Implementing the Stochastics Brane Calculus in a Generic Stochastic Abstract Machine

Marino Miculan   Ilaria Sambarino

Department of Mathematics and Computer Science, University of Udine, Italy

marino.miculan@uniud.it   ilaria.sambarino@gmail.com

In this paper, we deal with the problem of implementing an abstract machine for a stochastic version of the Brane Calculus. Instead of defining an *ad hoc* abstract machine, we consider the generic stochastic abstract machine introduced by Lakin, Paulevé and Phillips. The nested structure of membranes is flattened into a set of species where the hierarchical structure is represented by means of *names*. In order to reduce the overhead introduced by this encoding, we modify the machine by adding a *copy-on-write* optimization strategy. We prove that this implementation is adequate with respect to the stochastic structural operational semantics recently given for the Brane Calculus. These techniques can be ported also to other stochastic calculi dealing with nested structures.

1 Introduction

A fundamental issue in Systems Biology is modelling the membrane interaction machinery. Several models have been proposed in the literature [11, 14, 3]; among them, the Brane Calculus [4] has been arisen as a good model focusing on abstract membrane interactions, still being sound with respect to biological constraints (e.g. bitonality). In this calculus, a process represents a system of nested compartments, where active components are *on* membranes, not inside them. This reflects the biological evidence that functional molecules (proteins) are embedded in membranes, with consistent orientation.

In the original definition of the Brane Calculus [4] (which we will recall in Section 2) membranes interact according to three basic reaction rules corresponding to *phagocytosis*, *endo/exocytosis*, and *pinocytosis*. However, this semantics does not take into account quantitative aspects, like stochastic distributions, which are important for, e.g., implementing stochastic simulations.

A stochastic semantics for the Brane Calculus has been provided in [2], following an approach pioneered in [5] (but see also [8, 12] for Markov processes). Instead of giving a stochastic version $P \xrightarrow{a,r} Q$ of the reaction relation, in this semantics each process is given a *measure* of the stochastic distribution of the possible outcomes. More precisely, we define a relation $P \to \mu$ associating to a process P an action-indexed family of measures μ: for an action a, the measure μ_a specifies for each measurable set S of processes, the rate $\mu_a(S) \in \mathbb{R}^+$ of a-transitions from P to (elements of) S. An advantage of this approach is that we can apply results from measure theory for solving otherwise difficult issues, like instance-counting problems; moreover, process measures are defined *compositionally*, and in fact the relation $P \to \mu$ can be characterized by means of a set of rules in a GSOS-like format. We will recall this stochastic semantics and its main properties in Section 3.

In this paper, we use this new semantics for defining a *stochastic abstract machine* for the Brane Calculus, so that it can be effectively used for *in silico* simulations of membrane systems. Defining an *ad hoc* abstract machine for the Brane Calculus would be a complex task; instead, we take advantage of the *generic abstract machine for stochastic process calculi* (GSAM for short) introduced in [13, 10] as a general tool for simulating a broad range of calculi. This machine can be instantiated to a particular

calculus by defining a function for transforming a process of the calculus to a set of *species*, and another for computing the set of possible reactions between species.

An important aspect is that this abstract machine does not have a native notion of compartment, which is central in the Brane Calculus (as in any other model of membranes). To overcome this problem, we adopt a "flat" representation of membrane systems, used also in [10], where the hierarchical structure is represented by means of *names*: each name represents a compartment, and each species is labelled with the name of the compartment where it is located, and the name of its inner compartment (if any). So names and species are the nodes and the arcs of the tree, respectively. This technique can be used for representing any system with a tree-like structure of compartments.

However, this approach does not scale well, as the population of species may grow enormously: for instance, a population of n identical cells would lead to n species, all differing only for the name of its inner compartment, instead of a single specie with multiplicity n. For circumventing this problem, in Section 4 we introduce a variant of the GSAM with a *copy-on-write* optimization strategy—hence called COWGSAM. The idea is to keep a single copy of each species, with its multiplicity; when a reaction has to be applied, fresh copies of the compartments involved are generated on-the-fly, and reactions and rates are updated accordingly. In this way, the hierarchical structure is unfolded only if and when needed.

In Section 5 we show how the Brane Calculus can be represented in the COWGSAM, and we will prove that the abstract machine obtained in this way is adequate with respect to the stochastic semantics of the Brane Calculus; in this proof, we take advantage of the compositional definition of this semantics.

Conclusions and final remarks are in Section 6.

2 Brane Calculus

In this section we recall Cardelli's Brane Calculus [4] focusing on its basic version (without communication primitives, complexes and replication).

First, let us fix the notation we will use hereafter. Let S be a set of *sorts* (or "types"), ranged over by s,t, and T a set of S-sorted terms; for $t \in S$, $T_t \subseteq T$ denotes the set of terms of sort t. For A a set of symbols, A^* denotes the set of finite words (or lists) over A, and $\langle a_1, \ldots, a_n \rangle$ denotes a word in A^*. For a word $\langle t_1, \ldots, t_n \rangle$ in S^*, we define $T_{\langle t_1, \ldots, t_n \rangle} \triangleq T_{t_1} \times \cdots \times T_{t_n}$.

Syntax The sorts and the set $\mathbb{B}$ of terms of Brane Calculus are the following:

Sorts :: S	$t ::= \mathsf{sys} \mid \mathsf{mem}$
Membranes :: $\mathbb{B}_{\mathsf{mem}}$	$\sigma, \tau ::= \mathbf{0} \mid \sigma\|\tau \mid \eth_n.\sigma \mid \eth_n^\perp(\tau).\sigma \mid \eth_n.\sigma \mid \eth_n^\perp.\sigma \mid \circledcirc_n(\tau).\sigma$
Systems :: $\mathbb{B}_{\mathsf{sys}}$	$P, Q ::= \diamond \mid P \circ Q \mid \sigma(\!|P|\!)$

The subscripted names n are taken from a countable set Λ. By convention we shall use M, N, ... to denote generic Brane Calculus terms in $\mathbb{B}$.

A *membrane* can be either the empty membrane $\mathbf{0}$, or the parallel composition of two membranes $\sigma|\tau$, or the action-prefixed membrane $\varepsilon.\sigma$. Actions are: *phagocytosis* $\eth$, *exocytosis* $\eth$, and *pinocytosis* $\circledcirc$. Each action but pinocytosis comes with a matching co-action, indicated by the superscript $\perp$.

A *system* can be either the empty system $\diamond$, or the parallel composition $P \circ Q$, or the system nested within a membrane $\sigma(\!|P|\!)$. Notice that, differently from [4], pino actions are indexed by names in Λ. In [4], names are meant only to pair up an action with its corresponding co-action, hence a pino action does not need to be indexed by any name. Actually, names can be thought of as an abstract representation

$$\overline{\mho_n^\perp(\rho).\tau|\tau_0(\!|Q|\!) \circ \mho_n.\sigma|\sigma_0(\!|P|\!) \Longrightarrow \tau|\tau_0(\!|\rho(\!|\sigma|\sigma_0(\!|P|\!)|\!) \circ Q|\!)} \quad \text{(red-phago)}$$

$$\overline{\mho_n^\perp.\tau|\tau_0(\!|\mho_n.\sigma|\sigma_0(\!|P|\!) \circ Q|\!) \Longrightarrow \sigma|\sigma_0|\tau|\tau_0(\!|Q|\!) \circ P} \quad \text{(red-exo)}$$

$$\overline{\circledcirc(\rho).\sigma|\sigma_0(\!|P|\!) \Longrightarrow \sigma|\sigma_0(\!|\rho(\!|\diamond|\!) \circ P|\!)} \quad \text{(red-pino)} \qquad \frac{P \Longrightarrow Q}{\sigma(\!|P|\!) \Longrightarrow \sigma(\!|Q|\!)} \quad \text{(red-loc)}$$

$$\frac{P \Longrightarrow Q}{P \circ R \Longrightarrow Q \circ R} \quad \text{(red-comp)} \qquad \frac{P \equiv P' \quad P' \Longrightarrow Q' \quad Q' \equiv Q}{P \Longrightarrow Q} \quad \text{(red-equiv)}$$

Table 1: Reduction semantics for the Brane Calculus.

of particular protein conformational shapes; hence, each name can correspond to a different biological behaviour. Therefore, if we want to observe also kinetic properties of processes, it is important to keep track of names in pino actions.

Terms can be rearranged according to a structural congruence relation; the intended meaning is that two congruent terms actually denote the same system. Structural congruence $\equiv$ is the smallest equivalence relation over $\mathbb{B}$ which satisfies the axioms and rules listed below.

$$P \circ Q \equiv Q \circ P \qquad P \circ (Q \circ R) \equiv (P \circ Q) \circ R \qquad P \circ \diamond \equiv P$$

$$\sigma|\tau \equiv \tau|\sigma \qquad \sigma|(\tau|\rho) \equiv (\sigma|\tau)|\rho \qquad \sigma|0 \equiv \sigma$$

$$\mathbf{0}(\!|\diamond|\!) \equiv \diamond \qquad \frac{P \equiv Q}{P \circ R \equiv Q \circ R} \qquad \frac{\sigma \equiv \tau}{\sigma|\rho \equiv \tau|\rho} \qquad \frac{P \equiv Q \quad \sigma \equiv \tau}{\sigma(\!|P|\!) \equiv \tau(\!|Q|\!)}$$

$$\frac{\alpha \in \{\mho_n, \mho_n, \mho_n^\perp\}_{n\in\Lambda} \quad \sigma \equiv \tau}{\alpha.\sigma \equiv \alpha.\tau} \qquad \frac{\beta \in \{\mho_n^\perp, \circledcirc_n\}_{n\in\Lambda} \quad \rho \equiv v \quad \sigma \equiv \tau}{\beta(\rho).\sigma \equiv \beta(v).\tau}$$

Differently from [4], we allow to rearrange also the sub-membranes contained in co-phago and pino actions (by means of the last inference rule above).

Reduction Semantics The dynamic behaviour of Brane Calculus is specified by means of a reduction semantics, defined over a reduction relation ("reaction") $\Longrightarrow \subseteq \mathbb{B}_{\mathsf{sys}} \times \mathbb{B}_{\mathsf{sys}}$, whose rules are listed in Table 1. Notice that the presence of (red-phago/exo/pino) and (red-equiv) makes this not a *structural* presentation, since these rules are not primitive recursive in the syntax (i.e., structural recursive) as required by the SOS format.

3 Stochastic Structural Operational Semantics for the Brane Calculus

In this section we recall the stochastic structural operational semantics for the Brane Calculus, as defined in [2]. Following [5], we replace the classic "pointwise" rules of the form $P \xrightarrow{a,r} P'$ with rules of the form $P \to \mu$, where μ is an indexed class of *measures* on the *measurable space* of processes. We assume the reader to be familiar with basic notions from measure theory; for a brief summary, see Appendix A.

$$\frac{}{\mathbf{0} \to_{\mathsf{mem}} \omega_{\mathsf{mem}}} \text{ (zero)} \qquad \frac{\alpha \in \{\eth_n, \mho_n, \mho_n^\perp \mid n \in \Lambda\}}{\alpha.\sigma \to_{\mathsf{mem}} [\alpha]_\sigma} \text{ (pref)}$$

$$\frac{\beta \in \{\eth_n^\perp, \circledcirc_n \mid n \in \Lambda\}}{\beta(\tau).\sigma \to_{\mathsf{mem}} [\beta]_\sigma^\tau} \text{ (pref-arg)} \qquad \frac{\sigma \to_{\mathsf{mem}} \mu' \qquad \tau \to_{\mathsf{mem}} \mu''}{\sigma \mid \tau \to_{\mathsf{mem}} \mu'_\sigma \oplus_\tau \mu''} \text{ (par)}$$

$$\frac{}{\diamond \to_{\mathsf{sys}} \omega_{\mathsf{sys}}} \text{ (void)} \qquad \frac{\sigma \to_{\mathsf{mem}} \nu \qquad P \to_{\mathsf{sys}} \mu}{\sigma(\!(P)\!) \to_{\mathsf{sys}} \mu @_P^\sigma \nu} \text{ (loc)} \qquad \frac{P \to_{\mathsf{sys}} \mu' \qquad Q \to_{\mathsf{sys}} \mu''}{P \circ Q \to_{\mathsf{sys}} \mu'_P \otimes_Q \mu''} \text{ (comp)}$$

Table 2: Stochastic structural operational semantics for Brane Calculus

The set of action labels for the Brane Calculus will be denoted by $\mathbb{A}$ and can be partitioned with respect to the source sort (i.e., either systems or membranes), as follows:

$$\mathbb{A}_{\mathsf{sys}} \triangleq \{id : \mathsf{sys} \to \langle \mathsf{sys} \rangle\} \cup \{\mathsf{ph}_n : \mathsf{sys} \to \langle \mathsf{sys}, \mathsf{sys} \rangle \mid n \in \Lambda\} \cup$$
$$\{\mathsf{ph}_n^\perp : \mathsf{sys} \to \langle \mathsf{mem}, \mathsf{mem}, \mathsf{sys}, \mathsf{sys} \rangle \mid n \in \Lambda\} \cup$$
$$\{\mathsf{ex}_n : \mathsf{sys} \to \langle \mathsf{mem}, \mathsf{sys}, \mathsf{sys} \rangle \mid n \in \Lambda\}$$
$$\mathbb{A}_{\mathsf{mem}} \triangleq \{\eth_n, \mho_n, \mho_n^\perp : \mathsf{mem} \to \langle \mathsf{mem} \rangle \mid n \in \Lambda\} \cup$$
$$\{\eth_n^\perp, \circledcirc_n : \mathsf{mem} \to \langle \mathsf{mem}, \mathsf{mem} \rangle \mid n \in \Lambda\}$$

Let a range over $\mathbb{A}$, and $ar(a)$ denote its arity. To ease the reading in the following we will use the notation $\Delta_a(T, \Sigma)$ to denote the set of measures $\Delta(T_{\langle t_1, \dots t_n \rangle}, \bigotimes_{i=1}^n \Sigma_{t_i})$, for $ar(a) = t \to \langle t_1, \dots, t_n \rangle$.

Let $\mathbb{B}/_{\equiv}$ be the set of $\equiv$-equivalence classes on $\mathbb{B}$. For $M \in \mathbb{B}$, we denote by $[M]_{\equiv}$ the $\equiv$-equivalence class of M (sometimes dropping the equivalence symbol when clear from the context).

Definition 3.1 (Measurable space of terms). *The* measurable space of terms $(\mathbb{B}, \Pi)$ *is given by the measurable space over* $\mathbb{B}$ *where* Π *is the* σ-algebra generated by $\mathbb{B}/_{\equiv}$.

Notice that $\mathbb{B}/_{\equiv}$ is a denumerable partition of $\mathbb{B}$, hence it is a base (a generator such that all its elements are disjoint) for Π. Any element of Π can be obtained by a countable union of elements of the base, i.e., for all $\mathcal{M} \in \Pi$ there exist $\{M_i\}_{i \in I}$, for some countable I, such that $\mathcal{M} = \bigcup_{i \in I} [M_i]_{\equiv}$. As a consequence, in order to generate the whole Π we can simply compute all these unions, without the need of any closure by complement.

A similar argument holds for the product space $(\mathbb{B}_{\langle t_1, \dots, t_n \rangle}, \bigotimes_{i=1}^n \Pi_{t_i})$, where $t_i \in \{\mathsf{mem}, \mathsf{sys}\}$ $(1 \leq i \leq n)$; indeed $\bigotimes_{i=1}^n \Pi_{t_i}$ can be generated from the base $\mathbb{B}_{\langle t_1, \dots, t_n \rangle}/_{\equiv_{\langle t_1, \dots, t_n \rangle}}$, where $\equiv_{\langle t_1, \dots, t_n \rangle} \subseteq \mathbb{B}_{\langle t_1, \dots, t_n \rangle} \times \mathbb{B}_{\langle t_1, \dots, t_n \rangle}$ is defined by

$$\langle M_1, \dots, M_n \rangle \equiv_{\langle t_1, \dots, t_n \rangle} \langle N_1, \dots, N_n \rangle \qquad \text{iff} \qquad M_i \equiv N_i, \text{ for all } 1 \leq i \leq n,$$

which can be easily checked to be an equivalence relation. $\equiv_{\langle t_1, \dots, t_n \rangle}$-equivalence classes are rectangles, i.e. $[\langle M_1, \dots, M_n \rangle]_{\equiv_{\langle t_1, \dots, t_n \rangle}} = [M_1]_{\equiv} \times \cdots \times [M_n]_{\equiv}$, therefore the product measure $\bigotimes_{i=1}^n \Pi_{t_i}$ is well defined. For sake of simplicity in the following we write $[\langle M_1, \dots, M_n \rangle]_{\equiv}$ in place of $[\langle M_1, \dots, M_n \rangle]_{\equiv_{\langle t_1, \dots, t_n \rangle}}$, and $\mathbb{B}_{\langle t_1, \dots, t_n \rangle}/_{\equiv}$ in place of $\mathbb{B}_{\langle t_1, \dots, t_n \rangle}/_{\equiv_{\langle t_1, \dots, t_n \rangle}}$.

The operational semantics associates with each membrane a family of measures in $\Delta^{\mathbb{A}_{\mathsf{mem}}}(\mathbb{B}, \Pi)$, and with each system a family of measures in $\Delta^{\mathbb{A}_{\mathsf{sys}}}(\mathbb{B}, \Pi)$. This can be represented by two relations

$\to_{\text{mem}}\colon T_{\text{mem}} \to \Delta^{\mathbb{A}_{\text{mem}}}(\mathbb{B},\Pi)$, $\to_{\text{sys}}\colon T_{\text{sys}} \to \Delta^{\mathbb{A}_{\text{sys}}}(\mathbb{B},\Pi)$, defined by the SOS rules listed in Table 2. (In the following, for sake of readability, we will drop the indexes $_{\text{mem,sys}}$). In these rules we use some constants and operations over indexed families of measures, that we define next. For a set (of labels) A, let us denote by $\Delta^A(\mathbb{B},\Pi)$ the set $\prod_{a\in A}\Delta_a(\mathbb{B},\Pi)$ of A-indexed families of measures over $(\mathbb{B},\Pi)$. Given a family of measures $\mu \in \Delta^A(\mathbb{B},\Pi)$ and $a \in A$, the a-component of μ will be denoted as $\mu_a \in \Delta_a(\mathbb{B},\Pi)$.

Null: Let $\omega_{\text{mem}} \in \Delta^{\mathbb{A}_{\text{mem}}}(\mathbb{B},\Pi)$ be the constantly zero measure, i.e., for all $a\in\mathbb{A}_{\text{mem}}$ such that $ar(a) = t \to \langle t_1,\dots,t_n\rangle$ and $\mathcal{M} \in \bigotimes_{i=1}^n \Pi_{t_i}$: $(\omega_{\text{mem}})_a(\mathcal{M})=0$.

Prefix: For arbitrary $n \in \Lambda$, $\alpha \in \{\eth, \circlearrowleft, \circlearrowleft^\perp\}$, and $\beta \in \{\eth^\perp, \circledcirc\}$, let the constants $[\alpha_n]_\sigma, [\beta_n]_\sigma^\tau \in \Delta^{\mathbb{A}_{\text{mem}}}(\mathbb{B},\Pi)$ be defined, for arbitrary $X,Y \in \mathbb{B}_{\text{mem}}/_{\equiv}$, as follows:

$$([\alpha_n]_\sigma)_{\alpha_m}(X) = \begin{cases} \iota(n) & \text{if } n=m \text{ and } \sigma \in X \\ 0 & \text{otherwise} \end{cases}$$

$$([\alpha_n]_\sigma)_{\beta_m}(X \times Y) = 0$$

$$([\beta_n]_\sigma)_{\alpha_m}(X) = 0$$

$$([\beta_n]_\sigma^\tau)_{\beta_m}(X \times Y) = \begin{cases} \iota(n) & \text{if } n=m \text{ and } \sigma \in X,\ \tau \in Y \\ 0 & \text{otherwise} \end{cases}$$

Parallel: For $\mu,\mu' \in \Delta^{\mathbb{A}_{\text{mem}}}(\mathbb{B},\Pi)$, let $\mu_\sigma \oplus_\tau \mu' \in \Delta^{\mathbb{A}_{\text{mem}}}(\mathbb{B},\Pi)$ be defined, for $n \in \Lambda$, $\alpha \in \{\eth, \circlearrowleft, \circlearrowleft^\perp\}$, $\beta \in \{\eth^\perp, \circledcirc\}$, and $X,Y \in \mathbb{B}_{\text{mem}}/_{\equiv}$, as follows (where for all X,τ: $X_{|\tau} \triangleq \bigcup\{[\sigma]_{\equiv} \mid \sigma|\tau \in X\}$):

$$(\mu_\sigma \oplus_\tau \mu')_{\alpha_n}(X) = \mu_{\alpha_n}(X_{|\tau}) + \mu'_{\alpha_n}(X_{|\sigma})$$

$$(\mu_\sigma \oplus_\tau \mu')_{\beta_n}(X \times Y) = (\mu)_{\beta_n}(X_{|\tau} \times Y) + (\mu')_{\beta_n}(X_{|\sigma} \times Y)$$

Void: Let $\omega_{\text{sys}} \in \Delta^{\mathbb{A}_{\text{sys}}}(\mathbb{B},\Pi)$ be defined by $(\omega_{\text{sys}})_a(\mathcal{M}) = 0$ for any $a \in \mathbb{A}_{\text{sys}}$, such that $ar(a) = t \to \langle t_1,\dots,t_n\rangle$, and $\mathcal{M} \in \bigotimes_{i=1}^n \Pi_{t_i}$.

Nesting: For $\nu \in \Delta^{\mathbb{A}_{\text{mem}}}(\mathbb{B},\Pi)$ and $\mu \in \Delta^{\mathbb{A}_{\text{sys}}}(\mathbb{B},\Pi)$, let $\mu @_P^\sigma \nu \in \Delta^{\mathbb{A}_{\text{sys}}}(\mathbb{B},\Pi)$ be defined, for $X,Y \in \mathbb{B}_{\text{mem}}/_{\equiv}$ and $Z,W \in \mathbb{B}_{\text{sys}}/_{\equiv}$, as follows:

$$(\mu @_P^\sigma \nu)_{\text{ph}_n}(Z \times W) = \begin{cases} \nu_{\eth_n}([\sigma]_{\equiv}) & \text{if } \sigma\langle\!(P)\!\rangle \in Z \text{ and } \diamond \in W \\ 0 & \text{otherwise} \end{cases}$$

$$(\mu @_P^\sigma \nu)_{\text{ph}_n^\perp}(X \times Y \times Z \times W) = \begin{cases} \nu_{\eth_n^\perp}(X \times Y) & \text{if } P \in Z \text{ and } \diamond \in W \\ 0 & \text{otherwise} \end{cases}$$

$$(\mu @_P^\sigma \nu)_{\text{ex}_n}(X \times Z \times W) = \begin{cases} \nu_{\circlearrowleft_n}(X) & \text{if } P \in Z \text{ and } \diamond \in W \\ 0 & \text{otherwise} \end{cases}$$

$$(\mu @_P^\sigma \nu)_{id}(X) = \mu_{id}(X_{\sigma\langle\!(\rangle\!)}) + \sum_{X'\langle\!(X''\langle\!([\diamond]_{\equiv}\rangle\!)\circ[P]_{\equiv}\rangle\!)=X}^{n\in\Lambda} \nu_{\circledcirc_n}(X' \times X'') +$$

$$\sum_{X'|X''\langle\!(Y''\rangle\!)\circ Y'=X}^{n\in\Lambda} \frac{\mu_{\text{ex}_n}(X' \times Y' \times Y'') \cdot \nu_{\eth_n^\perp}(X'')}{\iota(n)}$$

87

Composition: For $\mu, \mu' \in \Delta^{\mathbb{A}_{\mathsf{sys}}}(\mathbb{B}, \Pi)$, let $\mu_P \otimes_Q \mu' \in \Delta^{\mathbb{A}_{\mathsf{sys}}}(\mathbb{B}, \Pi)$ be defined, for $X, Y \in \mathbb{B}_{\mathsf{mem}}/_{\equiv}$ and $Z, W \in \mathbb{B}_{\mathsf{sys}}/_{\equiv}$, as follows (where for all $W, Q, W_{\circ Q} \triangleq \bigcup\{[P]_{\equiv} \mid P \circ Q \in W\}$):

$$(\mu_P \otimes_Q \mu')_{\mathrm{ph}_n}(Z \times W) = \mu_{\mathrm{ph}_n}(Z \times W_{\circ Q}) + \mu'_{\mathrm{ph}_n}(Z \times W_{\circ P})$$

$$(\mu_P \otimes_Q \mu')_{\mathrm{ph}_n^{\perp}}(X \times Y \times Z \times W) = \mu_{\mathrm{ph}_n^{\perp}}(X \times Y \times Z \times W_{\circ Q}) + \mu'_{\mathrm{ph}_n^{\perp}}(X \times Y \times Z \times W_{\circ P})$$

$$(\mu_P \otimes_Q \mu')_{\mathrm{ex}_n}(X \times Z \times W) = \mu_{\mathrm{ex}_n}(X \times Z \times W_{\circ Q}) + \mu'_{\mathrm{ex}_n}(X \times Z \times W_{\circ P})$$

$$(\mu_P \otimes_Q \mu')_{id}(X) = \mu_{id}(X_{\circ Q}) + \mu'_{id}(X_{\circ P}) +$$

$$\sum_{X_1 \langle\!\langle X_2 \langle\!\langle Y_1 \rangle\!\rangle \circ Z_1 \rangle\!\rangle \circ Y_2 \circ Z_2 = X}^{n \in \Lambda} \frac{\mu_{\mathrm{ph}_n}(Y_1 \times Y_2) \cdot \mu'_{\mathrm{ph}_n^{\perp}}(X_1 \times X_2 \times Z_1 \times Z_2)}{\iota(n)} +$$

$$\sum_{X_1 \langle\!\langle X_2 \langle\!\langle Z_1 \rangle\!\rangle \circ Y_1 \rangle\!\rangle \circ Z_2 \circ Y_2 = X}^{n \in \Lambda} \frac{\mu_{\mathrm{ph}_n^{\perp}}(X_1 \times X_2 \times Y_1 \times Y_2) \cdot \mu'_{\mathrm{ph}_n}(Z_1 \times Z_2)}{\iota(n)}$$

These operators have nice algebraic properties (e.g., $\mu'_{\sigma} \mathbb{O}_{\tau} \mu'' = \mu''_{\tau} \mathbb{O}_{\sigma} \mu'$, $(\mu'_{\sigma} \mathbb{O}_{\tau} \mu'')_{\sigma | \tau} \mathbb{O}_{\rho} \mu''' = \mu'_{\sigma} \mathbb{O}_{\tau | \rho} (\mu''_{\tau} \mathbb{O}_{\rho} \mu''')$, ...), and respect the structural congruence (e.g., if $P \equiv P'$ and $Q \equiv Q'$ then $\mu'_P \otimes_Q \mu'' = \mu'_{P'} \otimes_{Q'} \mu''$). We refer to [2] for further details about these properties, very useful in calculations.

The next lemmata state that the stochastic transition relation $\rightarrow$ (and hence the operational semantics) is well-defined and consistent, that is, for each process we have exactly one family of measures of its continuations, and this family respects structural congruence.

Lemma 3.2 (Uniqueness). *For $a \in \mathbb{A}$ such that $ar(a) = t \rightarrow \langle t_1, \ldots, t_n \rangle$, and $M \in \mathbb{B}_t$, there exists a unique $\mu \in \Delta^{\mathbb{A}_t}(\mathbb{B}, \Pi)$ such that $M \rightarrow \mu$.*

Lemma 3.3. *If $M \equiv N$ and $M \rightarrow \mu$, then $N \rightarrow \mu$.*

This operational semantics can be used to define the "traditional" pointwise semantics:

$$M \xrightarrow{a,r} \langle M_1, \ldots, M_n \rangle \iff M \rightarrow \mu \text{ and } \mu_a([\langle M_1, \ldots, M_n \rangle]_{\equiv}) = r$$

and it is conservative with respect to the non-stochastic reduction semantics.

Proposition 3.4. *For all systems $P, Q \in \mathbb{B}_{\mathsf{mem}}$, if $P \rightarrow \mu$ and $\mu_{id}([Q]) > 0$ then $P \Longrightarrow Q$.*

4 The COW Generic Stochastic Abstract Machine

In this section we present a variant of the generic stochastic abstract machine (GSAM), oriented to systems with nested compartments.

The GSAM has been introduced in [13, 10] for simulating a broad range of process calculi with an arbitrary reaction-based simulation algorithm. Although it does not have a native notion of "compartment", nested systems can represented by "flattening" all compartments and species into a single multiset, where each species is tagged with names representing their position in the hierarchy, as shown in [10]. The idea is to represent each compartment as a different species, keeping track of their position in the hierarchy by means of *(node) names*. These names are ranged over by $x, y, z, \ldots$, and are different from names in actions. As an example, a system $\sigma \langle\!\langle \tau \langle\!\langle \rangle\!\rangle \rangle\!\rangle$ is represented as the multiset $\{\sigma \langle\!\langle_y^x \rangle\!\rangle \mapsto 1, \tau \langle\!\langle_z^y \rangle\!\rangle \mapsto 1\}$, which means "there is one cell with membrane σ located in the compartment x and whose compartment is y, and one cell with membrane τ positioned in y and whose compartment is z". Reactions can happen only if the names tagging the involved species match according to the required nesting structure.

Unfortunately, this approach does not scale well, as the population of species grows. Let us consider a system composed of n copies of the same cell, e.g., $n \cdot (\sigma ⦅ ⦆)$ (where n can be easily in the order of 10^3–10^4). In the original GSAM idea, this should be represented in the machine as a single species with multiplicity n, and each possible reaction is represented once but with propensity given by the law of mass action taking into account the species' multiplicity n. Instead, the "flat encoding" above yields n different species $\sigma ⦅_{y_1}^x ⦆, \ldots, \sigma ⦅_{y_n}^x ⦆$, each with multiplicity 1; the set of reactions explodes correspondingly.

For circumventing this problem we introduce a variant of the GSAM with a *copy-on-write* strategy—hence it is called COWGSAM. The idea is to keep a single copy of each species, with its multiplicity; the same applies to reactions. When a reaction has to be applied, the compartments involved are "unfolded", i.e., fresh copies of the compartments are generated and the reaction set is modified accordingly; then, the reaction can be applied. In this way, the hierarchical structure is unfolded only if and when needed.

In order to implement this idea, we have to modify the notion of machine term, reaction and reaction rule. The COWGSAM (with the Next Reaction method) is shown in Figure 1.

First, for generating fresh names, we have to keep track of those already allocated. To this end we introduce *environments*, which are finite sets of names. Then, the machine state is represented by a *machine term T*, i.e. a quadruple $E \vdash (t, S, R)$ where E is an environment; t is the current time; S is a finite function mapping each species I to its population $S(I)$ (if not null); and R maps each reaction O to its activity A, which is used to compute the next reaction. (Notice that the syntax of species I is left unspecified, as it depends on the specific process calculus one has to implement.) We say that a machine term $E \vdash (t, S, R)$ is *well-formed* if for all $x_i, x_j \in E : x_i = x_j \Rightarrow i = j$, and the free names in S, R appear in E. In the following, we assume that machine terms are well formed.

Each reaction is a quadruple (S_1, r, f, S_2), basically representing a reaction $S_1 \xrightarrow{r} S_2$, where

- S_1 and S_2 denote the *reactant* population and *product* population respectively;

- r denotes the rate (in s^{-1}) of the reaction;

- f is a function mapping machine terms to machine terms; this functions implements any global update of the machine term after the reaction (if needed).

The *transitions* of the abstract machine are represented by a relation $T \xrightarrow{a, O} T'$ between machine terms, indexed by the propensity a and reaction rules. This should be read as "T goes to T' with rate a, using the rule O". This relation is defined by (Reaction rule) in Figure 1, where the function $next(T)$ selects the next reaction, i.e. it returns a pair (O, t') where $O = (S_1, r, f, S_2)$ is the reaction to happen first among all possible reactions in T, and t' is the new time of the system. Once the reaction has been selected, we have first to create the separate (private) copies of the compartments involved, and to update the reaction set accordingly. This is done by the functions $cow(\_)$ and $dup(\_)$, which implement a deep copy-on-write: $cow(E \vdash (t, S, R), S_1)$ is a machine term $E' \vdash (t, S', R')$ representing the same state as $E \vdash (t, S, R)$, but in S' the species indicated in S_1 are unfolded; E' contains all names which have been generated in the process, and R' is the new set of reactions. (Actually, the freshly generated copies represent the instances which are *not* involved by the reaction; this simplifies the reaction application.) An example of the action of $cow(\_)$ is depicted in Figure 2.

At this point, the reaction is executed, by removing the reactants S_1 from the machine term (via the operation $\ominus$), adding the products S_2 (via the operation $\oplus$) and updating the current time of the machine. The function f performs any global "clean-up" and restructuring to the machine term that may be required by the reaction (e.g., garbage collection, elimination of names not used anymore,...). Moreover, since a reaction can rearrange the hierarchy structure, possibly creating new compartments and deleting others, we have to add to the environment any fresh name introduced in the products.

$$T ::= E \vdash (t, S, R) \qquad \text{(Machine term)}$$
$$E ::= x_1, \ldots, x_n \qquad \text{(Environments)}$$
$$S ::= \{I_1 \mapsto i_1, \ldots, I_N \mapsto i_N\} \qquad \text{(Populations)}$$
$$R ::= \{O_1 \mapsto A_1, \ldots, O_N \mapsto A_N\} \qquad \text{(Reactions)}$$
$$O ::= (S_1, r, f, S_2) \qquad \text{(Reaction)}$$

$$\frac{((S_1, r, f, S_2), a, t') = next(t, S, R) \quad E' \vdash (t, S', R') = cow(E \vdash (t, S, R), S_1)}{E \vdash (t, S, R) \xrightarrow{a, (S_1, r, f, S_2)} norm(E' \cup fn(S_2) \vdash (f(S_2 \oplus ((t', S', R') \ominus S_1))))} \qquad \text{(Reaction rule)}$$

$$next(t, S, R) \triangleq (O, a, t') \quad \text{if } R(O) = (t', a) \text{ and } t' = \min\{t \mid R(O) = (t, a)\}$$

$$cow(E \vdash (t, S, R), \emptyset) \triangleq E \vdash (t, S, R)$$
$$cow(E \vdash (t, S, R), \{\rho\, \mathbb{C}_z^y \mathbb{D} \mapsto j\} \cup S_1) \triangleq cow(E' \cup fn(S'') \vdash (t, S'', R'' \cup init(L' \cup L'', (t, S', R)), S_1) \quad \text{where}$$
$$\text{if } S(\sigma\, \mathbb{C}_y^x \mathbb{D}) > 1 \text{ for some } \sigma, x :$$
$$\text{let } i = S(\sigma\, \mathbb{C}_y^x \mathbb{D}) \text{ and } y' \notin E$$
$$(S', R') = dup(E \vdash (t, S\{\sigma\, \mathbb{C}_y^x \mathbb{D} \mapsto 1, \sigma\, \mathbb{C}_{y'}^x \mathbb{D} \mapsto i - 1\}, R), y, y')$$
$$L' = reactions(\sigma\, \mathbb{C}_{y'}^x \mathbb{D} \mapsto i - 1, S)$$
$$\text{otherwise let } (S', R') = (S, R), L' = \emptyset \text{ in}$$
$$\text{let } n = S(\rho\, \mathbb{C}_z^y \mathbb{D}), E' = E \cup fn(S'), \text{ and } z' \notin E'$$
$$(S'', R'') = dup(E' \vdash (t, S'\{\rho\, \mathbb{C}_z^y \mathbb{D} \mapsto j, \rho\, \mathbb{C}_{z'}^y \mathbb{D} \mapsto n - j\}, R'), z, z')$$
$$L'' = reactions(\rho\, \mathbb{C}_{z'}^y \mathbb{D} \mapsto n - j, S')$$

$$dup(E \vdash (t, S, R), y, y') \triangleq (S' \cup S'', R \cup R'' \cup init(L, (t, S', R)))$$
$$\text{where } S' = S \cup \{\rho\, \mathbb{C}_{w'}^{y'} \mathbb{D} \mapsto i \mid S(\sigma\, \mathbb{C}_y^z \mathbb{D}) > 0, S(\rho\, \mathbb{C}_w^y \mathbb{D}) = i\}, w' \notin E,$$
$$L = reactions(\rho\, \mathbb{C}_{w'}^{y'} \mathbb{D} \mapsto i, S'),$$
$$(S'', R'') = dup(E \cup \{w'\} \vdash (t, S, R), w, w')$$

Figure 1: The COW Generic Stochastic Abstract Machine, with the Next Reaction method.

Finally, the term can be "normalized" by collapsing equivalent copies of the same subtree into a single copy (with the right multiplicity), by the function $norm(\_)$. In its simplest form, $norm(\_)$ can be the identity, i.e., no normalization is performed at all. Although this is correct, it can lead to unnecessary copies of the same subtrees. We can define $norm(\_)$ such that

$$norm(\{I_1 \mapsto i_1, I_2 \mapsto i_2\} \cup S, R) = norm(\{I_1 \mapsto i_1 + i_2\} \cup S, R[I_2/I_1]) \quad \text{if } I_1 \equiv I_2$$

where the equivalence between species can be implemented by comparing the subtrees starting from I_1, I_2, e.g., by calculating a suitable hash value. We leave this refinement as future development.

In order to implement the Next Reaction method, each reaction O is associated with a pair $R(O) = (a, t)$, where a is the reaction *propensity* and t is the time at which the reaction is supposed to occur. The function $delay(r, a)$ computes a time interval from a random variable with rate r and propensity a.

$$\{I_1 \mapsto i_1, \dots, I_N \mapsto i_N\} \oplus (t, S, R) \triangleq I_1 \mapsto i_1 \oplus \dots \oplus I_N \mapsto i_N \oplus (t, S, R)$$

$$I \mapsto i \oplus (t, S, R) \triangleq \begin{cases} (t, S', R \cup updates(I, (t, S', R))) & \text{if } S(I) = i' \text{ and } S' = S\{I \mapsto i' + i\} \\ (t, S', R \cup init(L, (t, S', R))) & \text{if } I \notin dom(S), S' = S\{I \mapsto i\} \\ & \text{and } L = reactions(I \mapsto i, S) \end{cases}$$

$$(t, S, R) \ominus \{I_1 \mapsto i_1, \dots, I_N \mapsto i_N\} \triangleq (t, S, R) \ominus I_1 \mapsto i_1 \ominus \dots \ominus I_N \mapsto i_N$$

$$(t, S, R) \ominus I \mapsto i \triangleq (t, S', R \cup updates(I, (t, S', R)))$$
$$\text{if } S(I) = i', i' \geq i \text{ and } S' = S\{I \mapsto i' - i\}$$

$$init(L, (t, S, R)) \triangleq \{O \mapsto (t', a) \mid O \in L \text{ and } a = propensity(O, S) \text{ and}$$
$$O = (S_1, r, f, S_2) \text{ and } t' = t + delay(r, a)\}$$

$$updates(I, (t, S, R)) \triangleq \{O \mapsto (t', a') \mid R(O) = (t'', a) \text{ and } O = (S_1, r, f, S_2) \text{ and } S_1(I) > 0 \text{ and}$$
$$\text{if } t'' > t \text{ then } a' = propensity(O, S) \text{ and } t' = t + (a/a')(t'' - t)$$
$$\text{if } t'' = t \text{ then } a' = propensity(O, S) \text{ and } t' = t + delay(r, a)\}$$

$$propensity((S_1, r, f, S_2), S) \triangleq r \cdot \binom{S^*(I_1)}{j_1} \cdot \dots \cdot \binom{S^*(I_n)}{j_n} \text{ if } S_1 = \{I_1 \mapsto j_1, \dots, I_n \mapsto j_n\}$$

$$S^*(\sigma \mathbb{C}_y^x \mathbb{D}) \triangleq \begin{cases} S(\sigma \mathbb{C}_y^x \mathbb{D}) & \text{if } x = root \\ S(\sigma \mathbb{C}_y^x \mathbb{D}) \cdot S^*(\rho \mathbb{C}_x^z \mathbb{D}) & \text{if } x \neq root \text{ and } S(\rho \mathbb{C}_x^z \mathbb{D}) > 0 \end{cases}$$

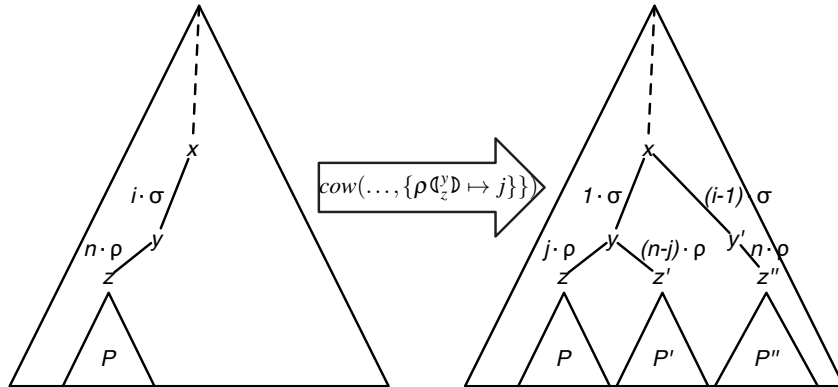Figure 1: The COW Generic Stochastic Abstract Machine (cont.).

Figure 2: Example copy-on-write of a subterm.

This general structure can be instantiated with a given process calculus, just by providing the definition for the missing parts. Given a set *Proc* of processes, we have to define:

1. the syntax of the species I (which may be different from that of processes);

2. a function $species_{E,x}(P)$ mapping a process $P \in Proc$ to a species set located in x;

3. a function $reactions(I \mapsto i, S)$ for computing the multiset of reactions between a (new) species I with multiplicity i and a population of (existing) species S.

The function *species* is used to initialise the abstract machine at the beginning of a simulation. If we aim to simulate the execution of a process $P \in Proc$, the corresponding initial state (rooted in *root*) is

$$(\!|P|\!)_{root} \triangleq fn(J) \vdash J \oplus (0, \emptyset, \emptyset) \quad \text{where } J = species_{\emptyset,root}(P).$$

The *reactions* function is used to adjust the set of possible reactions dynamically.

5 Implementing the Stochastic Brane Calculus in COWGSAM

In this section, we show how the COW Generic Stochastic Abstract Machine can be used to implement the Stochastic Brane Calculus, following the protocol described in the previous section.

5.1 Encoding of the Stochastic Brane Calculus

Syntax of species We define the species for the brane calculus, which in turn lead us to introduce *complexes* and *actions*. Notice that (despite the deceiving syntax) species are not systems and complexes are not membranes; nevertheless, actions are a subset of membranes.

$$I ::= C(\!|_y^x|\!) \qquad\qquad\qquad\qquad \text{(Species)}$$
$$C ::= A_1, \ldots, A_n \qquad\qquad\qquad\qquad \text{(Complexes)}$$
$$A ::= \eth_n.\sigma \mid \eth_n^\perp(\tau).\sigma \mid \eth_n.\sigma \mid \eth_n^\perp.\sigma \mid \circledcirc_n(\tau).\sigma \qquad \text{(Actions)}$$

Node names can appear in the species in S and in reactions R.

The *species* function We can now provide the definition of the translation of a process $P \in \mathbb{B}_{\text{sys}}$ into a species set. Basically, each compartment is assigned a different, fresh node name; therefore, the function $species_{E,x}(P)$ is parametric in the set E of allocated node names, and the name $x \in E$ to be used as the location of the system P. The name x changes as we descend the compartment hierarchy.

In order to capture correctly the multiplicity of each species, we assume that systems are in *normal form*. Basically, this form is a shorthand for products where n copies of the same system, i.e., $P \circ \cdots \circ P$, are represented as $n \cdot P$.

$$\text{Normal Systems} :: \mathbb{B}_{\text{sys}}^n \qquad Q ::= n_1 \cdot \sigma_1 (\!|Q_1|\!) \circ \ldots \circ n_k \cdot \sigma_k (\!|Q_k|\!)$$
$$\text{where } k \geq 0 \text{ and for } i \neq j : \sigma_i (\!|Q_i|\!) \not\equiv \sigma_j (\!|Q_j|\!)$$

A system in normal form can be translated into a system just by unfolding the products. For $Q \in \mathbb{B}_{\text{sys}}^n$ a system in normal form, let $\lceil Q \rceil \in \mathbb{B}_{\text{sys}}$ defined as follows:

$$\lceil \diamond \rceil = \diamond \qquad \lceil n \cdot \sigma (\!|Q'|\!) \circ Q'' \rceil = \overbrace{\sigma (\!|\lceil Q' \rceil|\!) \circ \cdots \circ \sigma (\!|\lceil Q' \rceil|\!)}^{n \text{ times}} \circ \lceil Q'' \rceil$$

Proposition 5.1. *For all $P \in \mathbb{B}_{\text{sys}}$, there exists a system in normal form $Q \in \mathbb{B}_{\text{sys}}^n$ such that $\lceil Q \rceil \equiv P$.*

As a consequence, we can give the definition of *species* on systems in normal form, as follows:

$$species_{E,x}(\diamond) \triangleq \emptyset$$
$$species_{E,x}(n \cdot \sigma (\!|Q_1|\!) \circ Q_2) \triangleq \{s(\sigma)(\!|_y^x|\!) \mapsto n\} \cup species_{E \uplus \{y\},y}(Q_1) \cup species_{E,x}(Q_2)$$
$$\text{with } y \notin E \text{ and } fn(species_{E \uplus \{y\},y}(Q_1)) \cap fn(species_{E,x}(Q_2)) \subseteq \{x\}$$

The condition in the second case ensures that two different compartments are never given the same name—any name clash has to be resolved by an α-conversion. The function $s(\_)$ converts a membrane into a set of complexes:

$$s(\mathbf{0}) \triangleq \emptyset \qquad s(\sigma|\sigma') \triangleq s(\sigma) \cup s(\sigma') \qquad s(\pi.\sigma) \triangleq \{\pi.\sigma\}$$

The *reactions* function   The next step is to define the function $reactions(I \mapsto i, S)$, for I a species with multiplicity i and S a population.

$$reactions_E(I_1 \mapsto i, S) \triangleq unary_E(I_1) \cup binary_E(I_1, S)$$

$$unary_E(I_1) \triangleq \{(\{I_1 \mapsto 1\}, r_n, id, \{(s(\sigma) \cup U_1')\mathbb{C}_y^x\mathbb{D} \mapsto 1, s(\rho)\mathbb{C}_w^y\mathbb{D} \mapsto 1\}) \mid$$
$$I_1 = U_1\mathbb{C}_y^x\mathbb{D}, \; U_1 = \{\circledcirc_n(\rho).\sigma\} \cup U_1', w \notin E\}$$

$$binary_E(I_1, S) \triangleq \{(\{I_1 \mapsto 1, I_2 \mapsto 1\}, r_n, f, \{(s(\tau) \cup U_1' \cup s(\sigma) \cup U_2')\mathbb{C}_y^x\mathbb{D} \mapsto 1\}) \mid I_2 \in dom(S),$$
$$(I_1 = U_1\mathbb{C}_y^x\mathbb{D}, I_2 = U_2\mathbb{C}_w^y\mathbb{D}) \vee (I_2 = U_1\mathbb{C}_y^x\mathbb{D}, I_1 = U_2\mathbb{C}_w^y\mathbb{D}),$$
$$U_1 = \{\circleddash_n^\perp.\tau\} \cup U_1', U_2 = \{\circleddash_n.\sigma\} \cup U_2', f = \lambda T.T[w := x]\} \; \uplus$$
$$\{(\{I_1 \mapsto 1, I_2 \mapsto 1\}, r_n, id,$$
$$\{(s(\tau) \cup U_1')\mathbb{C}_y^x\mathbb{D} \mapsto 1, s(\rho)\mathbb{C}_w^y\mathbb{D} \mapsto 1, (s(\sigma) \cup U_2')\mathbb{C}_z^w\mathbb{D} \mapsto 1\}) \mid I_2 \in dom(S),$$
$$(I_1 = U_1\mathbb{C}_y^x\mathbb{D}, I_2 = U_2\mathbb{C}_z^x\mathbb{D}) \vee (I_2 = U_1\mathbb{C}_y^x\mathbb{D}, I_1 = U_2\mathbb{C}_z^x\mathbb{D}),$$
$$U_1 = \{\circleddash_n^\perp(\rho).\tau\} \cup U_1', U_2 = \{\circleddash_n.\sigma\} \cup U_2', \; w \notin E\}$$

In the case of Brane Calculus, the unary reactions are only those arising from pinocytosis, while binary reactions arise from exocytosis and phagocytosis. In both cases, the multiplicity of each reactant is 1, so the multiplicity of I_1 is not relevant. Exocytosis merges two compartments; this is reflected by the fact that the "rearranging" function f substitutes every occurrence of the name w in T with x. On the other hand, pinocytosis and phagocytosis create new compartments; to represent the new structure, we choose a fresh name w representing the new intermediate nesting level, and reconnect the various compartments accordingly. Therefore, for any reaction $(S_1, r, f, S_2) \in reactions_E(I_1 \mapsto i, S)$, $fn(S_2) \setminus E$ is either $\emptyset$ (in the case of exocytosis) or the singleton $\{w\}$.

5.2 Adequacy results

Before proving the correctness of our implementation, we have to define how to translate a species set back to a system of the brane calculus.

Let S be a non empty species set. A *root name* of S, denoted by $root(S)$, is a name x such that $S(C\mathbb{C}_y^x\mathbb{D}) > 0$ for some C, y, and for all $z, C' : C'\mathbb{C}_x^z\mathbb{D} \notin dom(S)$. The next result states that $root(\_)$ is well defined on the species sets we encounter during a simulation.

Lemma 5.2. *For all* $P \in \mathbb{B}_{\mathsf{sys}}$:

1. *if* $P \neq \diamond$: $root(\langle\!\langle P\rangle\!\rangle_x) = x$.

2. *if* $\langle\!\langle P\rangle\!\rangle_x \xrightarrow{a,O} E \vdash (t, S, R)$ *and* $S \neq \emptyset$, *then* $root(S) = x$.

Proof. (1.) is trivial by definition. (2.) It is enough to check that the reaction rules introduced by $reactions(\_)$ do not change the name of the root, nor introduce new ones. $\square$

We can now define a function $[\![\,\_\,]\!]$ which maps complexes to membranes, and a function $[\![\,\_\,]\!]_x$ mapping species sets to systems; the latter is parametric in the name x of the root of the system.

$$[\![A_1,\ldots,A_n]\!] \triangleq A_1|\ldots|A_n \qquad [\![S]\!]_x \triangleq \prod_{C\mathfrak{C}_y^x\mathfrak{D}\in dom(S)} S(C\mathfrak{C}_y^x\mathfrak{D})\cdot([\![C]\!]\mathfrak{C}[\![S]\!]_y\mathfrak{D})$$

where the notation $n\cdot P$ is a shorthand for $P\circ\ldots\circ P$, n times.

Lemma 5.3. *For all $P\in\mathbb{B}_{\mathsf{sys}}$:*

1. $[\![([\![P]\!])_x]\!]_x \equiv P$.

2. if $([\![P]\!])_x \xrightarrow{a,O} E\vdash (t,S,R)$ then $[\![S]\!]_x$ is well defined.

Proof. (1.) is easy. (2.) It is enough to check that the reaction rules introduced by *reactions*($\_$) do not introduce loops (i.e., the order among names is well founded). $\qquad\square$

We can now state and prove the main results of this section.

Proposition 5.4 (Soundness). *For all $P\in\mathbb{B}_{\mathsf{sys}}$, if $([\![P]\!])_x \xrightarrow{a,O} E\vdash (t,S,R)$ then there exists μ such that $P\to\mu$ and $\mu_{id}([\![[\![S]\!]_x]\!]) = a$.*

Proof. The proof is by cases on which reaction rule O is selected by the function *next*. By additivity of measures, we can restrict ourselves to when the whole process P is the redex of the reduction. Let us see here the case when P is the redex of a (red-pino) (another is in Appendix B).

Let $P = \circledcirc_n(\rho).\sigma|\sigma_0\mathfrak{C}P'\mathfrak{D}$ and let us assume that σ_0 does not exhibit a $\circledcirc_n$ action. Then, the translation of P is $([\![P]\!])_x = E\vdash species_{\emptyset,x}(P)\oplus(0,\emptyset,\emptyset)$, where

$$\begin{aligned} species_{\emptyset,x}(P) &= \{s(\circledcirc_n(\rho).\sigma|\sigma_0)\mathfrak{C}_y^x\mathfrak{D}\mapsto 1\}\cup species_{\{y\},x}(P') \\ &= \{(s(\circledcirc_n(\rho).\sigma)\cup s(\sigma_0))\mathfrak{C}_y^x\mathfrak{D}\mapsto 1\}\cup species_{\{y\},x}(P') \\ &= \{(\{\circledcirc_n(\rho).\sigma\}\cup s(\sigma_0))\mathfrak{C}_y^x\mathfrak{D}\mapsto 1\}\cup species_{\{y\},x}(P') \end{aligned}$$

Let $I_1 = (\{\circledcirc_n(\rho).\sigma\}\cup s(\sigma_0))\mathfrak{C}_y^x\mathfrak{D}$ and $J_{P'} = species_{\{y\},x}(P')$; then $([\![P]\!])_x = I_1\mapsto 1\oplus J_{P'}\oplus(0,\emptyset,\emptyset) = J_{P'}\oplus (0,S',R')$ where

$$\begin{aligned} S' &= \{I_1\mapsto 1\} \\ R' &= init(L,(0,S',\emptyset)) = \{O_L\mapsto(t_1,a_1)\} \\ O_L &= (\{I_1\mapsto 1\},r_n,id,\{(s(\sigma)\cup s(\sigma_0))\mathfrak{C}_y^x\mathfrak{D}\mapsto 1, s(\rho)\mathfrak{C}_w^y\mathfrak{D}\mapsto 1\}) \\ L &= reactions(I_1\mapsto 1,\emptyset) = unary(I_1) = \{(\{I_1\mapsto 1\},r_n,id,\{(s(\sigma)\cup s(\sigma_0))\mathfrak{C}_y^x\mathfrak{D}\mapsto 1, s(\rho)\mathfrak{C}_w^y\mathfrak{D}\mapsto 1\})\} \end{aligned}$$

Now, the reaction O in $([\![P]\!])_x \xrightarrow{a,O} T$ is O_L (otherwise it would involve P', not the pino of the whole P). This means that $([\![P]\!])_x \xrightarrow{a,O} T$ is derived by means of an application of the (Reaction rule) as follows, where $S_1 = \{I_1\mapsto 1\}$ and $S_2 = \{(s(\sigma)\cup s(\sigma_0))\mathfrak{C}_y^x\mathfrak{D}\mapsto 1, s(\rho)\mathfrak{C}_w^y\mathfrak{D}\mapsto 1\}$:

$$\frac{((S_1,r_n,id,S_2),a_1,t_1)=next(0,S',R') \quad (E'\vdash(0,S'',R''))=cow(E\vdash(0,S',R'),S_1)}{E\vdash\{J_{P'}\}\oplus(0,S',R') \xrightarrow{a_1,(S_1,r_n,id,S_2)} norm(E\cup fn(S_2)\vdash(\{J_{P'}\}\oplus S_2\oplus((t_1,S'',R'')\ominus S_1)))}$$

where $S'' = S'$ and $R'' = R'$.

$$\{J_{P'}\} \oplus S_2 \oplus ((t_1, S'', R'') \ominus S_1) =$$
$$= J_{P'} \oplus \{(s(\sigma) \cup s(\sigma_0))(\!|_y^x|\!) \mapsto 1, s(\rho)(\!|_w^y|\!) \mapsto 1\} \oplus ((t_1, S', R') \ominus \{I_1 \mapsto 1\})$$
$$= J_{P'} \oplus \{(s(\sigma) \cup s(\sigma_0))(\!|_y^x|\!) \mapsto 1, s(\rho)(\!|_w^y|\!) \mapsto 1\} \oplus (t_1, \{I_1 \mapsto 0\}, \{O_L \mapsto (t_2, a_2)\})$$
$$= J_{P'} \oplus J' \oplus (t_1, \{I_1 \mapsto 0\}, \{O_L \mapsto (t_2, a_2)\})$$

Now let us define Q as $Q = \sigma|\sigma_0(\!|\rho(\!|\diamond|\!) \circ P'|\!)$, then $(\!|Q|\!)_x = species_{\emptyset, x}(Q) \oplus (0, \emptyset, \emptyset)$ where

$$species_{\emptyset, x}(Q) = species_{\emptyset, x}(\sigma|\sigma_0(\!|\rho(\!|\diamond|\!) \circ P'|\!))$$
$$= \{s(\sigma|\sigma_0)(\!|_y^x|\!) \mapsto 1\} \cup species_{\{x\}, y}(\rho(\!|\diamond|\!) \circ P')$$
$$= \{(s(\sigma) \cup s(\sigma_0))(\!|_y^x|\!) \mapsto 1\} \cup species_{\{x\}, y}(\rho(\!|\diamond|\!)) \cup species_{\{x\}, y}(P')$$
$$= \{(s(\sigma) \cup s(\sigma_0))(\!|_y^x|\!) \mapsto 1\} \cup \{s(\rho)(\!|_w^y|\!) \mapsto 1\} \cup \emptyset \cup J_{P'} = J' \cup J_{P'}$$

and hence $Q \equiv [\![J_{P'} \oplus J' \oplus (t_1, \{I_1 \mapsto 0\}, \{O_L \mapsto (t_2, a_2)\})]\!]_x$. It remains to prove that $r_n = \mu_{id}([Q])$. Let us notice that the derivation of $P \to \mu$ is actually as follows:

$$\text{(par)} \cfrac{\circledcirc_n(\rho).\sigma \to [\circledcirc_n]_\sigma^\rho \qquad \sigma_0 \to \mu''}{\circledcirc_n(\rho).\sigma|\sigma_0 \to [\circledcirc_n]_{\sigma \circledcirc_n(\rho).\sigma \circledcirc \sigma_0}^\rho \mu'' \qquad P' \to \mu'}$$
$$\text{(loc)} \cfrac{}{\circledcirc_n(\rho).\sigma|\sigma_0(\!|P'|\!) \to \mu}$$

where $\mu = \mu' @_{P'}^{\circledcirc_n(\rho).\sigma|\sigma_0}([\circledcirc_n]_{\sigma \circledcirc_n(\rho).\sigma \circledcirc \sigma_0}^\rho \mu'')$. Then:

$$\mu_{id}([\sigma|\sigma_0(\!|\rho(\!|\diamond|\!) \circ P'|\!)]) = (\mu' @_{P'}^{\circledcirc_n(\rho).\sigma|\sigma_0}([\circledcirc_n]_{\sigma \circledcirc_n(\rho).\sigma \circledcirc \sigma_0}^\rho \mu''))_{id}([\sigma|\sigma_0(\!|\rho(\!|\diamond|\!) \circ P'|\!)])$$
$$= \mu'_{id}([\sigma|\sigma_0(\!|\rho(\!|\diamond|\!) \circ P'|\!)]) + ([\circledcirc_n]_{\sigma \circledcirc_n(\rho).\sigma \circledcirc \sigma_0}^\rho \mu'')_{\circledcirc_n}([\sigma|\sigma_0] \times [\rho])$$
$$= ([\circledcirc_n]_{\sigma \circledcirc_n(\rho).\sigma \circledcirc \sigma_0}^\rho \mu'')_{\circledcirc_n}([\sigma|\sigma_0] \times [\rho])$$
$$= ([\circledcirc_n]_\sigma^\rho)_{\circledcirc_n}([\sigma|\sigma_0] \times [\rho]) + \mu''_{\circledcirc_n}([\sigma|\sigma_0] \times [\rho])$$
$$= r_n + \mu''_{\circledcirc_n}([\sigma|\sigma_0] \times [\rho]) = r_n$$

where the last equivalence holds because $\mu''_{\circledcirc_n}([\sigma|\sigma_0] \times [\rho]) = 0$ because we assumed that the reaction does not involve σ_0. $\qquad \square$

Proposition 5.5 (Progress). *For all processes P, Q, if $P \Longrightarrow Q$ then there exists a reaction O and a term T such that $(\!|P|\!)_x \xrightarrow{a, O} T$ and $Q \equiv [\![T]\!]_x$.*

Proof. By induction on the derivation of $P \Longrightarrow Q$. Let us see the case of (red-pin), the others being similar. Let $P = \circledcirc(\rho).\sigma|\sigma_0(\!|P'|\!)$ and $Q = \sigma|\sigma_0(\!|\rho(\!|\diamond|\!) \circ P'|\!)$. Then,

$$(\!|P|\!)_x = species_{\{y\}, x}(P') \oplus (0, \{(\{\circledcirc_n(\rho).\sigma\} \cup s(\sigma_0))(\!|_y^x|\!) \mapsto 1\}, \{O_L \mapsto (t_1, a_1)\})$$

where $O_L = (\{(\{\circledcirc_n(\rho).\sigma\} \cup s(\sigma_0))(\!|_y^x|\!) \mapsto 1\}, r_n, id, \{(s(\sigma) \cup s(\sigma_0))(\!|_y^x|\!) \mapsto 1, s(\rho)(\!|_w^y|\!) \mapsto 1\})$. Then, by the (Reaction rule) we can take $T = species_{\{x\}, y}(P') \oplus (t_1, \{(s(\sigma) \cup s(\sigma_0))(\!|_y^x|\!) \mapsto 1, s(\rho)(\!|_w^y|\!) \mapsto 1\}, \emptyset)$. It is easy to check that $Q \equiv [\![T]\!]_x$. $\qquad \square$

Proposition 5.6 (Completeness). *For all processes P, Q, if $P \to \mu$ and $\mu_{id}([Q]) > 0$ then for all node name x, there exists a reaction O and a term T such that $(\!|P|\!)_x \xrightarrow{a, O} T$, $Q \equiv [\![T]\!]_x$ and $a = \mu_{id}([Q])$.*

Proof. If $P \to \mu$ and $\mu_{id}([Q]) > 0$ then $P \Longrightarrow Q$ by Prop. 3.4. By Prop. 5.5, we have that for some a, O, T, $(\!|P|\!)_x \xrightarrow{a, O} T$ and $Q \equiv [\![T]\!]_x$. But then $a = \mu_{id}([Q])$ by soundness (Prop. 5.4). $\qquad \square$

5.3 Example

We conclude this section with an example. Let $P = 10000 \cdot \circlearrowright_n.\circlearrowright_m \mathbb{(}\circlearrowleft_k\mathbb{(D)D} \circ 100 \cdot ((\circlearrowleft_n^{\perp}(\circlearrowleft_m^{\perp}) \mid \circlearrowright^{\perp})\mathbb{(}\circlearrowleft_k\mathbb{(D)D})$.
Then, its reductions in the Brane Calculus are as follows:

$$10000 \cdot \circlearrowright_n.\circlearrowright_m \mathbb{(}\circlearrowleft_k\mathbb{(D)D} \circ 100 \cdot ((\circlearrowleft_n^{\perp}(\circlearrowleft_m^{\perp}) \mid \circlearrowright^{\perp})\mathbb{(}\circlearrowleft_k\mathbb{(D)D})$$

$$\Longrightarrow 9999 \cdot \circlearrowright_n.\circlearrowright_m \mathbb{(}\circlearrowleft_k\mathbb{(D)D} \circ 99 \cdot ((\circlearrowleft_n^{\perp}(\circlearrowleft_m^{\perp}) \mid \circlearrowright^{\perp})\mathbb{(}\circlearrowleft_k\mathbb{(D)D}) \circ \circlearrowright^{\perp}\mathbb{(}\circlearrowleft_m^{\perp}\mathbb{(}\circlearrowright_m \mathbb{(}\circlearrowleft_k\mathbb{(D)DD} \circ \circlearrowright_k\mathbb{(D)D}$$

$$\Longrightarrow 9999 \cdot \circlearrowright_n.\circlearrowright_m \mathbb{(}\circlearrowleft_k\mathbb{(D)D} \circ 99 \cdot ((\circlearrowleft_n^{\perp}(\circlearrowleft_m^{\perp}) \mid \circlearrowright^{\perp})\mathbb{(}\circlearrowleft_k^{\perp}()\mathbb{(D)D}) \circ \circlearrowright^{\perp}\mathbb{(}2 \cdot \circlearrowright_k\mathbb{(D)} \circ \diamond\mathbb{(}\diamond\mathbb{D)D}$$

The translation of P is $(\!|P|\!)_x = E \vdash species_{\emptyset,x}(P) \oplus (0,\emptyset,\emptyset)$, where

$$species_{\emptyset,x}(P) = species_{\{x\}}(10000 \cdot \circlearrowright_n.\circlearrowright_m \mathbb{(}\circlearrowleft_k\mathbb{(D)D} \circ 100 \cdot (\circlearrowleft_n^{\perp}(\circlearrowleft_m^{\perp}) \mid \circlearrowright^{\perp})\mathbb{(}\circlearrowleft_k\mathbb{(D)D})$$

$$= \{s(\circlearrowright_n.\circlearrowright_m)\mathbb{(}_y^x\mathbb{D} \mapsto 10000\} \cup species_{\{x,y\},y}(\circlearrowleft_k\mathbb{(D)}) \cup \{s(\circlearrowleft_n^{\perp}(\circlearrowleft_m^{\perp}) \mid \circlearrowright^{\perp})\mathbb{(}_z^x\mathbb{D} \mapsto 100\}$$

$$\cup species_{\{x,z\},z}(\circlearrowleft_k\mathbb{(D)})$$

$$= \{\{\circlearrowright_n.\circlearrowright_m\}\mathbb{(}_y^x\mathbb{D} \mapsto 10000, \{\circlearrowleft_n^{\perp}(\circlearrowleft_m^{\perp}), \circlearrowright^{\perp}\}\mathbb{(}_z^x\mathbb{D} \mapsto 100, \{\circlearrowright_k\}\mathbb{(}_w^y\mathbb{D} \mapsto 1,$$

$$\{\circlearrowright_k\}\mathbb{(}_v^z\mathbb{D} \mapsto 1\}$$

Let $I_1 = \{\circlearrowright_n.\circlearrowright_m\}\mathbb{(}_y^x\mathbb{D}$, $I_2 = \{\circlearrowleft_n^{\perp}(\circlearrowleft_m^{\perp}), \circlearrowright^{\perp}\}\mathbb{(}_z^x\mathbb{D}$, $I_3 = \{\circlearrowright_k\}\mathbb{(}_w^y\mathbb{D}$, $I_4 = \{\circlearrowright_k\}\mathbb{(}_v^z\mathbb{D}$, $r_n = 10s^{-1}$, $r_k = 5s^{-1}$ and $r_m = 5s^{-1}$; then

$$(\!|P|\!)_x = I_1 \mapsto 10000 \oplus I_2 \mapsto 100 \oplus I_3 \mapsto 1 \oplus I_4 \mapsto 1 \oplus (0,\emptyset,\emptyset)$$

$$= I_2 \mapsto 100 \oplus I_3 \mapsto 1 \oplus I_4 \mapsto 1 \oplus (0,S_1,R_1)$$

$$= I_3 \mapsto 1 \oplus I_4 \mapsto 1 \oplus (0,S_2,R_2)$$

$$= I_4 \mapsto 1 \oplus (0,S_3,R_3) = (0,S_4,R_4)$$

where

$$L_1 = reactions(I_1 \mapsto 10000, \emptyset) = \emptyset$$
$$S_1 = S\{I_1 \mapsto 10000\}$$
$$R_1 = R \cup init(L_1,(0,S_1,\emptyset)) = \emptyset$$

$$L_2 = reactions(I_2 \mapsto 100, S_1)$$
$$= (\{I_2 \mapsto 1, I_1 \mapsto 1\}, 10, id, \{\{\circlearrowright^{\perp}\}\mathbb{(}_z^x\mathbb{D} \mapsto 1, \{\circlearrowleft_m^{\perp}\}\}\mathbb{(}_w^z\mathbb{D} \mapsto 1, \{\circlearrowright_m\}\mathbb{(}_y^w\mathbb{D} \mapsto 1\})$$
$$S_2 = S_1\{I_2 \mapsto 100\} = \{I_1 \mapsto 10000, I_2 \mapsto 100\}$$
$$R_2 = R_1 \cup init(L_2,(0,S_2,R_1)) = \{O_{L_2} \mapsto (t_1,a_1)\}$$

$$L_3 = reactions(I_3 \mapsto 1, S_2) = \emptyset$$
$$S_3 = S_2\{I_3 \mapsto 1\} = \{I_1 \mapsto 10000, I_2 \mapsto 100, I_3 \mapsto 1\}$$
$$R_3 = R_2 \cup init(L_3,(0,S_3,R_2)) = R_2$$

$$L_4 = reactions(I_4 \mapsto 1, S_3) = \emptyset$$
$$S_4 = S_3\{I_4 \mapsto 1\} = \{I_1 \mapsto 10000, I_2 \mapsto 100, I_3 \mapsto 1, I_4 \mapsto 1\}$$
$$R_4 = R_3 \cup init(L_4,(0,S_4,R_3)) = R_3$$

with $O_{L_2} = (\{I_2 \mapsto 1, I_1 \mapsto 1\}, 10, id, \{\{\eth^\perp\}\mathfrak{C}_z^x\mathfrak{D} \mapsto 1, \{\eth_m^\perp\}\mathfrak{C}_w^z\mathfrak{D} \mapsto 1, \{\eth_m\}\mathfrak{C}_y^w\mathfrak{D} \mapsto 1\})$

$a_1 = propensity(O_{L_2}, S_2) = 10000000$

$t_1 = 0 + delay(10, 10000000)$.

Now, the reaction O in $(\!(P)\!)_x \xrightarrow{a,O} T$ is O_L. This means that $(\!(P)\!)_x \xrightarrow{a,O} T$ is derived by means of an application of the (Reaction rule) as follows, where $E = x, y, z, w$, $S_1 = \{I_2 \mapsto 1, I_1 \mapsto 1\}$, and $S_2 = \{\{\eth^\perp\}\mathfrak{C}_z^x\mathfrak{D} \mapsto 1, \{\eth_m^\perp\}\mathfrak{C}_w^z\mathfrak{D} \mapsto 1, \{\eth_m\}\mathfrak{C}_y^w\mathfrak{D} \mapsto 1\}$:

$$\frac{((S_1, 10, id, S_2), 10, t_1) = next(0, S_4, R_4) \quad (E' \vdash (0, S_5, R_5)) = cow(E \vdash (0, S_4, R_4), S_1)}{E \vdash (0, S_4, R_4) \xrightarrow{10,(S_1,10,id,S_2)} norm(E' \cup fn(S_2) \vdash (S_2 \oplus ((t_1, S_5, R_5) \ominus S_1)))}$$

where

$$S_5 = \{I_1 \mapsto 1, \{\eth_n.\eth_m\}\mathfrak{C}_{y'}^x\mathfrak{D} \mapsto 9999, I_2 \mapsto 1, \{\eth_n^\perp(\eth_m^\perp), \eth^\perp\}\mathfrak{C}_{z'}^x\mathfrak{D} \mapsto 99, I_3 \mapsto 1,$$

$$I_4 \mapsto 1, \{\eth_k\}\mathfrak{C}_{w'}^{y'}\mathfrak{D} \mapsto 1, \{\eth_k\}\mathfrak{C}_{v'}^{z'}\mathfrak{D} \mapsto 1\}$$

$$R_5 = \{O_{L_2} \mapsto (t_1, a_1), O_1 \mapsto (t_2, a_2), O_2 \mapsto (t_3, a_3), O_3 \mapsto (t_4, a_4)\}$$

with

$$O_1 = (\{\{\eth_n.\eth_m\}\mathfrak{C}_{y'}^x\mathfrak{D} \mapsto 1, I_2 \mapsto 1\}, 10, id, \{\{\eth^\perp\}\mathfrak{C}_z^x\mathfrak{D} \mapsto 1, \{\eth_m^\perp\}\mathfrak{C}_{w''}^z\mathfrak{D} \mapsto 1, \{\eth_m\}\mathfrak{C}_{y'}^{w''}\mathfrak{D} \mapsto 1\})$$

$$O_2 = (\{\eth_n^\perp(\eth_m^\perp), \eth^\perp\}\mathfrak{C}_{z'}^x\mathfrak{D} \mapsto 1, I_1 \mapsto 1\}, 10, id, \{\{\eth^\perp\}\mathfrak{C}_{z'}^x\mathfrak{D} \mapsto 1, \{\eth_m^\perp\}\mathfrak{C}_{w'''}^{z'}\mathfrak{D} \mapsto 1, \{\eth_m\}\mathfrak{C}_y^{w'''}\mathfrak{D} \mapsto 1\})$$

$$O_3 = (\{\{\eth_n.\eth_m\}\mathfrak{C}_{y'}^x\mathfrak{D} \mapsto 1, \{\eth_n^\perp(\eth_m^\perp), \eth^\perp\}\mathfrak{C}_{z'}^x\mathfrak{D} \mapsto 1\}, 10, id, \{\{\eth^\perp\}\mathfrak{C}_z^x\mathfrak{D} \mapsto 1, \{\eth_m^\perp\}\mathfrak{C}_{z''}^z\mathfrak{D} \mapsto 1, \{\eth_m\}\mathfrak{C}_y^{z''}\mathfrak{D}\})$$

$a_2 = propensity(O_1, S_5) = 99990$

$a_3 = propensity(O_2, S_5) = 990$

$a_4 = propensity(O_3, S_5) = 989901$

$t_2 = 0 + delay(10, 99990)$

$t_3 = 0 + delay(10, 990)$

$t_4 = 0 + delay(10, 989901)$

We can now compute the multiset of the new machine state:

$$S_2 \oplus ((t_1, S_5, R_5) \ominus S_1) = S_2 \oplus ((t_1, S_5, R_5) \ominus \{I_2 \mapsto 1, I_1 \mapsto 1\})$$

$$= S_2 \oplus ((t_1, S_6, R_6)) \ominus \{I_1 \mapsto 1\})$$

$$= \{\{\eth^\perp\}\mathfrak{C}_z^x\mathfrak{D} \mapsto 1, \{\eth_m^\perp\}\mathfrak{C}_w^z\mathfrak{D} \mapsto 1, \{\eth_m\}\mathfrak{C}_y^w\mathfrak{D} \mapsto 1\} \oplus (t_1, S_7, R_7)$$

$$= (t_1, S_8, R_8)$$

with

$a_2' = propensity(O_1, S_6) = 0$ $S_8 = S_6 \cup \{\{\eth^\perp\}\mathfrak{C}_z^x\mathfrak{D} \mapsto 1, \{\eth_m^\perp\}\mathfrak{C}_w^z\mathfrak{D} \mapsto 1, \{\eth_m\}\mathfrak{C}_y^w\mathfrak{D} \mapsto 1\}$

$t_2' = t_1 + (a_2/a_2')(t_2 - t_1)$ $R_6 = \{O_1 \mapsto (t_2', a_2')\}$

$a_3' = propensity(O_2, S_7) = 0$ $R_7 = R_6 \cup \{O_2 \mapsto (t_3', a_3')\}$

$t_3' = t_1 + (a_3/a_5)(t_3 - t_1)$ $R_8 = R_7 \cup \{\{\{\eth_m^\perp\}\mathfrak{C}_w^z\mathfrak{D} \mapsto 1, \{\eth_m\}\mathfrak{C}_y^w\mathfrak{D} \mapsto 1\}, 5, f, \{\emptyset\mathfrak{C}_w^z\mathfrak{D}\} \mapsto 1\}$

$S_6 = S_5 \setminus \{I_2 \mapsto 1\}$

$S_7 = S_6 \setminus \{I_1 \mapsto 1\}$ $f = \lambda T.T[y := z]$

6 Conclusions

In this paper, we have presented an abstract machine for the Stochastic Brane Calculus. Instead of defining an *ad hoc* machine, we have adopted the generic abstract machine for stochastic calculi (GSAM) recently introduced by Lakin, Paulevé and Phillips. According to the encoding technique we have adopted, membranes are flattened into a set of species, where the hierarchical structure is represented by means of *names*. In order to keep track of these names, and for dealing efficiently with multiple copies of the same species, we have introduced a new generic abstract machine, called COWGSAM, which extends the GSAM with a *name environment* and a *copy-on-write* optimization strategy. We have proved that the implementation of the Stochastic Brane Calculus in COWGSAM is adequate with respect to the stochastic structural operational semantics of the calculus given in [2].

We think that COWGSAM can be used for implementing other stochastic calculi dealing with nested structures, also beyond the models for systems biology. In particular, it is interesting to apply this approach to Stochastic Bigraphs [9], a general meta-model well-suited for representing a range of stochastic systems with compartments; in this way we would obtain a *General Stochastic Bigraphical Machine*, which could be instantiated to any given stochastic bigraphic reactive system. However, such a machine would not scale well, as in general the COW strategy may be not very useful; thus, we can restrict our attentions to smaller subsets of BRSs, specifically designed to some application domain. For biological applications, the bigraphic reactive systems considered in [1, 7] might be a more reasonable target.

Another interesting question is about the expressive power of GSAM and COWGSAM. We think that GSAM correspond to stochastic (multiset) Petri nets, but COWGSAM could go further thanks to the possibility of creating unlimited new names during execution. Further work include comparison with other stochastic simulation tools dealing with compartments, like BioPEPA [6].

References

[1] Giorgio Bacci, Davide Grohmann & Marino Miculan (2009): *Bigraphical models for protein and membrane interactions*. In: Gabriel Ciobanu, editor: *Proc. MeCBIC'09, EPTCS* 11, pp. 3–18. doi:10.4204/EPTCS.11.1.

[2] Giorgio Bacci & Marino Miculan (2012): *Measurable Stochastics for Brane Calculus*. *Theoretical Computer Science* 431, pp. 117–136. doi:10.1016/j.tcs.2011.12.055.

[3] Roberto Barbuti, Andrea Maggiolo-Schettini, Paolo Milazzo & Angelo Troina (2007): *The Calculus of Looping Sequences for Modeling Biological Membranes*. In: George Eleftherakis, Petros Kefalas, Gheorghe Paun, Grzegorz Rozenberg & Arto Salomaa, editors: *Workshop on Membrane Computing*, *Lecture Notes in Computer Science* 4860, Springer, pp. 54–76. doi:10.1007/978-3-540-77312-2_4.

[4] Luca Cardelli (2004): *Brane Calculi*. In: Vincent Danos & Vincent Schächter, editors: *Proc. CMSB*, *Lecture Notes in Computer Science* 3082, Springer, pp. 257–278. doi:10.1007/978-3-540-25974-9_24.

[5] Luca Cardelli & Radu Mardare (2010): *The Measurable Space of Stochastic Processes*. In: *Proc. QEST*, IEEE Computer Society, pp. 171–180. doi:10.1109/QEST.2010.30.

[6] Federica Ciocchetta & Maria Luisa Guerriero (2009): *Modelling Biological Compartments in Bio-PEPA*. *Electronic Notes in Theoretical Computer Science* 227, pp. 77–95. doi:10.1016/j.entcs.2008.12.105.

[7] Troels Christoffer Damgaard, Espen Højsgaard & Jean Krivine (2012): *Formal Cellular Machinery*. *Electronic Notes in Theoretical Computer Science* 284, pp. 55–74. doi:10.1016/j.entcs.2012.05.015.

[8] Holger Hermanns (2002): *Interactive Markov Chains: The Quest for Quantified Quality*, *Lecture Notes in Computer Science* 2428. Springer. doi:10.1007/3-540-45804-2.

[9] Jean Krivine, Robin Milner & Angelo Troina (2008): *Stochastic Bigraphs*. In: *Proc. 24th MFPS*, *ENTCS* 218, pp. 73–96. doi:10.1016/j.entcs.2008.10.006.

[10] Matthew R. Lakin, Loïc Paulevé & Andrew Phillips (2012): *Stochastic Simulation of Multiple Process Calculi for Biology*. Theoretical Computer Science 431, pp. 181–206. doi:10.1016/j.tcs.2011.12.057.

[11] Cosimo Laneve & Fabien Tarissan (2008): *A simple calculus for proteins and cells*. Theoretical Computer Science 404(1-2), pp. 127–141. doi:10.1016/j.tcs.2008.04.011.

[12] Prakash Panangaden (2009): *Labelled Markov Processes*. Imperial College Press, London, U.K.

[13] Loïc Paulevé, Simon Youssef, Matthew R. Lakin & Andrew Phillips (2010): *A generic abstract machine for stochastic process calculi*. In: Paola Quaglia, editor: *Proc. CMSB*, ACM, pp. 43–54. doi:10.1145/1839764.1839771.

[14] Aviv Regev, Ekaterina M. Panina, William Silverman, Luca Cardelli & Ehud Y. Shapiro (2004): *BioAmbients: an abstraction for biological compartments*. Theoretical Computer Science 325(1), pp. 141–167. doi:10.1016/j.tcs.2004.03.061.

A   Some measure theory

Given a set M, a family Σ of subsets of M is called a σ-*algebra* if it contains M and is closed under complements and (infinite) countable unions:

1. $M \in \Sigma$;
2. $A \in \Sigma$ implies $A^c \in \Sigma$, where $A^c = M \setminus A$;
3. $\{A_i\}_{i \in \mathbb{N}} \subset \Sigma$ implies $\bigcup_{i \in \mathbb{N}} A_i \in \Sigma$.

Since $M \in \Sigma$ and $M^c = \emptyset$, $\emptyset \in \Sigma$, hence Σ is nonempty by definition. A σ-algebra is closed under countable set-theoretic operations: is closed under finite unions ($A, B \in \Sigma$ implies $A \cup B = A \cup B \cup \emptyset \cup \emptyset \cup \cdots \in \Sigma$), countable intersections (by DeMorgan's law $A \cap B = (A^c \cup B^c)^c$ in its finite and inifite version), and countable subtractions ($A, B \in \Sigma$ implies $A \setminus B = A \cap B^c \in \Sigma$).

Definition A.1 (Measurable Space). *Given a set M and a σ-algebra on M, the tuple (M, Σ) is called a* measurable space*, the elements of Σ* measurable sets*, and M the* support-set.

A set $\Omega \subseteq 2^M$ is a *generator for the σ-algebra* Σ on M if Σ is the closure of Ω under complement and countable union; we write $\sigma(\Omega) = \Sigma$ and say that Σ is generated by Ω. Note that the σ-algebra generated by a Ω is also the smallest σ-algebra containing Ω, that is, the intersection of all σ-algebras that contain Ω. In particular it holds that a completely arbitrary intersection of σ-algebras is a σ-algebra. A σ-algebra generated by Ω, denoted by $\sigma(\Omega)$, is minimal in the sense that if $\Omega \subset \Sigma$ and Σ is a σ-algebra, then $\sigma(\Omega) \subset \Sigma$. If Ω is a σ-algebra then obviously $\sigma(\Omega) = \Omega$; if Ω is empty or $\Omega = \{\emptyset\}$, or $\Omega = \{M\}$, then $\sigma(\Omega) = \{\emptyset, M\}$; if $\Omega \subset \Sigma$ and Σ is a σ-algebra, then $\sigma(\Omega) \subset \Sigma$. A generator Ω for Σ is a *base for Σ* if it has disjoin elements. Note that if Ω is a base for Σ, all measurable sets in Σ can be decomposed into countable unions of elements in Ω.

A *measure* on a measurable space (M, Σ) is a function $\mu \colon \Sigma \to \mathbb{R}_\infty^+$, where $\mathbb{R}_\infty^+$ denotes the extended positive real line, such that

1. $\mu(\emptyset) = 0$;
2. for any disjoint sequence $\{N_i\}_{i \in I} \subseteq \Sigma$ with $I \subseteq \mathbb{N}$, it holds

$$\mu(\textstyle\bigcup_{i \in I} N_i) = \sum_{i \in I} \mu(N_i).$$

The triple (M, Σ, μ) is called a *measure space*. A measure space (M, Σ, μ) is called *finite* if $\mu(M)$ is a finite real number; it is called σ-*finite* if M can be decomposed into a countable union of measurable sets of finite measure, that is, $M = \bigcup_{i \in I} N_i$, for some $I \subseteq \mathbb{N}$ and $\mu(N_i) \in \mathbb{R}^+$ for each $i \in I$. A set in a measure

space has σ-*finite measure* if it is a countable union of sets with finite measure. Specifying a measure includes specifying its domain. If μ is a measure on a measurable space (M, Σ) and Σ' is a σ-algebra contained in Σ, then the restriction μ' of μ to Σ' is also a measure, and in particular a measure on (M', Σ'), for some $M' \subseteq M$ such that Σ' is a σ-algebra on M'.

Given two measurable spaces and measures on them, one can obtain the product measurable space and the product measure on that space. Let (M_1, Σ_1) and (M_2, Σ_2) be measurable spaces, and μ_1 and μ_2 be measures on these spaces. Denote by $\Sigma_1 \otimes \Sigma_2$ the σ-algebra on the cartesian product $M_1 \times M_2$ generated by subsets of the form $B_1 \times B_2$, said rectangles, where $B_1 \in \Sigma_1$ and $B_2 \in \Sigma_2$. The *product measure* $\mu_1 \otimes \mu_2$ is defined to be the unique measure on the measurable space $(M_1 \times M_2, \Sigma_1 \otimes \Sigma_2)$ such that, for all $B_1 \in \Sigma_1$ and $B_2 \in \Sigma_2$

$$(\mu_1 \otimes \mu_2)(B_1 \times B_2) = \mu_1(B_1) \cdot \mu_2(B_2)$$

The existence of this measure is guaranteed by the Hahn-Kolmogorov theorem. The uniqueness of the product measure is guaranteed only in the case that both (M_1, Σ_1, μ_1) and (M_2, Σ_2, μ_2) are σ-finite.

Let $\Delta(M, \Sigma)$ be the family of measures on (M, Σ). It can be organized as a measurable space by considering the σ-algebra generated by the sets $\{\mu \in \Delta(M, \Sigma) : \mu(S) \geq r\}$, for arbitrary $S \in \Sigma$ and $r > 0$.

Given two measurable spaces (M, Σ) and (N, Θ) a mapping $f \colon M \to N$ is *measurable* if for any $T \in \Theta$, $f^{-1}(T) \in \Sigma$. Measurable functions are closed under composition: given $f \colon M \to N$ and $g \colon N \to O$ measurable functions then $g \circ f \colon M \to O$ is also measurable.

B Proof of Prop. 5.4

Let $P = \eth_n^\perp.\tau | \tau_0 \langle \eth_n.\sigma | \sigma_0 \langle P' \rangle \circ P'' \rangle$; then, $\langle\!| P |\!\rangle_x = E \vdash species_{\emptyset,x}(P) \oplus (0, \emptyset, \emptyset)$, where

$\begin{aligned}
species_{\emptyset,x}(P) &= \{s(\eth_n^\perp.\tau | \tau_0) \langle_y^x \rangle\} \cup species_{\{x\},y}(\eth_n.\sigma | \sigma_0 \langle P' \rangle \circ P'') \\
&= \{(\{\eth_n^\perp.\tau\} \cup s(\tau_0)) \langle_y^x \rangle \mapsto 1\} \cup species_{\{x\},y}(\eth_n.\sigma | \sigma_0 \langle P' \rangle) \cup species_{\{x\},y}(P'') \\
&= \{(\{\eth_n^\perp.\tau\} \cup s(\tau_0)) \langle_y^x \rangle \mapsto 1\} \cup \{s(\eth_n.\sigma | \sigma_0) \langle_w^y \rangle \mapsto 1\} \cup species_{\{x,y\},w}(P') \cup species_{\{x\},y}(P'') \\
&= \{(\{\eth_n^\perp.\tau\} \cup s(\tau_0)) \langle_y^x \rangle \mapsto 1\} \cup \{(s(\eth_n.\sigma) \cup s(\sigma_0)) \langle_w^y \rangle \mapsto 1\} \cup species_{\{x,y\},w}(P') \cup species_{\{x\},y}(P'') \\
&= \{(\{\eth_n^\perp.\tau\} \cup s(\tau_0)) \langle_y^x \rangle \mapsto 1\} \cup \{(\{\eth_n.\sigma\} \cup s(\sigma_0)) \langle_w^y \rangle \mapsto 1\} \cup species_{\{x,y\},w}(P') \cup species_{\{x\},y}(P'')
\end{aligned}$

Let $I_1 = (\{\eth_n^\perp.\tau\} \cup s(\tau_0)) \langle_y^x \rangle$, $I_2 = (\{\eth_n.\sigma\} \cup s(\sigma_0)) \langle_w^y \rangle$, $J_{P'} = species_{\{x,y\},w}(P')$, $J_{P''} = species_{\{x\},y}(P'')$; then $\langle\!| P |\!\rangle_x = I_1 \mapsto 1 \oplus I_2 \mapsto 1 \oplus J_{P'} \oplus J_{P''} \oplus (0, \emptyset, \emptyset) = I_2 \mapsto 1 \oplus J_{P'} \oplus J_{P''} \oplus (0, S', R') = J_{P'} \oplus J_{P''} \oplus (0, S'', R'')$ where

$$L' = reactions(I_1 \mapsto 1, \emptyset) = \emptyset$$
$$S' = S\{I_1 \mapsto 1\}$$
$$R' = init(L', (0, S', \emptyset)) = \emptyset$$

$$L'' = reactions(I_2 \mapsto 1, S')$$
$$\quad = (\{I_2 \mapsto 1, I_1 \mapsto 1\}, r_n, f, \{(s(\tau) \cup s(\tau_0) \cup s(\sigma) \cup s(\sigma_0)) \langle_y^x \rangle \mapsto 1\})$$
$$S'' = S'\{I_2 \mapsto 1\} = \{I_1 \mapsto 1, I_2 \mapsto 1\}$$
$$R'' = init(L'', (0, S'', R')) = \{O_L \mapsto (t_1, a_1)\}$$

with $O_L = (\{I_1 \mapsto 1, I_2 \mapsto 1\}, r_n, f, \{(s(\tau) \cup s(\tau_0) \cup s(\sigma) \cup s(\sigma_0)) \langle_y^x \rangle \mapsto 1\})$. Now, the reaction O in $\langle\!| P |\!\rangle_x \xrightarrow{F,O} T$ is O_L. This means that $\langle\!| P |\!\rangle_x \xrightarrow{F,O} T$ is derived by means of an application of the (Reaction rule) as follows, where $S_1 = \{I_1 \mapsto 1, I_2 \mapsto 1\}$, $S_2 = \{(s(\tau) \cup s(\tau_0) \cup s(\sigma) \cup s(\sigma_0)) \langle_y^x \rangle \mapsto 1\}$ and $f = T[w := x]$:

$$\frac{((S_1,r_n,f,S_2),a_1,t_1)=next(0,S'',R'')\quad(E'\vdash(0,S''',R'''))=cow(E\vdash(0,S'',R''),S_1)}{E\vdash(0,S'',R'')\xrightarrow{a_1,(S_1,r_n,f,S_2)}norm(E'\cup fn(S_2)\vdash(f(S_2\oplus((t_1,S''',R''')\ominus S_1))))}$$

where $S'''=S''$ and $R'''=R''$.

$$
\begin{aligned}
(f(S_2\oplus&((t_1,S''',R''')\ominus S_1))))=\\
&=f((s(\tau)\cup s(\tau_0)\cup s(\sigma)\cup s(\sigma_0)\mathfrak{q}_y^x\mathfrak{D})\oplus((t_1,S''',R''')\ominus\{I_1\mapsto1,I_2\mapsto1\})\\
&=f((s(\tau)\cup s(\tau_0)\cup s(\sigma)\cup s(\sigma_0)\mathfrak{q}_y^x\mathfrak{D})\oplus((t_1,\{I_1\mapsto0,I_2\mapsto1\},\{O_L\mapsto(t_2,a_2)\})\ominus\{I_2\mapsto1\})\\
&=f((s(\tau)\cup s(\tau_0)\cup s(\sigma)\cup s(\sigma_0)\mathfrak{q}_y^x\mathfrak{D})\oplus(t_1,\{I_1\mapsto0,I_2\mapsto0\},\{O_L\mapsto(t_3,a_3)\})\\
&=(s(\tau)\cup s(\tau_0)\cup s(\sigma)\cup s(\sigma_0)\mathfrak{q}_y^x\mathfrak{D})\oplus(t_1,\{I_1\mapsto0,I_2\mapsto0\},\{O_L\mapsto(t_3,a_3)\})\\
&=J'\oplus(t_1,\{I_1\mapsto0,I_2\mapsto0\},\{O_L\mapsto(t_3,a_3)\})
\end{aligned}
$$

Now let us define Q as $Q=\sigma|\sigma_0|\tau|\tau_0\mathfrak{q}P''\mathfrak{D}\circ P'$, then $\langle\!\langle Q\rangle\!\rangle_x=species_{\emptyset,x}(Q)\oplus(0,\emptyset,\emptyset)$ where

$$
\begin{aligned}
species_{\emptyset,x}(Q)&=species_{\emptyset,x}(\sigma|\sigma_0|\tau|\tau_0\mathfrak{q}P''\mathfrak{D}\circ P')\\
&=species_{\emptyset,x}(\sigma|\sigma_0|\tau|\tau_0\mathfrak{q}P''\mathfrak{D})\cup species_{\emptyset,x}(P')\\
&=\{s(\sigma|\sigma_0|\tau|\tau_0)\mathfrak{q}_y^x\mathfrak{D}\mapsto1\}\cup species_{\{x\},y}(P'')\cup species_{\emptyset,x}(P')\\
&=\{s(\tau)\cup s(\tau_0)\cup s(\sigma)\cup s(\sigma_0)\mathfrak{q}_y^x\mathfrak{D}\mapsto1\}\cup J_{P''}\cup J_{P'}\\
&=J'\cup J_{P''}\cup J_{P'}
\end{aligned}
$$

And hence $Q\equiv[\![J_{P'}\oplus J_{P''}\oplus J'\oplus(t_1,\{I_1\mapsto0,I_2\mapsto0\},\{O_L\mapsto(t_3,a_3)\})]\!]_x$. It remains to prove that $r_n=\mu_{id}([Q])$. Let us notice that the derivation of $P\to\mu$ is actually as follows:

$$
\cfrac{\begin{array}{c}\text{(par)}\cfrac{\eth_n^\perp.\tau\to[\eth_n^\perp]_\tau\quad\tau_0\to\mu}{\eth_n^\perp.\tau|\tau_0\to[\eth_n^\perp]_\tau\eth_n^\perp.\tau\textcircled{\tiny D}\tau_0\mu}\\\text{(loc)}\end{array}\quad\text{(comp)}\cfrac{\cfrac{\text{(par)}\cfrac{\eth_n.\sigma\to[\eth_n]_\sigma\quad\sigma_0\to\mu''}{\eth_n.\sigma|\sigma_0\to[\eth_n]_\sigma\eth_n.\sigma\textcircled{\tiny D}\sigma_0\mu''}\quad P'\to\mu'}{\eth.\sigma|\sigma_0\mathfrak{q}P'\mathfrak{D}\to\mu'@_{P'}^{\eth_n.\sigma|\sigma_0}([\eth_n]_\sigma\eth_n.\sigma\textcircled{\tiny D}\sigma_0\mu'')}\text{(loc)}\quad P''\to\mu'''}{\eth_n.\sigma|\sigma_0\mathfrak{q}P'\mathfrak{D}\circ P''\to(\mu'@_{P'}^{\eth_n.\sigma|\sigma_0}([\eth_n]_\sigma\eth_n.\sigma\textcircled{\tiny D}\sigma_0\mu''))_{\eth_n.\sigma|\sigma_0\mathfrak{q}P'\mathfrak{D}}\otimes_{P''}\mu'''}}{\eth_n^\perp.\tau|\tau_0\mathfrak{q}\eth_n.\sigma|\sigma_0\mathfrak{q}P'\mathfrak{D}\circ P''\mathfrak{D}\to\nu}
$$

where $\nu=((\mu'@_{P'}^{\eth_n.\sigma|\sigma_0}([\eth_n]_\sigma\eth_n.\sigma\textcircled{\tiny D}\sigma_0\mu''))_{\eth_n.\sigma|\sigma_0\mathfrak{q}P'\mathfrak{D}}\otimes_S\mu''')@_{\eth_n.\sigma|\sigma_0\mathfrak{q}P'\mathfrak{D}\circ P''}^{\eth_n^\perp.\tau|\tau_0}([\eth_n^\perp]_\tau\eth_n^\perp.\tau\textcircled{\tiny D}\tau_0\mu)$,

$\mu_1=(\mu'@_{P'}^{\eth_n.\sigma|\sigma_0}([\eth_n]_\sigma\eth_n.\sigma\textcircled{\tiny D}\sigma_0\mu''))_{\eth_n.\sigma|\sigma_0\mathfrak{q}P'\mathfrak{D}}\otimes_{P''}\mu'''$ and $\mu_2=[\eth_n^\perp]_\tau\eth_n^\perp.\tau\textcircled{\tiny D}\tau_0\mu$. Then:

$$
\begin{aligned}
\nu_{id}([\sigma|\sigma_0|\tau|\tau_0\mathfrak{q}P''\mathfrak{D}\circ P'])&=(\mu_1@_{\eth_n.\sigma|\sigma_0\mathfrak{q}P'\mathfrak{D}\circ P''}^{\eth_n^\perp.\tau|\tau_0}\mu_2)_{id}([\sigma|\sigma_0|\tau|\tau_0\mathfrak{q}P''\mathfrak{D}\circ P'])\\
&=\mu_{1id}([\sigma|\sigma_0|\tau|\tau_0\mathfrak{q}P''\mathfrak{D}\circ P'])+\mu_{1ex_n}([\sigma|\sigma_0]\times[P']\times[P''])\cdot\mu_{2\eth_n^\perp}([\tau|\tau_0])/r_n\\
&=\mu_{1ex_n}([\sigma|\sigma_0]\times[P']\times[P''])\cdot\mu_{2\eth_n^\perp}([\tau|\tau_0])/r_n\\
&=((\mu'@_{P'}^{\eth_n.\sigma|\sigma_0}([\eth_n]_\sigma\eth_n.\sigma\textcircled{\tiny D}\sigma_0\mu''))_{ex_n}([\sigma|\sigma_0]\times[P']\times[P''])+\mu'''_{ex_n}([\sigma|\sigma_0]\times[P']\times[P'']))\cdot\mu_{2\eth_n^\perp}([\tau|\tau_0])/r_n\\
&=(([\eth_n]_\sigma\eth_n.\sigma\textcircled{\tiny D}\sigma_0\mu'')_{\eth_n}([\sigma|\sigma_0])+\mu'''_{ex_n}([\sigma|\sigma_0]\times[P']\times[P'']))\cdot([\eth_n^\perp]_\tau\eth_n^\perp.\tau\textcircled{\tiny D}\tau_0\mu)_{\eth_n^\perp}([\tau|\tau_0])/r_n\\
&=(([\eth_n]_\sigma)_{\eth_n}([\sigma|\sigma_0])+\mu''_{\eth_n}([\sigma|\sigma_0])+\mu'''_{ex_n}([\sigma|\sigma_0]\times[P']\times[P'']))\cdot((([\eth_n^\perp]_\tau)_{\eth_n^\perp}([\tau|\tau_0])+\mu_{\eth_n^\perp}([\tau|\tau_0]))/r_n\\
&=(r_n+\mu''_{\eth_n}([\sigma|\sigma_0])+\mu'''_{ex_n}([\sigma|\sigma_0]\times[P']\times[P'']))\cdot(r_n+\mu_{\eth_n^\perp}([\tau|\tau_0]))/r_n=r_n
\end{aligned}
$$

where the last equivalence holds because $\mu''_{\eth_n}([\sigma|\sigma_0])=\mu'''_{ex_n}([\sigma|\sigma_0]\times[P']\times[P''])=\mu_{\eth_n^\perp}([\tau|\tau_0])=0$ because we assumed that the reaction does not involve either σ_0 nor τ_0.