculus, proposed in [6], are inspired by membrane fusion and splitting. To make membrane splitting more controllable, in [6] two more basic operations are used: *budding*, consisting in splitting off one internal membrane, and *dripping*, consisting in splitting off zero internal membranes. Membrane fusion, or merging, is called *mating*.

**Definition 4.** *Let $Name$ be a denumerable set of names, ranged over by $n, m, \ldots$. The set of actions of MBD is defined by the following grammar:*

$$a \quad ::= \quad mate_n \mid mate_n^{\perp} \mid bud_n \mid bud_n^{\perp}(\sigma) \mid drip(\sigma)$$

Actions $mate_n$ and $mate_n^{\perp}$ will synchronize to obtain membrane fusion. Action $bud_n$ permits to split one internal membrane, and synchronizes with the co-action $bud_n^{\perp}$. Action *drip* permits to split off zero internal membranes. Actions $bud^{\perp}$ and *drip* are equipped with a process $\sigma$, that will be associated to the new membrane created by the membrane performing the action.

**Definition 5.** *The reaction relation for MBD is the least relation containing the following axioms, and satisfying the rules in Definition 3:*

(mate) $mate_n.\sigma|\sigma_0(\!| P |\!) \;\circ\; mate_n^{\perp}.\tau|\tau_0(\!| Q |\!) \rightarrow \sigma|\sigma_0|\tau|\tau_0(\!| P \circ Q |\!)$

(bud) $bud_n^{\perp}(\rho).\tau|\tau_0(\!| bud_n.\sigma|\sigma_0(\!| P |\!) \circ Q |\!) \rightarrow \rho(\!| \sigma|\sigma_0(\!| P |\!) |\!) \;\circ\; \tau|\tau_0(\!| Q |\!)$

(drip) $drip(\rho).\sigma|\sigma_0(\!| P |\!) \rightarrow \rho(\!| \; |\!) \;\circ\; \sigma|\sigma_0(\!| P |\!)$

## 3 A Causal Semantics for MBD: an informal explanation

In this section we provide a causal semantics for MBD.

To define a causal semantics for process calculi, we follow the approach used in [15] for CCS, and in [1] for the $\pi$-calculus. The idea consists in decorating the reaction relation with two pieces of information:

- a fresh name $k$, that is associated to the reaction and it is taken from the set of causes $\mathcal{K}$;
- a set $H \subseteq \mathcal{K}$, containing all the names associated to the already occurred reactions, that represent a cause for the current reaction.

To keep track of the names of the already occurred reactions that may represent a cause for the reactions that may happen in the future, the syntax of the terms of the calculus is enriched with such an information on causal dependencies. As in [1], for the sake of clarity we only keep track of the so called immediate causes, as the set of general causes can be reconstructed by transitive closure of the immediate causal relation. We will provide more explanation on this point with an example in the following part of the paper.

Now we start with an informal introduction of causality in MBD. First we discuss how the standard kinds of causality arising in most process calculi – i.e., those due to the prefix structure of processes and to the synchronization of two complementary actions – scale to Brane Calculi. Then we perform a design choice concerned with the semantics of calculi for membranes, and finally we discuss other features peculiar of the MBD operations.

### 3.1 Classical causal dependencies: structural and synchronization causality

We start the kind of causal dependencies that arises in all process calculi, namely, structural causality and synchronization causality.

Structural causality arises from the prefix structure of terms. Consider for example the following system:

$$drip(\sigma).drip(\rho)(\!|\quad|\!)$$

Such a system can first create a new membrane with process $\sigma$, followed by the creation of a second new membrane with process $\rho$; i.e., it can perform the sequence of reactions

$$drip(\sigma).drip(\rho)(\!|\quad|\!) \;\rightarrow drip(\rho)(\!|\quad|\!) \circ \sigma(\!|\quad|\!) \rightarrow 0(\!|\quad|\!) \circ \sigma(\!|\quad|\!) \circ \rho(\!|\quad|\!)$$

The creation of the first membrane is a necessary condition for the creation of the second membrane, hence we say that the execution of the $drip(\rho)$ operation *is caused by* the execution of the $drip(\sigma)$ operation.

To remember the fact that the action $drip(\sigma)$ will be a cause for the actions performed by the continuation of the prefix, we replace the $drip(\sigma)$ prefix with a causal operator containing the cause name associated to the $drip(\sigma)$ action. Thus, we obtain the following causal reactions:

$$drip(\sigma).drip(\rho)(\!|\ |\!) \xrightarrow{h;\emptyset} \{h\} :: drip(\rho)(\!|\ |\!) \circ \sigma(\!|\ |\!) \xrightarrow{k;\{h\}} \{k\} :: 0(\!|\ |\!) \circ \sigma(\!|\ |\!) \circ \rho(\!|\ |\!)$$

The decoration $h;\emptyset$ of the first reaction means that the first reaction is labeled with the causal name $h$ and that its set of causes is empty. The decoration $k;\{h\}$ of the second reaction means that the second reaction has associated the causal name $k$, and it is caused by the reaction named $h$ (i.e., the first reaction). The process $\{h\} :: drip(\rho)$ means that the first action performed by process $drip(\rho)$ is caused by the reaction named $h$. Note that – for the sake of brevity – process $\{k\} :: 0$ is decorated only with the immediate cause $\{k\}$, as the whole set of causes, i.e., $\{h, k\}$ can be easily constructed.

To lighten the notation, in the following we will drop the parentheses surrounding the set of causes, if this creates no confusion.

The other kind of causality, i.e., synchronization causality, arises when two processes synchronize on complementary actions. Consider the system

$$drip(\sigma_1).mate_n.drip(\tau_1)(\!|\ |\!) \circ drip(\sigma_2).mate_n^{\perp}.drip(\tau_2)(\!|\ |\!)$$

The mate reaction can be performed when both the actions $drip(\sigma_1)$ and $drip(\sigma_2)$ have been performed; hence, it is caused by both actions. We obtain the following:

$$\begin{array}{l}
drip(\sigma_1).mate_n.drip(\tau_1)(\!|\ |\!) \circ drip(\sigma_2).mate_n^{\perp}.drip(\tau_2)(\!|\ |\!) \xrightarrow{h_1;\emptyset} \\
h_1 :: mate_n.drip(\tau_1)(\!|\ |\!) \circ drip(\sigma_2).mate_n^{\perp}.drip(\tau_2)(\!|\ |\!) \circ \sigma_1(\!|\ |\!) \xrightarrow{h_2;\emptyset} \\
h_1 :: mate_n.drip(\tau_1)(\!|\ |\!) \circ h_2 :: mate_n^{\perp}.drip(\tau_2)(\!|\ |\!) \circ \sigma_1(\!|\ |\!) \circ \sigma_2(\!|\ |\!) \xrightarrow{k;h_1,h_2} \\
k :: (drip(\tau_1) \mid drip(\tau_2)(\!|\ |\!) \circ \sigma_1(\!|\ |\!) \circ \sigma_2(\!|\ |\!)
\end{array}$$

Hence, the (label $k$ of the) mate reaction will be an immediate cause for both $drip(\tau_1)$ and $drip(\tau_2)$, and the global set of causes of these two drip actions will be $\{h_1, h_2, k\}$.

### 3.2 How does the causes distribute over the parallel components of a membrane process?

When moving to consider the features of the causal relation peculiar of membrane calculi, a first question arises: if a process on a membrane performs an action, this action will be a cause only for its continuation (and eventually for the continuation of its synchronizing action), or for the whole process on the membrane? In other words, consider the system

$$mate_n \mid drip(\sigma)(\!|\ |\!) \circ mate_n^{\perp}(\!|\ |\!)$$

If the system performs the mate synchronization, then the drip action will be caused or not by the mate action? The assumption that the drip will be caused by the mate may have the following biological interpretation: when a membrane interaction operation is performed, all the membrane is involved, and at the end of the operation the structure of all the membrane has been affected. This assumption is considered in [14] in the definition of a causal semantics for Beta-binders [21, 22] , a bio-inspired process calculus roughly consisting of unnested compartments enclosing $\pi$-calculus processes. It is also used in [2] for the definition of a maximal parallelism semantics (i.e., step semantics with maximal progress: if an action can be performed in the current step, then it must be performed) for MBD. It is also the common approach used in the definition of the maximal parallelism semantics for Membrane Systems with evolving membranes (see, e.g., [19, 20]). In the present paper, we consider the opposite approach: in the above system, we consider the drip operation independent from the mate operation, as the drip operation can be executed regardless of the fact that the mate synchronization has been performed or not. The biological interpretation may be the following: the membrane proteins and the part of the lipid bilayer involved in the mate synchronization are different from the membrane proteins and the part of the bilayer that is performing the drip operation, and they lie in different parts of the membrane surface.

Thus, we consider the following causal reactions:

$$\begin{array}{l} mate_n \mid drip(\sigma)(\!|\quad|\!) \circ mate_n^{\perp}(\!|\quad|\!) \xrightarrow{h;\emptyset} \\ h :: 0 \mid drip(\sigma) \mid h :: 0(\!|\quad|\!) \xrightarrow{k;\emptyset} \\ h :: 0 \mid k :: 0 \mid h :: 0(\!|\quad|\!) \end{array}$$

Note that the information on the causes of the empty process 0 is completely irrelevant, hence in the following we will replace $H :: 0$ with 0.

### 3.3 Causal dependencies generated by the mate operation

Now we analyze the features peculiar of the MBD operations. According to the informal explanation above, a mate action turns out to be a cause for the continuations of the mate and the co-mate prefixes that synchronize to perform the operation. However, when considering the mate operation, a more subtle kind of causality, we call environment causality, is originated, e.g., between the mate action and the processes on the child membranes of the two membranes performing the mate and co-mate actions. This causality is due to the fact that the environment of such child membranes, i.e., the set of membranes with which they can interact, is increased by the execution of the mate action.

#### Mate followed by mate

Consider the following process:

$$\begin{array}{l} mate_n(\!|\ (mate_m \mid mate_o)(\!|\quad|\!) \ \circ\ mate_o^{\perp}(\!|\quad|\!)\ |\!) \circ \\ mate_n^{\perp}(\!|\ mate_m^{\perp}(\!|\quad|\!)\ |\!) \end{array}$$

Now the mate synchronization on $m$ cannot be performed, as the two membranes whose processes can synchronize on such an operation belong to different membranes. On the other hand, the mate synchronization on $o$ can take place, as both membranes whose processes can synchronize on such an operation belong to the same membrane.

However, if the mate synchronization on $n$ takes place, the two external membranes are fused; this results in a change of the environment of the child membrane; now the mate on $m$ can take place, as the two child membranes now belong to the same father membrane and can get in contact. Hence, the mate on $m$ causally depends on the mate on $n$.

To this aim, we decorate the processes of the child membranes of the external membrane performing a mate with label $k$ in the following way: the child membranes on the left are decorated with the enriched label $k_i^+$, whereas the child membrane on the right with the enriched complementary label $k_i^-$. [2] Note that we cannot simply decorate both groups of child membranes with label $k$, otherwise we are no longer able to distinguish between the synchronization on $m$, that is caused by $k$, and the synchronization on $o$ which has no causes.

The enriched labels are used in the following way: when two processes preceded by enriched labels synchronize on a mate operation, the label $k$ will be a cause for such a synchronization if one process in decorated with an enriched label and the synchronizing process is decorated with the complementary label.

We obtain the following causal reductions:

$$\begin{array}{l} mate_n(\!|\ (mate_m \mid mate_o)(\!|\quad|\!) \circ mate_o^{\perp}(\!|\quad|\!)\ |\!) \circ \\ mate_n^{\perp}(\!|\ mate_m^{\perp}(\!|\quad|\!)\ |\!) \\ \xrightarrow{h;\emptyset} \\ (0 \mid 0)(\!|\ (h_i^+ :: mate_m \mid h_i^+ :: mate_o)(\!|\quad|\!) \circ h_i^+ :: mate_o^{\perp}(\!|\quad|\!) \circ \\ h_i^- :: mate_m^{\perp}(\!|\quad|\!)\ |\!) \end{array}$$

Now, if the mate on m is executed, then it will be caused by $h$, as the *mate* and *comate* processes are labeled with $h_i^+$ and $h_i^-$, respectively:

$$\begin{array}{l} (0 \mid 0)(\!|\ (h_i^+ :: mate_m \mid h_i^+ :: mate_o)(\!|\quad|\!) \circ h_i^+ :: mate_o^{\perp}(\!|\quad|\!) \circ \\ h_i^- :: mate_m^{\perp}(\!|\quad|\!)\ |\!) \\ \xrightarrow{k;h} \\ (0 \mid 0)(\!|\ (0 \mid h_i^+ :: mate_o)(\!|\quad|\!) \circ h_i^+ :: mate_o^{\perp}(\!|\quad|\!) \circ \\ 0(\!|\quad|\!)\ |\!) \end{array}$$

On the other hand, if the mate on $o$ is executed, then it is not caused by $h_n$, because the *mate* and *comate* processes are labeled with the same label $h_i^+$.

[2] The $i$ in the labels $k_i^+$ and $k_i^-$ stands for "internal", and means that the action with label $k$ has been performed by the father membrane. The need for such a label will be made clear in the following.

**Mate followed by bud**

A similar problem arises between the father and the child membrane when a bud operation is performed. Consider the following process:

$$(mate_n \mid bud_m^{\perp}(\rho_1))(\!|\ bud_m(\!|\quad|\!) \circ bud_o(\!|\quad|\!)\ |\!)\circ$$
$$(mate_n^{\perp} \mid bud_o^{\perp}(\rho_2))(\!|\quad|\!)$$

Now only the bud on $m$ can be performed, as the membrane performing the bud on $o$ is not a child of the membrane performing the corresponding cobud. However, the bud on $o$ can be performed after the mate on $n$ is performed, hence the bud causally depends on the mate. Thus, besides decorating the children of a membrane performing a mate (resp comate) with complementary labels $k_i^+$ (resp. $k_i^-$), we also decorate the subprocesses in parallel with the subprocess performing the mate (resp. the comate) with $k_e^+$ (resp. $k_e^-$)[3]. When a bud is performed, it is caused by $k$ if the process performing the cobud on the father membrane is decorated, e.g., with $k_e^+$ and the process performing the bud on the child membrane is decorated with $k_e^-$.

Note that, in case of a mate followed by a drip, the decorated causes will give rise to no causal dependency: the drip is caused by the mate only if the mate (or the comate) is a prefix of the drip.

### 3.4 Causal dependencies generated by the bud and the drip operations

The bud (resp. drip) operation create a new membrane – whose membrane process is specified in the cobud (resp. drip) action – surrounding the child membrane that performs the synchronizing bud action (resp. with no children). As the new membrane does not exist before the bud (resp. drip) operation is performed, all the actions that such a membrane will perform are caused by the bud (resp. drip) operation.

Consider the following system:

$$bud_n^{\perp}(drip(\sigma))(\!|\ bud_n(\!|\quad|\!)\ |\!)$$

This system can perform the following causal reactions:

$$\begin{array}{l} bud_n^{\perp}(drip(\sigma))(\!|\ bud_n(\!|\quad|\!)\ |\!) \xrightarrow{h;\emptyset} \\ 0(\!|\quad|\!) \circ h :: drip(\sigma)(\!|\ 0(\!|\quad|\!)\ |\!) \xrightarrow{k;h} \\ 0(\!|\quad|\!) \circ 0(\!|\ 0(\!|\quad|\!)\ |\!) \circ k :: \sigma(\!|\quad|\!) \end{array}$$

We note that the bud operation generates no environmental cause. Regarding the child membranes, they are essentially divided into two sets, thus possibly preventing some mate (or bud) operation that was possible before to happen. On the other hand, the other processes in the father membrane are left unchanged.

[3] Here the $e$ means "external"

## 4 A causal semantics for MBD: the formal definition

In this section we provide a formal definition of the notions introduced in the previous section.

**Definition 6.** *Let $\mathcal{K}$ be a denumerable set of cause names, disjoint from the set Names. Let Deco($\mathcal{K}$) be the set*

$$Deco(\mathcal{K}) = \mathcal{K} \cup \{k_x^y \mid k \in \mathcal{K} \wedge x \in \{i, e\} \wedge y \in \{+, -\}\}$$

*The set of membrane processes with causes are defined by the following grammar:*

$$\tilde{\sigma}, \tilde{\tau} \quad ::= \quad 0 \mid \tilde{\sigma}|\tilde{\tau} \mid !\tilde{\sigma} \mid K :: a.\sigma$$

*Variables $a, b$ range over MBD actions specified in Definition 4, $K \subseteq Deco(\mathcal{K})$, and $a.\sigma$ is a sequential process as defined in Definition 1.*

*The set of systems with causes is defined as in Definition 1, but using processes with causes instead of processes to decorate membranes.*

The set of causes preceding the process 0 is useless, hence it has been omitted. We omit the $\tilde{}$ over processes if it is clear from the context that they are processes with causes.

To define the causal semantics, we need an auxiliary operator on processes (and on systems) permitting to add a set of causes in front of each sequential subprocess of the process (resp. of the processes associated to the most external membranes of the system.

**Definition 7.** *Given $K \subseteq Deco(\mathcal{K})$, the operator $K\Mapsto$ is inductively defined on processes with causes as follows:*

$$\begin{aligned} K\Mapsto 0 &= 0 \\ K\Mapsto (\sigma \mid \tau) &= K\Mapsto \sigma \mid K\Mapsto \tau \\ K\Mapsto !\sigma &= !K\Mapsto \sigma \\ K\Mapsto H :: a.\sigma &= H \cup K :: a.\sigma \end{aligned}$$

*The operator $K\Mapsto$ is inductively defined on systems as follows:*

$$\begin{aligned} K\Mapsto \diamond &= \diamond \\ K\Mapsto (P \circ Q) &= K\Mapsto P \;\circ\; K\Mapsto Q \\ K\Mapsto (!P) &= !K\Mapsto P \\ K\Mapsto (\sigma (\!| \; P \; |\!)) &= (K\Mapsto \sigma)(\!| \; P \; |\!) \end{aligned}$$

If the set $K$ is a singleton, often we omit the surrounding parenthesis in the operator $K\Mapsto$; thus we write, e.g., $k\Mapsto P$ instead of $\{k\}\Mapsto P$.

Now we are ready to define the causal semantics for MBD. We write $P \xrightarrow{k;H} P'$ to denote the fact that system $P$ performs an action – to which we associate the cause name $k$ – that is caused by the (previously occurred) actions whose action names form the set $H$. The cause name $k$ is a fresh name: this means that it does not occur in $P$ and that it has not used yet in the current computation.

The structural congruence relation is the that in Definition 2. The causal rules are obtained by decorating the rules in Definition 3 with the causal information.

**Definition 8.** *The causal reaction rules are the following:*

$$(\texttt{par})\ \frac{P \xrightarrow{k;H} Q}{P \circ R \xrightarrow{k;H} Q \circ R} \qquad (\texttt{brane})\ \frac{P \xrightarrow{k;H} Q}{\sigma(\!|\, P \,|\!) \xrightarrow{k;H} \sigma(\!|\, Q \,|\!)}$$

$$(\texttt{strucong})\ \frac{P' \equiv P \quad P \xrightarrow{k;H} Q \quad Q \equiv Q'}{P' \xrightarrow{k;H} Q'}$$

Now we are ready to define the causal reaction relation for MBD.

**Definition 9.** *The causal reaction relation for MBD is the least relation containing the following axioms, and satisfying the rules in Definition 8:*

$$\begin{array}{ll}(\texttt{kmate}) & (H_1 :: mate_n.\sigma)|\sigma_0(\!|\, P \,|\!) \ \circ \ (H_2 :: mate_n^{\perp}.\tau)|\tau_0(\!|\, Q \,|\!) \xrightarrow{k;H_1 \oplus_m H_2} \\ & ((k \cup H_1 \ominus_m H_2) \Rrightarrow \sigma \ \mid \ k_e^+ \Rrightarrow \sigma_0 \ \mid \\ & \ (k \cup H_2 \ominus_m H_1) \Rrightarrow \tau \ \mid \ k_e^- \Rrightarrow \tau_0) \quad (\!|\, k_i^+ \Rrightarrow (P) \ \circ \ k_i^- \Rrightarrow (Q) \,|\!)\end{array}$$

$$\begin{array}{ll}(\texttt{kbud}) & (H_1 :: bud_n^{\perp}(\rho).\tau)|\tau_0(\!|\, (H_2 :: bud_n.\sigma)|\sigma_0(\!|\, P \,|\!) \circ Q \,|\!) \xrightarrow{k;H_1 \oplus_b H_2} \\ & (k \cup H_1 \odot_b H_2) :: \rho(\!|\, ((k \cup H_2 \ominus_b H_1) \Rrightarrow \sigma)|\sigma_0(\!|\, P \,|\!) \,|\!) \circ \\ & ((k \cup H_1 \ominus_m H_2) \Rrightarrow \tau) \mid \tau_0(\!|\, Q \,|\!)\end{array}$$

$$\begin{array}{ll}(\texttt{kdrip}) & (H :: drip(\rho).\sigma|\sigma_0)(\!|\, P \,|\!) \xrightarrow{k;f_d(H)} \\ & k \cup f_d'(H) \Rrightarrow \rho(\!|\ |\!) \ \circ \ (k' \cup f_d''(H) \Rrightarrow \sigma) \mid \sigma_0(\!|\, P \,|\!)\end{array}$$

*The auxiliary functions are defined as follows:*

$$\begin{aligned} H_1 \oplus_m H_2 &= \{k \mid k \in (H_1 \cup H_2) \cap \mathcal{K}\} \cup \\ &\quad \{k \mid \{k_i^+, k_i^-\} \subseteq (H_1 \cup H_2)\} \\ H_1 \ominus_m H_2 &= \{k_x^y \in H_1 \mid k \notin (H_1 \oplus_m H_2)\} \end{aligned}$$

$$\begin{aligned} H_1 \oplus_b H_2 &= \{k \mid k \in (H_1 \cup H_2) \cap \mathcal{K}\} \cup \\ &\quad \{k \mid k_e^x \in H_1 \wedge k_e^y \in H_2 \wedge x \neq y\} \\ H_1 \odot_b H_2 &= \{k_i^y \in H_1 \mid k \notin (H_1 \oplus_b H_2)\} \\ H_1 \ominus_b H_2 &= \{k_x^y \in H_1 \mid k \notin (H_1 \oplus_b H_2)\} \end{aligned}$$

$$\begin{aligned} f_d(H) &= \{k \mid k \in H \cap \mathcal{K}\} \\ f_d'(H) &= \{k_i^x \mid k_i^x \in H \wedge x \in \{+,-\}\} \\ f_d''(H) &= \{k_y^x \mid k_y^x \in H \wedge x \in \{+,-\} \wedge y \in \{i,e\}\} \end{aligned}$$

The auxiliary functions describe the way in which the (decorated) causes propagate when a reduction is performed.

In the merge operation, the set of causes of the merge action, denoted by $H_1 \oplus_m H_2$, contains both the causes of the mate and the comate operation, as well as those causes $h$ such that $h_i^+$ decorates one of the merging membranes and $h_i^-$ decorates the other (this means that the two membranes have become sibling membranes by the execution of a mate operation with label $h$). The external decorated causes are not taken into account because they are concerned with bud operations between a father and a child membrane, and not with sibling membranes. To avoid redundancy, the set of causes $H_1 \ominus H_2$ of the continuation of the mate action is obtained by removing the decorated causes whose name appears as a cause of the current mate synchronization (and analogously for comate, bud and cobud actions).

In the bud operation, the set of causes of the bud actions, denoted by $H_1 \oplus_b H_2$, contains both the causes of the bud and the cobud operation, as well as those causes $h$ such that, e.g., $h_e^+$ decorates the father membrane and $h_i^-$ decorates the child membrane. The set of causes of the newly created membrane is denoted by $H_1 \odot H_2$, and contains only internal causes; they are needed because in the system

$$mate_n (\!|\ bud_o^\perp(mate_m)(\!|\ bud_o(\!|\quad|\!)\ |\!)\ |\!) \circ mate_n^\perp(\!|\ mate_m^\perp\ |\!)$$

the mate synchronization on $m$ can happen only if the mate synchronization on $n$ has been performed.

Regarding the drip operation, here there is no synchronization; hence, the set of causes labeling the reduction relation, represented by $f_d(H)$ is exactly the set of nondecorated causes. The newly created membrane is decorated with function $f'_d(H)$ containing only internal causes; they are needed because in the system

$$mate_n (\!|\ drip(mate_m)(\!|\quad|\!)\ |\!) \circ mate_n^\perp(\!|\ mate_m^\perp\ |\!)$$

the mate synchronization on $m$ can happen only if the mate synchronization on $n$ has been performed. The external causes are not needed because they are used for bud synchronization between father and child, and the newly created membrane has no child taken from the old membrane.

### 4.1 Properties of the causal semantics

The only interesting property enjoyed by the causal semantics is the retrievability of the interleaving semantics. We start defining the function *DropCause* which removes the causes from processes and systems.

**Definition 10.** *The function DropCause is defined inductively on processes with causes in the following way:*

$$\begin{aligned}
DropCause(0) &= 0 \\
DropCause(\tilde{\sigma}|\tilde{\tau}) &= DropCause(\tilde{\sigma}) \mid DropCause(\tilde{\tau}) \\
DropCause(!\tilde{\sigma}) &= !DropCause(\tilde{\sigma}) \\
DropCause(K :: a.\sigma) &= a.\sigma
\end{aligned}$$

*The function DropCause is defined inductively on systems with causes in the following way:*

$$
\begin{aligned}
DropCause(\diamond) &= \diamond \\
DropCause(P \circ Q) &= DropCause(P) \ \circ \ DropCause(Q) \\
DropCause(!P) &= !DropCause(P) \\
DropCause(\tilde{\sigma}(\!| P |\!)) &= DropCause(\tilde{\sigma})(\!| DropCause(P) |\!)
\end{aligned}
$$

**Theorem 1.** *Let $P$ be a system with causes. The following properties hold:*

- *if $P \xrightarrow{k;H} P'$ then $DropCause(P) \rightarrow DropCause(P')$;*
- *if $DropCause(P) \rightarrow Q$ then there exist a system with causes $P'$, a cause name $k$ and a set of cause names $H$ such that $P \xrightarrow{k;H} P'$ and $Q = DropCause(P')$.*

The so-called diamond property, stating that if two non-causally related actions can happen one after the other, then they can happen also in the other order, and at the end they reach the same system, does not hold. In our setting, the diamond property can be formally defined as follows: given a system $P$, if $P \xrightarrow{h;H} P' \xrightarrow{k;K} P''$ and $h \notin K$, then there exists a system $q$ such that $P \xrightarrow{k;K} Q \xrightarrow{h;H} P''$.

Consider e.g. the following system:

$$bud_m^{\perp}(0)(\!|\ mate_n(\!|\quad|\!)\ \circ\ (bud_m \mid mate_n^{\perp})(\!|\quad|\!)\ |\!)$$

This system can perform the mate action, followed by the bud action. Moreover, the two actions are independent, i.e., causally unrelated. However, if we first perform the bud action, then the submembrane $mate_n^{\perp})(\!|\quad|\!)$ is isolated from the other submembrane, and the merge can no longer take place. However, there is no reason to consider the bud action as causally dependent on the mate action, as the bud action can actually independently occur at the beginning of the computation. Nevertheless, there is a form of asymmetric conflict between the two actions: the occurrence of the bud action prevents the mate action to happen, but the vice versa does not hold. A similar phenomenon takes place, e.g., in Petri nets with read and inhibitor arcs (see [4] for a discussion on this topic). A possible way to capture this kind of asymmetric conflict is based on the following idea: if a mate synchronization with causal label $k$ is performed, we decorate the process of the father membrane of the two membrane that are fusing with a label, say $k_f$ (where the $f$ stands for "father"). When the bud operation is performed, the cobud prefix is decorated with $k_f$, whereas the bud prefix is decorate with $k_e^-$. As a but synchronization is performed in this situation, we get the information that – even if the bud does not causally depend from the mate – the bud synchronization cannot be swapped with the mate.

Even if there is no asymmetric conflict, there are situations where the diamond property does not hold. Even if the two actions can be performed in either order, the final states that are reached are different. Take the system

$$bud_n^{\perp}(0).\sigma(\!|\ (bud_n.\tau \mid drip(\rho))(\!|\quad|\!)\ |\!)$$

If the bud action is performed first, then the membrane with process $\rho$ will be dripped inside the newly created membrane, labeled with process 0. On the other hand, if the drip is performed first then the membrane with process $\rho$ remains inside the older membrane (with process $bud_n^{\perp}(0).\sigma$ or with the continuation $\sigma$.

## 5 Conclusion

In this paper we tackled the problem of defining a causal semantics for an instance of Brane Calculi, namely, the MBD calculus.

As already pointed out in [14], we think that the study of the causal dependencies that arise between the actions performed by a process is of primary importance for biologically inspired calculi, because of its possible application to the analysis of complex biological pathways.

This paper represents a first step in this direction, but a lot of work remains to be done. The next step is the study of the causal semantics for the PEP calculus, and its integration with the causal semantics for MBD. Then, we will move to the full Brane Calculus, that, besides the membrane-membrane interaction primitives of PEP and MBD, also contains objects representing free-floating molecules, and primitives for molecule-molecule and membrane-molecule interactions. When the definition of a causal semantics has been completed, we will start investigating the causal dependencies arising in biological pathways involving membranes, such as, e.g., the LDL Cholesterol Degradation Pathway [16], that has been modeled in the full Brane Calculus in [5].

We also plan to perform a thorough investigation of the properties that are enjoyed by the causal semantics, and possibly to refine the definition of the causal semantics in order to fulfill some of the properties. For example, a possible solution in order to obtain a causal semantics that partially enjoys the diamond property has been sketched in the previous section, and deserves further investigation.

We also plan to extend our investigation to other calculi/systems whose membranes are organized in a dynamically evolving hierarchical structure, such as, e.g., the Projective Brane Calculus [13] or Membrane Systems with active, evolving membranes.

## References

1. M. Boreale, D. Sangiorgi: A fully abstract semantics for causality in the $\pi$-Cclculus. *Acta Informatica*, 35, 5 (1998), 353–400. An extended abstract appeared in *Proc. STACS* 1995, 243–254.
2. N. Busi: On the computational power of the Mate/Bud/Drip Brane Calculus: interleaving vs. maximal parallelism. In *Proc 6th International Workshop on Membrane Computing* (WMC6), LNCS 3850, Springer, 2006.
3. N. Busi, R. Gorrieri: A Petri net semantics for $\pi$-calculus. In *Proc. Concur'95*, LNCS 962, Springer, 1995, 145–159.

4. N. Busi, G.M. Pinna: Comparing truly concurrent semantics for contextual place/transition nets with inhibitor and read arcs. *Fundam. Inform.*, 44, 3 (2000), 209–244.
5. N. Busi, C. Zandron: Modeling and analysis of biological processes by mem(brane) calculi and systems. In *Proceedings of the Winter Simulation Conference* (WSC 2006), ACM, 2006.
6. L. Cardelli: Brane Calculi - Interactions of biological membranes. In *Proc. Computational Methods in System Biology* 2004 (CMSB 2004), LNCS 3082, Springer, 2005.
7. L. Cardelli: *Abstract Machines for System Biology.* Draft, 2005.
8. L. Cardelli, A.D. Gordon: Mobile ambients. *Theoretical Computer Science*, 240, 1 (2000), 177–213.
9. Ph. Darondeau, P. Degano: Causal trees. In *Proc. ICALP'89*, LNCS 372, Springer, 1989, 234–248.
10. P. Degano, R. De Nicola, U. Montanari: Partial ordering descriptions and observations of nondeterministic concurrent processes. In *Proc. REX School/Workshop on Linear Time, Branching Time and Partial Order in Logic and Models of Concurrency*, LNCS 354, Springer, 1989, 438–466.
11. P. Degano, C. Priami: Causality for mobile processes. In *Proc. ICALP'95*, LNCS 944, Springer, 1995, 660–671.
12. P. Degano, C. Priami: Non interleaving semantics for mobile processes. *Theoretical Computer Science*, 216, 1-2 (1999), 237–270.
13. V. Danos, S. Pradalier: Projective brane calculus. In *Proc. Computational Methods in System Biology* 2004 (CMSB 2004), LNCS 3082, Springer, 2005.
14. M.L. Guerriero, C. Priami: *Causality and Concurrency in Beta-binders.* TR-01-2006 The Microsoft Research - University of Trento Centre for Computational and Systems Biology, 2006.
15. A. Kiehn: Proof systems for cause based equivalences. In *Proc. MFCS'93*, LNCS 711, Springer, 1993.
16. H. Lodish, A. Berk, P. Matsudaira, C.A. Kaiser, M. Krieger, M.P. Scott, S.L. Zipursky, J. Darnell: *Molecular Cell Biology.* W.H. Freeman and Company, 4th edition, 1999.
17. R. Milner: *Communication and Concurrency.* Prentice-Hall, 1989.
18. R. Milner, J. Parrow, D. Walker: A calculus of mobile processes. *Information and Computation*, 100 (1992), 1–77.
19. G. Păun: Computing with membranes. *Journal of Computer and System Sciences*, 61, 1 (2000), 108–143.
20. G. Păun: *Membrane Computing. An Introduction.* Springer, 2002.
21. C. Priami, P. Quaglia: Beta binders for biological interactions. In *Proc. of Computational Methods in Systems Biology*, LNCS 3082, Springer, 2005, 20–33.
22. C. Priami, P. Quaglia: Operational patterns in beta-binders. *T. Comp. Sys. Biology*, 1 (2005), 50–65.
23. A. Regev, E. M. Panina, W. Silverman, L. Cardelli, E. Shapiro: BioAmbients: An abstraction for biological compartments. *Theoretical Computer Science*, 325, 1 (2004), 141–167.