

A Global Occurrence Counting Analysis for Brane Calculi

C. Bodei¹, L. Brodo², R. Gori¹, D. Hermith³, and F. Levi¹

¹ Dipartimento di Informatica, Università di Pisa, Italy

² Dip. di Scienze Politiche, Scienze della Com. e Ing. dell'Informazione, Università di Sassari, Italy

³ Dip. di Ingegneria dell'informazione e Scienze Matematiche, Università di Siena, Italy

Abstract. We propose a polynomial static analysis for Brane Calculi [7], based on *Abstract Interpretation* [9] techniques. The analysis provides a description of the possible hierarchical structure of membranes and of the processes possibly associated to each membrane, together with global occurrence counting information. Our analysis can be applied in the biological setting to investigate systems in which the information on the number of membranes occurring in the system plays a crucial role.

1 Introduction

One of the challenges of systems biology is to understand the complex behaviour of biological systems and networks that interact in diverse ways. There is a large amount of data concerning single components and functions. The main difficulty is establishing their relationships and inferring the overall emerging behaviour of a system. Consider, for instance, the crucial problem of identifying biological pathways and reconstructing their inter-connections.

Computational frameworks and *in silico* investigation have been recently exploited to support *in vitro* or *in vivo* experiments. Often, these frameworks come with the drawback of high computational cost, due to the expensive inspection of the models that capture dynamic behaviour. Static analysis provides techniques able to reduce the computational cost, at the price of losing precision. In particular, they provide safe over-approximations of the dynamic behaviour: all the events that the analysis predicts *may* happen, while all the non predicted events will *never* happen.

In this paper we propose a *static analysis* for Brane Calculi [7], based on *Abstract Interpretation* (AI) [9] techniques. Brane calculi have been introduced to more closely model the behaviour of membrane-enclosed compartments. Therefore, these calculi are useful for modelling and reasoning about a large class of biological systems. Our analysis is based on the idea of enriching the standard information on control flow analysis (as the one in [3]) with global occurrence counting information. More specifically, we compute an abstract state describing the possible structure of all the derivatives of the analysed system. The abstract state provides information on the possible membrane hierarchy and on the processes that may be associated to each membrane, together with their multiplicity. The global occurrence counting information refers to the number of occurrences of membranes and processes, independently of their location. The analysis can be efficiently computed in polynomial time, thanks to the properties of the abstract semantics.

Our analysis can be applied to investigate systems in which the information on the number of membranes in the system plays a crucial role in biological terms. One could, for instance, be

interested in distinguishing between the case of one healthy cell that can let many molecules pass through the cell membrane, and the case in which an ill cell can let at most one molecule to pass. We illustrate our analysis by considering examples of communication via *mobile vesicles*. A mobile vesicle containing the substance to be transmitted springs from a membrane *Source* and eventually fuses with a membrane *Target* releasing its content X inside of it. We consider two different variants of this kind of communication by expanding the encoding proposed in [28], where there are two target membranes $Target_1$ and $Target_2$. In the first version, the membrane *Source* can communicate X either to $Target_1$ or $Target_2$, while in the second one, one instance of X is communicated to $Target_1$ and the other one to $Target_2$. Then, we show that our analysis faithfully predicts that X cannot be simultaneously present in both membranes in the first case, while it may occur in both in the second case, thanks to the occurrence counting information.

Related work Static analysis techniques have been applied to many biologically-oriented calculi (see, e.g. the survey in [16]). In particular, Control Flow Analysis (CFA) has been applied to Beta-binders [26] in [2], to Brane Calculi [3], and to BioAmbients [27] in [20,22,24,25]; while Abstract Interpretation to BioAmbients [14,13,15], and to Brane Calculi [4,5]. Many of these works are inspired by the application of static analysis techniques [1,19,17] to Mobile Ambients (MA) [8], from which, many bio-inspired calculi derive.

Part of the above mentioned static analyses [20,2,3,22] abstract, with different precision, the behaviour of the investigated system by providing an approximate description of structure of all derivatives. As a consequence, these approaches can be applied to establish invariant properties showing that certain events will not happen in any derivatives of the analysed system. These techniques have polynomial complexity but they provide a less precise description of the possible topological structure of derivatives given that they do not maintain any information on occurrence counting. The richer contextual CFA in [24], and the causality-oriented AI-based analysis in [4,5] improve the prediction accuracy, but, still, they are not able to observe the multiplicity.

Instead, in [14], the authors present a counting analysis for BioAmbients that is able to express that an ambient can reside in alternative locations. This analysis has exponential complexity and provides accurate information about the number of occurrences of ambients, by counting the local number inside any ambient rather than their global number.

There are several static analysis frameworks that include occurrence counting information applied to MA and to π -calculus [18]. In [19,21] analyses for MA are introduced that are rather expensive from a computational point of view. The authors propose in [19] an exponential analysis for counting the global number of occurrences of ambients. The approach based on CFA substantially differs from our analysis, which is computed by calculating an abstract semantics. At the expense of a higher complexity, the shape analysis in [21] uses context-dependent counts for inferring a more accurate description of the internal structure of an ambient, by taking care of the local multiplicity of ambients.

In [23,10,11] the author proposes a framework based on AI, applied to the π -calculus that, differently from the previous proposals, is non-uniform, i.e. the analysis can distinguish among recursive instances of agents. In this approach, the occurrence number of instances of agents is approximated by using a relational abstraction. The analysis is quite precise and efficient: its complexity is polynomial. For instance, the analysis described in [10] has a worst time cost of n^4 , where n is the number of processes in the initial configuration. This approach is adequate to capture mutual exclusion and other security properties of complex mobile

systems, formalised in π -calculus. In [1] the author proposes an adaptation of these techniques to MA, with a focus on security properties such as non-interference or confinement. In [12], the author proposes polynomial and precise analyses for MA and BioAmbients, based on both global and local counting. Specifically, the global analysis is the same as the one proposed in [10] for the π -calculus. This approach can handle mass preservation like invariants, which are ubiquitous in biological systems. In particular, it is able to preserve precision when dealing with continuations of replicated prefixes.

The static analyses proposed in [13,15,25] rely on a different approach since that they compute an abstract transition system to approximate the system behaviour, by still exploiting occurrence counting information. These techniques obviously provide useful information to verify temporal properties but at the price of a high complexity.

Structure of the Paper. In Section 2, we recall the semantics of Brane calculi. In Section 3, we introduce our running examples, based on hypothetical scenarios of communication via mobile vesicles in the style of [28]. In Section 4, we present the analysis and in Section 5 we apply it to our examples. Conclusions are drawn in Section 6.

For lack of space, we present the formal definitions only for the fragment of the calculus that includes the Phago/Exo/Pino (PEP) and Bud interactions. This fragment is sufficiently expressive to formalise our running examples. It is worth pointing out that our analysis can be easily extended to the full calculus.

2 An Overview on Brane Calculus

The Brane Calculi [7] are a family of calculi based on a set of primitives inspired by biological membrane interactions. As already mentioned, we focus here on the fragment of the calculus that includes the Phago/Exo/Pino (PEP) and Bud interactions.

The Phago/Exo/Pino(PEP) actions represent the biological processes of endocytosis and exocytosis. The first indicates the process of incorporating external material into a cell, by engulfing it with the cell membrane, while the second one indicates the reverse process. Endocytosis is rendered by two more basic operations: phagocytosis (*phago*), which consists in engulfing just one external membrane, and pinocytosis (*pino*), which consists in engulfing zero external membranes. Exocytosis is instead denoted by (*exo*). We further use the action (*bud*) to model the membrane splitting process and consists in the splitting off exactly one internal membrane. It is worth mentioning that the bud action can be encoded with a sequence of PEP actions [7]. However, from the analysis point of view it is more convenient to use the bud action as primitive.

We introduce the syntax and the semantics for the calculus, considering a *labelled version* of the calculus. As usual in static analysis, labels are exploited to support the analysis (presented in Sect. 4) and *do not affect* the dynamic semantics of the calculus.

A membrane system consists of nested membranes, where each membrane has associated a membrane process. The syntax of the labelled calculus is described in Tab. 1, where n is taken from a countable set \mathcal{N} of names, and where we write $P \in \mathbf{Sys}$ for *systems*, $\sigma \in \mathbf{Proc}$ for *membrane processes*, and $a \in \mathbf{Act}$ for *actions*. Each membrane is annotated with a *membrane label* $\Gamma \in \widehat{\mathbf{Lab}}_{\mathcal{M}}$ and each action is annotated with a *process label* $\lambda \in \mathbf{Lab}_{\mathcal{P}}$.

We therefore need two distinct sets of labels. We have the set of *process labels* $\mathbf{Lab}_{\mathcal{P}}$, ranged over by $\alpha, \beta, \gamma \dots$. Moreover, given a set of basic membrane labels $\mathbf{Lab}_{\mathcal{M}}$, we have

the associated set of *membrane labels* $\widehat{\text{Lab}}_{\mathcal{M}}$, ranged over by $\Delta, \Gamma, \Psi \dots$. The set $\widehat{\text{Lab}}_{\mathcal{M}}$ is inductively defined as follows: (i) $\text{Lab}_{\mathcal{M}} \subseteq \widehat{\text{Lab}}_{\mathcal{M}}$; (ii) if $\Gamma, \Delta \in \widehat{\text{Lab}}_{\mathcal{M}}$ and $\lambda, \mu \in \text{Lab}_{\mathcal{P}}$, then $\text{phago}(\Gamma, \Delta, \lambda, \mu)$, $\text{pino}(\Delta, \lambda)$ and $\text{bud}(\Gamma, \Delta, \lambda, \mu) \in \widehat{\text{Lab}}_{\mathcal{M}}$.

The system $\sigma\langle P \rangle^{\Gamma}$ describes a *membrane*, decorated by label Γ ⁴ that contains the system P and that performs the *membrane process* σ , describing its interaction capabilities. The construct $a^{\lambda}.\sigma$ defines a sequential process that executes an action a , decorated by label λ , and then behaves as the process σ . We adopt standard syntactical abbreviations: a^{λ} stands for $a^{\lambda}.0$, $\langle P \rangle^{\Gamma}$ stands for $0\langle P \rangle^{\Gamma}$, and $\sigma\langle \diamond \rangle^{\Gamma}$ is a shorthand for $\sigma\langle \diamond \rangle^{\Gamma}$.

$P, Q ::= \diamond \mid P \circ Q \mid !P \mid \sigma\langle P \rangle^{\Gamma}$ $\sigma, \tau ::= 0 \mid \sigma\tau \mid !\sigma \mid a^{\lambda}.\sigma$ $a, b ::= \text{phago}_n \mid \overline{\text{phago}}_n(\rho) \mid \text{exo}_n \mid \overline{\text{exo}}_n \mid \text{pino}(\rho) \mid \text{bud}_n \mid \overline{\text{bud}}_n(\sigma)$	systems Sys membrane processes Proc actions Act
<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> $(\text{PAR}) \frac{P \xrightarrow{l} Q}{P \circ R \xrightarrow{l} Q \circ R}$ </div> <div style="text-align: center;"> $(\text{BRANE}) \frac{P \xrightarrow{l} Q}{\sigma\langle P \rangle^{\Gamma} \xrightarrow{l} \sigma\langle Q \rangle^{\Gamma}}$ </div> <div style="text-align: center;"> $(\text{STRUCT}) \frac{P \equiv P' \wedge P' \xrightarrow{l} Q' \wedge Q' \equiv Q}{P \xrightarrow{l} Q}$ </div> </div>	
$(\text{PHAGO}) \text{phago}_n^{\lambda}.\sigma\langle P \rangle^{\Delta} \circ \overline{\text{phago}}_n^{\mu}(\rho).\tau\langle Q \rangle^{\Gamma} \xrightarrow{\text{phago}_1(\Delta, \Gamma, \lambda, \mu)} \tau\langle \rho\langle \sigma\langle P \rangle^{\Delta} \rangle^{\text{phago}(\Delta, \Gamma, \lambda, \mu)} \circ Q \rangle^{\Gamma}$ $(\text{EXO}) \overline{\text{exo}}_n^{\mu}.\tau\langle \text{exo}_n^{\lambda}.\sigma\langle P \rangle^{\Delta} \circ Q \rangle^{\Gamma} \xrightarrow{\text{exo}_1(\Delta, \Gamma, \lambda, \mu)} P \circ \sigma\langle \sigma_0\tau\langle Q \rangle^{\Gamma} \rangle^{\Gamma}$ $(\text{PINO}) \text{pino}^{\lambda}(\rho).\sigma\langle P \rangle^{\Delta} \xrightarrow{\text{pino}_1(\Delta, \lambda)} \sigma\langle \tau\langle \rho \rangle^{\text{pino}(\Delta, \lambda)} \circ P \rangle^{\Delta}$ $(\text{BUD}) \overline{\text{bud}}_n^{\mu}(\rho).\tau\langle \text{bud}_n^{\lambda}.\sigma\langle P \rangle^{\Delta} \circ Q \rangle^{\Gamma} \xrightarrow{\text{bud}_1(\Delta, \Gamma, \lambda, \mu)} \rho\langle \sigma\langle \sigma_0\langle P \rangle^{\Delta} \rangle^{\text{bud}(\Delta, \Gamma, \lambda, \mu)} \circ \tau\langle Q \rangle^{\Gamma}$	

Table 1: Syntax and Reduction Rules for (Labelled) Brane.

The semantics of the calculus is given by the reduction rules in Tab. 1, modulo the structural congruence rules, here omitted because standard (see [7]). The labelled transition relation is \xrightarrow{l} , where $P \xrightarrow{l} Q$ denotes that the system P evolves into the system Q performing a reaction described by the *transition label* $l \in \text{Lab}_{\mathcal{T}}$. The set of transition labels $\text{Lab}_{\mathcal{T}}$ (ranged over by $l_1, l_2 \dots$) is defined as follows:

$$\text{Lab}_{\mathcal{T}} = \{\text{pino}_1(\Delta, \lambda), \text{phago}_1(\Gamma, \delta, \lambda, \mu), \text{exo}_1(\Gamma, \Delta, \lambda, \mu), \text{bud}_1(\Gamma, \Delta, \lambda, \mu) \mid \Gamma, \Delta \in \widehat{\text{Lab}}_{\mathcal{M}}, \lambda, \mu \in \text{Lab}_{\mathcal{P}}\}$$

Besides the standard reduction rule for congruence (STRUCT), and the contextual rules to propagate reductions across parallel composition (PAR) and membrane nesting (BRANE), there are the axioms specific of the membrane actions.

Rule (PHAGO) models the inclusion of an external membrane, labelled by Δ , inside a membrane, labelled by Γ . The two membranes Δ and Γ exercise the actions phago_n^{λ} and

⁴ For brevity, from now on, we will usually write membrane Γ , instead of membrane labelled by Γ .

$\overline{\text{phago}}_n^\mu(\rho)$, respectively. Once engulfed, the membrane Δ is enclosed inside a new membrane with label $\text{phago}(\Delta, \Gamma, \lambda, \mu)$, which has associated the process ρ . The corresponding transition label is $\text{phago}_1(\Delta, \Gamma, \lambda, \mu)$. Rule (EXO) models the expulsion of the membrane Δ , outside the external membrane Γ , triggered by the actions exo_n^λ and $\overline{\text{exo}}_n^\mu$, respectively. The corresponding transition label is $\text{exo}_1(\Delta, \Gamma, \lambda, \mu)$. In the rule (PINO), the membrane Δ , creates a new empty membrane, labelled by $\text{pino}(\Delta, \lambda)$, inside itself. The action $\text{pino}^\lambda(\rho)$ is equipped with a process ρ that will be associated to the new membrane. The corresponding transition label is $\text{pino}_1(\Delta, \lambda)$. Finally, in the rule (BUD), the membrane Γ expels the child membrane Δ , performing the actions $\overline{\text{bud}}_n^\mu(\rho)$ and bud_n^λ , respectively. The membrane Δ is wrapped inside a new membrane with label $\text{bud}(\Delta, \Gamma, \lambda, \mu)$ and has associated the process ρ . The corresponding transition label is $\text{bud}_1(\Delta, \Gamma, \lambda, \mu)$.

The semantics of a system is defined as a *Labelled Transition System* (LTS). Given $P \in \text{Sys}$, we use $LTS(P) = (X, \xrightarrow{l}, P)$ ⁵ to denote the LTS obtained from the initial system P by applying the rules and axioms in Tab. 1. Moreover, to ensure the correctness of our analysis, we assume that the system P is *well labelled*, i.e. that the process labels occurring in P are all distinct. The analysis is not correct in general for every labelling of the initial systems.

Remark 1. It is worth briefly discussing the role of labels in our calculus. Process and membrane labels will be exploited in the analysis to maintain the information on the topological structure of systems. To decorate the new membranes introduced by reactions of the calculus, we adopt composite membrane labels that record the labels of the membrane and of the actions that interact. This labelling technique may introduce arbitrarily nested membrane labels (e.g. $\text{bud}(\Delta, \text{pino}(\Gamma, \delta), \lambda, \mu)$) and even an infinite number of membrane labels.

Furthermore, we decorate the reduction steps with transition labels giving information on the labels of the actions and on the membranes involved in the interaction. This information will be exploited in the analysis to establish a correspondence between reduction steps and abstract transitions.

3 Communication via Mobile Vesicles

To illustrate our analysis, we consider hypothetical scenarios of communication via mobile vesicles. A mobile vesicle containing the substance to be transmitted springs from a membrane *Source* and eventually fuses with a membrane *Target* releasing its content inside of it. In eucaryotic cells, a large variety of proteins is targeted to its final destination via mobile transport vesicles, i.e. small membrane-enclosed sacs separated from the cytosol by a lipid bilayer. Proteins can be contained in the vesicles (i.e. secretory proteins) or embedded in their membrane (i.e. transmembrane proteins). Through vesicular trafficking, proteins follow routes involving intracellular locations (e.g. endoplasmic reticulum, Golgi apparatus or lysosomes) as well as the plasma membrane, in the case of endo- and exocytosis. Since vesicular transport is essential in the organisation of eukaryotic cells, understanding the mechanisms that control vesicle budding and fusion is an active research topic in cell biology.

Roughly speaking, we are focussing on modelling the communication (through a vesicle) of a molecule X embedded in a *Source* membrane to specified *Target* compartments. More specifically we consider the encoding in Brane calculi presented in [28], where communication based upon the natural budding of mobile vesicle in a cell was modelled as follows. A

⁵ Here, $X \subseteq \text{Sys}$ stands for the set of systems that are reachable from system P .

vesicle containing (embedded in its membrane) a molecule that needs to be shuttled between two compartments, buds from a membrane *Source*. Then, it is engulfed by another compartment (the *Target* membrane) through phagocytosis (creating a coat membrane containing the vesicle) and, finally, the coat of the mobile vesicle is decomposed within the membrane *Target*, releasing the transported molecule in it. Here, we introduce a more general situation, by introducing two substantially different scenarios of communication via mobile vesicles. In both cases *Source* can communicate the molecule X , via a mobile vesicle, to different membranes $Target_1$ and $Target_2$. In the first version, the membrane *Source* can communicate the molecule X either to $Target_1$ or to $Target_2$, while in the second one, one instance of molecule X is communicated to membrane $Target_1$ and the other one to $Target_2$. Both the previous systems can be rendered along the lines of [28].

Example 1. To model the first scenario, we slightly extend the original model in [28]. In this case we have two vesicles and two target membranes, and each vesicle can be phagocytosed by one of the target membranes. Note that, in the biological setting, this particular kind of non-deterministic behaviour is of great interest because it may arise, e.g. in the extracellular environment of cells, where extracellular vesicles, which are small vesicles released by donor cells, can be taken up by any recipient target cell. This transport mechanism plays an important role in cell-to-cell communication.

The encoding of this scenario is reported in the upper part of Tab. 2, where the processes σ_S , σ_{target_i} , τ_X and τ'_i (with $i = 1, 2$) stand for membranes processes (not specified as not relevant at this level of abstraction), and where we decorate actions and membranes with basic membrane labels in $\text{Lab}_{\mathcal{M}}$ ⁶. The molecule X to be transmitted is enclosed inside a membrane labelled by F . Such membrane triggers the communication process, exercising the action bud_n^γ and leading to the gemmation of one of the two vesicles that will transport the molecule into one of the two targets.

For simplicity, we only illustrate the dynamic evolution of the first vesicle in the lower part of Tab. 2, where we underline the prefixes involved in each transition. The membrane F buds from the membrane *source* thus creating the first vesicle. Then the vesicle fuses with the corresponding membrane $target_1$, by means of actions $\text{phago}_{n_1}^{\mu_1}$ and $\text{exo}_{n_1}^{\nu_1}$. Note that the membrane created by the bud reaction is decorated with the label Π_1 , while the one created by the phago reaction with the label Π_{11} . Furthermore, the bud, phago and exo reactions are decorated with the transition labels l_1, l_{11} and l_{12} , respectively. The dynamic evolution of the second vesicle is analogous.

It should be clear that in this case the molecule X cannot be simultaneously present in membranes $target_1$ and $target_2$: either the molecule is transmitted to the first target or, alternatively, it is transmitted to the second one.

Example 2. To model the second scenario, we again extend the original model in [28]. In this case there are two membranes *Source*, each containing the molecule X that has to be transmitted. One instance of substance X is communicated via a mobile vesicle to membrane $Target_1$ while the other one membrane is communicated via a mobile vesicle to membrane $Target_2$. Also this case may arise in the biological setting, when different types of vesicles are simultaneously present, and each type of cell can selectively interact only with the correct target membrane, i.e. each kind of cargo is transported to the specific programmed location

⁶ We also assume the system S to be well labelled.

$S \stackrel{def}{=} \sigma_S(\text{Source} \circ \text{Target}_1 \circ \text{Target}_2)^{skin} \text{Target}_i \stackrel{def}{=} !\overline{\text{phago}}_{n_i}^{\delta_i}(\overline{\text{exo}}_{n_i}^{\beta_i}) \sigma_{\text{target}_i}(\emptyset)^{\text{target}_i}$	$Vesicle_i \stackrel{def}{=} \text{phago}_{n_i}^{\mu_i}.\text{exo}_{n_i}^{\nu_i} \tau'_i$	$Source \stackrel{def}{=} \sigma_{\text{Source}}(\sigma_X(\downarrow X)^\Gamma)^{\text{source}}$
$\sigma_X \stackrel{def}{=} \text{bud}_n^\gamma \tau_X$	$\sigma_{\text{source}} \stackrel{def}{=} !\overline{\text{bud}}_n^{\lambda_1}(\text{Vesicle}_1) !\overline{\text{bud}}_n^{\lambda_2}(\text{Vesicle}_2)$	
$S = \sigma_S(\text{Source} \circ \text{Target}_1 \circ \text{Target}_2)^{skin} \equiv$ $\sigma_S(!\overline{\text{bud}}_n^{\lambda_1}(\text{Vesicle}_1) !\overline{\text{bud}}_n^{\lambda_2}(\text{Vesicle}_2)(\text{bud}_n^\gamma \tau_X(\downarrow X)^\Gamma)^{\text{source}} \circ \text{Target}_1 \circ \text{Target}_2)^{skin} \xrightarrow{l_1}$ $\sigma_S(!\overline{\text{bud}}_n^{\lambda_1}(\text{Vesicle}_1) !\overline{\text{bud}}_n^{\lambda_2}(\text{Vesicle}_2)(\emptyset)^{\text{source}} \circ \text{Vesicle}_1(\tau_X(\downarrow X)^\Gamma)^{II_1} \circ \text{Target}_1 \circ \text{Target}_2)^{skin} \equiv$ $\sigma_S(\sigma_{\text{source}}(\emptyset)^{\text{source}} \circ \underline{\text{phago}}_{n_1}^{\mu_1}.\text{exo}_{n_1}^{\nu_1} \tau'_1(\tau_X(\downarrow X)^\Gamma)^{II_1} \circ \overline{\text{phago}}_{n_1}^{\delta_1}(\overline{\text{exo}}_{n_1}^{\beta_1}) \sigma_{\text{target}_1}(\emptyset)^{\text{target}_1} \circ \text{Target}_2)^{skin} \xrightarrow{l_{11}}$ $\sigma_S(\sigma_{\text{source}}(\emptyset)^{\text{source}} \circ !\overline{\text{phago}}_{n_1}^{\delta_1}(\overline{\text{exo}}_{n_1}^{\beta_1}) \sigma_{\text{target}_1}(\overline{\text{exo}}_{n_1}^{\beta_1}(\text{exo}_{n_1}^{\nu_1} \tau'_1(\tau_X(\downarrow X)^\Gamma)^{II_1})^{II_{11}})^{\text{target}_1} \circ \text{Target}_2)^{skin} \xrightarrow{l_{12}}$ $\sigma_S(\sigma_{\text{source}}(\emptyset)^{\text{source}} \circ !\overline{\text{phago}}_{n_1}^{\delta_1}(\overline{\text{exo}}_{n_1}^{\beta_1}) \sigma_{\text{target}_1}(\tau_X(\downarrow X)^\Gamma \circ \tau'_1(\emptyset)^{II_{11}})^{\text{target}_1} \circ \text{Target}_2)^{skin}$ <p>where</p> $l_1 = \text{bud}_1(\Gamma, \text{source}, \gamma, \lambda_1) \quad l_{11} = \text{phago}_1(II_1, \text{target}_1, \mu_1, \delta_1) \quad l_{12} = \text{exo}_1(II_1, II_{11}, \nu_1, \beta_1)$ $II_1 = \text{bud}(\Gamma, \text{source}, \gamma, \lambda_1) \quad II_{11} = \text{phago}(II_1, \text{target}_1, \mu_1, \delta_1)$		

Table 2: First Scenario: Encoding (Upper Part) and Evolution of the First Vesicle (Lower part).

via membrane fusion. This scenario may occur in intracellular vesicles in eukaryotic cells, where membrane-enveloped vesicles travel in between organelles in the cytoplasm.

The encoding of this second scenario is reported in Tab. 3, using process and membrane labels similar to the ones introduced in Tab. 2. Each membrane *Source* can communicate its molecule *X* to the corresponding *Target*. The dynamic evolution is similar to the previous one and it is not reported.

In this case, differently from the previous one, the molecules X will eventually end up in both membranes target₁ and target₂: one occurrence in target₁ and the other one in target₂.

$S' \stackrel{def}{=} \sigma_S(\text{Source}_1 \circ \text{Source}_2 \circ \text{Target}_1 \circ \text{Target}_2)^{skin} \text{Target}_i \stackrel{def}{=} !\overline{\text{phago}}_{n_i}^{\delta_i}(\overline{\text{exo}}_{n_i}^{\beta_i}) \sigma_{\text{target}_i}(\emptyset)^{\text{target}_i}$	$Vesicle_i \stackrel{def}{=} \text{phago}_{n_i}^{\mu_i}.\text{exo}_{n_i}^{\nu_i} \tau'_i$	$Source_i \stackrel{def}{=} \sigma_{\text{Source}_i}(\sigma_{X_i}(\downarrow X)^\Gamma)^{\text{source}}$
$\sigma_{X_i} \stackrel{def}{=} \text{bud}_n^\gamma \tau_{X_i}$	$\sigma_{\text{source}_i} \stackrel{def}{=} !\overline{\text{bud}}_n^{\lambda_i}(\text{Vesicle}_i)$	

Table 3: Second Scenario: Encoding.

4 The Abstraction

The analysis computes a description of the possible structure of all the derivatives of the system under investigation. Following the AI approach, the analysis result is calculated by

collecting all reachable abstract states representing approximate information on the evolution of the system. More specifically, *abstract states* provide information on the possible hierarchical structure of membranes, and on the processes that may be associated to each membrane, together with information about the possible number of occurrences of membrane and process labels. We prove that the analysis is a *safe over-approximation* of the concrete behaviour. Furthermore, we show that the properties of the abstract semantics help us in computing our analysis in polynomial time (see Theorem 1).

Abstract Membrane Labels. To guarantee that the analysis can be computed in a finite number of steps, we need an *abstraction of membrane labels*. In the abstract setting, the basic membrane labels are defined as $\text{Lab}_{\mathcal{M}}^{\circ} = \text{Lab}_{\mathcal{M}} \cup \{\text{@}\}$, where the special symbol @ represents the outermost membrane. Then, we derive the corresponding set of *abstract membrane labels* $\widehat{\text{Lab}}_{\mathcal{M}}^{\circ}$, ranged over by $\Gamma^{\circ}, \Delta^{\circ}, \dots$, defined as the least set s.t.: (i) $\text{Lab}_{\mathcal{M}}^{\circ} \subseteq \widehat{\text{Lab}}_{\mathcal{M}}^{\circ}$; and (ii) if $\Gamma^{\circ}, \Delta^{\circ} \in \widehat{\text{Lab}}_{\mathcal{M}}^{\circ}$ and $\lambda, \mu \in \text{Lab}_{\mathcal{P}}$ then $(\Gamma^{\circ}, \Delta^{\circ}, \lambda, \mu)$ and $(\Gamma^{\circ}, \lambda) \in \widehat{\text{Lab}}_{\mathcal{M}}^{\circ}$.

Note that in the previously introduced abstraction of membrane labels, arbitrarily nested membrane labels can still arise (e.g. $(\Gamma^{\circ}, (\Delta^{\circ}, \Theta^{\circ}, \nu, \pi), \lambda, \mu)$). As a consequence, we introduce further approximations to guarantee that the abstract membrane labels generated in the analysis are finite. We then consider the set of *abstract membrane labels parametric w.r.t.* the level of nesting depth $d \in \mathbb{N}^+$ defined as:

$$\widehat{\text{Lab}}_{\mathcal{M}}^d = \{\Delta^{\circ} \mid \Delta^{\circ} \in \widehat{\text{Lab}}_{\mathcal{M}}^{\circ} \text{ and } \text{depth}(\Delta^{\circ}) \leq d\} \cup \{(\top, \top, \lambda, \mu), (\top, \lambda) \mid \lambda, \mu \in \text{Lab}_{\mathcal{P}}\}$$

where $\text{depth}(\Delta^{\circ})$ gives the maximal number of nesting parenthesis levels occurring in Δ° . Intuitively, all the abstract membrane labels with depth greater than d are approximated with the following new special membrane labels: $(\top, \top, \lambda, \mu)$ and (\top, λ) .

This is formalised by introducing an *abstraction function* that maps a membrane label into an abstract membrane label with respect to a given parameter d .

Definition 1. Let $d \in \mathbb{N}^+$ and $\Delta \in \widehat{\text{Lab}}_{\mathcal{M}}$. The abstract version of Δ , denoted by $\Delta^{\bullet} \in \widehat{\text{Lab}}_{\mathcal{M}}^d$ ⁷, is inductively defined as follows,

1. $\Delta \in \text{Lab}_{\mathcal{M}} \Rightarrow \Delta^{\bullet} = \Delta$;
2. $\Delta = \#(\Gamma, \Psi, \lambda, \mu)$ with $\# \in \{\text{bud}, \text{phago}\} \Rightarrow \Delta^{\bullet} = \begin{cases} (\Gamma^{\bullet}, \Psi^{\bullet}, \lambda, \mu) & \text{if } \text{depth}((\Gamma^{\bullet}, \Psi^{\bullet}, \lambda, \mu)) \leq d \\ (\top, \top, \lambda, \mu) & \text{otherwise} \end{cases}$
3. $\Delta = \#(\Gamma, \lambda)$ with $\# \in \{\text{pino}\} \Rightarrow \Delta^{\bullet} = \begin{cases} (\Gamma^{\bullet}, \lambda) & \text{if } \text{depth}((\Gamma^{\bullet}, \lambda)) \leq d \\ (\top, \lambda) & \text{otherwise} \end{cases}$

By summarising, Δ° denotes a generic abstract membrane label, while Δ^{\bullet} exactly denotes the abstract membrane label that is the abstract version of the membrane label Δ .

Abstract States. An *abstract state* reports information on the parent-child relation between membranes and a description of the processes possibly associated to each membrane. Furthermore, it reports information about the possible number of occurrences of membrane and process labels. The occurrence counting information approximates the *global* number of membrane and process labels that may appear in any system.

⁷ For simplicity, we omit the explicit indication of the parameter d (assume fixed once for all).

To describe the structure of systems, we adopt an *abstract representation*, formally represented by a set of pairs, i.e. by a relation, that, for any abstract membrane label Δ° , gives: (i) the abstract membrane labels that *may* be child membranes of Δ° ; and (ii) the *sequential processes* that *may* be associated to membrane Δ° .

Definition 2 (Abstract Representation). *An abstract representation R° is a relation $R^\circ \subseteq \widehat{\text{Lab}}_{\mathcal{M}}^d \times (\widehat{\text{Lab}}_{\mathcal{M}}^d \cup \text{SProc})$, where $\text{SProc} = \{a^\lambda.\sigma \mid a^\lambda.\sigma \in \text{Proc}\}$ denotes the subset of sequential processes. We use \mathcal{R}° to denote the set of abstract representations.*

Given R° , if the pair $(\Delta^\circ, \Gamma^\circ) \in R^\circ$, then the abstract membrane Γ° *may* be a child membrane of the membrane Δ° . Similarly, if the pair $(\Delta^\circ, a^\lambda.\sigma) \in R^\circ$, then the sequential process $a^\lambda.\sigma$ *may* be associated with membrane Δ° .

To describe *occurrence counting* information, we adopt the set $\text{Mul} = \{1, \omega\}$ where each $x \in \text{Mul}$ denotes a *multiplicity* with the expected interpretation: 1 indicates at most one occurrence, while ω indicates any number of occurrences. The set of multiplicities Mul comes equipped with the standard order $1 \leq \omega$ and with the binary addition operator $+$, that, for any $x_1, x_2 \in \text{Mul}$, gives ω as a result.

Definition 3 (Occurrence Counting). *An occurrence counting function is a partial function $O^\circ : \widehat{\text{Lab}}_{\mathcal{M}}^d \cup \text{Lab}_{\mathcal{P}} \rightarrow \text{Mul}$. We use \mathcal{O}° for the set of occurrence counting functions.*

By using a standard notation, an occurrence counting function O° can be alternatively represented by a set of pairs: $\{(\ell, x) \mid \ell \in \text{dom}(O^\circ) \wedge O^\circ(\ell) = x\}$.

We rely on some auxiliary operators on occurrence counting functions. First, we introduce the substitution operator $O^\circ[x/\ell]$ that, applied to the occurrence counting function O° , returns the function where the multiplicity of $\ell \in \widehat{\text{Lab}}_{\mathcal{M}}^d \cup \text{Lab}_{\mathcal{P}}$ is replaced by $x \in \text{Mul}$.

Moreover, we define an operator \cup^+ that computes the addition of two functions $O_1^\circ, O_2^\circ \in \mathcal{O}^\circ$, the occurrence counting function $O_1^\circ \cup^+ O_2^\circ$ is defined as follows, where $\ell \in \widehat{\text{Lab}}_{\mathcal{M}}^d \cup \text{Lab}_{\mathcal{P}}$,

$$O_1^\circ \cup^+ O_2^\circ(\ell) = \begin{cases} O_1^\circ(\ell) + O_2^\circ(\ell) & \text{if } \ell \in \text{dom}(O_1^\circ) \cap \text{dom}(O_2^\circ) \\ O_1^\circ(\ell) & \text{if } \ell \in \text{dom}(O_1^\circ), \ell \notin \text{dom}(O_2^\circ) \\ O_2^\circ(\ell) & \text{if } \ell \in \text{dom}(O_2^\circ), \ell \notin \text{dom}(O_1^\circ) \end{cases}$$

We now have all the ingredients to define abstract states.

Definition 4 (Abstract State). *An abstract state is a pair $S^\circ = (R^\circ, O^\circ)$, where $R^\circ \in \mathcal{R}^\circ$ is an abstract representation and $O^\circ \in \mathcal{O}^\circ$ is an occurrence counting function. We use \mathcal{S}° for the set of abstract states.*

In standard AI style, the abstract states come equipped with an approximation order (denoted by \sqsubseteq°) that allows us to compare two approximations in terms of precision.

Definition 5 (Approximation Orders).

- Given $O_1^\circ, O_2^\circ \in \mathcal{O}^\circ$, we say that $O_1^\circ \sqsubseteq_O O_2^\circ$ iff for each $\ell \in \widehat{\text{Lab}}_{\mathcal{M}}^d \cup \text{Lab}_{\mathcal{P}}$ such that $\ell \in \text{dom}(O_1)$, we have $O_1(\ell) = x_1$ and $O_2(\ell) = x_2$ with $x_1 \leq x_2$.
- Given $S_1^\circ, S_2^\circ \in \mathcal{S}^\circ$, we say that $S_1^\circ \sqsubseteq^\circ S_2^\circ$ iff $S_1^\circ = (R_1^\circ, O_1^\circ)$ and $S_2^\circ = (R_2^\circ, O_2^\circ)$, $R_1^\circ \subseteq R_2^\circ$ and $O_1^\circ \sqsubseteq_O O_2^\circ$.

Given the previous orders, the corresponding least upper bounds (l.u.b.), \sqcup_O over occurrence counting functions and \sqcup° over abstract states, are defined as expected.

To formally relate systems and abstract states, we introduce a *translation function* t° that maps systems into abstract states. The function $t^\circ : \widehat{\text{Lab}}_{\mathcal{M}}^d \times \text{Sys} \rightarrow \mathcal{S}^\circ$, presented in Tab. 4, returns an abstract state, describing the system, with respect to an abstract membrane that represents the enclosing membrane. The definition relies, in turn, on a corresponding translation function for processes $t^\circ : \widehat{\text{Lab}}_{\mathcal{M}}^d \times \text{Proc} \rightarrow \mathcal{S}^{\circ 8}$.

$t^\circ(\Delta^\circ, \diamond)$	$= (\emptyset, \emptyset)$
$t^\circ(\Delta^\circ, P \circ Q)$	$= (R_1^\circ \cup R_2^\circ, O_1^\circ \cup^+ O_2^\circ)$ where $t^\circ(\Delta^\circ, P) = (R_1^\circ, O_1^\circ), t^\circ(\Delta^\circ, Q) = (R_2^\circ, O_2^\circ)$
$t^\circ(\Delta^\circ, !P)$	$= (R^\circ, O^\circ[\omega/\ell]_{\ell \in \text{dom}(O^\circ)})$ where $t^\circ(\Delta^\circ, P) = (R^\circ, O^\circ)$
$t^\circ(\Delta^\circ, \sigma(P)^\Gamma)$	$= \begin{cases} (R_1^\circ \cup R_2^\circ \cup \{(\Delta^\circ, \Gamma^\bullet)\}, O_1^\circ \cup^+ O_2^\circ \cup^+ \{(\Gamma^\bullet, 1)\}) & \text{if } \sigma \neq 0 \vee P \neq \diamond \\ (\emptyset, \emptyset) & \text{otherwise} \end{cases}$ where $t^\circ(\Gamma^\bullet, P) = (R_1^\circ, O_1^\circ)$ and $t^\circ(\Gamma^\bullet, \sigma) = (R_2^\circ, O_2^\circ)$
$t^\circ(\Delta^\circ, 0)$	$= (\emptyset, \emptyset)$
$t^\circ(\Delta^\circ, \sigma \tau)$	$= (R_1^\circ \cup R_2^\circ, O_1^\circ \cup^+ O_2^\circ)$ where $t^\circ(\Delta^\circ, \sigma) = (R_1^\circ, O_1^\circ)$ and $t^\circ(\Delta^\circ, \tau) = (R_2^\circ, O_2^\circ)$
$t^\circ(\Delta^\circ, !\sigma)$	$= (R^\circ, O^\circ[\omega/\ell]_{\ell \in \text{dom}(O^\circ)})$ where $t^\circ(\Delta^\circ, \sigma) = (R^\circ, O^\circ)$
$t^\circ(\Delta^\circ, a^\lambda.\sigma)$	$= (\{(\Delta^\circ, a^\lambda.\sigma)\}, \{(\lambda, 1)\})$

Table 4: Translation Function for Systems and Processes.

Based on the above defined translation function, it is immediate to derive a corresponding *abstraction function* that, given a system, returns the abstract state that is its *best approximation*. Intuitively, the best approximation is the *most precise* (with respect to the order \sqsubseteq°) abstract state that *safely represents* the information contained in the system.

Definition 6 (Abstraction function). We define $\alpha_{\text{Sys}} : \text{Sys} \rightarrow \mathcal{S}^\circ$ such that, given $P \in \text{Sys}$, $\alpha_{\text{Sys}}(P) = (R^\circ, O^\circ \cup^+ \{(\@, 1)\})$, where $t^\circ(\@, P) = (R^\circ, O^\circ)$.

The best approximation of a system is obtained by applying the translation function t° w.r.t. the abstract membrane label $\@$ representing the outermost membrane. Note that the previously introduced notions can be used to express the fundamental notion of safe approximation between abstract states and systems: an abstract state S° *safely approximates* a system P if and only if $\alpha_{\text{Sys}}(P) \sqsubseteq^\circ S^\circ$. Moreover, the abstraction function is exploited to compute the initial abstract state in the abstract semantics.

Example 3. Let us consider the system S introduced in Ex. 1 (see Tab. 2). Assuming the parameter for the depth of abstract membrane labels $d = 3$, the best approximation of S is given by the abstract state $\alpha_{\text{Sys}}(S) = S_0^\circ = (R_0^\circ, O_0^\circ)$, illustrated in Tab. 5. For convenience, both the abstract representation R_0° and the occurrence counting function O_0° are described by means of tables, on the left and on the right, respectively.

The table on the left contains one row for each abstract membrane label Δ° in the domain of R_0° . For each Δ° the corresponding row reports in the second column, the set of

⁸ For simplicity, we use t° for both abstract systems and processes.

membrane	children	processes
@	skin	
skin	source, target _i	σ_S
source	Γ	$\overline{\text{bud}}_n^{\lambda_i}(\text{Vesicle}_i)$
target _i		$\overline{\text{phago}}_{n_i}^{\delta_i}(\overline{\text{exo}}_{n_i}^{\beta_i}), \sigma_{\text{target}_i}$
Γ	X	$\text{bud}_n^\gamma, \tau_X$

membrane/process	multiplicity
@	1
skin, source	1
target ₁ , target ₂	1
Γ	1
γ	1
$\lambda_1, \lambda_2, \delta_1, \delta_2$	ω

Table 5: The Abstract State $\alpha_{\text{sys}}(S) = S_0^\circ = (R_0^\circ, O_0^\circ)$, where $i = 1, 2$.

abstract membrane labels that *may* be children of Δ° , and in the third column the set of sequential processes that *may* be associated to membrane Δ° . More formally, $\text{children}(\Delta^\circ) = \{\Theta^\circ \mid (\Delta^\circ, \Theta^\circ) \in R_0^\circ\}$ and $\text{processes}(\Delta^\circ) = \{\sigma \mid (\Delta^\circ, \sigma) \in R_0^\circ\}$. Hence, the third line can be read as the membrane *source* may include the membrane Γ , and it may have associated the processes $\overline{\text{bud}}_n^{\lambda_1}(\text{Vesicle}_1)$ and $\overline{\text{bud}}_n^{\lambda_2}(\text{Vesicle}_2)$.

The table on the right reports the multiplicities for each abstract membrane and process label in the domain of O_0° . For instance, the membrane labels *skin* and *source* have multiplicity 1, while the process labels λ_i and δ_i (with $i = 1, 2$) come with multiplicity ω . The corresponding prefixes occur indeed under the scope of a replication (see the rules in Tab. 4).

Abstract Transitions. The abstract semantics is given in terms of the abstract transition relation $\xrightarrow{l^\circ}_\circ$ among abstract states, where $l^\circ \in \text{Lab}_T^\circ$ is the *abstract transition label* describing the reaction. The abstract transitions are obtained by introducing inference rules for abstract states that model the abstract counterpart of the membrane interactions possible in the concrete system.

The set of *abstract transition labels* Lab_T° ⁹ (ranged over by $l_1^\circ, l_2^\circ, \dots$) is defined as in the concrete case, by replacing membrane labels with abstract membrane labels. Thus, we have:

$$\text{Lab}_T^\circ = \{\text{pino}_1(\Delta^\circ, \lambda), \text{bud}_1(\Gamma^\circ, \Delta^\circ, \lambda, \mu), \text{exo}_1(\Gamma^\circ, \Delta^\circ, \lambda, \mu), \text{phago}_1(\Gamma^\circ, \Delta^\circ, \lambda, \mu) \mid \Gamma^\circ, \Delta^\circ \in \widehat{\text{Lab}}_{\mathcal{M}}^d, \lambda, \mu \in \text{Lab}_{\mathcal{P}}\}$$

Due to the lack of space, we comment here only on the abstract inference rule, given in Tab. 6, corresponding to the (Bud) interaction. The abstract inference rules (PHAGO[°]), (EXO[°]), (PINO[°]) in Tab. 7, corresponding to (PHAGO), (EXO), (PINO) reactions can be derived in similar way from their concrete versions.

Rule (BUD[°]) uses an auxiliary operator *co* to modify an occurrence counting function O° according to a given multiplicity x .

$$\text{co}(O^\circ, x) = \begin{cases} O^\circ[\omega/\ell]_{\ell \in \text{dom}(O^\circ)} & \text{if } x = \omega \\ O^\circ & \text{otherwise} \end{cases}$$

Rule (BUD[°]) simulates the concrete (BUD) rule, by modelling the gemmation of a membrane Δ° from another membrane Γ° that may synchronise on actions $\overline{\text{bud}}_n^\mu(\rho)$ and bud_n^λ .

⁹ For simplicity, we omit the explicit indication of the parameter d when is clear from the context.

This requires that: (i) the abstract membrane Δ° is reported as a possible child of the membrane Γ° (i.e. $(\Gamma^\circ, \Delta^\circ) \in R^\circ$); (ii) according to the abstract representation R° , the actions cobud and bud may be associated to membranes Γ° and Δ° , respectively. Furthermore, it must be the case that the multiplicities of the process labels μ and λ associated to the actions are defined.

The abstract transition label l° is derived, as in the concrete case, by combining the labels of the membranes and of the actions involved. The resulting abstract state is obtained by enriching the abstract state (R°, O°) with information reporting the effects of the possible movement of the membrane Δ° out from the membrane Γ° . This requires to update both the abstract representation and the occurrence counting function. Note that the membrane introduced by the bud reaction is described by the abstract membrane label Π° , obtained by approximating the membrane $(\Delta^\circ, \Gamma^\circ, \lambda, \mu)$ according to its depth.

The abstract representation is extended by introducing the abstract membrane Π° as a possible child of the membrane Φ° (in turn, parent of Γ°), and Δ° as a possible child of membrane Π° . Moreover, we have to introduce information on the membrane processes that may be associated to membranes Γ° , Δ° and Π° . In the case of membrane Π° , this requires to add R_2° obtained by applying the translation function to process ρ related to cobud. Similarly, in the case of the membranes Γ° and Δ° the related abstract representations R_3° and R_4° are obtained by applying the continuations of the two coactions (σ and τ), respectively.

Finally, the occurrence counting function is updated by adding one occurrence of membrane Π° introduced by the bud reaction and the occurrence counting functions O_2° , O_3° , and O_4° , obtained by the translations the process ρ and of the continuations of the coactions. Note that *co* operator allows us to propagate the ω multiplicity, in the case of the continuations of prefixes under replication.

<div style="display: flex; justify-content: space-between;"> <div style="width: 15%;">(BUD$^\circ$)</div> <div style="width: 85%; text-align: center;"> $(\Phi^\circ, \Gamma^\circ) \in R^\circ, (\Gamma^\circ, \Delta^\circ) \in R^\circ,$ $(\Gamma^\circ, \overline{\text{bud}}_n^\mu(\rho).\tau) \in R^\circ, (\Delta^\circ, \text{bud}_n^\lambda.\sigma) \in R^\circ, O^\circ(\lambda) = x, O^\circ(\mu) = y$ </div> </div> <hr style="border: 0.5px solid black;"/> <div style="text-align: center;"> $(R^\circ, O^\circ) \xrightarrow{l^\circ} (R^\circ \cup R_1^\circ \cup R_2^\circ \cup R_3^\circ \cup R_4^\circ, O^\circ \cup^+ \{(\Pi^\circ, 1)\} \cup^+ O_2^\circ \cup^+ \text{co}(O_3^\circ, x) \cup^+ \text{co}(O_4^\circ, y))$ $R_1^\circ = \{(\Phi^\circ, \Pi^\circ), (\Pi^\circ, \Delta^\circ)\},$ $t^\circ(\Pi^\circ, \rho) = (R_2^\circ, O_2^\circ), t^\circ(\Delta^\circ, \sigma) = (R_3^\circ, O_3^\circ), t^\circ(\Gamma^\circ, \tau) = (R_4^\circ, O_4^\circ)$ </div> <div style="text-align: center; margin-top: 10px;"> where $l^\circ = \text{bud}_1(\Delta^\circ, \Gamma^\circ, \lambda, \mu)$ and $\Pi^\circ = \begin{cases} (\Delta^\circ, \Gamma^\circ, \lambda, \mu) & \text{if } (\Delta^\circ, \Gamma^\circ, \lambda, \mu) \in \widehat{\text{Lab}}_M^d, \\ (\top, \top, \lambda, \mu) & \text{otherwise} \end{cases}$ </div>
--

Table 6: Abstract Inference Rule for (BUD).

The Analysis. The analysis of a system P provides an *abstract state* describing the possible topological structure of all the derivatives of P together with occurrence counting information on membrane and process labels. We aim at calculating such abstract state by collecting all the abstract states that can be reached from the initial one $\alpha_{\text{sys}}(P)$, by applying the abstract inference rules.

(PHAGO ^o)	$(\Phi^o, \Delta^o) \in R^o, (\Phi^o, \Gamma^o) \in R^o,$ $(\Delta^o, \text{phago}_n^\lambda \cdot \sigma) \in R^o, (\Gamma^o, \overline{\text{phago}}_n^\mu(\rho) \cdot \tau) \in R^o, O^o(\lambda) = x, O^o(\mu) = y$
	$(R^o, O^o) \xrightarrow{l^o}_o (R^o \cup R_1^o \cup R_2^o \cup R_3^o \cup R_4^o, O^o \cup^+ \{(\Pi^o, 1)\} \cup^+ O_2^o \cup^+ \text{co}(O_3^o, x) \cup^+ \text{co}(O_4^o, y))$ $R_1^o = \{(\Gamma^o, \Pi^o), (\Pi^o, \Delta^o)\}$ $t^o(\Pi^o, \rho) = (R_2^o, O_2^o), t^o(\Delta^o, \sigma) = (R_3^o, O_3^o), t^o(\Gamma^o, \tau) = (R_4^o, O_4^o)$
	where $l^o = \text{phago}_1(\Delta^o, \Gamma^o, \lambda, \mu)$ and $\Pi^o = \begin{cases} (\Delta^o, \Gamma^o, \lambda, \mu) & \text{if } (\Delta^o, \Gamma^o \lambda, \mu) \in \widehat{\text{Lab}}_M^d \\ (\top, \top, \lambda, \mu) & \text{otherwise} \end{cases}$
(EXO ^o)	$(\Phi^o, \Gamma^o) \in R^o, (\Gamma^o, \Delta^o) \in R^o,$ $(\Gamma^o, \overline{\text{exo}}_n^\mu \cdot \tau) \in R^o, (\Delta^o, \text{exo}_n^\lambda \cdot \sigma) \in R^o, O^o(\lambda) = x, O^o(\mu) = y$
	$(R^o, O^o) \xrightarrow{l^o}_o (R^o \cup R_1^o \cup R_2^o \cup R_3^o \cup R_4^o, O^o \cup^+ \text{co}(O_3^o, x) \cup^+ \text{co}(O_4^o, y)),$ $R_1^o = \{(\Phi^o, \Theta^o) \mid (\Delta^o, \Theta^o) \in R^o\}, R_2^o = \{(\Gamma^o, \tau') \mid \tau' \in \text{sub}(C, x, \text{exo}_n^\lambda \cdot \sigma)\},$ $C = \{\sigma' \mid (\Delta^o, \sigma') \in R^o\} \quad t^o(\Gamma^o, \sigma) = (R_3^o, O_3^o), t^o(\Gamma^o, \tau) = (R_4^o, O_4^o)$
	where $l^o = \text{exo}_1(\Delta^o, \Gamma^o, \lambda, \mu)$ and $\text{sub}(C, x, a^\lambda \cdot \sigma) = \begin{cases} C \setminus \{a^\lambda \cdot \sigma\} & \text{if } x = 1 \\ C & \text{otherwise} \end{cases}$
(PINO ^o)	$(\Delta^o, \text{pino}^\lambda(\rho) \cdot \sigma) \in R^o$
	$(R^o, O^o) \xrightarrow{l^o}_o (R^o \cup R_1^o \cup R_2^o \cup R_3^o, O^o \cup^+ \{(\Pi^o, 1)\} \cup^+ O_2^o \cup^+ \text{co}(O_3^o, O^o(\lambda)))$ $R_1^o = \{(\Delta^o, \Pi^o)\}, t^o(\Pi^o, \rho) = (R_2^o, O_2^o), t^o(\Delta^o, \sigma) = (R_3^o, O_3^o)$
	where $l^o = \text{pino}_1(\Delta^o, \lambda)$ and $\Pi^o = \begin{cases} (\Delta^o, \lambda) & \text{if } (\Delta^o, \lambda) \in \widehat{\text{Lab}}_M^d \\ (\top, \lambda) & \text{otherwise} \end{cases}$

Table 7: Abstract Inference Rules for (PHAGO), (EXO), and (PINO).

Nevertheless, the application of the abstract inference rules without a strategy would lead us to have a correct, but very coarse approximation, especially as far as the counting information is concerned. The reason is that, in principle, any enabled reaction would be applied several times. As a consequence, infinite copies of the corresponding membranes and processes are introduced, even though there are cases in which this behaviour cannot occur in the dynamic evolution of the system. Our strategy for overcoming this problem consists in exploiting occurrence counting information to determine which abstract transitions apply to an abstract state. This allows us to more faithfully model the concrete behaviour and therefore to gain precision in our analysis.

We start by analysing the information given by *abstract transition labels*, and by distinguishing: (i) the abstract transitions that require updating the occurrence counting information more than once, from (ii) those that require to do it just once. Intuitively, the former model concrete transitions that may occur more than once in a concrete derivation, while the latter model concrete transitions that occur at most once.

To this aim, we need to define the *multiplicity* of abstract transition labels in a given abstract state, computed by the function $mul : \mathcal{S}^\circ \times \mathbf{Lab}_T^\circ \rightarrow \mathbf{Mul}$ defined as follows:

$$mul((R^\circ, O^\circ), l^\circ) = \begin{cases} \omega & \text{if } l^\circ = \mathbf{pino}_1(\Delta^\circ, \lambda) \wedge O^\circ(\lambda) = \omega, \\ \omega & \text{if } l^\circ = a_1(I^\circ, \Theta^\circ, \lambda, \mu), a_1 \in \{\mathbf{bud}_1, \mathbf{exo}_1, \mathbf{phago}_1\} \wedge O^\circ(\lambda) = O^\circ(\mu) = \omega, \\ 1 & \text{otherwise} \end{cases}$$

Note that the multiplicity assigned to an abstract transition label entirely depends on the multiplicity of labels associated to the actions that participate in the reaction. For any kind of reaction if all the involved actions have multiplicity ω , then also the associated abstract transition label has multiplicity ω . In this case indeed the reaction may be applied *more than once* in the corresponding concrete derivations. On the contrary, if at least one of the actions involved in the reaction has multiplicity 1, then the corresponding reaction may be executed *no more than once* in any derivation of the concrete system.

The multiplicity of transition labels is indeed exploited to compute the abstract semantics, where abstract states are enriched with information on the involved abstract transitions labels. More precisely, we have *configurations* in the form $T^\circ \triangleright S^\circ$, where $S^\circ \in \mathcal{S}^\circ$ is an abstract state and $T^\circ \subseteq \mathbf{Lab}_T^\circ$ is a set of abstract transition labels representing the reactions that have been already exercised. We use \mathcal{C}° to denote the set of configurations.

To describe the evolution of configurations we introduce two meta-inference rules that encode our strategy for the application of abstract rules. These rules allow us to define the evolution of a configuration $T^\circ \triangleright (R_1^\circ, O_1^\circ)$ into another configuration, whenever there exists an abstract reaction $(R_1^\circ, O_1^\circ) \xrightarrow{l^\circ} (R_2^\circ, O_2^\circ)$. The choice of the meta-inference rule depends on the multiplicity of the abstract transition label l° associated to the reaction, i.e. $mul((R_1^\circ, O_1^\circ), l^\circ)$.

$$\frac{(R_1^\circ, O_1^\circ) \xrightarrow{l^\circ} (R_2^\circ, O_2^\circ) \wedge (l^\circ \notin T^\circ \vee (l^\circ \in T^\circ \wedge mul((R_1^\circ, O_1^\circ), l^\circ) = \omega))}{T^\circ \triangleright (R_1^\circ, O_1^\circ) \xrightarrow{l^\circ} T^\circ \cup \{l^\circ\} \triangleright (R_2^\circ, O_2^\circ)} \quad (1)$$

$$\frac{(R_1^\circ, O_1^\circ) \xrightarrow{l^\circ} (R_2^\circ, O_2^\circ) \wedge (l^\circ \in T^\circ \wedge mul((R_1^\circ, O_1^\circ), l^\circ) = 1)}{T^\circ \triangleright (R_1^\circ, O_1^\circ) \xrightarrow{l^\circ} T^\circ \triangleright (R_2^\circ, O_1^\circ)} \quad (2)$$

We can apply the first rule (1), provided that either the reaction l° has never been applied before ($l^\circ \notin T^\circ$) or its multiplicity is ω . Thus, either the reaction associated to l° can be applied only once and it has not been realised or it can be realised any number of times. In both cases, the resulting configuration is obtained by recording that the reaction l° has now been performed, and by updating both the abstract representation and the occurrence counting information.

We can apply the second rule (2), if the reaction related to l° has multiplicity 1 and has already been applied ($l^\circ \in T^\circ$). In this case, it may indicate that the concrete reaction approximated by l° can be performed in another context, different from the one considered before. This requires updating the abstract representation reporting the effects of the move, while the occurrence counting information does not have to be modified since it already reports the correct multiplicities of the membrane and process labels involved in the move.

Example 4. To illustrate the application of meta-inference rule (1) let us consider the abstract state $\alpha_{\text{sys}}(S) = S_0^\circ = (R_0^\circ, O_0^\circ)$ of Ex. 3 (see Tab. 5) describing the best approximation of the system S , presented in Ex. 1. Note that we can apply the abstract rule

(BUD^o) to $\alpha_{\text{Sys}}(S)$, because its premises are fulfilled: (i) $(\text{skin}, \text{source}), (\text{source}, \Gamma) \in R_0^o$ and (ii) $(\Gamma, \text{bud}_n^\gamma), (\text{source}, \overline{\text{bud}}_n^{\lambda_1}(\text{Vesicle}_1)) \in R_0^o$. Furthermore, $O_0^o(\lambda_1)$ and $O_0^o(\gamma)$ are defined. As a consequence, we have $\alpha_{\text{Sys}}(S) \xrightarrow{l_1^o} S_1^o$, where $l_1^o = \text{bud}_1(\Gamma, \text{source}, \gamma, \lambda_1)$ and the state $S_1^o = (R_1^o, O_1^o)$ is the one depicted in Tab. 8. Hence, considering the configu-

membrane	children	processes	membrane/process	multipl.
@	skin		@	1
skin	source, target _i , Π_1^o	σ_S	skin, source,	1
source	Γ	$\overline{\text{bud}}_n^{\lambda_i}(\text{Vesicle}_i),$	target ₁ , target ₂ ,	1
target _i		$\text{phago}_{n_i}^{\delta_i}(\overline{\text{exo}}_{n_i}^{\beta_i}), \sigma_{\text{target}_i}$	Γ, Π_1^o	1
Γ	X	$\text{bud}_n^\gamma, \tau_X$	γ, μ_1	1
$\Pi_1^o = (\Gamma, \text{source}, \gamma, \lambda_1)$	Γ	$\text{phago}_{n_1}^{\mu_1}.\overline{\text{exo}}_{n_1}^{\nu_1}, \tau_1'$	$\lambda_1, \lambda_2, \delta_1, \delta_2$	ω

Table 8: The Abstract State $S_1^o = (R_1^o, O_1^o)$, where $i = 1, 2$.

ration $\emptyset \triangleright \alpha_{\text{Sys}}(S) = S_0^o$, we can apply meta-inference rule (1), since $l_1^o \notin \emptyset$, obtaining $\emptyset \triangleright \alpha_{\text{Sys}}(S) \xrightarrow{l_1^o} \{l_1^o\} \triangleright S_1^o$. Note that, since $O_0^o(\gamma) = 1$ (while $O_0^o(\lambda_1) = \omega$) we have that $\text{mul}((R_0^o, O_0^o), l_1^o) = 1$.

The analysis of a system P provides an abstract state that is obtained by collecting (taking the l.u.b.) all the abstract states that can be reached from the initial configuration $\emptyset \triangleright \alpha_{\text{Sys}}(P)$, by applying the meta-inference rules (1) and (2).

Definition 7 (The Analysis). We define a function $\mathcal{A}^o : \text{Sys} \rightarrow \mathcal{S}^o$ such that for $P \in \text{Sys}$ we have $\mathcal{A}^o(P) = \sqcup_{\{S^o | T^o \triangleright S^o \in X_P^o\}} S^o$, where $X_P^o = \text{lfp}(\mathcal{F}^o(\{\emptyset \triangleright \alpha_{\text{Sys}}(P)\}))^{10}$ and the function $\mathcal{F}^o : \wp(\mathcal{C}^o) \rightarrow \wp(\mathcal{C}^o)$ defined as $\mathcal{F}^o(X_1^o) = X_1^o \cup \{C_2^o \mid C_1^o \xrightarrow{l^o} C_2^o, C_1^o \in X_1^o\}$.

Despite the fact that the analysis involves a fixed point over a power domain, which seems to admit exponentially long increasing paths, the analysis can be computed in polynomial time. For its computation there is no need to deal with power sets: crafting a single maximal path is enough to compute the fixed point, which is a singleton set. This allows us to obtain a polynomial bound. It can be shown indeed that the analysis can be effectively computed by building a single maximal path starting from the initial configuration and ending into a final configuration, i.e. a configuration that cannot further evolve according to the meta-inference rules (1) and (2).

This property relies on the following result.

Theorem 1. Let $P \in \text{Sys}$ be well labelled and let $C_1^o, C_2^o \in \mathcal{C}^o$ be two configurations such that $C_1^o, C_2^o \in \text{lfp}(\mathcal{F}^o(\{\emptyset \triangleright \alpha_{\text{Sys}}(P)\}))$, as defined in Def. 7. If C_1^o and C_2^o are final configurations then $C_1^o = C_2^o$.

¹⁰ where lfp is the least fixed point.

The previous property allows us to calculate the analysis of a system without computing all the configurations that can be reached from the initial one. Indeed, $\mathcal{A}^\circ(P)$ can be computed by building a single path

$$T_0^\circ \triangleright S_0^\circ, T_1^\circ \triangleright S_1^\circ, \dots, T_m^\circ \triangleright S_m^\circ$$

where (i) $T_0^\circ = \emptyset$, $S_0^\circ = \alpha_{\text{sys}}(P)$; (ii) $T_m^\circ \triangleright S_m^\circ$ is a final configuration; and, (iii) for each $i \in [1, m]$ the corresponding configuration is obtained by applying the meta-inference rules (1) and (2) to the previous configuration $T_{i-1}^\circ \triangleright S_{i-1}^\circ$. Note that this path is an ascending chain since, for each $i \in [0, m-1]$, either $T_i^\circ \subset T_{i+1}^\circ$ and $S_i^\circ \sqsubseteq^\circ S_{i+1}^\circ$ or $T_i^\circ \subseteq T_{i+1}^\circ$ and $S_i^\circ \sqsubset^\circ S_{i+1}^\circ$. Hence, we have that the analysis of system P precisely coincides with the final state, i.e. $\mathcal{A}^\circ(P) = S_m^\circ$.

The above reasoning guarantees that the analysis can be computed in *polynomial time*, observing that the number of abstract membranes and transition labels arising in the computation of the analysis is polynomial, when fixing the maximum depth d to a constant value.

Finally, we present the main theorem that shows that the analysis of a system *safely approximates* its concrete behaviour, described by the concrete LTS. This means that each derivative P' of P is over-approximated by the abstract state calculated by the analysis of P .

Theorem 2 (Safety). *Let $P \in \text{Sys}$ be a well labelled system and let $\text{LTS}(P) = (X, \xrightarrow{\cdot}, P)$. We have that $(\sqcup_{P' \in X} \alpha_{\text{sys}}(P')) \sqsubseteq^\circ \mathcal{A}^\circ(P)$.*

5 Our Analysis at work

We now apply our analysis to the systems presented in Sect. 3 (assuming again $d = 3$ analogously as in Sect. 4). We illustrate in more the details the analysis of the system S of Ex. 1, whose first steps have been introduced in Ex. 3 and Ex. 4. Since the analysis of the system S' described in Ex. 2 is similarly obtained, we only comment its results.

Example 5. The analysis of the system S shown in Tab. 2 of Ex. 1 is computed starting from the initial configuration $\emptyset \triangleright \alpha_{\text{sys}}(S)$, where $\alpha_{\text{sys}}(S) = S_0^\circ = (R_0^\circ, O_0^\circ)$ is the abstract state of Tab. 5 (commented in Ex. 3).

The final configuration is given by $\{l_1^\circ, l_{11}^\circ, l_{12}^\circ, l_2^\circ, l_{21}^\circ, l_{22}^\circ\} \triangleright S^\circ$, where $S^\circ = (R^\circ, O^\circ)$ is the abstract state in Tab. 9, and the abstract transition labels for $i = 1, 2$ are:

$$l_i^\circ = \text{bud}_1(\Gamma, \text{source}, \gamma, \lambda_i), l_{i1}^\circ = \text{phago}_1(\Pi_i^\circ, \text{target}_i, \delta_i, \mu_i), l_{i2}^\circ = \text{exo}_1(\Pi_i^\circ, \Pi_{i1}^\circ, \nu_i, \beta_i).$$

Here, the abstract transition labels l_1° , l_{11}° and l_{12}° are the abstract versions of the transition labels l_{11} and l_{12} in Tab. 2. They represent the (bud), (phago) and (exo) reactions performed by the first vesicle, respectively. Analogously, the abstract transition labels l_2° , l_{21}° and l_{22}° represent the labels introduced by the similar evolution of the second vesicle. Note that in this case all the abstract transition labels have multiplicity 1, and consequently only the meta-inference rule (2) can be applied to the final configuration. As a consequence, at this point, no matter which reaction is applied, the final configuration cannot further evolve, and, in particular, the information counting information cannot be updated anymore.

Hence, we can conclude that $\mathcal{A}^\circ(S) = S^\circ = (R^\circ, O^\circ)$. For clarity, the membrane hierarchy described by abstract representation R° is shown in the tree in Fig. 1, where the nodes

membrane	children	processes	membrane/process	multipl.
@	skin		@	1
skin	source, target _i , Π _i ^o	σ _S	skin, source,	1
source	Γ	$\overline{\text{bud}}_n^{\lambda_i}(Vesicle_i)$,	target ₁ , target ₂ ,	1
target _i	Π _{i1} ^o , Γ	$\overline{\text{phago}}_{n_i}^{\delta_i}(\text{exo}_{n_i}^{\beta_i})$, σ _{target_i}	Γ, Π ₁ ^o , Π ₂ ^o	1
Γ	X	bud_n^{γ} , τ _X	Π ₁₁ ^o , Π ₂₁ ^o	1
Π _i ^o = (Γ, source, γ, λ _i)	Γ	$\underbrace{\text{phago}_{n_i}^{\mu_i} \cdot \text{exo}_{n_i}^{\nu_i}}_{Vesicle_i}$,	γ, μ ₁ , μ ₂ , ν ₁ , ν ₂	1
Π _{i1} ^o = (Π _i ^o , target _i , δ _i , μ _i)	Π _i ^o	$\text{exo}_{n_i}^{\nu_i}$, τ' _i	β ₁ , β ₂	1
		$\overline{\text{exo}}_{n_i}^{\beta_i}, Vesicle_i, \tau'_i$	λ ₁ , λ ₂ , δ ₁ , δ ₂	ω

Table 9: The Abstract State $S^\circ = (R^\circ, O^\circ)$, where $i = 1, 2$.

represent the abstract membrane labels and the edges represent the parent-child relation. It is worth noting that the information provided by R° predicts that the membrane Γ , which encloses the molecule X , *may* end up in membrane $target_1$, as well as in membrane $target_2$. However, the occurrence counting information expressed by O° guarantees that the membrane Γ will never reside at the same time inside the membranes $target_1$ and $target_2$. To point this out the two alternative inclusions of membrane Γ inside the membranes $target_1$ and $target_2$ the lines are displayed with dotted edges (blue in the pdf) in Fig. 1. Note that without applying the meta-inference rules (1) and (2), and by repeatedly updating the occurrence counting information, while applying abstract inference rules, we would obtain that $O^\circ(\Gamma) = \omega$, thus losing the information necessary to determine the alternative presence of membrane Γ in the two target membranes.

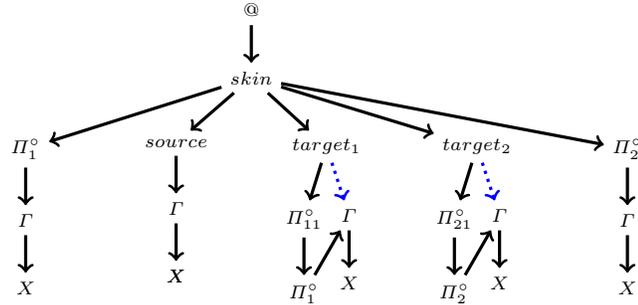


Fig. 1. The Membrane Hierarchy Tree described by R° .

Example 6. The analysis of the system S' described in Tab. 3 of Ex. 2 is given by the abstract state $S'^\circ = (R'^\circ, O'^\circ)$ illustrated in Tab. 10. The analysis result is obtained as in Ex. 5 starting from the corresponding initial configuration. Note that the abstract representation

R'° roughly describes the same information reported in Ex. 5, while the occurrence counting information O'° makes a difference. Given that the multiplicity of membrane Γ is in this case ω , the analysis reveals that Γ may be enclosed at the same time inside membranes $target_1$ and $target_2$ ¹¹.

We can then conclude that our analysis, thanks to the occurrence counting information, allows us to observe that the two biological systems introduced in Sect. 3 exhibit a different dynamical behaviour. In both cases, the analysis predicts that molecule X may end up both in $target_1$ and in $target_2$, but only in the first scenario the two inclusions are alternative.

Note that, in general, information on the possible presence/absence of a component in a membrane could also be exploited when developing a biological model, to detect errors in the model specification.

membrane	children	processes	membrane/process	multipl.
@	skin		@	1
skin	source, target _i , $\Pi_i^\circ, \Pi_{Dij}^\circ$	σ_S	skin,	1
source	Γ	$\overline{\text{bud}}_n^{\lambda_i}(Vesicle_i)$,	source, Γ	ω
target _i	$\Pi_{i1}^\circ, \Pi_{Dij1}^\circ, \Gamma$	$\text{phago}_{n_i}^{\delta_i}(\overline{\text{exo}}_{n_i}^{\beta_i}), \sigma_{target_i}$	γ_1, γ_2	1
Γ	X	$\text{bud}_n^{\gamma_i}, \tau_X$	target ₁ , target ₂ ,	1
$\Pi_i^\circ = (\Gamma, source, \gamma_i, \lambda_i)$	Γ	$\underbrace{\text{phago}_{n_i}^{\mu_i}.\overline{\text{exo}}_{n_i}^{\nu_i}}_{Vesicle_i}$,	Π_1°, Π_2°	1
$\Pi_{Dij}^\circ = (\Gamma, source, \gamma_i, \lambda_j)$	Γ	$\overline{\text{exo}}_{n_i}^{\nu_i}, \tau'_i$	$\Pi_{11}^\circ, \Pi_{21}^\circ$	1
$\Pi_{i1}^\circ = (\Pi_i^\circ, target_i, \delta_i, \mu_i)$	Π_i°	$\overline{\text{exo}}_{n_j}^{\nu_j}, Vesicle_j, \tau'_j$	$\Pi_{D12}^\circ, \Pi_{D21}^\circ$	1
$\Pi_{Dij1}^\circ = (\Pi_{Dij}^\circ, target_j, \delta_j, \mu_j)$	Π_{Dij}°	$\overline{\text{exo}}_{n_i}^{\beta_i}, Vesicle_i, \tau'_i$	$\Pi_{D121}^\circ, \Pi_{D211}^\circ$	1
		$\overline{\text{exo}}_{n_j}^{\beta_j}, Vesicle_j, \tau'_j$	$\mu_1, \mu_2, \nu_1, \nu_2$	1
			β_1, β_2	1
			$\lambda_1, \lambda_2, \delta_1, \delta_2,$	ω

Table 10: The Abstract State $S'^\circ = (R'^\circ, O'^\circ)$, where $i, j = 1, 2$ and $i \neq j$.

6 Conclusions

We presented an analysis based on *Abstract Interpretation* techniques for approximating the behaviour of biological systems described in Brane Calculi [7]. The analysis consists in two components. The first component, which over-approximates the possible membrane hierarchy, is obtained by adapting static analysis techniques used for process algebras handling biological compartments (see e.g. [20,2,3,4,5]). The less standard occurrence counting component of the analysis is used to predict whether some component may occur at most once in any system reachable from the initial one. The two components influence each other. In particular, the prediction on the possible membrane hierarchy is refined with global occurrence counting information, thus allowing us to increase the precision with respect to the previous static approaches for Brane Calculi [3,4,5]. Note that the analyses providing occurrence counting

¹¹ Note that giving the two membrane sources the same label does not influence the result.

information proposed for BioAmbients [14,13,15,25], the sibling bio-inspired calculus, cannot be straightforwardly adapted to Brane Calculi. A careful labelling technique for membranes is required indeed because of bitonality, i.e. the fact that brane interactions possibly introduce new membranes, in order not to mix what is inside a membrane with what is outside (entities can be shuttled inside or outside, only if wrapped by another membrane).

To validate the applicability of our analysis in the biological setting, we applied it to two minimal examples of communication via mobile vesicles that exhibit different dynamical behaviours with respect to the presence of a molecule X inside two target membranes: simultaneous in the second case, but not in the first one. Despite its simplicity, without the occurrence counting, the analysis failed to detect differences between the two cases, thus not predicting that, in the first case, X cannot be simultaneously present in the two target membranes, while in the second case, it may occur in both.

As future work, we would like to improve the accuracy of our approach. In particular, we would like to better analyse the systems that contain different instances of the same membrane or of the same process, and to better handle replication. One possible direction would be to refine our analysis by providing local occurrence counting information in the style of [14,15]. In this way, we could apply our approach to more complex biological case studies, such as the one modelled in [6], for investigating the relationships occurring among events. Note that the formalisation of the LDL Cholesterol Degradation pathway in Brane Calculi presented in [6] requires a version of the calculus with recursive definitions in place of replication. Recursive definitions have been shown useful to provide a more intuitive modelling of real systems with an infinite behaviour.

References

1. Bodei, C.: Control flow analysis for beta-binders with and without static compartments. *Theor. Comput. Sci.* 410(33-34), 3110–3127 (2009)
2. Bodei, C., Brodo, L.: Brane calculi systems: A static preview of their possible behaviour. *CoRR abs/1108.3429* (2011)
3. Bodei, C., Gori, R., Levi, F.: An analysis for causal properties of membrane interactions. *Electr. Notes Theor. Comput. Sci.* 299, 15–31 (2013)
4. Bodei, C., Gori, R., Levi, F.: Causal static analysis for brane calculi. *Theor. Comput. Sci.* 587, 73–103 (2015)
5. Busi, N., Zandron, C.: Modeling and analysis of biological processes by mem(brane) calculi and systems. In: *Proc. of the Winter Simulation Conference WSC 2006*. pp. 1646–1655. WSC (2006)
6. Cardelli, L.: Brane calculi. In: *Proc. of Computational Methods in Systems Biology (CMSB’04)*. *Lecture Notes in Computer Science*, vol. 3082, pp. 257–280. Springer, Springer (2005)
7. Cardelli, L., Gordon, A.: Mobile ambients. *Theor. Comp. Sci.* 240(1), 177–213 (2000)
8. Cousot, P., Cousot, R.: Static determination of dynamic properties of generalized type unions. In: *Language Design for Reliable Software*. pp. 77–94 (1977)
9. Cousot, P. (ed.): *Abstract Interpretation-Based Static Analysis of Mobile Ambients.*, LNCS, vol. 2126. Springer (2001)
10. Feret, J.: Occurrence counting analysis for the pi-calculus. *Electr. Notes Theor. Comput. Sci.* 39(2), 1–18 (2001)
11. Feret, J.: Abstract interpretation of mobile systems. *J. Log. Algebr. Program.* 63(1), 59–130 (2005)
12. Feret, J.: nalysis of mobile systems by abstract interpretation. Ph.D. thesis, *École Polytechnique, Paris, France* (2005)

13. Gori, R., Levi, F.: An analysis for proving temporal properties of biological systems. In: Proc. of Programming Languages and Systems (APLAS'06). LNCS, vol. 4279, pp. 234–252. Springer (2005)
14. Gori, R., Levi, F.: A new occurrence counting analysis for bioambients. In: Proc. of Programming Languages and Systems (APLAS'05). LNCS, vol. 3780, pp. 381–400. Springer (2005)
15. Gori, R., Levi, F.: Abstract interpretation based verification of temporal properties for bioambients. *Inf. Comput.* 208(8), 869–921 (2010)
16. Guerriero, M., Prandi, D., Priami, C., Quaglia, P.: Process calculi abstractions for biology. In: Algorithmic Bioprocesses Natural Computing Series. pp. 463–486. Springer (2009)
17. Levi, F., Maffei, S.: On abstract interpretation of mobile ambients. *Inf. Comput.* 188(2), 179–240 (2004)
18. Milner, R.: Communicating and mobile systems: the π -calculus. Cambridge University Press, Cambridge (1999)
19. Nielson, F., Hansen, R., Nielson, H.R.: Abstract interpretation of mobile ambients. *Sci. Comput. Program.* 47(2-3), 145–175 (2003)
20. Nielson, F., Nielson, H.R., Priami, C., da Rosa, D.S.: Control flow analysis for bioambients. *ENTCS* 180(3), 65–79 (2007)
21. Nielson, H.R., Nielson, F.: Shape analysis for mobile ambients. In: Proc. of the 27th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL'00). pp. 142–154. ACM Press (2000)
22. Nielson, H.R., Nielson, F., Pilegaard, H.: Spatial analysis of bioambients. In: Proc. of Static Analysis International Symposium (SAS'04). LNCS, vol. 3148, pp. 69–83. Springer (2004)
23. Palsberg, J. (ed.): Confidentiality Analysis of Mobile Systems, LNCS, vol. 1824. Springer (2000)
24. Pilegaard, H., Nielson, H.R., Nielson, F.: Context dependent analysis of bioambients. In: Simulation and Verification of Dynamic Systems. Dagstuhl Seminar Proceedings, vol. 06161 (2006)
25. Pilegaard, H., Nielson, H.R., Nielson, F.: Pathway analysis for bioambients. *J. Log. Algebr. Program.* 77(1-2), 92–130 (2008)
26. Priami, C., Quaglia, P.: Beta binders for biological interactions. In: Proc. of Computational Methods in Systems Biology. LNBI, vol. 3082. Springer (2005)
27. Regev, A., Panina, E., Silverman, W., Cardelli, L., Shapiro, E.: Bioambients: An abstraction for biological compartments. *Theor. Comput. Sci.* 325(1), 141–167 (2004)
28. Vitale, A., Mauri, G.: Communication via mobile vesicles in brane calculi. *ENTCS* 171(2), 187–196 (2007)