Abstract Interpretation of Mobile Ambients

Flemming Nielson, René Rydhof Hansen and
Hanne Riis Nielson

Department of Computer Science, Aarhus University

Abstract

We show how abstract interpretation can be expressed in a constraint-based formalism that is becoming increasingly popular for the analysis of functional and object-oriented languages. This is illustrated by developing analyses for the ambient calculus.

The first step of the development constructs an analysis for counting occurrences of processes inside other processes; we show that the analysis is semantically correct and that the set of acceptable solutions constitutes a Moore family. The second step considers a previously developed control flow analysis and shows how to induce it from the counting analysis; we show that its properties can be derived from those of the counting analysis using general results about abstract interpretation for constraint-based analyses.

1 Introduction

The ambient calculus was introduced in [3] as a natural language for modeling network security (e.g. firewalls) and mobility (e.g. autonomous agents). By employing static analysis it is possible to *automatically* validate certain security properties of such models (e.g. whether or not a firewall is protective cf. [9]) and thereby attaining a higher level of trust in the security of such models.

Abstract interpretation [5] is a powerful technique for performing static analysis by stepwise development. One starts with an overly precise and costly analysis and then develops more approximate and less costly analyses by carefully choosing appropriate Galois connections; in this way the semantic correctness of the initial analysis carries over to the approximate analyses. This technique has demonstrated its ability to deal successfully with a variety of programming languages. Recent papers have studied how to apply the technique to calculi of computation such as the π-calculus although it was assumed that processes were written in a "non-standard" syntax [13,14].

We show that abstract interpretation can be developed for the ambient calculus (that essentially extends a large subset of the π-calculus with operations for mobility) without the need to assume that processes are on a non-standard form. More importantly, we are able to perform the entire development by expressing the analyses in a constraint-based formulation that closely corresponds to formulations that have become popular for the analysis of functional and object-oriented languages and that have recently been applied also to the π-calculus [1,2]. This is likely to make abstract interpretation more accessible to a wider community because often abstract interpretation is being criticised for starting with a "low level" trace-based semantics. We refer to Section 2 for a review of the ambient calculus.

The first step is the development of an analysis for explicitly modelling which processes can be inside what other processes; in order to model accurately what happens when the only process inside some other process actually moves out of the process, the analysis incorporates a counting component. As is customary for constraint-based formulations (in particular our own based on flow logic) this takes the form of specifying a satisfaction relation $\mathcal{C} \models P$ for when the set $\mathcal{C}$ of descriptions of processes satisfies the demands of the program P; here $\mathcal{C}$ is a set of tuples that each describe a set of processes. If $\mathcal{C} \models P$ then $\mathcal{C}$ is said to be an *acceptable* analysis for P.

This approach is very natural for applications such as security and validation where information obtained by other means needs to be checked before it can be used – much the same ideas are found in type systems. We then show that the specification is semantically sound (meaning that, for a suitable extraction function η, if $\mathcal{C} \models P$ and $\eta(P) \in \mathcal{C}$ then $\mathcal{C}$ contains descriptions $\eta(Q)$ of all processes Q that P can evaluate to) and that the set of acceptable solutions has a least element (or more precisely that $\{\mathcal{C} \mid \mathcal{C} \models P \wedge \eta(P) \in \mathcal{C}\}$ constitutes a Moore family). The details are covered in Section 3.

The second step is to show that a previously developed control flow analysis [9] can in fact be induced from the counting analysis; to show this we first clarify what it means to induce one constraint-based analysis from another. We then show that semantic correctness and the existence of least solutions carry over from the counting analysis. This shows that abstract interpretation is a useful guide also when developing analyses of calculi of computation. It follows that the theoretical properties established in [9] from first principles actually fall out of the general development. We refer to Section 4 for the details.

A preliminary version of this paper appeared as [7]. More powerful analyses appear in [10,12].

2 The Ambient Calculus

The ambient calculus is a calculus of computation with special focus on mobility. It is based on traditional process algebras, such as the π-calculus, but the emphasis is on mobility and movement of processes rather than communication. In particular it extends the notion of mobility found in Java [6] by allowing *active* processes, as opposed to passive code, to move between administrative domains and sites. The calculus was introduced in [3] and a polyadic variant along with a typesystem was presented in [4].

In this paper we restrict our attention to the communication free fragment of the ambient calculus; as shown in [3] this fragment retains Turing-completeness.

2.1 Syntax

As in [3] the syntax of processes $P, Q \in \mathbf{Proc}$ is given by:

$$
\begin{array}{rcll}
P, Q & ::= & (\nu n)P & \text{restriction} \\
& | & \mathbf{0} & \text{inactivity} \\
& | & P \mid Q & \text{composition} \\
& | & !P & \text{replication} \\
& | & n^{\ell^a}[P] & \text{ambient} \\
& | & in^{\ell^i} n.P & \text{capability to enter } n \\
& | & out^{\ell^o} n.P & \text{capability to exit } n \\
& | & open^{\ell^p} n.P & \text{capability to open } n \\
\\
n & & & \text{names}
\end{array}
$$

The restriction $(\nu n)P$ introduces the new name n and limits its scope to the process P; $\mathbf{0}$ is the inactive process that does nothing; $P \mid Q$ stands for the processes P and Q running in parallel; replication provides recursion and iteration as $!P$ represents an unbounded number of copies of P running in parallel. By $n[P]$ we denote the ambient named n that has the process P running inside it. The capabilities $in n$ and $out n$ are used to move their enclosing ambients whereas $open n$ is used to dissolve the boundary of a sibling ambient; this will be made precise when we define the semantics below. We write $fn(P)$ for the free names of P. Trailing occurrences of the inactive process $\mathbf{0}$ will often be omitted.

To make the presentation of the analysis independent from the syntactic peculiarities of the ambient calculus, we have decided to base the analysis on labels

$$
\begin{array}{ll}
P \equiv P & P \mid Q \equiv Q \mid P \\
P \equiv Q \;\Rightarrow\; Q \equiv P & (P \mid Q) \mid R \equiv P \mid (Q \mid R) \\
P \equiv Q \;\wedge\; Q \equiv R \;\Rightarrow\; P \equiv R & !P \equiv P \mid !P \\
& \\
P \equiv Q \;\Rightarrow\; (\nu n)P \equiv (\nu n)Q & P \equiv Q \;\Rightarrow\; in^{\ell} n.P \equiv in^{\ell} n.Q \\
P \equiv Q \;\Rightarrow\; P \mid R \equiv Q \mid R & P \equiv Q \;\Rightarrow\; out^{\ell} n.P \equiv out^{\ell} n.Q \\
P \equiv Q \;\Rightarrow\; !P \equiv !Q & P \equiv Q \;\Rightarrow\; open^{\ell} n.P \equiv open^{\ell} n.Q \\
P \equiv Q \;\Rightarrow\; n^{\ell}[P] \equiv n^{\ell}[Q] & \\
& (\nu n)(\nu m)P \equiv (\nu m)(\nu n)P \\
P \mid \mathbf{0} \equiv P & (\nu n)(P \mid Q) \equiv P \mid (\nu n)Q \\
(\nu n)\mathbf{0} \equiv \mathbf{0} & \quad \text{if } n \notin \mathit{fn}(P) \\
!\mathbf{0} \equiv \mathbf{0} & (\nu n)(m^{\ell}[P]) \equiv m^{\ell}[(\nu n)P] \\
& \quad \text{if } n \neq m
\end{array}
$$

Fig. 1. Structural congruence.

that uniquely determine the program points of interest; consult [9,10,12] for alternative ways to present similar analyses. We write $\ell^a \in \mathsf{Lab}^a$ for labels on ambients, $\ell^i \in \mathsf{Lab}^i$ for labels on *in*-capabilities, $\ell^o \in \mathsf{Lab}^o$ for labels on *out*-capabilities and $\ell^p \in \mathsf{Lab}^p$ for labels on *open*-capabilities; it is helpful to assume that the sets Lab^a, Lab^i, Lab^o and Lab^p are pairwise disjoint. We use $\ell \in \mathsf{Lab} = \mathsf{Lab}^a \cup \mathsf{Lab}^i \cup \mathsf{Lab}^o \cup \mathsf{Lab}^p$ to range over their union so as to be able to dispense with the superscripts when no confusion can arise.

The syntax of names $n \in \mathsf{Nam}$ is left implicit but it is assumed that the set of names is countably infinite.

In the ambient calculus processes are identified up to α-renaming of their bound names (see below). This means that bound names can change dynamically during evaluation and thus are unsuitable for direct use in the analysis, because semantic correctness will be epxressed as a subject reduction result (see Sections 3.2 and 4.2). To overcome this problem we therefore introduce the notion of *stable names*: assume that the set of names is equipped with an equivalence relation $\equiv_\alpha$ such that each equivalence class $\{m \mid m \equiv_\alpha n\}$ is countably infinite and contains a unique representative $\lfloor n \rfloor$. We write SNam for the subset $\{\lfloor n \rfloor \mid n \in \mathsf{Nam}\}$ of representatives, called the *stable names*, and in examples we take $\lfloor n_i \rfloor = n$ etc. so that $n_i \equiv_\alpha m_j$ if and only if n equals m. We shall then identify processes up to a renaming of their bound names with names from the *same* equivalence classes. In [1,2,7] an alternative treatment of α-equivalence is explored: annotating bound names with labels, called "markers", and then using markers instead of stable names.

$$\overline{n^{\ell}[in^{\ell^i} m.P \mid Q] \mid m^{\ell'}[R] \rightarrow m^{\ell'}[n^{\ell}[P \mid Q] \mid R]}$$

$$\overline{m^{\ell'}[n^{\ell}[out^{\ell^o} m.P \mid Q] \mid R] \rightarrow n^{\ell}[P \mid Q] \mid m^{\ell'}[R]}$$

$$\overline{open^{\ell^p} n.P \mid n^{\ell}[Q] \rightarrow P \mid Q}$$

$$\frac{P \rightarrow Q}{n^{\ell}[P] \rightarrow n^{\ell}[Q]} \qquad \frac{P \rightarrow Q}{(\nu n)P \rightarrow (\nu n)Q}$$

$$\frac{P \rightarrow Q}{P \mid R \rightarrow Q \mid R} \qquad \frac{P \equiv P' \qquad P' \rightarrow Q' \qquad Q' \equiv Q}{P \rightarrow Q}$$

Fig. 2. Reduction relation.

2.2 Semantics

The semantics is given by a structural congruence $P \equiv Q$ and a reduction relation $P \rightarrow Q$ in the manner of the π-calculus. The congruence relation of Figure 1 is a straightforward modification of the corresponding table in [3].

The reduction relation is given in Figure 2 and is a straightforward modification of the corresponding table in [3]. A pictorial representation of the three basic rules is given in Figure 3.

Example 1 *The following example shows how a firewall modeled in the ambient calculus works. For brevity it is a scaled down version of the firewall example that appears in [3] in that only two keys are used as opposed to three. Note however that this firewall is still* protective *in the sense of [9].*

Consider the ambient w (representing the firewall) that contains a probe p:

$$w^{\mathsf{w}}[p^{\mathsf{p}}[out^{\ell_1^o} w.in^{\ell_2^i} k.in^{\ell_4^i} w] \mid open^{\ell_5^p} k.P]$$

The ambient can use the probe to fetch ambients with name k that are willing to be fetched; as an example we have:

$$k^{\mathsf{k}}[open^{\ell_3^p} p \mid Q]$$

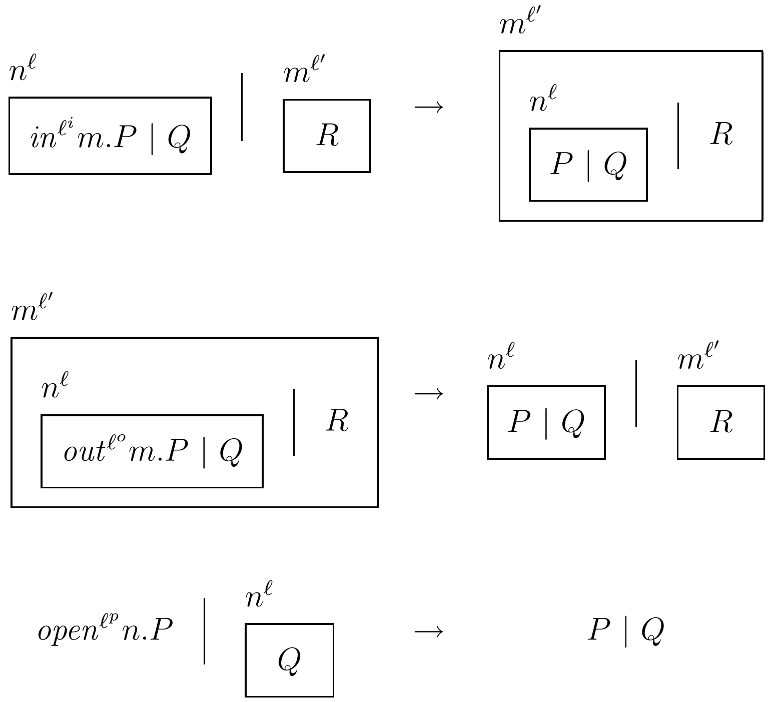

Fig. 3. Pictorial representation of the basic reduction rules.

We illustrate the fact that w can use p to fetch k by the reduction sequence:

$$
\begin{aligned}
& w^{\mathsf{w}}[p^{\mathsf{p}}[out^{\ell_1^o} w.in^{\ell_2^i} k.in^{\ell_4^i} w] \mid open^{\ell_5^p} k.P] \mid k^{\mathsf{k}}[open^{\ell_3^p} p \mid Q] \\
& \rightarrow w^{\mathsf{w}}[open^{\ell_5^p} k.P] \mid p^{\mathsf{p}}[in^{\ell_2^i} k.in^{\ell_4^i} w] \mid k^{\mathsf{k}}[open^{\ell_3^p} p \mid Q] \\
& \rightarrow w^{\mathsf{w}}[open^{\ell_5^p} k.P] \mid k^{\mathsf{k}}[p^{\mathsf{p}}[in^{\ell_4^i} w] \mid open^{\ell_3^p} p \mid Q] \\
& \rightarrow w^{\mathsf{w}}[open^{\ell_5^p} k.P] \mid k^{\mathsf{k}}[in^{\ell_4^i} w \mid Q] \\
& \rightarrow w^{\mathsf{w}}[k^{\mathsf{k}}[Q] \mid open^{\ell_5^p} k.P] \\
& \rightarrow w^{\mathsf{w}}[Q \mid P]
\end{aligned}
$$

The reduction sequence shows that w sends p into k; k opens p and enters w and finally k is opened. ■

Throughout the development we consider a process, $P_\star$, executing in an environment represented by the ambient $n_\star$ with label $\ell_\star^a$ such that neither (names equivalent to) $n_\star$ nor $\ell_\star^a$ occurs inside $P_\star$. This amounts to a system of the form $n_\star^{\ell_\star^a}[P_\star]$ where there is exactly one occurrence of the name $n_\star$ (modulo $\equiv_\alpha$) and of the label $\ell_\star^a$. Clearly the set $\mathbf{Lab}_\star$ of labels and the set $\mathbf{SNam}_\star$ of representatives of names occuring in $n_\star^{\ell_\star^a}[P_\star]$ will be finite subsets of **Lab** and **SNam**, respectively, and we shall occasionally restrict **Lab** and **SNam** to be these finite subsets.

3 Occurrence Counting Analysis

In this section we present an analysis that counts occurrences of ambients. It can be seen as an abstract collecting semantics in that it is very precise (on processes without replication) but its results are rather unwieldy, and the analysis is therefore unsuitable for implementation. Nonetheless, it is a useful intermediate step in developing more approximate analyses, in the spirit of abstract interpretation, cf. [5]. As motivated in Section 2, an ambient will be identified by its label $\ell^a \in \mathsf{Lab}^a$ and a capability by its label $\ell \in \mathsf{Lab}^i \cup \mathsf{Lab}^o \cup \mathsf{Lab}^p$.

3.1 Specification of the Analysis

The specification of the analysis is split into three: first the abstract domains are presented; then we define and discuss the extraction and representation functions used in the analysis; and finally the flow logic specification is given.

3.1.1 Domains of the analysis

The analysis operates on sets of representations of processes. A process can be represented by a triple $(\hat{I}, \hat{H}, \hat{A}) \in \mathsf{InAmb} \times \mathsf{OfNam} \times \mathsf{Accum}$; the individual components of the triple are described below.

For each ambient the set of ambients and capabilities contained in it is recorded in the following component:

$$\hat{I} \in \mathsf{InAmb} = \mathcal{P}(\mathsf{Lab}^a \times (\mathsf{Lab}^a \cup \mathsf{Lab}^i \cup \mathsf{Lab}^o \cup \mathsf{Lab}^p))$$

If a process contains an ambient labelled ℓ^a enclosing a capability or ambient labelled ℓ then $(\ell^a, \ell) \in \hat{I}$ should hold in order for $(\hat{I}, \hat{H}, \hat{A})$ to be a correct representation of the process.

Each occurrence of an ambient has a name and to keep track of this information we have the component:

$$\hat{H} \in \mathsf{OfNam} = \mathcal{P}(\mathsf{Lab}^a \times \mathsf{SNam})$$

If a process contains an ambient labelled ℓ^a with name n then $(\ell^a, \lfloor n \rfloor) \in \hat{H}$ should hold in order for $(\hat{I}, \hat{H}, \hat{A})$ to be a correct representation of the process. Note that $\hat{H}$ may be taken to be a partial function, $\hat{H} \in \mathsf{Lab}^a \to_{\mathrm{par}} \mathsf{SNam}$, when all ambients are uniquely labelled.

Furthermore the representation contains information about the number of occurrences of each ambient called the *multiplicity* of the ambient. The occurrence counting information is recorded in the $\hat{A}$-component:

$$\hat{A} \in \textsf{\textbf{Accum}} = \textsf{\textbf{Lab}}^a \to_{\mathrm{par}} (\textsf{\textbf{Mult}} \setminus \{0\})$$

where **Mult** $= \{0, 1, \omega\}$ and ω should be read as two or more, 1 as exactly one, and 0 as exactly zero. Thus, 0 represents *absence* of an ambient, but rather than explicitly assigning multiplicity 0 to ambients that do not occur, we simply do not include them in the domain, $\mathrm{dom}(\hat{A})$, of the *partial* function $\hat{A}$; hence all $(\hat{I}, \hat{H}, \hat{A})$ of interest will be finitary.

The **Mult** domain is equipped with an "addition" operator $\oplus$:

$\oplus$	0	1	ω
0	0	1	ω
1	1	ω	ω
ω	ω	ω	ω

If a process contains an ambient labelled ℓ^a then $\hat{A}(\ell^a)$ should be 1 or ω (depending on the actual number of occurrences of ℓ^a) in order for $(\hat{I}, \hat{H}, \hat{A})$ to be an acceptable representation of the process. We write $\perp$ for the partial function with $\mathrm{dom}(\perp) = \emptyset$.

We say that $(\hat{I}, \hat{H}, \hat{A})$ is *compatible* whenever

if $(\ell^a, \ell) \in \hat{I}$ or $(\ell^a, n) \in \hat{H}$ then $(\{\ell^a, \ell\} \cap \textsf{\textbf{Lab}}^a) \subseteq \mathrm{dom}(\hat{A})$,
if $\ell^a \in \mathrm{dom}(\hat{A})$ then there exists n such that $(\ell^a, n) \in \hat{H}$, and
if $\hat{A}(\ell^a) = 1$ then $\{n \mid (\ell^a, n) \in \hat{H}\}$ is a singleton and $\{\ell \mid (\ell, \ell^a) \in \hat{I}\}$ is a singleton or empty.

This says that the ambient labels with non-0 multiplicity are the same as the ambient labels with names and that they include all ambient labels in the triple; it also says that ambient labels occurring only once have exactly one name and at most one parent. We define **Count** to be the set of compatible triples:

$$\textsf{\textbf{Count}} = \{(\hat{I}, \hat{H}, \hat{A}) \in \textsf{\textbf{InAmb}} \times \textsf{\textbf{OfNam}} \times \textsf{\textbf{Accum}} \mid (\hat{I}, \hat{H}, \hat{A}) \text{ is compatible}\}.$$

Finally, a proposed analysis $\mathcal{C}$ is a set of representations of processes:

$$\mathcal{C} \in \textsf{\textbf{CountSet}} = \mathcal{P}(\textsf{\textbf{Count}}).$$

Clearly **CountSet** is a complete lattice under the subset ordering.

$$\begin{tikzcd}
\mathcal{P}(\textsf{\textbf{Proc}}) & & \textsf{\textbf{CountSet}} \\
\textsf{\textbf{Proc}} \arrow[u, "\{\cdot\}"] \arrow[urr, "\beta^{\mathrm{OC}}"'] \arrow[rr, "\eta^{\mathrm{OC}}"'] & & \textsf{\textbf{Count}} \arrow[u, "\{\cdot\}"']
\end{tikzcd}$$

Fig. 4. Extraction (η^{OC}) and representation (β^{OC}) functions [8].

Variations. The notion of compatible could be made more demanding, e.g. by requiring that whenever $(\ell^a, \ell) \in \hat{I}$ then $(\ell^a_\star, \ell) \in \hat{I}^*$ (writing $\hat{I}^*$ for the reflexive and transitive closure of the relation $\hat{I}$ and recalling that $\ell^a_\star$ is the label of the top-level environment). As it stands, there are more compatible triples than are constructed by the representation function η^{OC} to be defined in Figure 5; we abstain from using the more demanding notion of compatible in order not to overly complicate the specification of the analysis in Figures 6 and 7.

In this paper occurrences are counted *globally* with respect to the entire system considered. A more precise, and much more costly, approach where occurrences are counted *locally* with respect to their parent is considered in [12], where the precision furthermore is increased by recording which capabilities block what other capabilities; roughly, this would correspond to extending **InAmb** to be $\mathcal{P}(\textsf{\textbf{Lab}} \times \textsf{\textbf{Lab}})$.

3.1.2 Extraction and representation functions

We already informally explained the demands on a triple $(\hat{I}, \hat{H}, \hat{A})$ in order to represent a process $P_\star$. The desired triple is formally given as $\eta^{\mathrm{OC}}(P_\star)$ where the extraction function η^{OC} is defined in Figure 5. Its functionality is summarised in Figure 4: it maps processes to their canonical (abstract) representation, and it will be invariant under α-equivalence and $\equiv$-equivalence, cf. Lemma 2.

The definition of η^{OC} makes use of the operator $\uplus$ in order to combine representations of processes; it is defined by

$$(\hat{I}_1, \hat{H}_1, \hat{A}_1) \uplus (\hat{I}_2, \hat{H}_2, \hat{A}_2) = (\hat{I}_1 \cup \hat{I}_2, \hat{H}_1 \cup \hat{H}_2, \hat{A}_1 \oplus \hat{A}_2)$$

$$\begin{array}{ll}
\eta^{\mathrm{OC}}_{\ell}((\nu n)P) & = \eta^{\mathrm{OC}}_{\ell}(P) \\
\eta^{\mathrm{OC}}_{\ell}(\mathbf{0}) & = (\emptyset, \emptyset, \bot) \\
\eta^{\mathrm{OC}}_{\ell}(P \mid Q) & = \eta^{\mathrm{OC}}_{\ell}(P) \uplus \eta^{\mathrm{OC}}_{\ell}(Q) \\
\eta^{\mathrm{OC}}_{\ell}(!P) & = \text{ let } (\hat{I}, \hat{H}, \hat{A}) = \eta^{\mathrm{OC}}_{\ell}(P) \\
 & \quad \text{ in } (\hat{I}, \hat{H}, [\mathrm{dom}(\hat{A}) \mapsto \omega]) \\
\eta^{\mathrm{OC}}_{\ell}(n^{\ell^a}[P]) & = \eta^{\mathrm{OC}}_{\ell^a}(P) \uplus (\{(\ell, \ell^a)\}, \{(\ell^a, \lfloor n \rfloor)\}, [\{\ell^a\} \mapsto 1]) \\
\eta^{\mathrm{OC}}_{\ell}(in^{\ell^i} n.P) & = \eta^{\mathrm{OC}}_{\ell}(P) \uplus (\{(\ell, \ell^i)\}, \emptyset, \bot) \\
\eta^{\mathrm{OC}}_{\ell}(out^{\ell^o} n.P) & = \eta^{\mathrm{OC}}_{\ell}(P) \uplus (\{(\ell, \ell^o)\}, \emptyset, \bot) \\
\eta^{\mathrm{OC}}_{\ell}(open^{\ell^p} n.P) & = \eta^{\mathrm{OC}}_{\ell}(P) \uplus (\{(\ell, \ell^p)\}, \emptyset, \bot) \\
 & \\
\eta^{\mathrm{OC}}(P_\star) & = \eta^{\mathrm{OC}}_{\ell^a_\star}(P_\star) \uplus (\emptyset, \{(\ell^a_\star, \lfloor n_\star \rfloor)\}, [\{\ell^a_\star\} \mapsto 1])
\end{array}$$

Fig. 5. Extraction function for occurrence counting.

where $\oplus$ is extended to **Accum** $\times$ **Accum** as follows:

$$(\hat{A}_1 \oplus \hat{A}_2)(\ell) = \begin{cases} \hat{A}_1(\ell) \oplus \hat{A}_2(\ell) & \text{if } \ell \in \mathrm{dom}(\hat{A}_1) \ \wedge \ \ell \in \mathrm{dom}(\hat{A}_2) \\ \hat{A}_1(\ell) & \text{if } \ell \in \mathrm{dom}(\hat{A}_1) \ \wedge \ \ell \notin \mathrm{dom}(\hat{A}_2) \\ \hat{A}_2(\ell) & \text{if } \ell \notin \mathrm{dom}(\hat{A}_1) \ \wedge \ \ell \in \mathrm{dom}(\hat{A}_2) \\ \text{undef} & \text{if } \ell \notin \mathrm{dom}(\hat{A}_1) \ \wedge \ \ell \notin \mathrm{dom}(\hat{A}_2) \end{cases}$$

Clearly $\uplus$ produces a compatible triple from two compatible triples. The notation $[S \mapsto \mathsf{c}]$ is used for the partial function that is only defined on S and that gives the constant value c for all arguments in S. Notice that all ambients "inside" a replication are assigned the multiplicity ω as is natural due to the congruence axiom $!P \equiv P \mid !P$ that ensures that $!P \equiv P \mid \cdots \mid P \mid !P$.

It is clear that the extraction function η^{OC} only produces compatible triples. The representation function β^{OC} (cf. Figure 4) for a process is then defined in terms of the extraction function η^{OC}:

$$\beta^{\mathrm{OC}}(P_\star) = \{\eta^{\mathrm{OC}}(P_\star)\}.$$

3.1.3 The flow logic

The specification of the occurrence counting analysis is given in Figures 6 and 7 in the form of a flow logic and is explained below. It is intended to ensure that the set $\mathcal{C}$ of triples is closed under reduction and therefore contains non-trivial clauses only in the case of capabilities; ensuring that a triple describing the

$$
\begin{array}{lll}
\mathcal{C} \models^{\mathrm{OC}} (\nu n)P & \text{iff} & \mathcal{C} \models^{\mathrm{OC}} P \\
\mathcal{C} \models^{\mathrm{OC}} \mathbf{0} & \text{always} & \\
\mathcal{C} \models^{\mathrm{OC}} P \mid Q & \text{iff} & \mathcal{C} \models^{\mathrm{OC}} P \ \wedge\ \mathcal{C} \models^{\mathrm{OC}} Q \\
\mathcal{C} \models^{\mathrm{OC}} !P & \text{iff} & \mathcal{C} \models^{\mathrm{OC}} P \\
\mathcal{C} \models^{\mathrm{OC}} n^{\ell^a}[P] & \text{iff} & \mathcal{C} \models^{\mathrm{OC}} P
\end{array}
$$

$$
\begin{array}{l}
\mathcal{C} \models^{\mathrm{OC}} in^{\ell^i} n.P \ \text{iff} \quad \mathcal{C} \models^{\mathrm{OC}} P \ \wedge\ \forall (\hat{I}, \hat{H}, \hat{A}) \in \mathcal{C} : \forall \ell^a, \ell', \ell'' : \\
\quad \Big((\ell^a, \ell^i) \in \hat{I} \ \wedge\ (\ell'', \ell^a) \in \hat{I} \ \wedge\ (\ell'', \ell') \in \hat{I} \ \wedge\ (\ell', \lfloor n \rfloor) \in \hat{H} \Big) \Rightarrow \\
\quad\quad \left(\begin{array}{l}
\text{case } \hat{A}(\ell^a) \text{ of} \\
1 : (\hat{I} \setminus \{(\ell'', \ell^a)\} \cup \{(\ell', \ell^a)\}, \hat{H}, \hat{A}) \in \mathcal{C} \ \wedge \\
\quad (\hat{I} \setminus \{(\ell'', \ell^a), (\ell^a, \ell^i)\} \cup \{(\ell', \ell^a)\}, \hat{H}, \hat{A}) \in \mathcal{C} \\
\omega : (\hat{I} \cup \{(\ell', \ell^a)\}, \hat{H}, \hat{A}) \in \mathcal{C} \\
\quad (\hat{I} \setminus \{(\ell'', \ell^a)\} \cup \{(\ell', \ell^a)\}, \hat{H}, \hat{A}) \in \mathcal{C} \\
\quad (\hat{I} \setminus \{(\ell^a, \ell^i)\} \cup \{(\ell', \ell^a)\}, \hat{H}, \hat{A}) \in \mathcal{C} \\
\quad (\hat{I} \setminus \{(\ell'', \ell^a), (\ell^a, \ell^i)\} \cup \{(\ell', \ell^a)\}, \hat{H}, \hat{A}) \in \mathcal{C}
\end{array} \right)
\end{array}
$$

Fig. 6. Specification of the analysis with occurrence counting (part 1).

initial process P is part of $\mathcal{C}$ is expressed using the extraction function defined above, by demanding that $\eta^{\mathrm{OC}}(P) \in \mathcal{C}$, and is not part of the specification of the analysis in the approach taken here.

Case $in^{\ell^i} n.P$**:** The clause for $in^{\ell^i} n.P$ first checks the subprocess P. Then all representations of processes in $\mathcal{C}$ are considered. Each time the capability occurs inside an ambient ℓ^a with a sibling ℓ' with the name $\lfloor n \rfloor$, a demand on $\mathcal{C}$ is made depending on the multiplicity of ℓ^a.

Case $in^{\ell^i} n.P$**,** $\hat{A}(\ell^a) = 1$**:** If ℓ^a has the multiplicity 1 we know that ℓ^a is the only ambient with that name inside its parent, the ambient ℓ''. In order to represent the effect of executing an *in*-capability we must represent the fact that the ambient ℓ^a is no longer directly inside its previous parent ℓ'', i.e.

$$\hat{I} \setminus \{(\ell'', \ell^a)\} \ldots$$

Furthermore, ℓ^a has not merely disappeared but has moved inside its previous sibling, ℓ'. So far this yields the formula

$$\hat{I} \setminus \{(\ell'', \ell^a)\} \cup \{(\ell', \ell^a)\} \tag{1}$$

for the new $\hat{I}$-component. Since we do not know whether or not a capability labelled ℓ^i occurs inside the ambient ℓ^a somewhere else in the process we

$$
\begin{array}{l}
\mathcal{C} \models^{\mathrm{OC}} out^{\ell^o} n.P \quad \text{iff} \quad \mathcal{C} \models^{\mathrm{OC}} P \ \wedge \ \forall (\hat{I}, \hat{H}, \hat{A}) \in \mathcal{C} : \forall \ell^a, \ell', \ell'' : \\
\qquad \Big((\ell^a, \ell^o) \in \hat{I} \ \wedge \ (\ell', \ell^a) \in \hat{I} \ \wedge \ (\ell'', \ell') \in \hat{I} \ \wedge \ (\ell', \lfloor n \rfloor) \in \hat{H} \Big) \Rightarrow \\
\qquad \left(\begin{array}{l}
\text{case } \hat{A}(\ell^a) \text{ of} \\
1 : (\hat{I} \setminus \{(\ell', \ell^a)\} \cup \{(\ell'', \ell^a)\}, \hat{H}, \hat{A}) \in \mathcal{C} \\
\quad (\hat{I} \setminus \{(\ell', \ell^a), (\ell^a, \ell^o)\} \cup \{(\ell'', \ell^a)\}, \hat{H}, \hat{A}) \in \mathcal{C} \\
\omega : (\hat{I} \cup \{(\ell'', \ell^a)\}, \hat{H}, \hat{A}) \in \mathcal{C} \\
\quad (\hat{I} \setminus \{(\ell', \ell^a)\} \cup \{(\ell'', \ell^a)\}, \hat{H}, \hat{A}) \in \mathcal{C} \\
\quad (\hat{I} \setminus \{(\ell^a, \ell^o)\} \cup \{(\ell'', \ell^a)\}, \hat{H}, \hat{A}) \in \mathcal{C} \\
\quad (\hat{I} \setminus \{(\ell', \ell^a), (\ell^a, \ell^o)\} \cup \{(\ell'', \ell^a)\}, \hat{H}, \hat{A}) \in \mathcal{C}
\end{array} \right)
\end{array}
$$

$$
\begin{array}{l}
\mathcal{C} \models^{\mathrm{OC}} open^{\ell^p} n.P \ \text{iff} \quad \mathcal{C} \models^{\mathrm{OC}} P \ \wedge \ \forall (\hat{I}, \hat{H}, \hat{A}) \in \mathcal{C} : \forall \ell^a, \ell' : \\
\quad \Big((\ell^a, \ell^p) \in \hat{I} \ \wedge \ (\ell^a, \ell') \in \hat{I} \ \wedge \ (\ell', \lfloor n \rfloor) \in \hat{H} \Big) \Rightarrow \\
\quad \left(\begin{array}{l}
\text{case } \hat{A}(\ell') \text{ of} \\
1 : \forall X \subseteq \{(\ell^a, \ell^p)\} : \\
\quad (\hat{I} \setminus (\{(\ell', \ell) \in \hat{I} | \, \ell \in \textsf{Lab}\} \cup \{(\ell^a, \ell')\} \cup X) \cup \{(\ell^a, \ell) | \, (\ell', \ell) \in \hat{I}\}, \\
\quad\ \hat{H} \setminus \{(\ell', m) | \, m \in \textsf{SNam}\}, \hat{A} \setminus \{\ell'\}) \in \mathcal{C} \\
\omega : \forall Z \subseteq \{(\ell', \ell) \in \hat{I} | \, \ell \in \textsf{Lab}\} : \forall X \subseteq \{(\ell^a, \ell^p)\} : \forall \mathsf{c} \in \{1, \omega\} : \\
\quad \forall Y \subseteq Z \text{ s.t. } Y \supseteq Z \cap \{(\ell', \ell) \in \hat{I} \mid \hat{A}(\ell) = 1\} : \\
\quad \text{case } \mathsf{c} \text{ of} \\
\quad \omega : \forall V \subseteq \{(\ell', \lfloor n \rfloor)\} \text{ s.t. } \exists m \in \textsf{SNam} : (\ell', m) \in \hat{H} \setminus V : \\
\qquad \forall W \subseteq \{(\ell^a, \ell')\} \text{ s.t. } \exists \ell \colon (\ell, \ell') \in ((\hat{I} \setminus (Y \cup X)) \cup \{(\ell^a, \ell) | \, (\ell', \ell) \in Z\}) \setminus W : \\
\qquad (((\hat{I} \setminus (Y \cup X)) \cup \{(\ell^a, \ell) | \, (\ell', \ell) \in Z\}) \setminus W, \hat{H} \setminus V, \hat{A}[\ell' \mapsto \mathsf{c}]) \in \mathcal{C} \\
\quad 1 : \forall V \mathrel{\underset{\cdot\cdot}{\subset}} \{(\ell', m) \mid (\ell', m) \in \hat{H}\} : \\
\qquad \forall W \mathrel{\underset{\cdot\cdot}{\subset}} \{(\ell, \ell') \mid (\ell, \ell') \in ((\hat{I} \setminus (Y \cup X)) \cup \{(\ell^a, \ell) | \, (\ell', \ell) \in Z\})\} : \\
\qquad (((\hat{I} \setminus (Y \cup X)) \cup \{(\ell^a, \ell) | \, (\ell', \ell) \in Z\}) \setminus W, \hat{H} \setminus V, \hat{A}[\ell' \mapsto \mathsf{c}]) \in \mathcal{C}
\end{array} \right)
\end{array}
$$

Fig. 7. Specification of the analysis with occurrence counting (part 2).

have to consider two cases, one where (ℓ^a, ℓ^i) should not be removed from the representation (as in (1) above) and one where it should be:

$$\hat{I} \setminus \{(\ell'', \ell^a), (\ell^a, \ell^i)\} \cup \{(\ell', \ell^a)\} \tag{2}$$

Thus, in order to represent the effect of an *in*-capability, it is demanded that the representations (1) and (2) both are in $\mathcal{C}$.

Case $in^{\ell^i} n.P$, $\hat{A}(\ell^a) = \omega$: In the case where ℓ^a has multiplicity ω the general pattern is much as above, but we cannot make any assumption on the number of ambients labeled ℓ^a that occur inside the ambient ℓ'' currently being analysed. This is due to the fact that the analysis counts labels *globally* and hence we do not know which part of the global count that is pertinent to the situation at hand. Therefore we have two cases to consider: one that represents the case that two or more ambients labelled ℓ^a occur inside ℓ'' and one that represents the case that ℓ^a occurs exactly once inside ℓ''. As for multiplicity 1 we have subcases depending on whether or not ℓ^i still occurs in the process.

Case $out^{\ell^o} n.P$: The clause for *out*-capabilities is similar and will not be explained.

Case $open^{\ell^p} n.P$: The clause for $open^{\ell^p} n.P$ first checks the subprocess P. Then all representations of processes in $\mathcal{C}$ are considered. Each time the capability occurs with a sibling ℓ' with the name $\lfloor n \rfloor$, a demand on $\mathcal{C}$ is made depending on the multiplicity of ℓ'.

Case $open^{\ell^p} n.P$, $\hat{A}(\ell') = 1$: If ℓ' has the multiplicity 1 we know that opening ℓ' destroys it, thus it should be removed from the representation, i.e.

$$\hat{I} \setminus \{(\ell^a, \ell')\} \ldots$$

Furthermore, all the capabilities and ambients inside it must be removed as well, yielding

$$\hat{I} \setminus (\{(\ell', \ell) \in \hat{I} | \, \ell \in \textsf{Lab}\} \cup \{(\ell^a, \ell')\}) \ldots$$

only to be added again at a higher level (the same level as the now opened parent):

$$\hat{I} \setminus (\{(\ell', \ell) \in \hat{I} | \, \ell \in \textsf{Lab}\} \cup \{(\ell^a, \ell')\}) \cup \{(\ell^a, \ell) | \, (\ell', \ell) \in \hat{I}\}$$

As for *in*-capabilities we have to consider further two cases depending on whether or not ℓ^p has to be removed from ℓ^a; to express this more succinctly than before we use X (either empty or a singleton) to denote the information that is to be removed. Putting all of the above together, we arrive at the following formulation of the new $\hat{I}$ component:

$$\begin{aligned} &\forall X \subseteq \{(\ell^a, \ell^p)\} : \\ &\quad \hat{I} \setminus (\{(\ell', \ell) \in \hat{I} | \, \ell \in \textsf{Lab}\} \cup \{(\ell^a, \ell')\} \cup X) \cup \{(\ell^a, \ell) | \, (\ell', \ell) \in \hat{I}\} \end{aligned}$$

Finally we also need to update the $\hat{H}$ and $\hat{A}$ components, which is straightforward as we only need to remove the stable name of the ambient that was opened and to make sure that its counting information is "reset to zero", i.e. removed.

	in Y	in Z
Only in this ℓ'	yes	yes
Both inside and outside this ℓ'	no	yes
Not in this ℓ'	no	no

Fig. 8. Purpose of Y and Z.

Case $open^{\ell^p} n.P$**,** $\hat{A}(\ell') = \omega$**:** If ℓ' has the multiplicity ω we again do not know what part of the global count is pertinent to the situation at hand. Therefore we do not know exactly which ambients and capabilities reported to be inside an ambient labelled ℓ' are in fact inside the ambient actually opened. We therefore consider all possible combinations Z:

$$Z \subseteq \{(\ell', \ell) \in \hat{I} | \, \ell \in \mathsf{Lab}\}$$

Thus a given Z represents a guess at the set of ambients and capabilities actually inside the particular ambient that is opened. Of these some may occur *only* inside the ambient actually opened. These ambients and capabilities must explicitly be removed from the ambient that was opened, but since we do not know exactly which ambients and capabilities that should be included, we again consider all possible combinations Y (ensuring that they include all ambients with multiplicity 1):

$$\forall Y \subseteq Z \text{ s.t. } Y \supseteq Z \cap \{(\ell', \ell) \in \hat{I} \mid \hat{A}(\ell) = 1\}$$

In Figure 8 the purpose of Y and Z is summarised. The specification then ensures that ambients and capabilities in Z move inside the parent ambient ℓ^a and that the ones from Y are removed from being inside ℓ' (and where X is defined as above):

$$(\hat{I} \setminus (X \cup Y)) \cup \{(\ell^a, \ell) | \, (\ell', \ell) \in Z\}$$

Since the ambient ℓ' was destroyed by opening it, its new multiplicity intuitively is "$\mathsf{c} = \omega - 1$"; formally we use c to denote the possible solutions to $\mathsf{c} + 1 = \omega$ and note that they are $\mathsf{c} = 1$ and $\mathsf{c} = \omega$ (but not $\mathsf{c} = 0$).

Case $open^{\ell^p} n.P$**,** $\hat{A}(\ell') = \omega$**,** $\mathsf{c} = \omega$**:** For $\mathsf{c} = \omega$ we consider the possibility that the ambient actually opened may be the only ambient with that name so that it may be possible to remove the name from $\hat{H}$; we use V to express the possibilities and take care of compatibility by insisting that the name is not removed if it is the only name possible:

$$\forall V \subseteq \{(\ell', \lfloor n \rfloor)\} \text{ s.t. } \exists m \in \mathsf{SNam} : (\ell', m) \in \hat{H} \setminus V :$$
$$\ldots \hat{H} \setminus V \ldots$$

In a similar way the ambient actually opened may be the only ambient labelled ℓ' occurring inside ℓ^a so that it may be possible to remove the link (ℓ^a, ℓ') from $\hat{I}$; we use W to express the possibilities and take care of compatibility by insisting that the link is not removed if it is the only link possible:

$$\begin{array}{l}\forall W \subseteq \{(\ell^a, \ell')\} \text{ s.t. } \exists \ell \colon (\ell, \ell') \in ((\hat{I} \setminus (Y \cup X)) \cup \{(\ell^a, \ell) | \, (\ell', \ell) \in Z\}) \setminus W \colon \\ \qquad \ldots ((\hat{I} \setminus (Y \cup X)) \cup \{(\ell^a, \ell) | \, (\ell', \ell) \in Z\}) \setminus W \ldots\end{array}$$

Finally the $\hat{A}$ component is updated with the new multiplicity of ℓ':

$$\hat{A}[\ell' \mapsto \mathsf{c}]$$

Putting it all together we arrive at the following representation that must be in $\mathcal{C}$:

$$(((\hat{I} \setminus (Y \cup X)) \cup \{(\ell^a, \ell) | \, (\ell', \ell) \in Z\}) \setminus W, \hat{H} \setminus V, \hat{A}[\ell' \mapsto \mathsf{c}])$$

with V, W, X, Y and Z as discussed above.

Case $open^{\ell^p} n.P$**,** $\hat{A}(\ell') = \omega$**,** $\mathsf{c} = 1$**:** For $\mathsf{c} = 1$ the reasoning is analogous to that for $\mathsf{c} = \omega$ above; we take care to ensure compatibility by insisting that $\hat{H}$ records exactly one name for ℓ' and that $\hat{I}$ records exactly one parent for ℓ'. To express this succinctly we use the notation

$$S \mathrel{\dot{\subset}} T \qquad \equiv \qquad (S \subseteq T) \quad \wedge \quad (T \setminus S \text{ is a singleton})$$

meaning that S contains all elements but one of those in T.

The specification is well-defined because it is defined compositionally and because all triples constructed for inclusion in $\mathcal{C}$ are compatible.

Variations. The analysis could be made more demanding, e.g. by considering the case $\mathsf{c} = 1$ only when $\{(\ell', m) \in \hat{H} \mid m \in \textsf{\textbf{SNam}}\}$ contains at most two elements. As it stands, using the terminology of Subsection 4.1, the occurrence counting analysis approximates, but is not induced from, the collecting semantics as usually considered in abstract interpretation; we abstain from developing a stronger analysis so as not to overly complicate the specification in Figures 6 and 7.

3.2 Properties of the Analysis

In this section the formal properties of the occurrence counting analysis are discussed. We first show the semantic correctness of the analysis and next show the existence of least (or best) solutions.

3.2.1 Semantic correctness

Since we have employed an operational semantics for the ambient calculus it is natural to express semantic correctness in the form of a subject reduction result in the manner known from type systems. This leads to a "local" notion of correctness that shows that each reduction of the semantics is adequately mimicked in the analysis; as a preparation we first show an auxiliary result for the structural congruence.

Lemma 2 *If* $P, Q \in \textsf{Proc}$ *then*

$$\beta^{\mathrm{OC}}(P) \subseteq \mathcal{C} \ \wedge \ \mathcal{C} \models^{\mathrm{OC}} P \ \wedge \ P \equiv Q \Rightarrow \beta^{\mathrm{OC}}(Q) \subseteq \mathcal{C} \ \wedge \ \mathcal{C} \models^{\mathrm{OC}} Q.$$

PROOF. It suffices to prove the following statements:

(1) If $P \equiv Q$ then $\mathcal{C} \models^{\mathrm{OC}} P \Leftrightarrow \mathcal{C} \models^{\mathrm{OC}} Q$.

(2) If $P \equiv Q$ then $\forall \ell \in \textsf{Lab}^a : \ \eta_\ell^{\mathrm{OC}}(P) = \eta_\ell^{\mathrm{OC}}(Q)$.

Both are proved by induction in the inference $P \equiv Q$ (defined in Figure 1); the details may be found in Appendix A. ■

Proposition 3 ("Local" correctness) *If* $P, Q \in \textsf{Proc}$ *then*

$$\beta^{\mathrm{OC}}(P) \subseteq \mathcal{C} \ \wedge \ \mathcal{C} \models^{\mathrm{OC}} P \ \wedge \ P \rightarrow Q \Rightarrow \beta^{\mathrm{OC}}(Q) \subseteq \mathcal{C} \ \wedge \ \mathcal{C} \models^{\mathrm{OC}} Q.$$

PROOF. It suffices to prove the following statements:

(1) $\mathcal{C} \models^{\mathrm{OC}} P \ \wedge \ P \rightarrow Q \Rightarrow \mathcal{C} \models^{\mathrm{OC}} Q$

(2) $\forall \ell \in \textsf{Lab}^a : \forall (\hat{I}, \hat{H}, \hat{A}) \in \textsf{Count} :$

$$((\hat{I}, \hat{H}, \hat{A}) \uplus \eta_\ell^{\mathrm{OC}}(P)) \in \mathcal{C} \ \wedge \ \mathcal{C} \models^{\mathrm{OC}} P \ \wedge \ P \rightarrow Q$$
$$\Rightarrow ((\hat{I}, \hat{H}, \hat{A}) \uplus \eta_\ell^{\mathrm{OC}}(Q)) \in \mathcal{C}$$

Both are proved by induction in the inference $P \rightarrow Q$ (defined in Figure 2) and rely on Lemma 2; the details may be found in Appendix B. ■

An alternative approach to semantic correctness is motivated by characterising the set of "reachable" configurations in the manner known from model checking. This leads to a "global" notion of correctness that shows that the set of processes reachable under reduction is accounted for by the analysis.

Corollary 4 ("Global" correctness) *If $P, Q \in$* **Proc** *then*

$$\beta^{\mathrm{OC}}(P) \subseteq \mathcal{C} \ \wedge \ \mathcal{C} \models^{\mathrm{OC}} P \ \wedge \ P \rightarrow^* Q \ \Rightarrow \ \beta^{\mathrm{OC}}(Q) \subseteq \mathcal{C}.$$

PROOF. A straightforward proof by induction in the length of the derivation sequence $P \rightarrow^* Q$, using Proposition 3, shows that

$$\beta^{\mathrm{OC}}(P) \subseteq \mathcal{C} \ \wedge \ \mathcal{C} \models^{\mathrm{OC}} P \ \wedge \ P \rightarrow^* Q \ \Rightarrow \ \beta^{\mathrm{OC}}(Q) \subseteq \mathcal{C} \ \wedge \ \mathcal{C} \models^{\mathrm{OC}} Q$$

from which the corollary trivially follows. ∎

3.2.2 Least solutions

We next show that the set of acceptable solutions constitutes a Moore family; this corresponds to the model intersection property as known from logic. Recall that a subset of a complete lattice, $Y \subseteq L$, is a Moore family if whenever $Y' \subseteq Y$ then $\sqcap Y' \in Y$.

Proposition 5 (Moore family) *The set*

$$\left\{\mathcal{C} \,\middle|\, \mathcal{C} \models^{\mathrm{OC}} P \ \wedge \ \beta^{\mathrm{OC}}(P) \subseteq \mathcal{C}\right\}$$

is a Moore family for every P.

PROOF. Let $(\mathcal{C}_j)_{j \in J}$ be a family of elements in the set, i.e.

$$\forall j \in J : \mathcal{C}_j \models^{\mathrm{OC}} P \qquad \wedge \qquad \forall j \in J : \beta^{\mathrm{OC}}(P) \subseteq \mathcal{C}_j,$$

and show that also $\bigcap_{j \in J} \mathcal{C}_j$ is in the set, i.e.

$$(\bigcap_{j \in J} \mathcal{C}_j) \models^{\mathrm{OC}} P \qquad \wedge \qquad \beta^{\mathrm{OC}}(P) \subseteq (\bigcap_{j \in J} \mathcal{C}_j).$$

The second conjunct is immediate by the properties of intersection (recalling that the intersection over an empty index set yields **Count**). For the first conjunct we proceed by structural induction in P. The cases not involving capabilities are immediate. The cases involving capabilities are rather similar: for each $(\hat{I}, \hat{H}, \hat{A}) \in \bigcap_{j \in J} \mathcal{C}_j$ a family of triples $(\hat{I}_k, \hat{H}_k, \hat{A}_k)_{k \in K}$ is constructed in Figures 6 and 7 and it is demanded that

$$\forall k \in K : (\hat{I}_k, \hat{H}_k, \hat{A}_k) \in \bigcap_{j \in J} \mathcal{C}_j.$$

However, when $(\hat{I}, \hat{H}, \hat{A}) \in \bigcap_{j \in J} \mathcal{C}_j$ also $(\hat{I}, \hat{H}, \hat{A}) \in \mathcal{C}_j$ for all $j \in J$ and it then follows from the assumptions that $(\hat{I}_k, \hat{H}_k, \hat{A}_k) \in \mathcal{C}_j$ for all $j \in J$ and $k \in K$; the result then follows from the properties of intersection. ∎

$$
\begin{aligned}
\beta^{\mathrm{CF}}_{\ell}((\nu n)P) &= \beta^{\mathrm{CF}}_{\ell}(P)\\
\beta^{\mathrm{CF}}_{\ell}(\mathbf{0}) &= (\emptyset, \emptyset)\\
\beta^{\mathrm{CF}}_{\ell}(P \mid Q) &= \beta^{\mathrm{CF}}_{\ell}(P) \sqcup \beta^{\mathrm{CF}}_{\ell}(Q)\\
\beta^{\mathrm{CF}}_{\ell}(!P) &= \beta^{\mathrm{CF}}_{\ell}(P)\\
\beta^{\mathrm{CF}}_{\ell}(n^{\ell^a}[P]) &= \beta^{\mathrm{CF}}_{\ell^a}(P) \sqcup (\{(\ell, \ell^a)\}, \{(\ell^a, \lfloor n \rfloor)\})\\
\beta^{\mathrm{CF}}_{\ell}(in^{\ell^i} n.P) &= \beta^{\mathrm{CF}}_{\ell}(P) \sqcup (\{(\ell, \ell^i)\}, \emptyset)\\
\beta^{\mathrm{CF}}_{\ell}(out^{\ell^o} n.P) &= \beta^{\mathrm{CF}}_{\ell}(P) \sqcup (\{(\ell, \ell^o)\}, \emptyset)\\
\beta^{\mathrm{CF}}_{\ell}(open^{\ell^p} n.P) &= \beta^{\mathrm{CF}}_{\ell}(P) \sqcup (\{(\ell, \ell^p)\}, \emptyset)\\
\\
\beta^{\mathrm{CF}}(P_\star) &= \beta^{\mathrm{CF}}_{\ell^a_\star}(P_\star) \sqcup (\emptyset, \{(\ell^a_\star, \lfloor n_\star \rfloor)\})
\end{aligned}
$$

Fig. 9. Representation Function for CFA.

The Moore family property implies that the counting analysis admits an analysis of every process, and that every process has a *least* (or "best") analysis.

The set $\mathcal{C}$ may be infinite because it is a subset of the infinite set $\mathbf{Count}$. Luckily, it is possible to restrict the set $\mathbf{Count}$ to be finite by restricting all ambient and capability labels to be those occurring in the program $P_\star$ of interest and similarly for representations of names; this defines the finite sets $\mathbf{Count}_\star$ and $\mathbf{CountSet}_\star$ of which $\mathcal{C}$ is a member. It follows that the least solution

$$\mathcal{C}_\star = \bigcap \left\{ \mathcal{C} \in \mathbf{CountSet}_\star \,\middle|\, \mathcal{C} \models^{\mathrm{OC}} P_\star \ \wedge\ \beta^{\mathrm{OC}}(P_\star) \subseteq \mathcal{C} \right\}$$

is in fact computable. A standard worklist algorithm, cf. [8], can be used to find solutions. However, such a computation of $\mathcal{C}_\star$ is likely to require exponential time and space and we are unaware of any algorithms or methods for reducing this complexity significantly. This makes the analysis unsuitable for practical use in automated tools, such as firewall verifiers etc. This suggests developing a coarser but more efficient analysis and is the subject of Section 4.

4 Control Flow Analysis

A coarser analysis can be obtained from the counting analysis by dispensing with the $\hat{A}$ component and by merging the resulting pairs $(\hat{I}, \hat{H})$ into one (by taking their least upper bound). In other words, the analysis works on pairs $(\hat{I}, \hat{H}) \in \mathbf{InAmb} \times \mathbf{OfNam}$ where $\mathbf{InAmb} \times \mathbf{OfNam}$ is a complete lattice under the componentwise partial order. We say that a pair $(\hat{I}, \hat{H})$ is *compatible* whenever

$$
\begin{array}{ll}
(\hat{I},\hat{H}) \models^{\mathrm{CF}} (\nu n)P & \text{iff} \quad (\hat{I},\hat{H}) \models^{\mathrm{CF}} P \\
(\hat{I},\hat{H}) \models^{\mathrm{CF}} \mathbf{0} & \text{always} \\
(\hat{I},\hat{H}) \models^{\mathrm{CF}} P \mid Q & \text{iff} \quad (\hat{I},\hat{H}) \models^{\mathrm{CF}} P \ \wedge\ (\hat{I},\hat{H}) \models^{\mathrm{CF}} Q \\
(\hat{I},\hat{H}) \models^{\mathrm{CF}} !P & \text{iff} \quad (\hat{I},\hat{H}) \models^{\mathrm{CF}} P \\
(\hat{I},\hat{H}) \models^{\mathrm{CF}} n^{\ell^a}[P] & \text{iff} \quad (\hat{I},\hat{H}) \models^{\mathrm{CF}} P \\
(\hat{I},\hat{H}) \models^{\mathrm{CF}} \mathit{in}^{\ell^i} n.P & \text{iff} \quad (\hat{I},\hat{H}) \models^{\mathrm{CF}} P \ \wedge \\
\multicolumn{2}{l}{\quad \forall \ell^a, \ell', \ell'' \in \mathsf{Lab}^a : ((\ell^a,\ell^i) \in \hat{I} \ \wedge\ (\ell'',\ell^a) \in \hat{I} \ \wedge\ (\ell'',\ell') \in \hat{I}} \\
\multicolumn{2}{l}{\qquad \wedge\ (\ell', \lfloor n \rfloor) \in \hat{H}) \ \Rightarrow\ (\ell',\ell^a) \in \hat{I}} \\
(\hat{I},\hat{H}) \models^{\mathrm{CF}} \mathit{out}^{\ell^o} n.P & \text{iff} \quad (\hat{I},\hat{H}) \models^{\mathrm{CF}} P \ \wedge \\
\multicolumn{2}{l}{\quad \forall \ell^a, \ell', \ell'' \in \mathsf{Lab}^a : ((\ell^a,\ell^o) \in \hat{I} \ \wedge\ (\ell',\ell^a) \in \hat{I} \ \wedge\ (\ell'',\ell') \in \hat{I}} \\
\multicolumn{2}{l}{\qquad \wedge\ (\ell', \lfloor n \rfloor) \in \hat{H}) \ \Rightarrow\ (\ell'',\ell^a) \in \hat{I}} \\
(\hat{I},\hat{H}) \models^{\mathrm{CF}} \mathit{open}^{\ell^p} n.P & \text{iff} \quad (\hat{I},\hat{H}) \models^{\mathrm{CF}} P \ \wedge \\
\multicolumn{2}{l}{\quad \forall \ell^a, \ell' \in \mathsf{Lab}^a : ((\ell^a,\ell^p) \in \hat{I} \ \wedge\ (\ell^a,\ell') \in \hat{I} \ \wedge\ (\ell', \lfloor n \rfloor) \in \hat{H})} \\
\multicolumn{2}{l}{\qquad \Rightarrow \left\{(\ell^a,\ell) \mid (\ell',\ell) \in \hat{I}\right\} \subseteq \hat{I}}
\end{array}
$$

Fig. 10. Specification of CFA.

$$\text{if } (\ell^a,\ell) \in \hat{I} \text{ or } (\ell,\ell^a) \in \hat{I} \text{ then there exists } n \text{ such that } (\ell^a,n) \in \hat{H}$$

(recalling that $\ell^a \in \mathsf{Lab}^a$ and $\ell \in \mathsf{Lab}$) and we write $\mathsf{InAmb} \ddot{\times} \mathsf{OfNam}$ for the set of compatible pairs. If Y is a set of compatible pairs and $\bigsqcup Y$ is their least upper bound as calculated in $\mathsf{InAmb} \times \mathsf{OfNam}$ then clearly $\bigsqcup Y$ is compatible. It follows that when $\mathsf{InAmb} \ddot{\times} \mathsf{OfNam}$ is equipped with the componentwise partial order it has the same least upper bounds as in $\mathsf{InAmb} \times \mathsf{OfNam}$; in particular all subsets have least upper bounds. It follows (see e.g. Lemma A.2 in [8]) that $\mathsf{InAmb} \ddot{\times} \mathsf{OfNam}$ is a complete lattice as well.

In keeping with the counting analysis, the coarser analysis is defined by a representation function and a specification. The representation function is shown in Figure 9 and the analysis specification in Figure 10. The specification of the analysis mostly amounts to recursive checks of subprocesses. The case for *in*-capabilities states that if some ambient, labelled ℓ^a, has an *in*-capability (denoted by $(\ell^a,\ell^i) \in \hat{I}$) and has a sibling (denoted $(\ell'',\ell^a) \in \hat{I} \ \wedge\ (\ell'',\ell') \in \hat{I}$) with the right name (denoted $(\ell', \lfloor n \rfloor) \in \hat{H}$) then the result of performing the *in*-capability should also be recorded in $\hat{I}$ (denoted $(\ell',\ell^a) \in \hat{I}$). The cases for the remaining capabilities are similar.

Note that the representation function only produces compatible pairs and that the pairs constructed in the analysis are unproblematic as far as compatibility is concerned.

Example 6 *Recall the scaled down firewall process of Example 1 and let P and Q equal* $\mathbf{0}$ *for simplicity:*

$$w^{\mathsf{w}}[p^{\mathsf{p}}[out^{\ell_1^o} w.in^{\ell_2^i} k.in^{\ell_4^i} w] \mid open^{\ell_5^p} k.\mathbf{0}] \mid k^{\mathsf{k}}[open^{\ell_3^p} p \mid \mathbf{0}]$$

The least solution $(\hat{I}, \hat{H})$ to the analysis of the above process is:

$$\begin{aligned}
\hat{I} = \{&(\ell_\star^a, \mathsf{k}), (\ell_\star^a, \mathsf{p}), (\ell_\star^a, \mathsf{w}),\\
&(\mathsf{k}, \mathsf{k}), (\mathsf{k}, \mathsf{p}), (\mathsf{k}, \mathsf{w}), (\mathsf{k}, \ell_1^o), (\mathsf{k}, \ell_2^i), (\mathsf{k}, \ell_3^p), (\mathsf{k}, \ell_4^i),\\
&(\mathsf{p}, \ell_1^o), (\mathsf{p}, \ell_2^i), (\mathsf{p}, \ell_4^i),\\
&(\mathsf{w}, \mathsf{k}), (\mathsf{w}, \mathsf{p}), (\mathsf{w}, \mathsf{w}), (\mathsf{w}, \ell_1^o), (\mathsf{w}, \ell_2^i), (\mathsf{w}, \ell_3^p), (\mathsf{w}, \ell_4^i), (\mathsf{w}, \ell_5^p)\}\\
\hat{H} = \{&(\ell_\star^a, n_\star), (\mathsf{k}, k), (\mathsf{p}, p), (\mathsf{w}, w)\}
\end{aligned}$$

The above pair is clearly compatible (although not very informative). ■

In the ambient calculus, capabilities are used to change the hierarchy or configuration of ambients. Thus capabilities can be seen as the active control component of the ambient calculus, whereas ambients themselves can be seen as the passive data component. The analysis described above computes a conservative approximation of all the possible ways capabilities may be used to change the hierarchy of ambients. It therefore seems sensible to refer to the analysis as a control flow analysis.

The control flow analysis was originally developed in [9] where the formal properties of the analysis were also established and where the analysis was used to validate the protectiveness of a proposed firewall. Below we first develop a notion of how to induce constraint-based analyses and then show that the analysis can also be induced from the counting analysis, whence the formal properties of the control flow analysis also follow from the general framework to be presented here.

4.1 Systematic Construction of Analyses

As we have seen, it may be necessary to simplify an analysis in order for it to be of practical use. Rather than doing this in an ad-hoc manner, and thus maybe “re-invent” the analysis for every specific use of it, it would be of great value to be able to systematically construct or derive analyses that meet specific criteria regarding complexity and precision.

The framework presented in this paper lends itself to a systematic approach analogous to the use of Galois connections in abstract interpretation. In what follows we assume that $\sqsubseteq_{\mathcal{A}}$ is a partial ordering of the complete lattice $\mathcal{A}$ and that $\sqsubseteq_{\mathcal{A}'}$ is a partial ordering of the complete lattice $\mathcal{A}'$; we shall write $\sqsubseteq$ for both orderings where no confusion can arise. As usual a Galois connection between $\mathcal{A}$ and $\mathcal{A}'$, denoted $\mathcal{A}' \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \mathcal{A}$, is a pair of monotone functions, $\alpha : \mathcal{A}' \to \mathcal{A}$ and $\gamma : \mathcal{A} \to \mathcal{A}'$, such that $\mathrm{id}_{\mathcal{A}'} \sqsubseteq \gamma \circ \alpha$ and $\alpha \circ \gamma \sqsubseteq \mathrm{id}_{\mathcal{A}}$. A Galois connection is a Galois insertion when additionally $\alpha \circ \gamma = \mathrm{id}_{\mathcal{A}}$. We are now in a position to define two ways in which satisfaction relations can be related to each other.

Definition 7 (Induced and Approximate Satisfaction Relation) *Let $\mathcal{A}' \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \mathcal{A}$ be a Galois connection, and let $\models: \mathcal{A} \times \textsf{Proc} \to \{\mathsf{tt}, \mathsf{ff}\}$ and $\models': \mathcal{A}' \times \textsf{Proc} \to \{\mathsf{tt}, \mathsf{ff}\}$ be satisfaction relations.*

(1) The relation $\models$ is said to be induced *from the relation $\models'$ when*

$$\forall A \in \mathcal{A}, P \in \textsf{Proc} : \gamma(A) \models' P \iff A \models P,$$

(2) The relation $\models$ is said to approximate *the relation $\models'$ when*

$$\forall A \in \mathcal{A}, P \in \textsf{Proc} : \gamma(A) \models' P \Longleftarrow A \models P.$$

Clearly, if a satisfaction relation $\models$ is induced from $\models'$ then it also approximates $\models'$. Analogously, representation functions may be induced (but we dispense with defining the notion of a representation function approximating another because this concept seems to be too weak to be useful):

Definition 8 (Induced Representation Function) *Let $\mathcal{A}' \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \mathcal{A}$ be a Galois connection, then a representation function, $\beta : \textsf{Proc} \to \mathcal{A}$ is said to be* induced *from a representation function, $\beta' : \textsf{Proc} \to \mathcal{A}'$ whenever:*

$$\alpha \circ \beta' = \beta.$$

An induced satisfaction relation inherits several important formal properties from the original satisfaction relation, such as correctness and the Moore family property; the "global" notion of correctness is also inherited in the case of an approximate satisfaction relation. We first show that correctness is preserved:

Proposition 9 (Preservation of "local" correctness) *Given $\models$, β, $\models'$ and β' such that $\models$ and β are induced from $\models'$ and β' respectively, via the Galois connection $\mathcal{A}' \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \mathcal{A}$. Then*

$$\beta'(P) \sqsubseteq A' \ \wedge \ A' \models' P \ \wedge \ P \to Q \Rightarrow \beta'(Q) \sqsubseteq A' \ \wedge \ A' \models' Q$$

implies

$$\beta(P) \sqsubseteq A \;\wedge\; A \models P \;\wedge\; P \to Q \;\Rightarrow\; \beta(Q) \sqsubseteq A \;\wedge\; A \models Q$$

for all P, Q, A and A'.

PROOF. We calculate as follows:

$$\begin{aligned}
&\beta(P) \sqsubseteq A \;\wedge\; A \models P \;\wedge\; P \to Q \\
&\quad\Rightarrow\; \beta'(P) \sqsubseteq \gamma(A) \;\wedge\; \gamma(A) \models' P \;\wedge\; P \to Q \\
&\quad\Rightarrow\; \beta'(Q) \sqsubseteq \gamma(A) \;\wedge\; \gamma(A) \models' Q \\
&\quad\Rightarrow\; \beta(Q) \sqsubseteq A \;\wedge\; A \models Q
\end{aligned}$$

This concludes the proof. ■

Proposition 10 (Preservation of "global" correctness) *Assume $\models$, β, $\models'$ and β' are given such that $\models$ approximates $\models'$ and β is induced from β', via the Galois connection $\mathcal{A}' \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \mathcal{A}$. Then*

$$\beta'(P) \sqsubseteq A' \;\wedge\; A' \models' P \;\wedge\; P \to^{*} Q \;\Rightarrow\; \beta'(Q) \sqsubseteq A'$$

implies

$$\beta(P) \sqsubseteq A \;\wedge\; A \models P \;\wedge\; P \to^{*} Q \;\Rightarrow\; \beta(Q) \sqsubseteq A$$

for all P, Q, A and A'.

PROOF. We calculate as follows:

$$\begin{aligned}
&\beta(P) \sqsubseteq A \;\wedge\; A \models P \;\wedge\; P \to^{*} Q \\
&\quad\Rightarrow\; \beta'(P) \sqsubseteq \gamma(A) \;\wedge\; \gamma(A) \models' P \;\wedge\; P \to^{*} Q \\
&\quad\Rightarrow\; \beta'(Q) \sqsubseteq \gamma(A) \\
&\quad\Rightarrow\; \beta(Q) \sqsubseteq A
\end{aligned}$$

This concludes the proof. ■

Next we show that the Moore family property is also preserved:

Proposition 11 (Preservation of Moore family) *Let $\models$, β, $\models'$ and β' be given such that $\models$ and β are induced from $\models'$ and β' respectively, via the Galois* *connection* $\mathcal{A}' \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \mathcal{A}$. *If*

$$\{A' \in \mathcal{A}' \mid A' \models' P \ \wedge \ \beta'(P) \sqsubseteq A'\}$$

is a Moore family for every P*, then*

$$\{A \in \mathcal{A} \mid A \models P \ \wedge \ \beta(P) \sqsubseteq A\}$$

is also a Moore family for every P*.*

PROOF. Let $(A_j)_{j\in J}$ be a family of elements in the set, i.e.

$$\forall j \in J : A_j \models P \qquad \wedge \qquad \forall j \in J : \beta(P) \sqsubseteq A_j$$

and show that also $\sqcap_{j\in J} A_j$ is in the set, i.e.

$$\sqcap_{j\in J} A_j \models P \qquad \wedge \qquad \beta(P) \sqsubseteq \sqcap_{j\in J} A_j.$$

We then calculate as follows using that a concretization function γ is completely multiplicative (see e.g. Lemma 4.22 in [8]):

$$\begin{aligned}
&\forall j \in J : (A_j \models P \ \wedge \ \beta(P) \sqsubseteq A_j) \\
&\quad \Rightarrow \forall j \in J : (\gamma(A_j) \models' P \ \wedge \ \beta'(P) \sqsubseteq \gamma(A_j)) \\
&\quad \Rightarrow \sqcap_j \gamma(A_j) \models' P \ \wedge \ \beta'(P) \sqsubseteq \sqcap_j \gamma(A_j) \\
&\quad \Rightarrow \gamma(\sqcap_j A_j) \models' P \ \wedge \ \beta'(P) \sqsubseteq \gamma(\sqcap_j A_j) \\
&\quad \Rightarrow \sqcap_j A_j \models P \ \wedge \ \beta(P) \sqsubseteq \sqcap_j A_j
\end{aligned}$$

This concludes the proof. ■

Unfortunately, we cannot state an analogue of Proposition 11 in the case where $\models$ merely approximates $\models'$. Specifically it is the last implication in the above proof that cannot be proved in that case. Intuitively this is because the greatest lower bound calculated in $\mathcal{A}'$ cannot, in general, be lifted back to $\mathcal{A}$ in such a way that it can still be guaranteed to be an acceptable analysis result. As a concrete counter example, consider the case where $\mathcal{A}' = \mathcal{A} = \mathcal{P}(\{1, 2\})$, $A' \models' P$ holds for all A', $A \models P$ demands that $A \neq \bot$, $\beta'(P) = \bot$, $\beta(P) = \bot$ and both α and γ are the identities; then $\models$ approximates $\models'$ and $\{A' \in \mathcal{A}' \mid A' \models' P \ \wedge \ \beta'(P) \sqsubseteq A'\}$ is a Moore family unlike what is the case for $\{A \in \mathcal{A} \mid A \models P \ \wedge \ \beta(P) \sqsubseteq A\}$.

In order to show that the control flow analysis is induced from the counting analysis we need a Galois connection. In the following $\sqsubseteq$ is the componentwise ordering on $\textsf{\textbf{InAmb}} \times \textsf{\textbf{OfNam}}$ and we define $(\alpha^{\rm CF}, \gamma^{\rm CF})$ by:

$$\alpha^{\rm CF}(\mathcal{C}) = \bigsqcup\{(\hat{I}, \hat{H}) \,|\, (\hat{I}, \hat{H}, \hat{A}) \in \mathcal{C}\}$$

$$\gamma^{\rm CF}(\hat{I}, \hat{H}) = \{(\hat{I}', \hat{H}', \hat{A}') \,|\, (\hat{I}', \hat{H}') \sqsubseteq (\hat{I}, \hat{H}) \ \wedge \ (\hat{I}', \hat{H}', \hat{A}') \text{ is compatible}\}$$

These functions are clearly monotone and it is straightforward to check that $\gamma^{\rm CF}(\alpha^{\rm CF}(\mathcal{C})) \supseteq \mathcal{C}$ whenever $\mathcal{C} \in \textsf{\textbf{CountSet}}$ (meaning that all triples in $\mathcal{C}$ are compatible) and that $\alpha^{\rm CF}(\gamma^{\rm CF}(\hat{I}, \hat{H})) \sqsubseteq (\hat{I}, \hat{H})$. This shows that $(\alpha^{\rm CF}, \gamma^{\rm CF})$ is a Galois connection between $\textsf{\textbf{CountSet}}$ and $\textsf{\textbf{InAmb}} \times \textsf{\textbf{OfNam}}$.

Recall that the least upper bounds in $\textsf{\textbf{InAmb}} \ddot{\times} \textsf{\textbf{OfNam}}$ are the same as those calculated in $\textsf{\textbf{InAmb}} \times \textsf{\textbf{OfNam}}$. It follows that $\alpha^{\rm CF}(\mathcal{C})$ is compatible whenever $\mathcal{C} \in \textsf{\textbf{CountSet}}$ and this shows that $(\alpha^{\rm CF}, \gamma^{\rm CF})$ is a Galois connection also between $\textsf{\textbf{CountSet}}$ and $\textsf{\textbf{InAmb}} \ddot{\times} \textsf{\textbf{OfNam}}$ (see Figure 11). This Galois connection turns out to be a Galois insertion as well. To see this first note that if $(\hat{I}, \hat{H})$ is compatible then so is $(\hat{I}, \hat{H}, [\{\ell^a \mid (\ell^a, n) \in \hat{H}\} \mapsto \omega])$. For compatible $(\hat{I}, \hat{H})$ this means that

$$\gamma^{\rm CF}(\hat{I}, \hat{H}) \ni (\hat{I}, \hat{H}, [\{\ell^a \mid (\ell^a, n) \in \hat{H}\} \mapsto \omega])$$

and it follows that $\alpha^{\rm CF}(\gamma^{\rm CF}(\hat{I}, \hat{H})) = (\hat{I}, \hat{H})$.

The following proposition then states that the control flow analysis is *induced* (in the sense of Definition 7(1)) from the analysis with occurrence counting:

Proposition 12 *If* $P \in \textsf{\textbf{Proc}}$ *and* $(\hat{I}, \hat{H})$ *is compatible then*

$$\gamma^{\rm CF}(\hat{I}, \hat{H}) \models^{\rm OC} P \iff (\hat{I}, \hat{H}) \models^{\rm CF} P$$

and

$$\alpha^{\rm CF}(\beta^{\rm OC}(P)) = \beta^{\rm CF}(P).$$

PROOF. It suffices to prove

(1) $\forall (\hat{I}, \hat{H}) \in \textsf{\textbf{InAmb}} \ddot{\times} \textsf{\textbf{OfNam}} : \ \gamma^{\rm CF}(\hat{I}, \hat{H}) \models^{\rm OC} P \iff (\hat{I}, \hat{H}) \models^{\rm CF} P$

(2) $\forall \ell \in \textsf{\textbf{Lab}}^a : \ \alpha^{\rm CF}(\{\eta^{\rm OC}_\ell(P)\}) = \beta^{\rm CF}_\ell(P)$

by structural induction in P; the details may be found in Appendix C. ■

$$\begin{tikzcd}
\textsf{CountSet} \arrow[rr, shift right, "\alpha^{\mathrm{CF}}"'] & & \textsf{InAmb} \mathbin{\dddot{\times}} \textsf{OfNam} \arrow[ll, shift right, "\gamma^{\mathrm{CF}}"'] \\
& \textsf{Proc} \arrow[ul, "\beta^{\mathrm{OC}}"] \arrow[ur, "\beta^{\mathrm{CF}}"'] &
\end{tikzcd}$$

Fig. 11. A Galois connection.

The semantic correctness and Moore family property of the control flow analysis now follows from the similar results for the occurrence counting analysis:

Corollary 13 ("Local" and "global" correctness) *If* $P, Q \in \textsf{Proc}$ *then*

$$\beta^{\mathrm{CF}}(P) \sqsubseteq (\hat{I}, \hat{H}) \wedge (\hat{I}, \hat{H}) \models^{\mathrm{CF}} P \wedge P \rightarrow Q \;\Rightarrow\; \beta^{\mathrm{CF}}(Q) \sqsubseteq (\hat{I}, \hat{H}) \wedge (\hat{I}, \hat{H}) \models^{\mathrm{CF}} Q$$

$$\beta^{\mathrm{CF}}(P) \sqsubseteq (\hat{I}, \hat{H}) \wedge (\hat{I}, \hat{H}) \models^{\mathrm{CF}} P \wedge P \rightarrow^{*} Q \;\Rightarrow\; \beta^{\mathrm{CF}}(Q) \sqsubseteq (\hat{I}, \hat{H})$$

hold for all $(\hat{I}, \hat{H}) \in \textsf{InAmb} \mathbin{\dddot{\times}} \textsf{OfNam}$.

PROOF. Immediate from Propositions 3, 9, 10 and 12 and Corollary 4. ■

Corollary 14 (Moore family) *The set*

$$\left\{ (\hat{I}, \hat{H}) \in \textsf{InAmb} \mathbin{\dddot{\times}} \textsf{OfNam} \,\middle|\, (\hat{I}, \hat{H}) \models^{\mathrm{CF}} P \;\wedge\; \beta^{\mathrm{CF}}(P) \sqsubseteq (\hat{I}, \hat{H}) \right\}$$

is a Moore family for every P.

PROOF. Immediate from Propositions 5, 11 and 12. ■

In [11] it is shown that for a given system, $n_{\star}^{\ell_a \star}[P_{\star}]$, of size s, it is possible to devise an $O(s^3)$ algorithm for computing the least solution to the control flow analysis.

5 Conclusion

We have shown how to develop a theory of abstract interpretation in a constraint based manner (based on flow logic) so as to make it useful for analysing calculi of computation and without the need to require processes to be on a "non-standard" form. The development mimics the familiar use of abstract interpretation to induce abstract transfer functions from concrete ones, but has been expressed in a specification oriented manner as known from type systems.

Specifically we developed a counting analysis for the ambient calculus. We proved it semantically correct with respect to the "official" operational semantics and showed that the set of acceptable solutions constitute a Moore family so that a least solution always exists. Next a previously developed control flow analysis was shown to be induced from the counting analysis and its properties derived using the general theory of abstract interpretation in constraint-based form.

In our view the development demonstrates that abstract interpretation and constraint-based analyses naturally complement the use of type systems for analysing calculi of computation.

Furthermore we believe that static analysis has an important role to play in automated validation of software, e.g. security validation. In order to implement tools that are of practical use it is often necessary to develop less complex, and thus less precise, analyses. The framework presented in this paper shows how such analyses can be constructed in a systematic way that preserves important formal properties. As an example the control flow analysis described in Section 4 (derived from the occurrence counting analysis from Section 3) forms the basis of a tool for validating firewalls as described in [9]. A web front-end for the firewall tool is available via the Flow Logic web page: `http://www.daimi.au.dk/~fn/FlowLogic.html`.

Acknowledgements

The authors would like to thank Jacob Grydholt Jensen for cooperating on an earlier version [7].

References

[1] C. Bodei, P. Degano, F. Nielson, and H. Riis Nielson. Control Flow Analysis for the π-calculus. In *Proc. of Concurrency Theory, CONCUR'98*, volume 1466 of *Lecture Notes in Computer Science*, pages 84–98. Springer Verlag, 1998.

[2] C. Bodei, P. Degano, F. Nielson, and H. Riis Nielson. Static Analysis of Processes for No Read-Up and No Write-Down. In *Proc. FoSSaCS'99*, volume 1578 of *Lecture Notes in Computer Science*, pages 120–134. Springer Verlag, 1999.

[3] L. Cardelli and A. D. Gordon. Mobile Ambients. In *Proc. FoSSaCS'98*, volume 1378 of *Lecture Notes in Computer Science*, pages 140–155. Springer Verlag, 1998.

[4] L. Cardelli and A. D. Gordon. Types for Mobile Ambients. In *Conference Record of the Annual ACM Symposium on Principles of Programming Languages, POPL'99*, pages 79–92. ACM Press, 1999.

[5] P. Cousot and R. Cousot. Systematic Design of Program Analysis Frameworks. In *Conference Record of the Annual ACM Symposium on Principles of Programming Languages, POPL'79*, pages 269–282. ACM Press, 1979.

[6] Li Gong. *Inside Java 2 Platform Security*. Addison-Wesley, 1999.

[7] R. R. Hansen, J. G. Jensen, F. Nielson, and H. Riis Nielson. Abstract Interpretation of Mobile Ambients. In Agostino Cortesi and Gilbert Filé, editors, *Proc. Static Analysis Symposium, SAS'99*, volume 1694 of *Lecture Notes in Computer Science*, pages 134–148. Springer Verlag, 1999.

[8] F. Nielson, H. Riis Nielson, and C. Hankin. *Principles of Program Analysis*. Springer, 1999.

[9] F. Nielson, H. Riis Nielson, R. R. Hansen, and J. G. Jensen. Validating Firewalls in Mobile Ambients. In *Proc. of Concurrency Theory, CONCUR'99*, Lecture Notes in Computer Science, pages 463–477. Springer Verlag, 1999.

[10] F. Nielson, H. Riis Nielson, and M. Sagiv. A Kleene analysis of mobile ambients. In *Proc. European Symposium on Programming, ESOP'00*, volume 1782 of *Lecture Notes in Computer Science*, pages 305–319. Springer Verlag, 2000.

[11] F. Nielson and H. Seidl. Control-flow analysis in cubic time. In *Proc. ESOP'01*, Lecture Notes in Computer Science. Springer-Verlag, 2001.

[12] H. Riis Nielson and F. Nielson. Shape Analysis for Mobile Ambients. In *Conference Record of the Annual ACM Symposium on Principles of Programming Languages, POPL'00*, pages 142–154. ACM Press, 2000.

[13] A. Venet. Abstract Interpretation of the π-calculus. In *Proc. Analysis and Verification of Multiple-Agent Languages*, volume 1192 of *Lecture Notes in Computer Science*, pages 51–75. Springer Verlag, 1997.

[14] A. Venet. Automatic Determination of Communication Topologies in Mobile Systems. In *Proc. Static Analysis Symposium, SAS'98*, volume 1503 of *Lecture Notes in Computer Science*, pages 152–167. Springer Verlag, 1998.

A Proof of Lemma 2

For the statement

(1) if $P \equiv Q$ then $\mathcal{C} \models^{\mathrm{OC}} P \Leftrightarrow \mathcal{C} \models^{\mathrm{OC}} Q$

we proceed by induction in the inference $P \equiv Q$ and only consider three illustrative cases.

First suppose that $!P \equiv (P \mid !P)$. Using Figure 6 we calculate that

$$\begin{array}{c} \mathcal{C} \models^{\mathrm{OC}} !P \quad \Leftrightarrow \quad \mathcal{C} \models^{\mathrm{OC}} P \quad \Leftrightarrow \quad \mathcal{C} \models^{\mathrm{OC}} P \;\wedge\; \mathcal{C} \models^{\mathrm{OC}} P \\ \Leftrightarrow \quad \mathcal{C} \models^{\mathrm{OC}} P \;\wedge\; \mathcal{C} \models^{\mathrm{OC}} !P \quad \Leftrightarrow \quad \mathcal{C} \models^{\mathrm{OC}} (P \mid !P) \end{array}$$

and this shows the result.

Next suppose that $n^{\ell^a}[P] \equiv n^{\ell^a}[Q]$ because $P \equiv Q$. Using Figure 6 and the induction hypothesis we calculate that

$$\mathcal{C} \models^{\mathrm{OC}} n^{\ell^a}[P] \quad \Leftrightarrow \quad \mathcal{C} \models^{\mathrm{OC}} P \quad \Leftrightarrow \quad \mathcal{C} \models^{\mathrm{OC}} Q \quad \Leftrightarrow \quad \mathcal{C} \models^{\mathrm{OC}} n^{\ell^a}[Q]$$

as was to be shown.

Finally suppose that $in^{\ell^i} n.P \equiv in^{\ell^i} n.Q$ because $P \equiv Q$. Using Figure 6 and the induction hypothesis we calculate that

$$\begin{array}{rcl} \mathcal{C} \models^{\mathrm{OC}} in^{\ell^i} n.P & \Leftrightarrow & \mathcal{C} \models^{\mathrm{OC}} P \;\wedge\; \forall (\hat{I}, \hat{H}, \hat{A}) \in \mathcal{C} : \forall \ell^a, \ell', \ell'' : \cdots \\ & \Leftrightarrow & \mathcal{C} \models^{\mathrm{OC}} Q \;\wedge\; \forall (\hat{I}, \hat{H}, \hat{A}) \in \mathcal{C} : \forall \ell^a, \ell', \ell'' : \cdots \\ & \Leftrightarrow & \mathcal{C} \models^{\mathrm{OC}} in^{\ell^i} n.Q \end{array}$$

using that the formula $\forall (\hat{I}, \hat{H}, \hat{A}) \in \mathcal{C} : \forall \ell^a, \ell', \ell'' : \cdots$ of Figure 6 is independent of P and Q.

For the statement

(2) if $P \equiv Q$ then $\forall \ell \in \mathbf{Lab}^a : \; \eta_\ell^{\mathrm{OC}}(P) = \eta_\ell^{\mathrm{OC}}(Q)$.

we proceed by induction on $P \equiv Q$ and consider the same three cases as above.

First suppose that $!P \equiv (P \mid !P)$. Using Figure 5 we write

$$(\hat{I}, \hat{H}, \hat{A}) = \eta_\ell^{\mathrm{OC}}(P)$$

and calculate that

$$\begin{array}{rcl} \eta_\ell^{\mathrm{OC}}(!P) & = & (\hat{I}, \hat{H}, [\mathrm{dom}(\hat{A}) \mapsto \omega]) \\ & = & (\hat{I}, \hat{H}, \hat{A}) \uplus (\hat{I}, \hat{H}, [\mathrm{dom}(\hat{A}) \mapsto \omega]) \\ & = & \eta_\ell^{\mathrm{OC}}(P) \uplus \eta_\ell^{\mathrm{OC}}(!P) \\ & = & \eta_\ell^{\mathrm{OC}}(P \mid !P) \end{array}$$

and this shows the result.

Next suppose that $n^{\ell^a}[P] \equiv n^{\ell^a}[Q]$ because $P \equiv Q$. Using Figure 5 and the induction hypothesis we calculate that

$$\begin{array}{rcl} \eta_\ell^{\mathrm{OC}}(n^{\ell^a}[P]) & = & \eta_{\ell^a}^{\mathrm{OC}}(P) \uplus (\{(\ell, \ell^a)\}, \{(\ell^a, \lfloor n \rfloor)\}, [\{\ell^a\} \mapsto 1]) \\ & = & \eta_{\ell^a}^{\mathrm{OC}}(Q) \uplus (\{(\ell, \ell^a)\}, \{(\ell^a, \lfloor n \rfloor)\}, [\{\ell^a\} \mapsto 1]) \\ & = & \eta_\ell^{\mathrm{OC}}(n^{\ell^a}[Q]) \end{array}$$

as was to be shown.

Finally suppose that $in^{\ell^i} n.P \equiv in^{\ell^i} n.Q$ because $P \equiv Q$. Using Figure 5 and the induction hypothesis we calculate that

$$\begin{array}{rcl} \eta_\ell^{\mathrm{OC}}(in^{\ell^i} n.P) & = & \eta_\ell^{\mathrm{OC}}(P) \uplus (\{(\ell, \ell^i)\}, \emptyset, \bot) \\ & = & \eta_\ell^{\mathrm{OC}}(Q) \uplus (\{(\ell, \ell^i)\}, \emptyset, \bot) \\ & = & \eta_\ell^{\mathrm{OC}}(in^{\ell^i} n.Q) \end{array}$$

and this concludes the proof of Lemma 2.

B Proof of Proposition 3

We first prove

(1) $\mathcal{C} \models^{\mathrm{OC}} P \ \wedge \ P \rightarrow Q \ \Rightarrow \ \mathcal{C} \models^{\mathrm{OC}} Q$

by induction in the inference $P \rightarrow Q$ (see Figure 2). Essentially the proof simply amounts to observing that the capabilities present in Q are also present in P. We consider two illustrative cases.

First suppose that

$$\mathcal{C} \models^{\mathrm{OC}} open^{\ell^p} n.P' \mid n^{\ell}[Q']$$

and that

$$open^{\ell^p} n.P' \mid n^{\ell}[Q'] \rightarrow P' \mid Q'$$

and show that $\mathcal{C} \models^{\mathrm{OC}} P' \mid Q'$. Expanding $\mathcal{C} \models^{\mathrm{OC}} \mathit{open}^{\ell^p} n.P' \mid n^{\ell}[Q']$ using Figure 7 we obtain

$$\forall(\hat{I},\hat{H},\hat{A}) \in \mathcal{C} : \forall \ell^a, \ell' : \cdots$$

$$\mathcal{C} \models^{\mathrm{OC}} P'$$

$$\mathcal{C} \models^{\mathrm{OC}} Q'$$

from which $\mathcal{C} \models^{\mathrm{OC}} P' \mid Q'$ easily follows.

Next consider the case where

$$\mathcal{C} \models^{\mathrm{OC}} P$$

and

$$P \to Q \quad \text{because} \quad P \equiv P' \ \wedge\ P' \to Q' \ \wedge\ Q' \equiv Q$$

and show that $\mathcal{C} \models^{\mathrm{OC}} Q$. From $\mathcal{C} \models^{\mathrm{OC}} P$ and $P \equiv P'$ it follows by Lemma 2 that $\mathcal{C} \models^{\mathrm{OC}} P'$; from the induction hypothesis we then have $\mathcal{C} \models^{\mathrm{OC}} Q'$; using Lemma 2 once more we obtain $\mathcal{C} \models^{\mathrm{OC}} Q$ as desired.

Next we prove

$$\begin{array}{l} \textit{(2)}\ \forall \ell \in \mathbf{Lab}^a : \forall(\hat{I},\hat{H},\hat{A}) \in \mathbf{Count} : \\ \qquad ((\hat{I},\hat{H},\hat{A}) \uplus \eta^{\mathrm{OC}}_{\ell}(P)) \in \mathcal{C} \ \wedge\ \mathcal{C} \models^{\mathrm{OC}} P \ \wedge\ P \to Q \\ \qquad\quad \Rightarrow\ ((\hat{I},\hat{H},\hat{A}) \uplus \eta^{\mathrm{OC}}_{\ell}(Q)) \in \mathcal{C} \end{array}$$

by induction in the inference $P \to Q$; we consider all cases in the order listed in Figure 2.

In the case for *in*-capabilities we write

$$(\hat{I}',\hat{H}',\hat{A}') = (\hat{I},\hat{H},\hat{A}) \uplus \eta^{\mathrm{OC}}_{\ell}(n^{\ell^a_n}[\mathit{in}^{\ell^i} m.P' \mid Q'] \mid m^{\ell^a_m}[R'])$$
$$(\hat{I}'',\hat{H}'',\hat{A}'') = (\hat{I},\hat{H},\hat{A}) \uplus \eta^{\mathrm{OC}}_{\ell}(m^{\ell^a_m}[n^{\ell^a_n}[P' \mid Q'] \mid R'])$$

and assume that

$$(\hat{I}',\hat{H}',\hat{A}') \in \mathcal{C}$$

$$\mathcal{C} \models^{\mathrm{OC}} n^{\ell^a_n}[\mathit{in}^{\ell^i} m.P' \mid Q'] \mid m^{\ell^a_m}[R']$$

$$n^{\ell^a_n}[\mathit{in}^{\ell^i} m.P' \mid Q'] \mid m^{\ell^a_m}[R'] \to m^{\ell^a_m}[n^{\ell^a_n}[P' \mid Q'] \mid R']$$

and must show that

$$(\hat{I}'',\hat{H}'',\hat{A}'') \in \mathcal{C}.$$

We calculate that

$$
\begin{aligned}
&(\hat{I}', \hat{H}', \hat{A}') \\
&\quad = (\hat{I}, \hat{H}, \hat{A}) \uplus \eta^{\mathrm{OC}}_{\ell^a_n}(P') \uplus \eta^{\mathrm{OC}}_{\ell^a_n}(Q') \uplus \eta^{\mathrm{OC}}_{\ell^a_m}(R') \\
&\qquad \uplus(\{(\ell, \ell^a_n), (\ell^a_n, \ell^i), (\ell, \ell^a_m)\}, \{(\ell^a_n, \lfloor n \rfloor), (\ell^a_m, \lfloor m \rfloor)\}, \{[\{\ell^a_n, \ell^a_m\} \mapsto 1]\})
\end{aligned}
$$

and that

$$
\begin{aligned}
&(\hat{I}'', \hat{H}'', \hat{A}'') \\
&\quad = (\hat{I}, \hat{H}, \hat{A}) \uplus \eta^{\mathrm{OC}}_{\ell^a_n}(P') \uplus \eta^{\mathrm{OC}}_{\ell^a_n}(Q') \uplus \eta^{\mathrm{OC}}_{\ell^a_m}(R') \\
&\qquad \uplus(\{(\ell^a_m, \ell^a_n), (\ell, \ell^a_m)\}, \{(\ell^a_n, \lfloor n \rfloor), (\ell^a_m, \lfloor m \rfloor)\}, \{[\{\ell^a_n, \ell^a_m\} \mapsto 1]\}).
\end{aligned}
$$

It is immediate that $\hat{H}'' = \hat{H}'$ and $\hat{A}'' = \hat{A}'$. We also have that

$$
\hat{I}'' \in \left\{
\begin{array}{l}
\hat{I}' \cup \{(\ell^a_m, \ell^a_n)\}, \\
\hat{I}' \setminus \{(\ell, \ell^a_n)\} \cup \{(\ell^a_m, \ell^a_n)\}, \\
\hat{I}' \setminus \{(\ell^a_n, \ell^i)\} \cup \{(\ell^a_m, \ell^a_n)\}, \\
\hat{I}' \setminus \{(\ell, \ell^a_n), (\ell^a_n, \ell^i)\} \cup \{(\ell^a_m, \ell^a_n)\}
\end{array}
\right\}
$$

which may be strengthened to

$$
\hat{I}'' \in \left\{
\begin{array}{l}
\hat{I}' \setminus \{(\ell, \ell^a_n)\} \cup \{(\ell^a_m, \ell^a_n)\}, \\
\hat{I}' \setminus \{(\ell, \ell^a_n), (\ell^a_n, \ell^i)\} \cup \{(\ell^a_m, \ell^a_n)\}
\end{array}
\right\}
$$

in the case where $\hat{A}'(\ell^a_n) = 1$ because then the identified occurrence of (ℓ, ℓ^a_n) is the only such occurrence. Since $(\hat{I}', \hat{H}', \hat{A}') \in \mathcal{C}$ and

$$
(\ell^a_n, \ell^i) \in \hat{I}' \ \wedge\ (\ell, \ell^a_n) \in \hat{I}' \ \wedge\ (\ell, \ell^a_m) \in \hat{I}' \ \wedge\ (\ell^a_m, \lfloor m \rfloor) \in \hat{H}'
$$

it follows that the triple $(\hat{I}'', \hat{H}'', \hat{A}'')$ is demanded to be in $\mathcal{C}$ by the specification in Figure 6. This concludes the consideration of *in*-capabilities.

The case for *out*-capabilities is analogous and is omitted.

In the case for *open*-capabilities we write

$$
\begin{aligned}
(\hat{I}', \hat{H}', \hat{A}') &= (\hat{I}, \hat{H}, \hat{A}) \uplus \eta^{\mathrm{OC}}_{\ell}(\mathit{open}^{\ell^p} n.P' \mid n^{\ell^a}[Q']) \\
(\hat{I}'', \hat{H}'', \hat{A}'') &= (\hat{I}, \hat{H}, \hat{A}) \uplus \eta^{\mathrm{OC}}_{\ell}(P' \mid Q')
\end{aligned}
$$

and assume that

$$(\hat{I}', \hat{H}', \hat{A}') \in \mathcal{C}$$

$$\mathcal{C} \models^{\mathrm{OC}} open^{\ell^p} n.P' \mid n^{\ell^a}[Q']$$

$$open^{\ell^p} n.P' \mid n^{\ell^a}[Q'] \to P' \mid Q'$$

and must show that

$$(\hat{I}'', \hat{H}'', \hat{A}'') \in \mathcal{C}.$$

We calculate that

$$\begin{aligned}(\hat{I}', \hat{H}', \hat{A}') = (\hat{I}, \hat{H}, \hat{A}) \uplus \eta_{\ell}^{\mathrm{OC}}(P') \uplus \eta_{\ell^a}^{\mathrm{OC}}(Q') \\ \uplus(\{(\ell, \ell^a), (\ell, \ell^p)\}, \{(\ell^a, \lfloor n \rfloor)\}, \{[\{\ell^a\} \mapsto 1]\})\end{aligned}$$

and that

$$(\hat{I}'', \hat{H}'', \hat{A}'') = (\hat{I}, \hat{H}, \hat{A}) \uplus \eta_{\ell}^{\mathrm{OC}}(P') \uplus \eta_{\ell}^{\mathrm{OC}}(Q').$$

In the case where $\hat{A}'(\ell^a) = 1$ we have identified the unique occurrence of ℓ^a and it follows that $\hat{H}'' = \hat{H}' \setminus \{(\ell^a, m) | \, m \in \mathbf{SNam}\}$ and that $\hat{A}'' = \hat{A}' \setminus \{\ell^a\}$ and that furthermore

$$\hat{I}'' \in \left\{ \begin{array}{l} \hat{I}' \setminus (\{(\ell^a, \ell') \in \hat{I}' | \, \ell' \in \mathbf{Lab}\} \cup \{(\ell, \ell^a), (\ell, \ell^p)\}) \cup \{(\ell, \ell') | \, (\ell^a, \ell') \in \hat{I}'\}, \\ \hat{I}' \setminus (\{(\ell^a, \ell') \in \hat{I}' | \, \ell' \in \mathbf{Lab}\} \cup \{(\ell, \ell^a)\}) \cup \{(\ell, \ell') | \, (\ell^a, \ell') \in \hat{I}'\} \end{array} \right\}$$

depending on whether or not there may be other occurrences of the *open*-capability. Since $(\hat{I}', \hat{H}', \hat{A}') \in \mathcal{C}$ and

$$(\ell, \ell^p) \in \hat{I}' \ \wedge \ (\ell, \ell^a) \in \hat{I}' \ \wedge \ (\ell^a, \lfloor n \rfloor) \in \hat{H}'$$

it follows that the triple $(\hat{I}'', \hat{H}'', \hat{A}'')$ is demanded to be in $\mathcal{C}$ by the specification in Figure 7.

In the case where $\hat{A}'(\ell^a) = \omega$ we take

$$
\begin{aligned}
Z &= \left\{ (\ell^a, \ell') \;\middle|\; \begin{array}{l} (\ell^a, \ell') \text{ is in the } \hat{I}\text{-component of } \eta^{\mathrm{OC}}_{\ell^a}(Q') \wedge \\ (\ell^a, \ell') \text{ is not in the } \hat{I}\text{-component of } \eta^{\mathrm{OC}}_{\ell}(Q') \end{array} \right\} \\
Y &= \{(\ell^a, \ell') \in Z \mid (\ell^a, \ell') \notin \hat{I}''\} \\
X &= \begin{cases} \emptyset & \text{if } (\ell, \ell^p) \in \hat{I}'' \\ \{(\ell, \ell^p)\} & \text{if } (\ell, \ell^p) \notin \hat{I}'' \end{cases} \\
\mathsf{c} &= \hat{A}''(\ell^a) \\
V &= \begin{cases} \emptyset & \text{if } (\ell^a, \lfloor n \rfloor) \in \hat{H}'' \\ \{(\ell^a, \lfloor n \rfloor)\} & \text{if } (\ell^a, \lfloor n \rfloor) \notin \hat{H}'' \end{cases} \\
W &= \begin{cases} \emptyset & \text{if } (\ell, \ell^a) \in \hat{I}'' \\ \{(\ell, \ell^a)\} & \text{if } (\ell, \ell^a) \notin \hat{I}'' \end{cases}
\end{aligned}
$$

and note that $Y \supseteq Z \cap \{(\ell^a, \ell') \in \hat{I}' \mid \hat{A}'(\ell') = 1\}$. Furthermore, we check that

$$
\begin{aligned}
\hat{I}'' &= ((\hat{I}' \setminus (Y \cup X)) \cup \{(\ell, \ell') | \, (\ell^a, \ell') \in Z\}) \setminus W \\
\hat{H}'' &= \hat{H}' \setminus V \\
\hat{A}'' &= \hat{A}'[\ell^a \mapsto \mathsf{c}].
\end{aligned}
$$

For $\mathsf{c} = \omega$ we observe that

$$
\begin{aligned}
&\exists m \in \textbf{SNam} : (\ell^a, m) \in \hat{H}' \setminus V \\
&\exists \ell' : (\ell', \ell^a) \in ((\hat{I} \setminus (Y \cup X)) \cup \{(\ell^a, \ell) | \, (\ell', \ell) \in Z\}) \setminus W
\end{aligned}
$$

and for $\mathsf{c} = 1$ we observe that

$$
\begin{aligned}
&V \mathrel{\underset{\cdot\cdot}{\subset}} \{(\ell^a, m) \mid (\ell^a, m) \in \hat{H}'\} \\
&W \mathrel{\underset{\cdot\cdot}{\subset}} \{(\ell', \ell^a) \mid (\ell', \ell^a) \in ((\hat{I}' \setminus (Y \cup X)) \cup \{(\ell, \ell') | \, (\ell^a, \ell') \in Z\}
\end{aligned}
$$

(where the reason for the different formulation in Figure 7 is the desire only to produce compatible triples). It follows that the triple $(\hat{I}'', \hat{H}'', \hat{A}'')$ is demanded to be in $\mathcal{C}$ by the specification in Figure 7 in all cases.

We are left with the four cases for reduction in context. First consider the case where

$$((\hat{I},\hat{H},\hat{A}) \uplus \eta^{\mathrm{OC}}_{\ell}(n^{\ell^a}[P'])) \in \mathcal{C}$$

$$\mathcal{C} \models^{\mathrm{OC}} n^{\ell^a}[P']$$

$$n^{\ell^a}[P'] \to n^{\ell^a}[Q'] \quad \text{because} \quad P' \to Q'$$

(which clearly also establishes $\mathcal{C} \models^{\mathrm{OC}} P'$) and where we must show that

$$((\hat{I},\hat{H},\hat{A}) \uplus \eta^{\mathrm{OC}}_{\ell}(n^{\ell^a}[P'])) \in \mathcal{C}.$$

Using Figure 5 we calculate

$$\begin{array}{l} (\hat{I},\hat{H},\hat{A}) \;\uplus\; \eta^{\mathrm{OC}}_{\ell}(n^{\ell^a}[P']) \\ = ((\hat{I},\hat{H},\hat{A}) \uplus (\{(\ell,\ell^a)\}, \{(\ell^a, \lfloor n \rfloor)\}, [\{\ell^a\} \mapsto 1])) \;\uplus\; \eta^{\mathrm{OC}}_{\ell^a}(P') \end{array}$$

and the induction hypothesis ensures that whenever the above element is in $\mathcal{C}$ then also

$$\begin{array}{l} ((\hat{I},\hat{H},\hat{A}) \uplus (\{(\ell,\ell^a)\}, \{(\ell^a, \lfloor n \rfloor)\}, [\{\ell^a\} \mapsto 1])) \;\uplus\; \eta^{\mathrm{OC}}_{\ell^a}(Q') \\ = (\hat{I},\hat{H},\hat{A}) \;\uplus\; \eta^{\mathrm{OC}}_{\ell}(n^{\ell^a}[Q']) \end{array}$$

is in $\mathcal{C}$ thereby proving the result.

The cases where

$$(\nu n)P' \to (\nu n)Q' \quad \text{because} \quad P' \to Q'$$

and

$$P' \mid R \to Q' \mid R \quad \text{because} \quad P' \to Q'$$

are entirely analogous. Finally, the case where

$$P \to Q \quad \text{because} \quad P \equiv P' \;\wedge\; P' \to Q' \;\wedge\; Q' \equiv Q$$

makes use of Lemma 2 but is otherwise analogous. This concludes the proof of Proposition 3.

C Proof of Proposition 12

We first prove

(1) $\forall (\hat{I},\hat{H}) \in \textsf{\textbf{InAmb}} \mathbin{\ddot{\times}} \textsf{\textbf{OfNam}} : \;\; \gamma^{\mathrm{CF}}(\hat{I},\hat{H}) \models^{\mathrm{OC}} P \iff (\hat{I},\hat{H}) \models^{\mathrm{CF}} P$

by structural induction in P. Only the cases for capabilities are non-trivial.

Consider the proof of "$\Leftarrow$" in the case of *in*-capabilities: Let $(\hat{I}, \hat{H})$ be compatible such that $(\hat{I}, \hat{H}) \models^{\mathrm{CF}} in^{\ell^i} n.P$ and show that $\gamma^{\mathrm{CF}}(\hat{I}, \hat{H}) \models^{\mathrm{OC}} in^{\ell^i} n.P$. From Figure 10 it follows that $(\hat{I}, \hat{H}) \models^{\mathrm{CF}} P$ and by the induction hypothesis we have $\gamma^{\mathrm{CF}}(\hat{I}, \hat{H}) \models^{\mathrm{OC}} P$. Next consider $(\hat{I}', \hat{H}', \hat{A}') \in \gamma^{\mathrm{CF}}(\hat{I}, \hat{H})$ and ℓ^a, ℓ', ℓ'' such that

$$(\ell^a, \ell^i) \in \hat{I}' \ \wedge \ (\ell'', \ell^a) \in \hat{I}' \ \wedge \ (\ell'', \ell') \in \hat{I}' \ \wedge \ (\ell', \lfloor n \rfloor) \in \hat{H}'.$$

By definition of γ^{CF} it follows that $(\hat{I}', \hat{H}', \hat{A}')$ is compatible and that

$$(\ell^a, \ell^i) \in \hat{I} \ \wedge \ (\ell'', \ell^a) \in \hat{I} \ \wedge \ (\ell'', \ell') \in \hat{I} \ \wedge \ (\ell', \lfloor n \rfloor) \in \hat{H}$$

so that by Figure 10 also $(\ell', \ell^a) \in \hat{I}$. All of the four triples

$$(\hat{I}' \cup \{(\ell', \ell^a)\}, \hat{H}', \hat{A}')$$

$$(\hat{I}' \setminus \{(\ell'', \ell^a)\} \cup \{(\ell', \ell^a)\}, \hat{H}', \hat{A}')$$

$$(\hat{I}' \setminus \{(\ell^a, \ell^i)\} \cup \{(\ell', \ell^a)\}, \hat{H}', \hat{A}')$$

$$(\hat{I}' \setminus \{(\ell'', \ell^a), (\ell^a, \ell^i)\} \cup \{(\ell', \ell^a)\}, \hat{H}', \hat{A}')$$

considered in Figure 6 are clearly compatible and only add the pair (ℓ', ℓ^a) already known to be in $\hat{I}$. It follows that all the triples are elements of $\gamma^{\mathrm{CF}}(\hat{I}, \hat{H})$ and this concludes the proof that $\gamma^{\mathrm{CF}}(\hat{I}, \hat{H}) \models^{\mathrm{OC}} in^{\ell^i} n.P$.

Conversely, consider the proof of "$\Rightarrow$": Let $(\hat{I}, \hat{H})$ be compatible such that $\gamma^{\mathrm{CF}}(\hat{I}, \hat{H}) \models^{\mathrm{OC}} in^{\ell^i} n.P$ and show that $(\hat{I}, \hat{H}) \models^{\mathrm{CF}} in^{\ell^i} n.P$. From Figure 6 it follows that $\gamma^{\mathrm{CF}}(\hat{I}, \hat{H}) \models^{\mathrm{OC}} P$ and by the induction hypothesis we have $(\hat{I}, \hat{H}) \models^{\mathrm{CF}} P$. Next consider ℓ^a, ℓ', ℓ'' such that

$$(\ell^a, \ell^i) \in \hat{I} \ \wedge \ (\ell'', \ell^a) \in \hat{I} \ \wedge \ (\ell'', \ell') \in \hat{I} \ \wedge \ (\ell', \lfloor n \rfloor) \in \hat{H}.$$

We already remarked that

$$(\hat{I}, \hat{H}, [\{\ell^a \mid (\ell^a, n) \in \hat{H}\} \mapsto \omega]) \in \gamma^{\mathrm{CF}}(\hat{I}, \hat{H})$$

and from Figure 6 it follows that

$$(\hat{I} \cup \{(\ell', \ell^a)\}, \hat{H}, [\{\ell^a \mid (\ell^a, n) \in \hat{H}\} \mapsto \omega]) \in \gamma^{\mathrm{CF}}(\hat{I}, \hat{H}).$$

Given the definition of γ^{CF} it follows that $(\ell', \ell^a) \in \hat{I}$ and this concludes the proof that $(\hat{I}, \hat{H}) \models^{\mathrm{CF}} in^{\ell^i} n.P$.

The proof for *out*-capabilities is analogous and we dispense with the details.

Next, consider the proof of "$\Leftarrow$" in the case of *open*-capabilities: Let $(\hat{I}, \hat{H})$ be compatible such that $(\hat{I}, \hat{H}) \models^{\mathrm{CF}} open^{\ell^p} n.P$ and let us show that we also have $\gamma^{\mathrm{CF}}(\hat{I}, \hat{H}) \models^{\mathrm{OC}} open^{\ell^p} n.P$. From Figure 10 it follows that $(\hat{I}, \hat{H}) \models^{\mathrm{CF}} P$ and by the induction hypothesis we have $\gamma^{\mathrm{CF}}(\hat{I}, \hat{H}) \models^{\mathrm{OC}} P$. Next consider $(\hat{I}', \hat{H}', \hat{A}')$ in $\gamma^{\mathrm{CF}}(\hat{I}, \hat{H})$ and ℓ^a, ℓ' such that

$$(\ell^a, \ell^p) \in \hat{I}' \ \wedge \ (\ell^a, \ell') \in \hat{I}' \ \wedge \ (\ell', \lfloor n \rfloor) \in \hat{H}'.$$

By definition of γ^{CF} it follows that $(\hat{I}', \hat{H}', \hat{A}')$ is compatible and that

$$(\ell^a, \ell^p) \in \hat{I} \ \wedge \ (\ell^a, \ell') \in \hat{I} \ \wedge \ (\ell', \lfloor n \rfloor) \in \hat{H}$$

so that by Figure 10 also $\{(\ell^a, \ell) | \, (\ell', \ell) \in \hat{I}\} \subseteq \hat{I}$. Clearly all the triples considered in Figure 7 for inclusion into $\gamma^{\mathrm{CF}}(\hat{I}, \hat{H})$ are compatible and at most add the pairs

$$\{(\ell^a, \ell) | \, (\ell', \ell) \in Z\} \subseteq \{(\ell^a, \ell) | \, (\ell', \ell) \in \hat{I}'\} \subseteq \{(\ell^a, \ell) | \, (\ell', \ell) \in \hat{I}\}$$

already known to be in $\hat{I}$. It follows that all the triples are elements of $\gamma^{\mathrm{CF}}(\hat{I}, \hat{H})$ and this concludes the proof that $\gamma^{\mathrm{CF}}(\hat{I}, \hat{H}) \models^{\mathrm{OC}} open^{\ell^p} n.P$.

Conversely, consider the proof of "$\Rightarrow$": Let $(\hat{I}, \hat{H})$ be compatible such that $\gamma^{\mathrm{CF}}(\hat{I}, \hat{H}) \models^{\mathrm{OC}} open^{\ell^p} n.P$ and show that $(\hat{I}, \hat{H}) \models^{\mathrm{CF}} in^{\ell^i} n.P$. From Figure 7 it follows that $\gamma^{\mathrm{CF}}(\hat{I}, \hat{H}) \models^{\mathrm{OC}} P$ and by the induction hypothesis we have $(\hat{I}, \hat{H}) \models^{\mathrm{CF}} P$. Next consider ℓ^a, ℓ' such that

$$(\ell^a, \ell^p) \in \hat{I} \ \wedge \ (\ell^a, \ell') \in \hat{I} \ \wedge \ (\ell', \lfloor n \rfloor) \in \hat{H}.$$

We already remarked that

$$(\hat{I}, \hat{H}, [\{\ell^a \mid (\ell^a, n) \in \hat{H}\} \mapsto \omega]) \in \gamma^{\mathrm{CF}}(\hat{I}, \hat{H}).$$

For this choice of element in $\gamma^{\mathrm{CF}}(\hat{I}, \hat{H})$ in Figure 7 we take:

$$\begin{aligned} Z &= \{(\ell', \ell) \in \hat{I} | \, \ell \in \textsf{\textbf{Lab}}\} \\ Y &= \emptyset \\ X &= \emptyset \\ \mathsf{c} &= \omega \\ V &= \emptyset \\ W &= \emptyset \end{aligned}$$

It then follows that

$$(((\hat{I} \setminus (Y \cup X)) \cup \{(\ell^a, \ell) | \, (\ell', \ell) \in Z\}) \setminus W, \hat{H} \setminus V, \hat{A}[\ell' \mapsto \mathsf{c}]) \in \gamma^{\mathrm{CF}}(\hat{I}, \hat{H}).$$

Given the definition of γ^{CF} it follows that

$$\{(\ell^a, \ell) | \, (\ell', \ell) \in \hat{I}\} = \{(\ell^a, \ell) | \, (\ell', \ell) \in Z\} \subseteq \hat{I}$$

and this concludes the proof that $(\hat{I}, \hat{H}) \models^{\mathrm{CF}} open^{\ell p} n.P$.

We next prove

(2) $\forall \ell \in \mathsf{Lab}^a : \;\; \alpha^{\mathrm{CF}}(\{\eta^{\mathrm{OC}}_\ell(P)\}) = \beta^{\mathrm{CF}}_\ell(P)$

by structural induction in P. For the induction step we use that

$$\begin{aligned} & \alpha^{\mathrm{CF}}(\{(\hat{I}_1, \hat{H}_1, \hat{A}_1) \uplus (\hat{I}_2, \hat{H}_2, \hat{A}_2)\}) \\ = \;\; & \alpha^{\mathrm{CF}}(\{(\hat{I}_1, \hat{H}_1, \hat{A}_1)\}) \sqcup \alpha^{\mathrm{CF}}(\{(\hat{I}_2, \hat{H}_2, \hat{A}_2)\}). \end{aligned}$$

This concludes the proof of Proposition 12. ■