A new Occurrence Counting analysis for BioAmbients

Roberta Gori[1] and Francesca Levi[2]

[1] Department of Computer Science, University of Pisa, Italy gori@di.unipi.it
[2] DISI, University of Genova, Italy levifran@disi.unige.it

Abstract. This paper concerns the application of formal methods to biological systems, modelled specifically in BioAmbients [30]. BioAmbients [30] is a variant of the Mobile Ambients (MA)[7] calculus, designed precisely for more faithfully capturing basic biological concepts. We propose a new static analysis for BioAmbients which computes approximate information about the run-time behaviour of a system. The analysis is derived following the abstract interpretation approach and introduces two main novelties with respect to the analyses in literature [25, 24, 26, 27]: (i) it records information about the number of occurrences of objects; (ii) it maintains more detailed information about the possible contents of ambients, at any time. In this way, the analysis gives substantially more precise results and captures both the quantitative and causal aspect which are really important for reasoning on the temporal and spatial structure of biological systems. The interest of the analysis is demonstrated by considering a few simple examples which point out the limitations of the existing analyses for BioAmbients.
Keywords: *Mobile Ambients and BioAmbients calculus, static analysis, abstract interpretation.*

1 Introduction

In the past few years several models, originally developed by computer scientists for describing systems of interacting components, have been successfully used for describing biological systems. This is an exciting and interesting application especially because the simulation and verification tools, designed for these formal models, can be used for understanding the behavior of complex biological systems. Such verification techniques can offer biologists very important instruments to replace expensive experiments in vitro or guide the biologists in their experiments by making predictions on the possible results.

Several models and languages, adequate for modeling different aspects of biological systems, have been proposed. They include traditional specification languages for concurrent and reactive systems [21, 16, 15] and also process calculi, designed for modeling distributed and mobile systems, which can successfully describe both the molecular and biochemical aspect. This approach is pioneered by the application of stochastic π-calculus [31, 29], see for example the modeling of the RTK/MAPK pathway. New process calculi have also been proposed in order to faithfully model biological structures such as compartments and membranes, which play a key role in the organization of biomolecular systems. Among them, BioAmbients [30], Beta-Binders [28], and Brane calculi [6].

BioAmbients (BA) is a variant of Mobile Ambients (MA)[7], a very popular calculus designed to model distributed and mobile processes. The key concept of MA is that of *ambient*. An ambient represents a bounded location where computation happens; ambients are organized into a hierarchy, that can be dynamically modified as a consequence of an ambient movement or dissolution. The concepts of ambient and of ambient movement permit to naturally represent important aspects of molecular systems, such as localization, compartmentalization and hierarchy. With the aim of better capturing basic biological concepts, minor modifications are introduced in BA with respect to MA. Ambients are nameless; the primitive for opening is replaced by a primitive of merge, which realizes the fusion of two ambients; capabilities have corresponding co-capabilities; new primitives for communication and choice are introduced.

For BA both verification and simulation methods have been proposed that are essential for a practical application of the model. A stochastic simulation tool has been implemented by extending that of biochemical stochastic π-calculus [31, 29]; tools for automatic verification include model checking [19, 20] and static analysis [25, 24, 26, 27]. *Static analysis* is a formal technique for computing safe approximations of the system (run-time) semantics, and it has been typically applied in the MA setting for verifying security properties. In our opinion, this technique of approximation is essential for dealing with the intrinsic complexity of biological systems.

The proposed analyses are obtained by naturally adapting to its variant BA existing Control Flow Analyses in Flow Logic style of MA. More in details, the analysis of [25] is derived from [23] and predicts the run-time behavior of processes, by giving information about the evolution of the ambients hierarchy, and about which capabilities may be exercised inside any ambient. The proposals of [24, 27] refine the analysis of [23] by introducing more information about the possible shape of processes and about the context, along the lines of various analyses for π-calculus or MA [5, 12, 17].

These analyses give an over-approximation of the behavior of a process, and as usual guarantee invariant properties showing that certain events *will not happen* in *each* state of the system. In particular, they can be applied to establish whether an ambient *will never* end up inside another one; and similarly whether a capability *will never* be exercised inside a given ambient. This kind of information is crucial, when considering security guarantees, and also in the setting of biological systems; for example, in [27] this is enough for distinguishing a system describing a normal LDL degradation process from one presenting mutations or defects. Nonetheless, we believe that different and more detailed kinds of information would be very useful for biologists in order to argue about the spatial and temporal evolution of biological systems.

First, we observe that *quantitative information* plays an essential role in modeling and observing biological systems, as demonstrated by the following example.

Example 1. The system described below (inspired from the porin example in [30]) models the movement of molecules across membrane-bound compartments, specifically a cell. The cell and the molecules are described by ambients, labeled *cell* and *mol*, respectively; their local processes, e.g. P and M, describe their possible interactions.

$$SYS ::= [M]^{mol} \mid \ldots \mid [M]^{mol} \mid [P]^{cell}$$

$$M ::= \mathtt{rec}X.\, \mathtt{in}\, m.\, \mathtt{out}\, n.\, X$$

$$P ::= \mathtt{rec}Y.\, (\overline{\mathtt{in}}\, m.\, Y + \overline{\mathtt{out}}\, n.\, Y) \quad (1)$$

$$P ::= \mathtt{rec}Y.\, \overline{\mathtt{in}}\, m.\, \overline{\mathtt{out}}\, n.\, Y \quad (2)$$

Process M models the ability of molecules of entering and exiting from the membrane, any number of times. The complementary process P gives the permission to ambients *mol* to enter and exit from the *cell*, and thus regulates the crossing of the membrane. Specifically, when process (1) is running inside *cell*, then, at any time, a molecule can exit from or enter inside *cell*. Therefore, when several molecules are present, as described in process SYS, the cell may contain *any number* of molecules. By contrast, when process (2) is running inside *cell*, then, no other molecule can enter, after one has entered inside the *cell*. Thus, no matter how many molecules are present, *just one* molecule can reside inside the cell, at the time.

It is clear that processes (1) and (2) produce a substantially different behaviour for ambient *cell*. Unfortunately, the existing analyses for BA [25, 24, 26, 27] can not capture this relevant difference; in fact, in both cases, they report that ambients *mol may reside inside ambients cell* without giving any information about the possible number of occurrences. Even occurrence counting analyses of MA [17, 14] would be too coarse to model this difference, because they are designed for approximating the number of ambients which occur in the *whole* system. □

Another limitation of the existing analyses [25, 24, 26, 27] is that they do not maintain sufficiently precise information about the possible contents of ambients, at any computation step. This has serious consequences on their ability to capture *causality aspects* which are essential for understanding the temporal and spatial structure of biological systems, such as pathways and networks of proteins. This point is illustrated by the following example.

Example 2. The system describes a simplified version of a bi-substrate enzymatic reaction, modeled as the movement of molecular ambients [30], where two molecules mol_1 and mol_1 interact with an *enzyme*.

$$SYS' = [M_1]^{mol_1} \mid \ldots \mid [M_1]^{mol_1} \mid [M_2]^{mol_2} \mid \ldots \mid [M_2]^{mol_2} \mid [E]^{e} \mid \ldots \mid [E]^{e}$$
$$M_1 ::= \mathtt{rec}Y.\, \mathtt{in}\, m_1.\, (\mathtt{out}\, n_1.\, P_1 + \mathtt{out}\, q_1.\, Y)$$
$$M_2 ::= \mathtt{rec}Y.\, \mathtt{in}\, m_2.\, (\mathtt{out}\, n_2.\, P_2 + \mathtt{out}\, q_2.\, Y)$$
$$E ::= \mathtt{rec}Y.\, \overline{\mathtt{in}}\, m_1.\, \overline{\mathtt{in}}\, m_2.\, (\overline{\mathtt{out}}\, n_2.\, \overline{\mathtt{out}}\, n_1.\, Y + \overline{\mathtt{out}}\, q_2.\, \overline{\mathtt{out}}\, q_1.\, Y)$$

The enzyme and its substrates are modeled by ambients, labeled e, mol_1 and mol_2, respectively. Processes M_1 and M_2 describe the movements of ambients mol_1 and mol_2, respectively; process E describes how the molecules bind to the enzyme and how their products are released. Specifically, the enzyme-substrate binding is modeled as entry of the substrate ambient inside the enzyme ambient, and it follows a precise order (mol_1 and then mol_2). When both molecules are inside the enzyme there are two possible evolutions: both molecules either exit unbind or exit and release their products P_1 and P_2 (these steps follow the inverse order). This process can iterate forever.

This enzymatic reaction has a crucial feature. Not only the binding of both substrates is necessary for the release of their products, but also it has to follow a precise order. This can be formalized by the following property: for each state the binding of mol_1 with e (shown by the presence of mol_1 inside e) is *necessary* for the binding of mol_2 with e (shown by the presence of mol_2 inside e). Such a property cannot be proved with the existing analyses for BA [25, 24, 26, 27], which give too coarse information about the possible run-time nesting of ambients. In fact, they report that both ambients mol_1 and mol_2 may reside inside ambient e without saying whether the presence of one molecule depends on that of the other.

Moreover, a typical way to test whether both substrates mol_1 and mol_2 are *necessary* for the release of the products is to simulate an experiment where either mol_1 or mol_2 are removed. Unfortunately, using the analyses of [25, 24, 26, 27] it is not possible to observe a change in the release of the products.

□

Based on these motivations we propose a new analysis for BA following the *Abstract Interpretation* [9, 10] approach to program analysis, more specifically in the style of pre-

vious proposals for MA [14, 13, 17]. The analysis refines the existing analyses of BA [25, 24, 26, 27] by introducing two (strictly related) novelties: (i) quantitative information is modeled by recording information about the number of occurrences of ambients and processes which may appear in any location; (ii) more detailed information about the possible contents of ambients, at any time, is obtained by pushing forward the idea of continuations proposed in [17]. In this way, we obtain a more informative analysis which can be successfully applied to prove the properties of Examples 1 and 2.

This gain in precision is obviously paid in terms of complexity (in the worst case, the analysis is exponential); by contrast, the existing analyses [25, 24, 26, 27] are associated with polynomial time algorithms. A great advantage of the abstract interpretation theory is that it offers the possibility to systematically define further approximations (e.g. new weaker analyses) by means of *widening operators* [11]. We show that this approach can be profitably applied also to our analysis by introducing a simple parametric widening which turns out to be polynomial in the size of a chosen partition of abstract labels. We then apply the widening to Example 2 for showing that it still gives better results w.r.t. the existing analyses [25, 24, 26, 27].

The paper is organized as follows. Section 2 introduces the syntax and the semantics of the BioAmbients calculus. In Section 3 we presents our analysis and in Section 4 the corresponding widening operator.

2 Syntax and Semantics

For lack of space, we consider here a simplified version of BioAmbients [30] without communication primitives; the analysis can be extended in a simple way to the full calculus.

In Control Flow Analysis (see for instance [23, 12]) typically processes are labeled and α-conversion is treated in a particular way, based on a given partition of labels and names. This modification supports simpler specifications of abstractions. We therefore consider the following sets of names and labels. Let $\mathcal{N}$ (ranged over by $n, m, h, k, \ldots$) be the set of *names* such that $\mathcal{N} = \uplus_{i=1}^{\omega} \mathcal{N}_i$, where $\uplus$ denotes disjoint union and each $\mathcal{N}_i$ is an infinite set. Similarly, let $\mathcal{L}$ (ranged over by $\lambda, \mu, \ldots$) be the set of *labels* such that $\mathcal{L} = \uplus_{i=1}^{\omega} \mathcal{L}_i \cup \{\top\}$, where each $\mathcal{L}_i$ is an infinite set and $\top$ is a distinct symbol used to model the outermost ambient. Moreover, we consider *composite labels* (in the following referred to as labels) $\widehat{\mathcal{L}} = \wp(\mathcal{L}) \setminus \emptyset$. We adopt meta-variables $\Psi, \Gamma, \Delta, \ldots$ to range over $\widehat{\mathcal{L}}$ and we use for simplicity λ for the singleton $\{\lambda\}$. We also consider a set of recursion variables $\mathcal{V}$ (ranged over by $X, Y, Z, \ldots$).

The syntax of (labelled) *processes* is defined in Table 1. The constructs for inactivity, parallel composition, restriction are standard (see for instance π-calculus [22]). The inactive process is denoted by 0; parallel composition is denoted by $P \mid Q$; the restriction operator, denoted by $(\nu n)\, P$, creates a new name n with scope P. Operator $\mathtt{rec}X^{\lambda}.P$ defines a recursive process (for convenience we adopt recursion in place of standard replication $!P$). Specific to the ambient calculi, are the ambient construct, $[P]^{\Psi}$, and the capability prefix $M^{\lambda}.P$, where M is an action or co-action[1]. Specifically, process $[P]^{\Psi}$ defines an ambient (labelled) Ψ where process P runs. Finally, process $\Sigma_{i \in I}^{\lambda} M_i.P_i$ defines a capability choice primitive with the obvious meaning.

[1] Notice that we adopt a notation for coactions in the style of Safe Ambients [18] in place of the standard one.

M,N::=		(*capabilities*)
	$\texttt{in}\, n$	enter
	$\overline{\texttt{in}}\, n$	co-enter
	$\texttt{out}\, n$	exit
	$\overline{\texttt{out}}\, n$	co-exit
	$\texttt{merge}\, n$	merge
	$\overline{\texttt{merge}}\, n$	co-merge

P,Q::=		(*processes*)
	0	inactivity
	$(\nu n)\, P$	restriction
	$P \mid Q$	parallel composition
	X	recursion variable
	$\texttt{rec}X^{\lambda}.\, P$	recursive process
	$[P]^{\Psi}$	ambient
	$M^{\lambda}.\, P$	capability prefix
	$\Sigma^{\lambda}_{i\in I} M_i.\, P_i$	capability choice

Table 1. BioAmbients Processes.

For processes we adopt standard syntactical conventions. We often omit the trailing 0 in processes, and we assume that parallel composition has the least syntactic precedence. The operator $(\nu n)P$ acts as static binder for name n, and thus produces the standard notion of free and bound names of a process; similarly, $\texttt{rec}X.\, P$ is a binder for X with scope P. A process is *closed on recursion variables* if it has no free recursion variables. In the following, we assume that processes are closed on recursion variables. Moreover, since processes are labelled, with $\Lambda(P)$ we indicate the set of labels of a process P.

As usual, we identify processes which are α-convertible, that is that can be made syntactically equals by a change of bound names. In typical Control Flow Analysis style [12, 17], we however discipline α-conversion by assuming that a bound name m can be replaced only with a name n provided that $n, m \in \mathcal{N}_i$. Similarly, we also identify *re-labeled* processes, i.e. processes that can be made syntactically equals by changing labels, requiring that a label λ can be replaced with a label μ, provided that $\lambda, \mu \in \mathcal{L}_i$.

The semantics of BA is given in the form of a standard reduction relation; the rules are reported in Table 2. In order to compact several rules together we introduce a special notation for capability prefix and capability choice. We write $+M^{\lambda}.\, P$ to denote both process $M^{\lambda}.\, P$ and process $\Sigma^{\lambda}_{i\in I} M_i.\, Q_i$, where $M = M_i$ and $P = Q_i$ for some $i \in I$.

The reduction axioms (In), (Out) and (Merge) define the basic interactions; they model the movement of an ambient, in or out, of another ambient and the merge of two ambients. They differ from those of MA mainly because ambients are nameless (labels are attached to processes as comments and do not influence the interaction). Moreover, the primitive *merge* replaces the standard primitive of opening. Notice that, when two ambients labelled Ψ and Δ are merged, the new ambient is labelled $\Psi \cup \Delta$ showing that it is the result of their fusion. Another difference with MA, common instead with its variant Safe Ambients [18], is that we prefer to view the unfolding of recursion as a reduction rule, e.g. (Rec), rather than as a step of structural congruence.

The inference rules (Res), (Par), (Amb) and (Cong) are standard; they handle reductions in contexts and permit to apply structural congruence. Structural congruence is needed to bring the participants of a potential interaction into contiguous positions; it includes standard rules for commuting the positions of components appearing in parallel and in a choice, and rules for stretching the scope of a restrictions. For lack of space, we omit the presentation of structural congruence (e.g. relation $\equiv$) and we refer to [30]. In the following, we say that a process P is *active* if either $P = \Sigma^{\lambda}_{i\in I} M_i.\, Q_i$, $P = M^{\lambda}.\, Q$ or $P = \texttt{rec}X^{\lambda}.\, P$. Moreover, we use $\mathcal{P}$ and $\mathcal{AP}$ to denote the set of processes and the subset of active processes, respectively.

$[+\mathtt{in}\, m^{\lambda}.\, P \mid Q]^{\Psi} \mid [+\overline{\mathtt{in}}\, m^{\mu}.\, R \mid S]^{\Delta} \to [[P \mid Q]^{\Psi} \mid R \mid S]^{\Delta}$	(In)
$[[+\mathtt{out}\, m^{\lambda}.\, P \mid Q]^{\Psi} \mid +\overline{\mathtt{out}}\, m^{\mu} R \mid S]^{\Delta} \to [P \mid Q]^{\Psi} \mid [R \mid S]^{\Delta}$	(Out)
$[+\mathtt{merge}\, m^{\lambda}.\, P \mid Q]^{\Psi} \mid [+\overline{\mathtt{merge}}\, m^{\mu}.\, R \mid S]^{\Delta} \to [P \mid Q \mid R \mid S]^{\Psi \cup \Delta}$	(Merge)
$\mathtt{rec} X^{\lambda}.\, P \to P[\mathtt{rec} X^{\lambda}.\, P/X]$	(Rec)
$P \to Q \Rightarrow (\nu n)\, P \to (\nu n)\, Q$	(Res)
$P \to Q \Rightarrow P \mid R \to Q \mid R$	(Par)
$P \to Q \Rightarrow [P]^{\Psi} \to [Q]^{\Psi}$	(Amb)
$(P' \to Q',\ P \equiv P',\ Q' \equiv Q) \Rightarrow P \to Q$	(Cong)

Table 2. Reduction Rules of BioAmbients

The collecting semantics. The collecting semantics is defined as the least fixed-point of a function, which collects all the states (namely processes) reachable from the initial process. The *concrete domain* is therefore $\mathcal{A} = (\wp(\mathcal{P}), \subseteq)$.

Definition 1 (Collecting Semantics). *Let $P \in \mathcal{P}$ be a process such that $\top \notin \Lambda(P)$. We define $\mathfrak{S}_{Coll}[\![P]\!] = lfp\ F(P)$ for the function $F : \mathcal{P} \to (\mathcal{A} \to \mathcal{A})$ such that $F(P) = \Psi_P$ and, for $Ss \in \wp(\mathcal{P})$,*

$$\Psi_P(Ss) = \{P\} \bigcup_{\{P_2 \mid P_1 \to P_2,\ P_1 \in Ss\}} \{P_2\}.$$

3 The Abstraction

Our analysis is designed to prove properties that are true in all the states reachable from the initial state. To this aim, it computes an over-approximation of the following information about any reachable state: for each ambient, which ambients may be contained and which capabilities may be exercised inside, and their number of occurrences. Following the abstract interpretation approach of [17] we define the analysis by giving the abstract states (the abstract processes) and the abstract transitions (the abstract reduction steps among processes). To formally prove the correctness of the analysis, we introduce a corresponding abstract domain, equipped with an ordering expressing precision of approximations, and we formalize its relation with the concrete one through a Galois connection [9, 10].

The abstraction is parametric with respect to the choice of *abstract names* and *labels*, defined by a partition of names and labels. For these purposes, we first consider an abstract partition of labels $\mathcal{L}$, given by $\mathcal{L}^{\circ} = \uplus_i \mathcal{L}_i^{\circ} \cup \{\top\}$, where $i \in \{1, \ldots, h\}$ for some h, $\mathcal{L}_i^{\circ}$ is a (possible) infinite set of labels, $\uplus_i \mathcal{L}_i^{\circ} \cup \{\top\} = \mathcal{L}$ and $\mathcal{L}^{\circ}$ is congruent with $\mathcal{L}$, i.e., $\lambda, \mu \in \mathcal{L}_i$ implies that $\lambda, \mu \in \mathcal{L}_j^{\circ}$ for some j. We consider, then, *abstract labels*

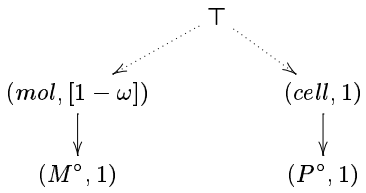


Fig. 1. State S° graphically

$\widehat{\mathcal{L}}^\circ = \wp(\mathcal{L}^\circ_{/\cong}) \setminus \emptyset$ (ranged over by $\Psi^\circ, \Gamma^\circ, \Delta^\circ, \ldots$), where $\cong$ is the obvious equivalence induced by the partition. For names we proceed in a similar way by considering an abstract partition of names $\mathcal{N}^\circ = \uplus_i \mathcal{N}^\circ_i$, where $i \in \{1, \ldots, h\}$ for some h, $\uplus_i \mathcal{N}^\circ_i = \mathcal{N}$, such that $\mathcal{N}^\circ$ is congruent with $\mathcal{N}$, i.e., $n, m \in \mathcal{N}_i$ implies that $n, m \in \mathcal{N}^\circ_i$. We therefore consider *abstract names* $\widehat{\mathcal{N}}^\circ = \wp(\mathcal{N}^\circ_{/\cong} *) \setminus \emptyset$ (ranged over by $A^\circ, B^\circ, C^\circ, \ldots$). For convenience, we assume that Δ° stands for the abstract label (namely its equivalence class) corresponding to label Δ; similarly for abstract names.

The abstract partitions of names and labels naturally induce a corresponding notion of abstract processes; built following the syntax of Table 1 by using names $\widehat{\mathcal{N}}^\circ$ and labels $\widehat{\mathcal{L}}^\circ$. As usual, $\mathcal{P}^\circ$ and $\mathcal{AP}^\circ$ stands for the set of abstract and active abstract processes, respectively. Similarly, P° stands for the abstract process corresponding to P (this is obtained in the obvious way).

Abstract domain and Galois connection. Abstract states are the key concept behind the abstraction and are designed precisely to represent approximate information about "concrete" states (e.g. processes) according to the following intuitive ideas. An *abstract state* reports: (i) the abstract labels of the ambients that may appear; and (ii) for each of them, one or more *configurations* describing the possible contents of the ambients with that label. More in details, a configuration contains the *abstract labels* of the ambients and the *active abstract processes*, which may appear at top- level, and their number of occurrences. For representing occurrence counting information, we adopt the following set $\mathcal{M} = \{0, 1, [0-\omega], [1-\omega]\}$. Each $\mathtt{m} \in \mathcal{M}$ denotes a *multiplicity*, with the following meaning: 0 and 1 indicate zero and exactly one respectively, the interval $[1-\omega]$ at least one while the interval $[0-\omega]$ indicate 0 or more.

Example 3. Consider the system (already described in Example 1),

$$\begin{array}{ll} SYS ::= & [M]^{mol} \mid \ldots \mid [M]^{mol} \mid [P]^{cell} \\ M ::= & \mathtt{rec}X.\, \mathtt{in}\, m.\, \mathtt{out}\, n.\, X \\ P ::= & \mathtt{rec}Y.\, \overline{\mathtt{in}}\, m.\, \overline{\mathtt{out}}\, n.\, Y \end{array}$$

Moreover, assume that abstract names and labels are defined by the following equivalence classes $\{\{n, m\}\}$ (ranged over by m) and $\{\{mol\}, \{cell\}\}$, respectively. With respect to this partition of labels and names, the *best approximation* of SYS is given by the following abstract state (graphically represented also in Figure 1)

$$S^\circ = \{(\top, C^\circ_0), (mol, C^\circ_1), (cell, C^\circ_2)\} \quad C^\circ_0 = \{(mol, [1-\omega]), (cell, 1)\} \quad C^\circ_1 = \{(M^\circ, 1)\}$$
$$C^\circ_2 = \{(P^\circ, 1)\} \quad M^\circ ::= \mathtt{rec}X.\, \mathtt{in}\, m.\, \mathtt{out}\, m.\, X \qquad P^\circ ::= \mathtt{rec}Y.\, \overline{\mathtt{in}}\, m.\, \overline{\mathtt{out}}\, m.\, Y$$

Configuration C°_0 reports information about the possible internal process of ambient $\top$ (a special symbol representing the outermost ambient). More in details, pair $(cell, 1)$ says

that *exactly one* ambient *cell* may appear at top-level, while pair $(mol, [1-\omega])$ says that *at least one* ambient *mol* may appear at top-level. Ambients *cell* and *mol* may appear in parallel inside ambient $\top$; this is shown by a dotted line that connects these ambients with their father in Figure 1. Configurations C_1° and C_2° describe the possible internal processes of ambients *mol* and *cell*, respectively. In C_1° pair $(M^\circ, 1)$ says that, inside *any* ambient *mol*, *exactly* one process abstracted by M° may be running. In this sense, the counting of occurrences is local, being $[1-\omega]$ the global number of occurrences of processes M°. Similarly, in C_2° pair $(P^\circ, 1)$ says that, inside *any* (in this case one) ambient *cell*, *exactly* one process abstracted by P° may be running.

Consider then a minor modification of SYS, where more than one ambient *cell* may appear, $SYS_1 = [M]^{mol} \mid \ldots \mid [M]^{mol} \mid [P]^{cell} \mid \ldots \mid [P]^{cell}$. Now the best approximation is

$$S_1^\circ = \{(\top, \{(mol, [1-\omega]), (cell, [1-\omega])\}), (mol, C_1^\circ), (cell, C_2^\circ)\}.$$

The only difference between S° and S_1° concerns the multiplicity of ambients *cell*, which is now $[1-\omega]$. It is clear that state S_1° is also a correct approximation for process SYS; it is however less precise than S°, which predicts *exactly* one occurrence of ambients *cell* at top-level.

It is worth noticing that in abstract states S° and S_1° exactly one configuration describes the possible internal processes of each abstract label (and thus of the related ambients). It may be convenient however to adopt several different configurations, as illustrated by the following system,

$$SYS_2 ::= [M]^{mol} \mid \ldots \mid [M]^{mol} \mid [\overline{\mathtt{out}}\, m.\, P \mid [\mathtt{out}\, m.\, M]^{mol}]^{cell}$$

This process is a derivative of SYS and describes the situation where: one ambient *mol* has moved inside ambient *cell* and is ready to exit; the remaining ambients *mol* are still in the initial situation. Process SYS_2 could be approximated by the following abstract state,

$$S_2^\circ = \{(\top, C_0^\circ), (mol, C_4^\circ), (cell, C_5^\circ)\}$$
$$C_5^\circ = \{(\overline{\mathtt{out}}\, m.\, P^\circ, 1), (mol, 1)\} \quad C_4^\circ = \{(M^\circ, [0-\omega]), (\mathtt{out}\, m.\, M^\circ, [0-\omega])\}$$

Configuration C_5° describes the possible contents of ambients *cell* and shows that: exactly one ambient *mol* and exactly one process abstracted by $\overline{\mathtt{out}}\, m.\, P^\circ$ may appear; these processes may be running in parallel inside an ambient *cell*. Configuration C_4° reports the information about the possible contents of ambients *mol*; it describes both those at top-level (that contain the recursive process) and the one, residing inside ambient *cell* (where process $\mathtt{out}\, m.\, M^\circ$ is running). The configuration says that, inside any ambient *mol*, zero or more processes abstracted by M° and $\mathtt{out}\, m.\, M^\circ$ may appear (in particular they may be running in parallel). Notice that, since all the ambients *mol* are identified, the multiplicity, for each process, is $[0-\omega]$ showing that it may be the case that the process does not appear.

The information about ambients *mol* in state S_2° is rather approximate. Better results can be obtained by adopting distinct configurations to describe the different instances of ambients *mol*, as in the following abstract state,

$$S_3^\circ = \{(\top, C_0^\circ), (cell, C_5^\circ), (mol, C_6^\circ), (mol, C_7^\circ)\} \quad C_6^\circ = \{(M^\circ, 1)\} \quad C_7^\circ = \{(\mathtt{out}\, m.\, M^\circ, 1)\}$$

In this case, ambients *mol* are described by two configurations, C_6° and C_7° . Their interpretation is that any ambient *mol*, contains *either* exactly one process abstracted by M° or exactly one process abstracted by $\mathtt{out}\, m.\, M^\circ$. In this way, more precise information about the multiplicity of processes M° and $\mathtt{out}\, m.\, M^\circ$ is achieved; also, it is possible to argue that the two processes cannot run in parallel inside the same instance of ambient *mol*. This state is graphically represented in Figure 2 where these processes are connected with their enclosing ambient *mol* by a plain line precisely for showing that they cannot be in parallel.

□

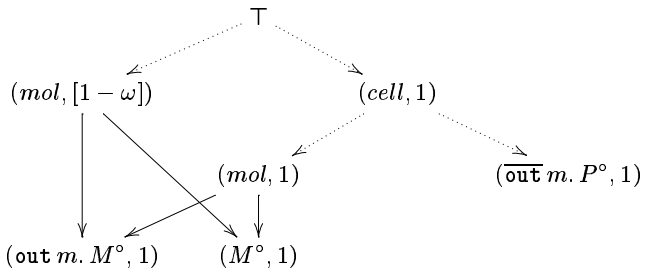
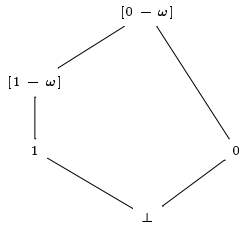


Fig. 2. State S_3° graphically



$+^\circ$	0	1	$[1-\omega]$	$[0-\omega]$
0	0	1	$[1-\omega]$	$[0-\omega]$
1	1	$[1-\omega]$	$[1-\omega]$	$[1-\omega]$
$[1-\omega]$	$[1-\omega]$	$[1-\omega]$	$[1-\omega]$	$[1-\omega]$
$[0-\omega]$	$[0-\omega]$	$[1-\omega]$	$[1-\omega]$	$[0-\omega]$

$-^\circ$	1
0	0
1	0
$[1-\omega]$	$[0-\omega]$
$[0-\omega]$	$[0-\omega]$

Table 3. Occurrence Counting.

We introduce the formal definitions. In the following, we use $\widehat{\mathcal{PL}} = \widehat{\mathcal{L}}^\circ \cup \mathcal{AP}^\circ$ to denote the set of abstract labels and abstract active processes; also, we use e to denote a generic element of $\widehat{\mathcal{PL}}$. Moreover, we use $(e, \mathtt{m})$ to denote a generic element of $\mathcal{E} = (\widehat{\mathcal{L}}^\circ \times \mathcal{M}) \cup (\mathcal{AP}^\circ \times \mathcal{M})$.

Definition 2 (Abstract States). *An* abstract state S° *is a set of pairs* (Ψ°, C°) *where* $C^\circ \in \wp(\mathcal{E})$ *is a* configuration*, such that: (i) if* $(e, \mathtt{m}), (e, \mathtt{m}') \in C^\circ$*, then* $\mathtt{m} = \mathtt{m}'$*; and (ii) for each* $(e, \mathtt{m}) \in C^\circ$*,* $\mathtt{m} \neq 0$*.*

Notice that in configurations, no pair $(e, 0)$ can appear, recording explicitly that there are no occurrences of element e. However, in the following with an abuse of notation we may write $(e, 0) \in C^\circ$ in place of $(e, \mathtt{m}) \notin C^\circ$ for any $\mathtt{m} \in \mathcal{M}$. This notation simplifies the definition of some operators over configurations and states. In the following, $\mathcal{S}^\circ$ and $\mathcal{C}^\circ$ stand for the set of abstract states and of configurations, respectively.

Following the intuitive ideas explained in Example 3 we introduce two information orders on configurations and abstract states which formalize precision of approximations. To this aim, we assume that the domain $\mathcal{M}$ of multiplicity comes equipped with the obvious (information) order $\leq_m$ and with the set of operations $+^\circ$ and $-^\circ$, reported in Table 3.

Definition 3 (Ordering).

- *We say that* $C_1^\circ \leq^c C_2^\circ$ *iff, for each* $(e, \mathtt{m}) \in C_1^\circ$ *there exists* $(e, \mathtt{m}') \in C_2^\circ$ *such that* $\mathtt{m} \leq_m \mathtt{m}'$*;*
- *We say that* $S_1^\circ \leq^s S_2^\circ$ *iff, for each* $(\Psi^\circ, C_1^\circ) \in S_1^\circ$*, there exists* $(\Psi^\circ, C_2^\circ) \in S_2^\circ$ *such that* $C_1^\circ \leq^c C_2^\circ$*.*

$\leq^s$ is a pre-order. We consider the order $\subseteq^\circ$ induced by the pre-order $\leq^s$, namely the order obtained considering classes of abstract states modulo the equivalence induced by $\leq^s$. For a sake of simplicity in the rest of the paper the domain $\mathcal{S}^\circ_{/\cong^s}$ and the equivalence class $[S^\circ]_{\cong^s}$ will be simply indicated by $\mathcal{S}^\circ$ and S° respectively.

Given the ordering over abstract states, it is immediate to define the *abstract domain*, $\mathcal{A}^\circ = (\mathcal{S}^\circ, \subseteq^\circ)$. Notice that the concrete domain records sets of states (e.g. processes); while in the abstract domain only one abstract states collects all the information.

The relation between the concrete and the abstract domain is formalized by establishing a Galois connection. To this aim, we first introduce a function that, given a process reports its *best approximation*, that is the best abstract state which has enough information about the process. This is derived along the lines of the intuitive ideas explained in Example 3. More in details, given a process P, we proceed as follows

1. we take its abstract version P° where labels $\widehat{\mathcal{L}}$ and names $\widehat{\mathcal{N}}$ are replaced with their abstract versions $\widehat{\mathcal{L}}^\circ$ and $\widehat{\mathcal{N}}^\circ$, respectively;
2. we produce a representation of P° in terms of set of configurations where explicit information about the nesting of ambients and processes and about their quantities is properly introduced.

Formally, we define $\alpha^{aux} : \mathcal{P} \to \mathcal{S}^\circ$ as $\alpha^{aux}(P) = \delta^\circ(\top, P^\circ)$ where $\delta^\circ : (\widehat{\mathcal{L}}^\circ \times \mathcal{P}^\circ) \to \mathcal{S}^\circ$ is an *auxiliary translation function*, giving an abstract state representing the abstract process with respect to the label of the enclosing ambient (in this case $\top$).

The translation function $\delta^\circ : (\widehat{\mathcal{L}}^\circ \times \mathcal{P}^\circ) \to \mathcal{S}^\circ$ exploits an additional function $\eta^\circ : \mathcal{P}^\circ \to (\mathcal{C}^\circ \times \mathcal{S}^\circ)$, which intuitively gives: (i) an abstract configuration reporting the processes and ambients occurring at top level; (ii) an abstract state representing the internal ambients. Having in mind this interpretation we define

$$\delta^\circ(\Psi^\circ, P^\circ) = \{(\Psi^\circ, C^\circ)\} \cup S^\circ \quad \text{where} \quad \eta^\circ(P^\circ) = (C^\circ, S^\circ).$$

Function η° is reported in Table 4 and uses an operator $\cup^+$ between configurations, which simply realizes the union of two configurations by summing the multiplicity in the obvious way. Given $C_1^\circ, C_2^\circ \in \wp(\mathcal{E})$, we define

$$C_1^\circ \cup^+ C_2^\circ = \{(e, \mathtt{m}) \mid (e, \mathtt{m}_i) \in C_i^\circ, \text{ for each } i \in \{1,2\}, \ \mathtt{m} = \mathtt{m}_1 +^\circ \mathtt{m}_2\}.$$

As an example, it is not difficult to check that for the system of Example 3, we have $\delta^\circ(\top, SYS^\circ) = S^\circ$, where S° is the state of Figure 1.

Based on function α^{aux} it is immediate to derive the following abstraction and concretization functions between sets of processes and abstract states. This permits precisely to formalize when an abstract state is a *safe over-approximation* of a set of concrete states. We use $\cup^\circ$ between abstract states to denote the l.u.b. with respect to $\subseteq^\circ$; it realizes indeed the union of configurations.

DRes$^\circ$	$\eta^\circ((\nu M^\circ)P^\circ)$	$= \eta^\circ(P^\circ)$
DAmb$^\circ$	$\eta^\circ([P^\circ]^{\Delta^\circ})$	$= (\{(\Delta^\circ, 1)\}, \delta^\circ(\Delta^\circ, P^\circ))$
DZero$^\circ$	$\eta^\circ(0)$	$= (\emptyset,\ \emptyset)$
DPar$^\circ$	$\eta^\circ(P_1 \mid P_2)$	$= (C_1{}^\circ \cup^+ C_2{}^\circ, S_1{}^\circ \cup^\circ S_2{}^\circ)$ $\eta^\circ(P_i) = (C_i^\circ, S_i^\circ)$ for $i \in \{1,2\}$
DRec$^\circ$	$\eta^\circ(\mathtt{rec}X^{\Psi^\circ}.P^\circ)$	$= (\{(\mathtt{rec}X^{\Psi^\circ}.P^\circ, 1)\}, \emptyset)$
DPref$^\circ$	$\eta^\circ(M^{\Psi^\circ}.P^\circ)$	$= (\{(M^{\Psi^\circ}.P^\circ, 1)\}, \emptyset)$
DSum$^\circ$	$\eta^\circ(\Sigma_{i\in I}^{\Psi^\circ} M_i^\circ.P_i^\circ)$	$= (\{(\Sigma_{i\in I}^{\Psi^\circ} M_i^\circ.P_i^\circ, 1)\}, \emptyset)$

Table 4. Abstract Translation Function.

Definition 4. *Let $Ss \in \wp(\mathcal{P})$ and $S^\circ \in \mathcal{S}^\circ$. We define $\alpha^\circ : \wp(\mathcal{P}) \to \mathcal{S}^\circ$ and $\gamma^\circ : \mathcal{S}^\circ \to \wp(\mathcal{P})$ as follows,*

$$\alpha^\circ(Ss) = \bigcup^\circ_{P\in Ss} \alpha^{aux}(P) \qquad \gamma^\circ(S^\circ) = \bigcup_{\{P \mid \alpha^{aux}(P) \subseteq^\circ S^\circ\}} P$$

Theorem 1. *The pair of functions $(\alpha^\circ, \gamma^\circ)$ of Def. 4 is a Galois connection between $\langle \mathcal{A}, \subseteq\rangle$ and $\langle \mathcal{A}^\circ, \subseteq^\circ\rangle$.*

Abstract semantics. We introduce the *abstract transitions*. They use the following operators which realize the removal from a configuration of one occurrence of an object e and similarly of a set of objects. For $PL^\circ \subseteq \widehat{\mathcal{PL}}$ we have

$$C^\circ \backslash^\circ e = C^\circ \setminus \{(e, \mathtt{m})\} \cup \{(e, \mathtt{m}-^\circ 1)\} \qquad C^\circ \backslash^\circ PL^\circ = C^\circ \backslash^\circ_{e\in PL^\circ} e.$$

The *abstract transitions* are defined by the rules of Table 5; they realize the unfolding of recursion, the movements of ambients, in and out, and the merge of two ambients (reflecting rules (Rec), (In), (Out) and (Merge) of Table 2). Due to the implicit representation of parallel composition, ambient and restriction in abstract states there are no abstract transitions corresponding to the structural rules and to structural congruence of the reduction semantics.

Rule **Rec**$^\circ$ models the unfolding of recursion and is applicable to a recursive process $T^\circ = \mathtt{rec}X^{\Delta^\circ}.P^\circ$ which is running inside an ambient labeled Γ° (this means that there exists a configuration C° for Γ° that contains process T°). The resulting abstract state is obtained by adding a configuration representing the ambient labeled Γ° where replication has been unfolded. More specifically, the configuration is $(C^\circ \backslash^\circ T^\circ) \cup^+ \delta^\circ(\Gamma^\circ, P^\circ[T^\circ/X])$ where the translation of the unfolded process is added and the recursive process T° is removed (according to their multiplicities).

The rules **In**$^\circ$, **Out**$^\circ$, **Merge**$^\circ$ are similar; as an example we comment **In**$^\circ$. The rule models the movement of an ambient labeled Ψ° inside an ambient labeled Δ°. It is applicable whenever: (i) they are possible siblings meaning that they may be enclosed, at the same time, inside an ambient (labeled Γ°); (ii) they offer the right action or coaction. Formally, it must be the case that: (i) there exists a configuration C° for Γ° which contains *both* Ψ° and Δ°; (ii) there exist configurations $C_1{}^\circ$ and $C_2{}^\circ$ for Ψ° and Δ°, where capabilities $\mathtt{in}\, M^\circ$ and $\overline{\mathtt{in}}\, M^\circ$ are ready to fire, respectively. If Ψ° and Δ° happen to be the same label, then the movement is possible only if their multiplicities are greater than 1.

The resulting abstract state is obtained as follows. Abstract configurations are added for modeling the ambients labeled Ψ° and Γ° which have participated to the movement:

1. $(C_1^\circ \backslash^\circ T^\circ) \cup^+ \delta^\circ(\Psi^\circ, P^\circ)$ describes the local process of ambient Ψ° and is obtained by removing the executed process T° and by adding its continuation (according to their multiplicities);
2. $(C_2^\circ \backslash^\circ T'^\circ) \cup^+ \delta^\circ(\Delta^\circ, Q^\circ) \cup^+ (\Psi^\circ, 1)$ describes the local process of ambient Δ° and is obtained similarly as in the previous case. The only relevant difference is that one occurrence of ambient Ψ° is added for modeling the movement.

Similarly, abstract configuration $C^\circ \backslash^\circ \Psi^\circ$ is added for the ambient labeled Γ°, taking into account precisely that one ambient labeled Ψ° has moved somewhere-else.

Rec°
$$\frac{(\Gamma^\circ, C^\circ) \in S^\circ \qquad (T^\circ, \mathtt{m}) \in C^\circ \qquad T^\circ = \mathtt{rec}\, X^{\Delta^\circ}.\, P^\circ}{S^\circ \mapsto^\circ S^\circ \cup^\circ \{(\Gamma^\circ, (C^\circ \backslash^\circ T^\circ) \cup^+ \delta^\circ(\Gamma^\circ, P^\circ[T^\circ/X]))\}}$$

In°
$$\frac{\begin{array}{l}(\Psi^\circ, C_1^\circ) \in S^\circ \qquad (T^\circ, \mathtt{m}_1) \in C_1^\circ \qquad T^\circ = +\mathtt{in}\, M^{\circ\Theta^\circ}.\, P^\circ \\ (\Delta^\circ, C_2^\circ) \in S^\circ \qquad (T'^\circ, \mathtt{m}_2) \in C_2^\circ \qquad T'^\circ = +\overline{\mathtt{in}}\, M^{\circ\Phi^\circ}.\, Q^\circ \\ (\Gamma^\circ, C^\circ) \in S^\circ \qquad (\Psi^\circ, \mathtt{m}_3) \in C^\circ \qquad (\Delta^\circ, \mathtt{m}_4) \in C^\circ \\ \Delta^\circ = \Psi^\circ \to \mathtt{m}_3 = \mathtt{m}_4 >_m 1\end{array}}{\begin{array}{l}S^\circ \mapsto^\circ S^\circ \cup^\circ \{(\Psi^\circ, (C_1^\circ \backslash^\circ T^\circ) \cup^+ \delta^\circ(\Psi^\circ, P^\circ))\} \cup^\circ \\ \quad \{(\Delta^\circ, (C_2^\circ \backslash^\circ T'^\circ) \cup^+ \delta^\circ(\Delta^\circ, Q^\circ) \cup^+ (\Psi^\circ, 1))\} \cup^\circ \{(\Gamma^\circ, C^\circ \backslash^\circ \Psi^\circ)\}\end{array}}$$

Out°
$$\frac{\begin{array}{l}(\Psi^\circ, C_1^\circ) \in S^\circ \qquad (T^\circ, \mathtt{m}_1) \in C_1^\circ \qquad T^\circ = +\mathtt{out}\, M^{\circ\Theta^\circ}.\, P^\circ \\ (\Delta^\circ, C_2^\circ) \in S^\circ \qquad (T'^\circ, \mathtt{m}_2) \in C_2^\circ \qquad T'^\circ = +\overline{\mathtt{out}}\, M^{\circ\Phi^\circ}.\, Q^\circ \qquad (\Psi^\circ, \mathtt{m}_3) \in C_2{}^\circ \\ (\Gamma^\circ, C^\circ) \in S^\circ \qquad (\Delta^\circ, \mathtt{m}_4) \in C^\circ\end{array}}{\begin{array}{l}S^\circ \mapsto^\circ S^\circ \cup^\circ \{(\Psi^\circ, (C_1^\circ \backslash^\circ T^\circ) \cup^+ \delta^\circ(\Psi^\circ, P^\circ))\} \cup^\circ \\ \quad \{(\Delta^\circ, (C_2^\circ \backslash^\circ \{T'^\circ, \Psi^\circ\}) \cup^+ \delta^\circ(\Delta^\circ, Q^\circ))\} \cup^\circ \{(\Gamma^\circ, C^\circ \cup^+ (\Psi^\circ, 1))\}\end{array}}$$

Merge°
$$\frac{\begin{array}{l}(\Psi^\circ, C_1^\circ) \in S^\circ \qquad (T^\circ, \mathtt{m}_1) \in C_1^\circ \qquad T^\circ = +\mathtt{merge}\, M^{\circ\Theta^\circ}.\, P^\circ \\ (\Delta^\circ, C_2^\circ) \in S^\circ \qquad (T'^\circ, \mathtt{m}_2) \in C_2^\circ \qquad T'^\circ = +\overline{\mathtt{merge}}\, M^{\circ\Phi^\circ}.\, Q^\circ \\ (\Gamma^\circ, C^\circ) \in S^\circ \qquad (\Psi^\circ, \mathtt{m}_3) \in C^\circ \qquad (\Delta^\circ, \mathtt{m}_4) \in C^\circ \\ \Delta^\circ = \Psi^\circ \to \mathtt{m}_3 = \mathtt{m}_4 >_m 1\end{array}}{\begin{array}{l}S^\circ \mapsto^\circ S^\circ \cup^\circ \{(\Psi^\circ \cup \Delta^\circ, (C_2^\circ \backslash^\circ T'^\circ) \cup^+ (C_1^\circ \backslash^\circ T^\circ) \cup^+ \delta^\circ(\Psi^\circ, P^\circ) \cup^+ \delta^\circ(\Delta^\circ, Q^\circ))\} \cup^\circ \\ \quad \{(\Gamma^\circ, (C^\circ \backslash^\circ \{\Delta^\circ, \Psi^\circ\}) \cup^+ (\Psi^\circ \cup \Delta^\circ, 1))\}\end{array}}$$

Table 5. Abstract Transitions.

The abstract semantics is then defined as the least fixed-point of a function, which computes starting from the abstraction of the initial process, an abstract state that is the union of all the reachable abstract states.

Definition 5 (The abstract semantics). *Let P be a process such that $\top \notin \Lambda(P)$. We define $\mathfrak{S}_{Coll^\circ}[\![P]\!] = \mathit{lfp}\, F^\circ(\alpha^{aux}(P))$ for the function $F^\circ : \mathcal{S}^\circ \to (\mathcal{A}^\circ \to \mathcal{A}^\circ)$ such*

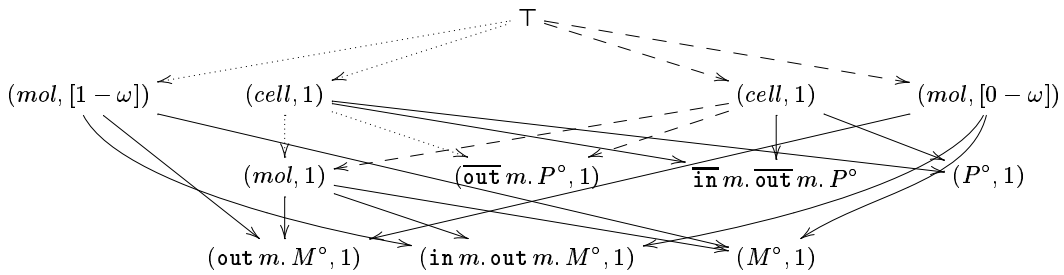


Fig. 3. The analysis of SYS.

that $F^\circ(S^\circ) = \Psi^\circ_{S^\circ}$ *and*

$$\Psi^\circ_{S^\circ}(S^\circ_1) = S^\circ \cup^\circ \bigcup^\circ_{\{S_2{}^\circ \mid S_1{}^\circ \mapsto^\circ S_2{}^\circ\}} S_2{}^\circ.$$

The following theorem shows the correctness of the analysis.

Theorem 2 (Safeness). *Let P be a process such that $\top \notin \Lambda(P)$. We have*

$$\alpha^\circ(\mathfrak{S}_{Coll}[\![P]\!]) \subseteq^\circ \mathfrak{S}_{Coll^\circ}[\![P]\!].$$

Example 4. We apply the abstraction to process SYS of Example 3 (and also Example 1). The analysis computes the following abstract state (also represented in Figure 3), namely $\mathfrak{S}_{Coll^\circ}[\![P]\!] = S^\circ_{SYS}$

$$S^\circ_{SYS} = \{(\top, C^\circ_0), (mol, C^\circ_1), (cell, C^\circ_2), (cell, C^\circ_3), (cell, C^\circ_4), (\top, C^\circ_5), (mol, C^\circ_6), (mol, C^\circ_7)\}$$

$$C^\circ_0 = \{(mol, [1-\omega]), (cell, 1)\} \qquad C^\circ_1 = \{(M^\circ, 1)\} \qquad C^\circ_2 = \{(P^\circ, 1)\}$$

$$C^\circ_3 = \{(\overline{\mathtt{in}}\, m.\, \overline{\mathtt{out}}\, m.\, P^\circ, 1)\} \quad C^\circ_4 = \{(\overline{\mathtt{out}}\, m.\, P^\circ, 1), (mol, 1)\}$$

$$C^\circ_5 = \{(mol, [0-\omega]), (cell, 1)\} \quad C^\circ_6 = \{(\mathtt{in}\, m.\, \mathtt{out}\, m.\, M^\circ, 1)\} \quad C^\circ_7 = \{(\mathtt{out}\, m.\, M^\circ, 1)\}$$

State S°_{SYS} reports approximate information about all the derivatives of SYS, including obviously the configurations C°_0, C°_1 and C°_2 describing the initial state of Figure 1. The other configurations are added by performing the abstract transitions. Configurations C°_3 and C°_6 model the unfolding of the recursive processes inside ambients *cell* and *mol*, respectively; they are added by $\mathbf{Rec}^\circ$ steps. In both cases the recursive process, P° or M°, is deleted from the configuration precisely because it has multiplicity one (and thus is it has been consumed). Configurations C°_4, C°_5 and C°_7 are introduced by the execution of rule $\mathbf{In}^\circ$, modelling the movement of one ambient *mol* inside ambient *cell*. Configuration C°_4 describes the situation where ambient *cell* contains exactly one ambient *mol* and in parallel process $\overline{\mathtt{out}}\, m.\, P^\circ$; configuration C°_5 shows that any number (including zero) of ambients *mol* may appear at top level, because one has moved somewhereelse; configuration C°_7 shows that exactly one process $\mathtt{out}\, m.\, M^\circ$ is running inside *mol*. In configurations C°_4 and C°_7, processes $\overline{\mathtt{in}}\, m.\, \overline{\mathtt{out}}\, m.\, P^\circ$ and $\mathtt{in}\, m.\, \mathtt{out}\, m.\, M^\circ$, respectively, are deleted because of their multiplicity as explained above. No other configurations are needed to describe the execution of $\mathbf{Out}^\circ$ modeling the movement of ambient *mol* out from ambient *cell*.

This analysis is able to capture the relevant feature of SYS and therefore to establish the property discussed in Example 1: *just one* molecule can reside inside the cell, at the time. In fact, in any configuration of *cell* either there are no occurrences of ambients *mol* or there is exactly one occurrence.

It is worth stressing that different configurations are needed to better approximate the possible contents of ambients, in particular of ambient *cell* (e.g. configurations C_2°, C_3° and C_4°). This permits to have a very precise information about its possible contents at any computation step, and consequently to substantially restrict the space of possible interactions. More in details, the analysis captures that, when one ambient *mol* resides inside *cell*, it must be the case that the process running in parallel is $\overline{\mathtt{out}}\, m.\, P^\circ$; and thus no other ambient is authorized to enter (capability $\overline{\mathtt{in}}\, m$ indeed has been consumed). By using weaker analyses where all the configurations of a given ambient are merged (as in the widening of Section 4 or in analyses in literature [25, 24, 26, 27, 17]), it would not possible to derive this information. Moreover, as we have already pointed out, the occurrence counting analysis of [14] cannot prove such a property since it would count the number of ambients *mol* appearing in the whole system.

We conclude by observing that the analysis of the minor modification of SYS, where several ambients *cell* may appear, establishes this property in a similar way (see SYS_1 in Example 3).

□

We conclude by briefly discussing the complexity of the analysis. As a measure of complexity we can take the maximal number of iterations of the fixed-point operator in the worst case. This is the maximal number of different configurations we can have in an abstract state, and therefore they are exponential in the size of the abstract process.

4 A widening operator

We present here a *widening operator* [11] which formalizes a natural and simple way for further approximating the analysis of Section 3. Widening operators are introduced precisely to speed up convergency of a fix-point computation. In this setting, a simple widening can be defined by a function $\omega : \mathcal{A} \to \mathcal{A}$, reporting an approximation of an abstract state; this means that $\omega(S_1^\circ) = S_2^\circ$ implies $S_1^\circ \subseteq^\circ S_2^\circ$. This function is intended to be applied at each iteration of the fixed-point computation of function $\Psi_{S^\circ}^\circ$ which defines the analysis (see Def. 5).

The widening operator defined below is based on the simple idea to have *at most one* configuration describing the possible content of any abstract label. This means that all the configurations describing a given label are put together. Moreover, in order to be able to completely tune the complexity related to the size of the set of abstract labels, we make the widening operator parametric w.r.t. a partition of $\widehat{\mathcal{L}}^\circ$, the set of abstract labels. Let $\tilde{\mathcal{L}}^\circ = \widehat{\mathcal{L}}^\circ_{/\cong}$, where $\cong$ is the equivalence induced by the chosen partition.

To formalize the widening operator, we use $\cup^c$ to denote the l.u.b. of two configurations (according to the ordering of Def. 3); also, we extend $\cup^c$ to abstract states for performing the merge of all configurations related to equivalent abstract labels.

Definition 6 (Widening). *Consider $\tilde{\mathcal{L}}^\circ$ a partition of $\widehat{\mathcal{L}}^\circ$. We define the widening operator $\omega : \mathcal{A} \to \mathcal{A}$ as*

$$\omega(S^\circ) = \cup^c S^\circ =$$
$$\{(\Delta^\circ, \cup^c_{j\in\{1,\ldots,k\}} C_j) \mid (\Gamma^\circ, C_s) \in S^\circ, \Delta^\circ \cong \Gamma^\circ \Rightarrow \exists l \in \{1,\ldots,k\},\ (\Delta^\circ, C_l) = (\Gamma^\circ, C_s)\}$$

Using the widening operator ω we define a new fixed-point operator $\Psi^{\omega}_{S^\circ} : \mathcal{A} \to \mathcal{A}$, where the widening is applied at each iteration,

$$\Psi^{\omega}_{S^\circ}(S^\circ_1) = \omega(S^\circ_2) \text{ if } \Psi^{\circ}_{S^\circ}(S^\circ_1) = S^\circ_2.$$

In the following, $\mathfrak{S}_{Coll^\omega}[\![P]\!]$ stands for the result of the fixed-point computation of $\Psi^{\omega}_{S^\circ}$. In this way we obtain a new analysis which is *polynomial* w.r.t the cardinality of $\tilde{\mathcal{L}}^\circ$, a completely tunable parameter.

The following example shows that this new analysis still gives interesting results.

Example 5. Consider the system presented in Example 2,

$$\begin{array}{l} SYS' ::= [M_1]^{mol_1} \mid \ldots \mid [M_1]^{mol_1} \mid [M_2]^{mol_2} \mid \ldots \mid [M_2]^{mol_2} \mid [E]^e \mid \ldots \mid [E]^e \\ M_1 ::= \mathtt{rec}Y.\, \mathtt{in}\, m_1.\, (\mathtt{out}\, n_1.\, P_1 + \mathtt{out}\, q_1.\, Y) \\ M_2 ::= \mathtt{rec}Y.\, \mathtt{in}\, m_2.\, (\mathtt{out}\, n_2.\, P_2 + \mathtt{out}\, q_2.\, Y) \\ E ::= \mathtt{rec}Y.\, \overline{\mathtt{in}}\, m_1.\, \overline{\mathtt{in}}\, m_2.\, (\overline{\mathtt{out}}\, n_2.\, \overline{\mathtt{out}}\, n_1.\, Y + \overline{\mathtt{out}}\, q_2.\, \overline{\mathtt{out}}\, q_1.\, Y) \end{array}$$

Consider the following partition of names $\{\{m_1, q_1, n_1\}, \{m_2, q_2, n_2\}\}$ (ranged over by abstract names s and r respectively) and of labels $\{\{\{mol_1\}\}, \{\{mol_2\}\}, \{\{e\}\}, \{s \in \widehat{\mathcal{L}}^\circ \mid cardinality(s) > 1\}\}$ designed for distinguishing mol_1, mol_2 and e. If we apply the analysis of Section 3 we have $\mathfrak{S}_{Coll^\circ}[\![SYS']\!] = S^\circ$ where [2]

$$\begin{array}{l} S^\circ = \{\ (\top, \{(mol_1, [1-\omega]), (e, [1-\omega]), (mol_2, [1-\omega])\}), \\ \quad (\top, \{(mol_1, [0-\omega]), (e, [1-\omega]), (mol_2, [1-\omega])\}), \\ \quad (\top, \{(mol_1, [0-\omega]), (e, [1-\omega]), (mol_2, [0-\omega])\}), \\ \quad (mol_1, \{(M_1^\circ, 1)\}), (mol_1, \{(\mathtt{in}\, s.\, R_1, 1)\}), (mol_1, \{(R_1, 1)\}), (mol_1, \{(P_1, 1)\}), \\ \quad (mol_2, \{(M_2^\circ, 1)\}), (mol_2, \{(\mathtt{in}\, r.\, R_2, 1)\}), (mol_2, \{(R_2, 1)\}), (mol_2, \{(P_2, 1)\}) \\ \quad (e, \{(E^\circ, 1)\}), (e, \{((\overline{\mathtt{in}}\, s.\, \overline{\mathtt{in}}\, r.\, R_3, 1)\}), (e, \{((\overline{\mathtt{in}}\, r.\, R_3, 1), (mol_1, 1)\}), \\ \quad (e, \{(R_3, 1), (mol_1, 1), (mol_2, 1)\}), (e, \{((\overline{\mathtt{out}}\, s.\, E^\circ), 1), (mol_1, 1)\})\} \\ \\ \quad R_1 = (\mathtt{out}\, s.\, P_1 + \mathtt{out}\, s.\, {M_1}^\circ), R_2 = (\mathtt{out}\, r.\, P_2 + \mathtt{out}\, r.\, {M_2}^\circ) \\ \quad R_3 = (\overline{\mathtt{out}}\, r.\, \overline{\mathtt{out}}\, s.\, E^\circ + \overline{\mathtt{out}}\, r.\, \overline{\mathtt{out}}\, s.\, E^\circ). \end{array}$$

The analysis establishes that the binding of mol_1 and mol_2 follows a precise order; indeed, there is no configuration for ambient e showing the presence of mol_2 without that of mol_1. The use of different configurations is essential for this very precise prediction, similarly as explained in Example 4.The widening of Definition 6 as well as the existing analysis [25, 24, 26, 27] do not give sufficiently precise information for proving this property.

However, when compared to the existing analyses [25, 24, 26, 27] the widening gives more precise predictions due to the different treatment of continuations in the style of [17]. This causal aspect is very useful in this setting; for instance, it is adequate for analyzing the system SYS' where one of the two molecules has been removed (with the aim of proving that both are necessary). As an example we consider the case where no molecule mol_2 is present, as modelled by the system $SYS'' ::= [M_1]^{mol_1} \mid \ldots \mid [M_1]^{mol_1} \mid [E]^e \mid \ldots \mid [E]^e$ (the symmetric case is analogous). We have $\mathfrak{S}_{Coll^\omega}[\![SYS'']\!] = S^\circ_1$ where

$$\begin{array}{l} S^\circ_1 = \{\ (\top, \{(mol_1, [0-\omega]), (e, [1-\omega])\}), \\ \quad (mol_1, \{(M_1^\circ, [0-\omega]), (\mathtt{in}\, s.\, R_1, [0-\omega]), (R_1, [0-\omega])\}), \\ \quad (e, \{(E^\circ, [0-\omega]), (\overline{\mathtt{in}}\, s.\, \overline{\mathtt{in}}\, r.\, R_3, [0-\omega]), (\overline{\mathtt{in}}\, r.\, R_3, [0-\omega]), (mol_1, [0-\omega])\})\} \end{array}$$

The widening shows that none of products P_1 and P_2 is released (in particular it reports that process P_1 cannot run at top-level inside ambient mol_1 as instead happens in the abstract state S° result of the analysis of system SYS'). This property cannot be established by applying

[2] For simplicity we are assuming that $P_i = 0$ for each $i \in \{1, 2\}$.

the existing analyses [25, 24, 26, 27], precisely because these proposals have a far less precise prediction about the local processes of ambients. Specifically, they do not capture that the continuation of capability $\overline{\texttt{in}}\, r$, inside ambient e, cannot be exercised; and consequently that capability $\overline{\texttt{out}}\, s$ cannot be consumed by ambient mol_1 to exit from ambient e. □

5 Conclusions and Related Works

Our analysis introduces many novelties w.r.t. the analyses presented in the literature [25, 24, 26, 27]. In particular: (i) it gives very precise information on occurrence counting (which is local in contrast to standard global information [17, 14]); (ii) it permits to obtain more detailed information about the processes and ambients which may reside inside an ambient, at any time. This is obtained by adopting different configurations to describe ambients in different stages of evolution and by adapting the treatment of continuations of [17]. As a consequence, the analysis better captures also causality aspects. Causality issues have been considered in a few type systems [1–3] for MA or for its variant Safe Ambients [18]. The types of [1, 3] describe the possible contents of ambients by means of a sort of traces and probably could give interesting results when applied to biological systems. They however lack occurrence counting information.

Our analysis is rather expensive from a computational point of view w.r.t. the proposals of [25, 24, 26, 27]. In our opinion, this additional complexity is motivated by the need of capturing *quantitative* and *causality* information which are fundamental in the biological systems setting. Examples 1 and 2 (then commented in Sections 3 and 4) demonstrate the relevance of this information and show the limitations of the existing analyses. Moreover, it is worth noting that also the occurrence counting analysis of [14] for MA has an exponential complexity even if it does not report sufficiently precise information for proving the properties of Examples 1 and 2.

Moreover, our approach offers several possibilities for tuning the precision, and therefore to find out the right balance between precision and computational cost. The abstraction is parametric, in the sense that one can choose *which part of the system he is interested in* by defining equivalence classes of ambients labels and names. Further approximations can easily be derived by following the widening approach of abstract interpretation. The polynomial widening of Section 4 is an interesting example, especially because it gives better results with respect to the existing analyses [25, 24, 26, 27]. This is discussed in Section 4 by considering Example 2.

There are several interesting directions for future works. First of all, we intend to implement an abstract interpretation framework for computing analyses of BA. Furthermore, we would like to design new analyses which to take into account the stochastic and temporal aspects. In particular, in this setting, it seems very important to be able to establish *temporal properties* much more general than invariant properties; in particular to observe the possible evolution paths of a biological system. This is motivated by the variety of recent works concerning temporal logics and model checking for biological systems [19, 20, 8, 4].

Acknowledgements. We would like to thank the anonymous referees for their useful comments.

References

1. T. Amtoft. *Causal Type System for Ambient Movements.* Submitted for publication, 2003.
2. T. Amtoft, A. J. Kfoury and S. M. Pericas-Geertsen. *What are Polymorphically-Typed Ambients?* Proceedings of ESOP'01, LNCS 2028, 206-220, Springer Verlag, 2001.
3. T. Amtoft, H. Makholm and J.B. Wells. *PolyA: True Type Polymorphism for Mobile Ambients.* Proc. of TCS'04, 591–604, Kluwer, 2004.
4. R. Barbuti, S. Cataudella, A. Maggiolo-Schettini, P. Milazzo and A.Troina, *A Probabilistic Calculus for Molecular Systems* Proc. of Workshop CS & P, Informatik Berichte 170, Humboldt University press, vol 202–216, 2004.
5. C. Bodei, P. Degano, C. Priami and N. Zannone. *An enhanced cfa for security policies.* Proc. of WITS'03, 2003.
6. L. Cardelli. *Membrane Interactions.* Proc. of BioCONCUR '03, Electronic Notes in Computer Science, 2003.
7. L. Cardelli and A.D. Gordon. *Mobile ambients.* Theoretical Computer Science 240, 177–213, 2000.
8. N. Chabrier and F. Fages. *Symbolic model-checking of biochemical networks.* In Proceedings of the First International Workshop on Computational Methods in Systems Biology, 149-162, 2003.
9. P. Cousot and R. Cousot. Abstract Interpretation: A Unified Lattice Model for Static Analysis of Programs by Construction or Approximation of Fixpoints. In *Proc. Fourth ACM Symp. Principles of Programming Languages*, pages 238–252, 1977.
10. P. Cousot and R. Cousot. Systematic Design of Program Analysis Frameworks. In *Proc. Sixth ACM Symp. Principles of Programming Languages*, pages 269–282, 1979.
11. P. Cousot and R. Cousot. Comparing the Galois Connection and Widening/Narrowing Approaches to Abstract Interpretation. In M. Bruynooghe and M. Wirsing, editors, *Proc. of PLILP'92*, volume 631 of *Lecture Notes in Computer Science*, pages 269–295. Springer-Verlag, 1992.
12. P. Degano, F. Levi and C. Bodei. *Safe Ambients: Control Flow Analysis and Security.* Proc. of ASIAN '00, LNCS 1961, 199-214, Springer Verlag, 2000.
13. J. Feret. *Abstract Interpretation-Based Static Analysis of Mobile Ambients.* Proc. of SAS'01, LNCS 2126, 412-430, Springer Verlag, 2001.
14. R. R. Hansen and J. G. Jensen and F. Nielson and H. R.Nielson. *Abstract Interpretation of Mobile Ambients.* Proc. of SAS'99, LNCS 1694, 135-148, Springer-Verlag, 1999.
15. N. Kam, D. Harel, H. Kugler, R. Marelly, A. Pnueli, E.J.A. Hubbard and M.J. Stern. *Formal Modeling of C. elegans Development: A Scenario-Based Approach.* Proc. 1st Int. Workshop on Computational Methods in Systems Biology (ICMSB 2003), LNCS 2602, Springer-Verlag, 4-20, 2003.
16. R. Hofestadt and S.Thelen. *Quantitative modeling of biochemical networks* . Silico Biology, volume1, 39-53, 1998.
17. F. Levi and S. Maffeis. *On Abstract Interpretation of Mobile Ambients.* Information and Computation 188, 179–240, 2004.
18. F. Levi and D. Sangiorgi. *Mobile Safe Ambients.* TOPLAS, 25(1), 1–69. ACM Press, 2003.
19. R. Mardare and C. Priami. *Logical Analysis of Biological Systems.* Fundamenta Informaticae, 64, 271–285, 2005.
20. R. Mardare, O. Vagin, P. Quaglia and C. Priami. *Model Checking Biological Systems described using Ambient Calculus.* Proc. of CMSB'04, LNCS 3082, 85–103, 2004.
21. H.Matsuno, A.Doi, M.Nagasaki and S.Miyano. *Hybrid petri net representation of gene regulatory network.* Pacific Symposium on Biocomputing (5), 338-349, 2000.
22. R. Milner, J. Parrow and D. Walker. *A Calculus of Mobile Processes.* Information and Computation, 100, 1-77, 1992.

23. F. Nielson, H.R. Nielson, R.R. Hansen. *Validating firewalls using flow logics.* Theoretical Computer Science, 283(2), 381-418, 2002.
24. F. Nielson, H.R. Nielson and H. Pilegaard. *Spatial Analysis of BioAmbients.* Proc. of SAS'04, LNCS 3148, pp. 69–83, Springer-Verlag, 2004.
25. F. Nielson, H.R. Nielson, C. Priami and D. Schuch da Rosa. *Control Flow Analysis for BioAmbients.* BioCONCUR'03, Electronic Notes in Computer Science, 2003.
26. F. Nielson, H.R. Nielson, C. Priami and D. Schuch da Rosa. *Static Analysis for Systems Biology.* Proc. of the winter International Symposium on Information and Communication Technologies, 1–6, Trinity College Dublin, 2004.
27. H. Pilegaard, F. Nielson and H.R. Nielson. *Static Analysis of a Model of the LDL Degradation Pathway.* Proc. of CMSB'05, to appear.
28. C.Priami and P. Quaglia. *Beta binders for biological interactions.* Proceedings of Computational Methods in System Biology, CMSB'04, LNBI, to appear.
29. C. Priami, A. Regev, W. Silverman and E. Shapiro. Application of a stochastic name-passing calculus to representation and simulation of molecular processes. Information Processing Letters, 80 (1), 25–31, 2001.
30. A. Regev, E. M. Panina, W. Silverman, L. Cardelli and E. Shapiro. BioAmbients: an Abstraction for Biological Compartments. Theoretical Computer Science, 325, 141–167, 2004.
31. A. Regev, W. Silverman and E. Shapiro. Representation and Simulation of Biochemical Processes using the pi-calculus process algebra. Proc. of the Pacific Symposium on Biocomputing 2001, 6, 459–470, 2001.