ELSEVIER
Available online at www.sciencedirect.com
SCIENCE DIRECT®
Theoretical Computer Science 330 (2005) 475–499
Theoretical Computer Science
www.elsevier.com/locate/tcs

Elimination of spatial connectives in static spatial logics

Étienne Lozes*

LIP, ENS Lyon, 46, allée d'Italie, Lyon 69364, France

Abstract

The recent interest for specification on resources yields so-called *spatial logics*, that is specification languages offering new forms of reasoning: the local reasoning through the separation of the resource space into two disjoint subspaces, and the contextual reasoning through hypothetical extension of the resource space.

We consider two resource models and their related logics:

- The static ambient model, proposed as an abstraction of semistructured data (Proc. ESOP'01, Lecture Notes in Computer Science, vol. 2028, Springer, Berlin, 2001, pp. 1–22 (invited paper)) with the static ambient logic (SAL) that was proposed as a request language, both obtained by restricting the mobile ambient calculus (Proc. FOSSACS'98, Lecture Notes in Computer Science, vol. 1378, Springer, Berlin, 1998, pp. 140–155) and logic (Proc. POPL'00, ACM Press, New York, 2000, pp. 365–377) to their purely static aspects.
- The memory model and the assertion language of separation logic, both defined in Reynolds (Proc. LICS'02, 2002) for the purpose of the axiomatic semantic of imperative programs manipulating pointers.

We raise the questions of the expressiveness and the minimality of these logics. Our main contribution is a minimalisation technique we may apply for these two logics. We moreover show some restrictions of this technique for the extension $\mathrm{SAL}^{\forall}$ with universal quantification, and we establish the minimality of the adjunct-free fragment ($\mathrm{SAL}_{\mathrm{int}}$).

Keywords: Spatial logics; Separation logic; Mobile ambients; Minimality

* Tel.: +4 7272 8796.
E-mail address: elozes@ens-lyon.fr (E. Lozes).

0304-3975/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2004.10.006

1. Introduction

The mobile ambients calculus (MA) [7] is a proposal for a new paradigm in the field of concurrency models. Its originality is to set as data the notion of *location*, and as notion of computation the reconfiguration of the hierarchy of locations. The calculus has a spatial part expressing the topology of locations as a labelled unordered tree with binders, and a dynamic part describing the evolution of this topology. The basic connectives for the spatial part are 0, defining the empty tree, $a[P]$, defining the tree rooted at a with subtree P, $P|Q$ for the tree consisting of the two subtrees P and Q in parallel, and $(\nu n)P$ for the tree P in which the label (or name) n has been hidden. Leaving out from MA all capabilities, we get rid of the dynamics of the calculus, working with what we call *static ambients* (SA).

Type systems are commonly used to express basic requirements on programs. In the case of SA, the static ambient logic (SAL) [8] provides a very flexible descriptive framework. Seeing SAL as a request language, one may ask a structure P to match some specification $\mathcal{A}$, written

$$P \vDash \mathcal{A}\,.$$

The SAL approach is however much more intensional than it is the case for standard type systems. Indeed, the whole spatial structure of the calculus is reflected in the logic. For instance, the formula $n[\mathcal{A}]$ is satisfied by structures of the form $n[P]$ with $P \vDash \mathcal{A}$. Finally, AL includes *adjunct connectives* for every spatial construct. For instance, the *guarantee* operator

$$\mathcal{A} \triangleright \mathcal{B}$$

specifies that a process is able to satisfy $\mathcal{B}$ when it is extended by any process satisfying $\mathcal{A}$. SA, associated to SAL, has appeared to be an interesting model for *semistructured data* [6] such as XML documents, due to the underlying tree structure. Data are modelled by unordered labelled trees, where the binders may represent pointers [5], and the logic is used as the basis for a language for queries involving such data. For instance, the process of Fig. 1 represents a database containing the two authors Cardelli and Gordon with one copy of their paper about ambients stored at Cardelli's and linked to Gordon's. Query

$$\text{И} ptr.ptr \circledR \big(Cardelli[\top]|\top\big)$$

asks whether the database contains some author named Cardelli.

Separation logic (SL) [18] is a proposal for a new assertion language in Hoare's approach of imperative programs verification. Indeed, imperative programming languages manipulating pointers allow one to change the value a variable refers to without explicitly mentioning this variable. Such multiple accesses to data make the axiomatic semantics [14] of these programs difficult to handle using classical logic as an assertion language [17]. SL nicely handles the subtleties of pointer manipulation, providing two new connectives: a separative conjunction $P * Q$ asserting that P and Q hold in separate parts of the memory, and a separating implication $P \mathrel{-\!\!*} Q$ allowing one to introduce 'spatial hypotheses' about the memory. For instance, the judgement

$$\big\{(x \mapsto -) * \big((x \mapsto e) \mathrel{-\!\!*} \phi\big)\big\} x := e \big\{\phi\big\}$$

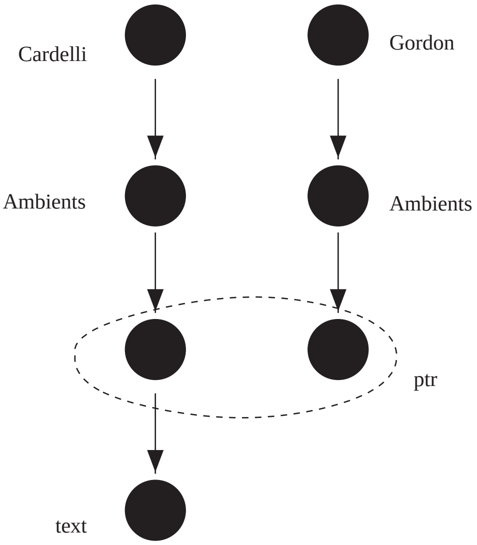

Fig. 1. $(\nu ptr)(Cardelli[Ambients[ptr[text[\mathbf{0}]]]]|Gordon[Ambients[ptr[\mathbf{0}]]])$.

is the transposition of the classical backward reasoning $\{\phi[e/x]\}x \ := \ e\{\phi\}$ in Hoare logic.

Both specification languages rely on classical logic reasoning extended by two non-standard operations: splitting of the resource space and separated assertions $(|, *)$ on each subspace, and extension of the resource space assuming some hypothesis $(\triangleright, -\!\!*)$. These two aspects are the main novelties of the so-called *spatial logics*. The interest of these connectives has been illustrated in several ways. For mobile ambients, it is known that the connective $\triangleright$ coupled with $\diamond$ can express the action modalities [19], persistence, and other strong properties [13]. For SL, the proof of an in-place reversal of a list turns out to require complex invariants in the standard classical logic, whereas it has a simple formulation in SL using $*$, as one of the many examples presented in [17].

Although spatial connectives evidently bring a real ease to the formulation of complex properties of the structures, their actual contribution to the expressiveness of the logic is not so clear. For instance, the formula $x \hookrightarrow \mathsf{nil} * y \hookrightarrow \mathsf{nil}$ expresses that both x and y points to nil, but from distinct locations, which can also be expressed as $x \hookrightarrow \mathsf{nil} \wedge y \hookrightarrow \mathsf{nil} \wedge x \neq y$ without requiring $*$; the formula $n[0]\triangleright n[0]$ tells that after extension of the structure adding $n[0]$, one exactly has $n[0]$, which means that the structure was initially empty, hence this formula is equivalent to 0. On the other hand, it has been established for the mobile ambient case, i.e. in a dynamic setting, that guarantee brings some extra expressive power [13].

This paper studies the contribution of spatial connectives in the expressiveness of static spatial logics. This question is important since spatial connectives introduce a lot of complication from the model-checking point of view. Indeed, separated conjunctions $*$ and $|$ forces to try all the splitting of the structure, which may be costly for wide structures. Even worse, the spatial implications $-\!\!*$ and $\triangleright$ considerably complicate the model-checking by introducing the need to seek a representative testing set [3,4], when it is not an undecidable problem [4,12]. The expressiveness of spatial connectives is also important from theoretical issues. For instance, the proof of an in-place reversal of a list is derivable, through heavy formulations, in classical Hoare logic as well, and the question is open whether SL can prove programs on which classical reasoning would fail.

Several kinds of quantification can be taken under consideration for our spatial logics:

- Absence of quantification, as it is the case for SL (in this work).
- Classical quantification ($\forall$, $\exists$), which defines the logic $\mathrm{SAL}^{\forall}$.
- Fresh quantification [11] ($\text{И}n.\mathcal{A}$), which is the way SAL handles name generation. This quantification is related to α conversion of bound names. It is complementary to the spatial connective $n®\mathcal{A}$ that forces the process to reveal a hidden name by calling it n.

We establish that the contribution of spatial connectives depends on the forms of quantification supported by the logic.

Indeed, in quantifier-free logics, adjuncts do not increase the expressiveness of the logic (Theorem 4.4). Neither does the separated conjunction ($*$) for SL, since it only expresses separation, so that SL assertions can be translated into a classical logic (Theorem 8.1). In a different way, | brings extra expressiveness to SAL, namely the power of counting, so it cannot be eliminated, and actually the adjunct-free fragment of SAL is minimal (Theorem 7.1). The proof of these elimination results goes through the intensive use of intensional partial equivalences on models; such equivalences are common for the study of the expressiveness of a logic (see [13,19] for spatial logic cases), but were also exploited for decidability issues in [3,4]. Two properties justify the encoding: a property we call *precompactness*, which expresses finiteness of behaviours, and the existence of *characteristic formulas* for the classes of partial intensional equivalence.

When classical quantifiers are taken under consideration, more complex properties can be expressed through adjuncts, and they cannot be taken out freely (Theorem 6.1). This difference of nature of the logic was already observed from the decidability aspect [3,4,10], which implied the absence of an effective adjuncts elimination. Our result shows that the adjuncts elimination is impossible even theoretically.

Finally, we establish the quite surprising result that adjuncts elimination is still possible in presence of fresh quantification (Theorem 5.3), essentially due to prenex forms for И (Proposition 5.2). This result underlines the fundamental difference between classical quantification and fresh quantification.

Related work: Apart from [16], this is, to our knowledge, the first results studying precisely the expressiveness and minimality of spatial logics. Other works about expressiveness only give some hints. A first result about the separation power of AL is presented in [19]. Other examples of expressive formulas for AL are shown in [13], such as formulas for persistence and finiteness.

A compilation result has been derived for a spatial logic for trees without quantification and private names [16]. In that work, the target logic includes some new features such as Presburger arithmetic, and the source logic includes a form of Kleene star.

The setting in which we obtain our encoding is rather different in the dynamic case (see [13]). There, the presence of adjuncts considerably increases the expressive power of the logic. For instance, $\triangleright$ allows one to construct formulas to characterise processes of the form

$\mathsf{open}\, n.P$, and, using the @ connective, we may define a formula to capture processes of the form $\mathsf{out}\, n.P$.

The use of a partial intensional equivalence and the notion of precompactness is original. Intensional bisimilarity plays an important role in the characterisation of the separation power of the logic [19]. Our proof suggests that it is also a powerful and meaningful concept for the study of expressiveness.

The presence of the $\triangleright$ connective in the logic is crucial with respect to decidability issues. The undecidability of the model-checking of SAL with classical quantification has been established in [10]. Quite unexpected decidability results for spatial logics with $\triangleright$ and without quantification were then established in [3] and [4]. These works are closely related to the present study; roughly, the decidability result of Calcagno et al. [3] relies on finiteness of *processes*, whereas our encoding exploits finiteness of *observations*. For this reason, our approach is more general and cuts out decidability issues. Actually, the undecidability of the model-checking problem for SAL has been recently established [12]. This last work studies many variations around SAL, derives decidability results with $\triangleright$ and И, and presents a prenex form result similar to ours.

Outline: We introduce SA, SAL and its adjunct-free fragment ($\mathrm{SAL}_{\mathrm{int}}$) in Section 2. We prove adjunct elimination for quantifier-free formulas in Section 4, based on the notion of intensional bisimilarity, discussed in Section 3. The general result for SAL is then established in Section 5, based on prenex forms. We discuss the adjunct elimination for $\mathrm{SAL}^{\forall}$ in Section 6, and show minimality of $\mathrm{SAL}_{\mathrm{int}}$ in Section 7; in Section 8, we introduce SL and a classical fragment of it (CL), which we prove to be as expressive as SL. Section 9 gives concluding remarks.

2. Background

In this section, we define the model of static ambients (SA) and its logic SAL. We also define the intensional fragment ($\mathrm{SAL}_{\mathrm{int}}$) of SA.

In all what follows, we assume an infinite set $\mathcal{N}$ of names, ranged over by n, m. Tree terms are defined by the following grammar:

$$P ::= P|P \,\big|\, n[P] \,\big|\, (\nu n)P \,\big|\, \mathbf{0}\,.$$

The set $\mathrm{fn}(P) \subset \mathcal{N}$ of free names of P is defined by saying that ν is the only binder on trees. We call *static ambients* tree terms quotiented by the smallest congruence $\equiv$ (called *structural congruence*) satisfying the axioms of Fig. 2. Formulas, ranged over with $\mathcal{A}$, $\mathcal{B}$, ..., are defined in Fig. 3 . These formulas form *the static ambient logic*, and we call *intensional fragment* the subset of the formulas not using the connectives $\triangleright$, @, and $\oslash$ (adjuncts). We note them, respectively, SAL and $\mathrm{SAL}_{\mathrm{int}}$.

We will say that $\mathcal{A}$ is *quantifier-free* if $\mathcal{A}$ does not contain any И quantification. The set of free names of a formula $\mathcal{A}$, written $\mathrm{fn}(\mathcal{A})$ is the set of names appearing in $\mathcal{A}$ that are not bound by a И quantification. $\mathcal{A}(n \leftrightarrow n')$ is the formula $\mathcal{A}$ in which names n and n' are swapped.

$$\begin{array}{ccc} P\mid\mathbf{0}\equiv P & (\nu n)\,\mathbf{0}\equiv\mathbf{0} & \\ \Big(P|Q\Big)|R\equiv P|\Big(Q|R) & (\nu n)\,m[P]\equiv m[(\nu n)P] & (n\neq m) \\ P\mid Q\equiv Q\mid P & (\nu n)P\mid Q\equiv(\nu n)\Big(P\mid Q) & (n\notin \mathrm{fn}(Q)) \end{array}$$

Fig. 2. Structural congruence on SA.

$$\begin{array}{lllll} \mathcal{A} & ::= \mathcal{A}\wedge\mathcal{A}\,\Big|\,\neg\mathcal{A}\,\Big|\,\text{И}n.\mathcal{A}\,\Big|\,0 & \Big|\,\mathcal{A}|\mathcal{A} & \Big|\,n[\mathcal{A}] & \Big|\,n\circledR\mathcal{A} \quad \text{(intensional fragment)} \\ & & \Big|\,\mathcal{A}\triangleright\mathcal{A} & \Big|\,\mathcal{A}@n & \Big|\,\mathcal{A}\oslash n \quad \text{(adjuncts)} \end{array}$$

Fig. 3. SAL and the intensional fragment $\mathrm{SAL}_{\mathrm{int}}$.

Definition 2.1 (*Satisfaction*). We define the relation $\vDash\ \subset (SA\times\mathrm{SAL})$ by induction on the formula as follows:

- $P\vDash\mathcal{A}_1\wedge\mathcal{A}_2$ if $P\vDash\mathcal{A}_1$ and $P\vDash\mathcal{A}_2$;
- $P\vDash\neg\mathcal{A}$ if $P\nvDash\mathcal{A}$;
- $P\vDash\text{И}n.\mathcal{A}$ if $\forall n'\in\mathcal{N}-(\mathrm{fn}(P)\cup\mathrm{fn}(\mathcal{A}))$, $P\vDash\mathcal{A}(n\leftrightarrow n')$;
- $P\vDash\mathcal{A}_1|\mathcal{A}_2$ if there is P_1, P_2 s.t. $P\equiv P_1|P_2$ and $P_i\vDash\mathcal{A}_i$ for $i=1,2$;
- $P\vDash 0$ if $P\equiv\mathbf{0}$;
- $P\vDash n[\mathcal{A}]$ if there is P' such that $P\equiv n[P']$ and $P'\vDash\mathcal{A}$;
- $P\vDash n\circledR\mathcal{A}$ if there is P' such that $P\equiv(\nu n)P'$ and $P'\vDash\mathcal{A}$;
- $P\vDash\mathcal{A}_1\triangleright\mathcal{A}_2$ if for all Q such that $Q\vDash\mathcal{A}_1$, $P|Q\vDash\mathcal{A}_2$;
- $P\vDash\mathcal{A}@n$ if $n[P]\vDash\mathcal{A}$;
- $P\vDash\mathcal{A}\oslash n$ if $(\nu n)P\vDash\mathcal{A}$.

We note $\mathcal{A}\dashv\vdash\mathcal{B}$ if for all $P\in SA$, $P\vDash\mathcal{A}$ iff $P\vDash\mathcal{B}$. A context is a formula containing a *hole*; if $\mathcal{C}$ is a context, $\mathcal{C}[\mathcal{A}]$ stands for the formula obtained by replacing the hole with $\mathcal{A}$ in $\mathcal{C}$.

Lemma 2.2. *For all $\mathcal{A}$, $\mathcal{B}$, and all context $\mathcal{C}$, if $\mathcal{A}\dashv\vdash\mathcal{B}$, then $\mathcal{C}[\mathcal{A}]\dashv\vdash\mathcal{C}[\mathcal{B}]$.*

Remark 2.1.

- The formula $\bot$, that no process satisfies, can be defined as $0\wedge\neg 0$. As e.g. in [8], other derived connectors include $\vee$, and $\blacktriangleright$: P satisfies $\mathcal{A}\blacktriangleright\mathcal{B}$ iff there exists Q satisfying $\mathcal{A}$ such that $P\mid Q$ satisfies $\mathcal{B}$.
- If $P\vDash\mathcal{A}$ and $P\equiv Q$, then $Q\vDash\mathcal{A}$. Moreover, $\vDash$ is *equivariant*, that is $P\vDash\mathcal{A}$ iff $P(n\leftrightarrow n')\vDash\mathcal{A}(n\leftrightarrow n')$ for any n, n'.
- For any P, there is a characteristic formula (for $\equiv$) $\mathcal{A}_P$, using the same tree representation, such that for all Q, $Q\vDash\mathcal{A}_P$ iff $Q\equiv P$. In particular, two static ambients are logically equivalent if and only if they are structurally congruent.

3. Intensional bisimilarity

In this section and the following, we will give a first illustration of our minimalisation method on the case of SAL and $\mathrm{SAL}_{\mathrm{int}}$. This minimalisation transforms a formula from a logic to the other; however, it does not proceed as a dictionary, that is we do not show that the connectives from the original logic are some syntactic sugar for some fixed construction in the target logic. The translation actually goes through the exploration of all behaviours a process may have with respect to a formula. Roughly, we translate a formula $\mathcal{A}$ into an exhaustive disjunction

$$\mathcal{A} \rightsquigarrow \bigvee_{C \in \mathsf{Behaviours}(\mathcal{A})} F_C$$

of all the behaviours that lead to the acceptance of $\mathcal{A}$.

The bottleneck of this embedding is to define what are these behaviours. By behaviours, we refer to equivalence classes of some observational equivalence. In this section, we will hence introduce a notion of partial observation over trees corresponding to logical testing. This model equivalence can be seen as the adapted *game* for this logic (in the sense of Ehrenfreucht–Fraïssé), or as the static *intensional bisimilarity* [19]. Observations are taken from the logic to which we want to reduce to, in this setting $\mathrm{SAL}_{\mathrm{int}}$. Each connective defines a simulation rule in a very natural way. Then we show that this observational equivalence is enough to ensure model equivalence with respect to the logic we want to minimalize, that is SAL (Proposition 3.4) in this setting. We then give a compact representation of the observational equivalence classes as some symbolic sets we call *signatures*.

We will assume in the remainder some fixed set $N \subset \mathcal{N}$.

3.1. Definition

We now introduce the intensional bisimilarity. Intuitively, $\simeq_{i,N}$ equates processes that may not be distinguished by logical tests involving at most i steps where the names used for the tests are picked in N.

Definition 3.1 (*Intensional bisimilarity*). We define the family $(\simeq_{i,N})_{i\in\mathbb{N}}$ of symmetric relations over SA by induction on i $: \simeq_{0,N} \overset{\text{def}}{=} \mathrm{SA} \times \mathrm{SA}$, and for any $i \geqslant 1$, $\simeq_{i,N}$ is the greatest relation such that if $P \simeq_{i,N} Q$, then the following conditions hold:

- if $P \equiv \mathbf{0}$ then $Q \equiv \mathbf{0}$;
- for all P_1, P_2, if $P \equiv P_1|P_2$ then there is Q_1, Q_2 such that $Q \equiv Q_1|Q_2$ with $P_\varepsilon \simeq_{i-1,N} Q_\varepsilon$, $\varepsilon = 1, 2$;
- for all $n \in N$ and for all P', if $P \equiv n[P']$, then there is Q' such that $Q \equiv n[Q']$ and $P' \simeq_{i-1,N} Q'$;
- for all $n \in N$ and for all P', if $P \equiv (\nu n)P'$, then there is Q' such that $Q \equiv (\nu n)Q'$ and $P' \simeq_{i-1,N} Q'$.

Lemma 3.2. *For all i, $\simeq_{i,N}$ is an equivalence relation.*

We shall write $\mathrm{SA}_{/\simeq_{i,N}}$ for the quotient of SA induced by $\simeq_{i,N}$: it will be ranged over by equivalence classes called C, C_1, C_2.

We may observe that the bisimilarities define a stratification of observations on terms, namely $\simeq_{i',N'} \subseteq \simeq_{i,N}$ for $i \leqslant i'$ and $N \subseteq N'$. This may be understood in a topological setting. Given a fixed N, we consider the ultrametric distance over models defined by

$d(P, Q) = 2^{-i}$ if i is the smallest natural for which $P \not\simeq_{i,N} Q$, and $d(P, Q) = 0$ if $P \simeq_{\omega,N} Q$ where $\simeq_{\omega,N} = \bigcap_{i \in \mathbb{N}} \simeq_{i,N}$. We call it the N-topology. It somehow captures the granularity of the logical observations with respect to their cost.

3.2. *Correction*

The key step in proving correction of the intensional bisimilarities with respect to the logic is their congruence properties for the connectives admitting an adjunct.

Lemma 3.3. *If $P \simeq_{i,N} Q$, then*:
- *for all R, $P|R \simeq_{i,N} Q|R$;*
- *for all $n \in \mathcal{N}$, $n[P] \simeq_{i,N} n[Q]$;*
- *for all $n \in N$, $(\nu n)P \simeq_{i,N} (\nu n)Q$.*

Proof. By induction on i. □

Note that the last point cannot be improved: consider $N = \{n\}$, $P \equiv m_1[\mathbf{0}]$, $Q \equiv m_2[\mathbf{0}]$. Then $P \simeq_{2,N} Q$, but $(\nu m_1)P \not\simeq_{2,N} (\nu m_1)Q$. For this reason, $\simeq_{i,N}$ is not a pure congruence.

We note $s(\mathcal{A})$ the size of $\mathcal{A}$, defined as the number of its connectives.

Proposition 3.4 (*Correction*). *For all P, Q, i such that $P \simeq_{i,N} Q$, for all quantifier free formula $\mathcal{A}$ such that $s(\mathcal{A}) \leqslant i$ and $\mathrm{fn}(\mathcal{A}) \subseteq N$,*

$$P \vDash \mathcal{A} \qquad \textit{iff} \qquad Q \vDash \mathcal{A}.$$

Proof. By induction on $\mathcal{A}$. For the adjuncts, apply the congruence properties of Lemma 3.3, and for the other connectives use the definition of $\simeq_{i,N}$. □

3.3. *Signature functions*

Definition 3.5 (*Signature*). For $i \geqslant 1$, we set
- $z_i^N(P) = 0$ if $P \equiv \mathbf{0}$, otherwise $\neg 0$;
- $p_i^N(P) = \{(C_1, C_2) \in (\mathrm{SA}_{/\simeq_{i-1,N}})^2 : P \equiv P_1|P_2 \text{ and} P_i \in C_i\}$;
- $a_i^N(P) = [n, C]$ if there is P' s.t. $P \equiv n[P']$, $n \in N$ and $P \in C$, $C \in \mathrm{SA}_{/\simeq_{i-1,N}}$, otherwise $a_i^N(P) = \mathsf{noobs}$, where noobs is a special constant;
- $r_i^N(P) = \{(n, C) \in N \times \mathrm{SA}_{/\simeq_{i-1,N}} : \exists P'.P \equiv (\nu n)P' \text{and} P' \in C\}$.

We call *signature of P at (i, N)* the fourtuple $\chi_i^N(P) = [z_i^N(P), p_i^N(P), a_i^N(P), r_i^N(P)]$.

The following lemma says that the signature actually collects all the information that may be obtained from the bisimilarity tests.

Lemma 3.6. *Assume $i \geqslant 1$. Then $P \simeq_{i,N} Q$ iff $\chi_i^N(P) = \chi_i^N(Q)$.*

4. Adjuncts elimination on quantifier-free formulas

In this section, we show that the quantifier free formulas of SAL have equivalent formulas in $\mathrm{SAL}_{\mathrm{int}}$. This result is then extended to all formulas of SAL in the next section.

In all what follows, we will assume N is a *finite* subset of $\mathcal{N}$; it is intended to bound the free names of the considered formulas. The encoding result is based on two key properties:

- Precompactness of the N-topology. In other words, when i, N are fixed, only a finite number of behaviours may be observed.
- Existence of intensional characteristic formulas for the classes of $\simeq_{i,N}$.

The first property basically says the following: if we fix some formula $\mathcal{A}$, then we may finitely list all the behaviours a process P may have with respect to $\mathcal{A}$. Then we may tag the ones corresponding to an acceptance and the ones corresponding to a rejection, and from the second property, we may express this by some formula in $\mathrm{SAL}_{\mathrm{int}}$.

Here is the proof with more details.

Lemma 4.1. *The codomain of χ_i^N is finite.*

Proof. We reason by induction on i. First notice that the codomain of χ_i^N is:

$$\begin{aligned}\mathbf{codom}\,\chi_i^N = &\{0, \neg 0\} \times \left(\mathrm{SA}_{/\simeq_{i-1,N}}\right)^2 \times \left(\{\mathsf{noobs}\} + N \times \mathrm{SA}_{/\simeq_{i-1,N}}\right)\\ &\times \mathcal{P}\left(N \times \mathrm{SA}_{/\simeq_{i-1,N}}\right)\end{aligned}$$

hence $\mathbf{codom}\,\chi_i^N$ is finite iff $\mathrm{SA}_{/\simeq_{i-1,N}}$ is finite too (here we use that N is finite). For $i = 1$, $\mathrm{SA}_{/\simeq_{0,N}} = \{\mathrm{SA}\}$, hence χ_0^N is finite, and so is $\mathbf{codom}\,\chi_1^N$. For $i \geqslant 2$, we have by induction $\mathbf{codom}\,\chi_{i-1}^N$ finite. By Lemma 3.6, there is an injection of $\mathrm{SA}_{/\simeq_{i-1,N}}$ into $\mathbf{codom}\,\chi_{i-1}^N$, so $\mathrm{SA}_{/\simeq_{i-1,N}}$ is finite, and so is $\mathbf{codom}\,\chi_i^N$. □

Here is an immediate consequence of Lemma 4.1:

Proposition 4.2 (*Precompactness*). *For all i, the number of classes of $\simeq_{i,N}$ is finite.*

These results roughly say that only a finite amount of information is needed to capture a given bisimilarity class. The next result makes it more precise: this information may be collected in a single formula of $\mathrm{SAL}_{\mathrm{int}}$.

Proposition 4.3 (*Characteristic formulas*). *For any $i \in \mathbb{N}$ and for any process P, there is a formula $\mathcal{A}_P^{i,N} \in \mathrm{SAL}_{\mathrm{int}}$ such that*

$$\forall Q \qquad Q \vDash \mathcal{A}_P^{i,N} \qquad \Leftrightarrow \qquad Q \simeq_{i,N} P\,.$$

Proof. By induction on i. For $i = 0$, we may take $\mathcal{A}_P^{i,N} = \top$. Then assume $i \geqslant 1$, and we have formulas $\mathcal{A}_P^{i-1,N}$ for all P. This obviously gives a characteristic formula $\mathcal{A}_C^{i-1,N}$ for any class C of $\mathrm{SA}_{/\simeq_{i-1,N}}$. Let us consider some fixed P. We set

$$\begin{aligned}
\mathcal{A}_z &= 0 \text{ if} z_i^N(P) = 0, \text{ otherwise}\neg 0;\\
\mathcal{A}_p &= \bigwedge_{(C_1,C_2)\in p_i^N(P)} \mathcal{A}_{C_1}^{i-1,N} | \mathcal{A}_{C_2}^{i-1,N} \wedge \neg \bigvee_{(C_1,C_2)\notin p_i^N(P)} \mathcal{A}_{C_1}^{i-1,N} | \mathcal{A}_{C_2}^{i-1,N};\\
\mathcal{A}_a &= \begin{cases} \bigwedge_{n\in N} \neg n[\top] \text{ if } a_i^N(P) = \mathsf{noobs},\\ n[\mathcal{A}_C^{i-1,N}] \text{ if } a_i^N(P) = [n, C]; \end{cases}\\
\mathcal{A}_r &= \bigwedge_{[n,C]\in r_i^N(P)} n \circledR \mathcal{A}_C^{i-1,N} \wedge \neg \bigvee_{[n,C]\notin r_i^N(P)} n \circledR \mathcal{A}_C^{i-1,N};\\
\mathcal{A}_P^{i,N} &= \mathcal{A}_z \wedge \mathcal{A}_p \wedge \mathcal{A}_a \wedge \mathcal{A}_r,
\end{aligned}$$

where the finiteness of the conjunctions and disjunctions is ensured by Lemma 4.1.

Then $Q \vDash \mathcal{A}_P^{i,N}$ iff $\chi_i^N(Q) = \chi_i^N(P)$, hence the result. □

The precompactness property says that if we bound the granularity of the observations, only finitely many distinct situations may occur. The characteristic formula property says that each of these situations is expressible in the intensional fragment. The idea of the encoding is then just to logically enumerate all these possible situations.

Theorem 4.4. *For all quantifier-free formula* $\mathcal{A} \in \mathrm{SAL}$, *there is a formula* $[\mathcal{A}] \in \mathrm{SAL}_{\mathrm{int}}$ *such that*

$$\mathcal{A} \dashv\vdash [\,\mathcal{A}\,].$$

Proof. We define $[\mathcal{A}]$ as follows:

$$[\,\mathcal{A}\,] \overset{\text{def}}{=} \bigvee \mathcal{A}_C^{i,N} \qquad \text{for } C \in \mathrm{SA}_{/\simeq_{i,N}}, C \vDash \mathcal{A}$$

for $i = s(\mathcal{A})$ and $N = \mathrm{fn}(\mathcal{A})$. The disjunction is finite by Proposition 4.2. $P \vDash [\,\mathcal{A}\,]$ iff there is Q such that $Q \vDash \mathcal{A}$ and $P \simeq_{i,N} Q$, that is, by Proposition 3.4, $P \vDash \mathcal{A}$. □

Effectiveness of the encoding: Due to its finiteness, the construction of our proof could seem to be effective. However, this cannot be the case due to an undecidability result for the model-checking problem on SAL [12]. This is quite surprising, since only an effective enumeration of the bisimilarity classes is missing to make the proof constructive. Moreover, such an enumeration exists for SA without name restriction, via testing sets as defined in [3]. This reveals an unexpected richness of SA compared to pure trees.

5. Adjuncts elimination and fresh quantifier

In this section, we establish the adjunct elimination for the full SAL. The result we already obtained for quantifier-free formulas easily extends to formulas in prenex forms. So our efforts will focus on establishing the existence of an equivalent formula in prenex form for any formula of SAL. Intuitively, prenex forms can be generated by pulling out the fresh quantifiers. We actually show how to swap the order between a quantifier and another

$$
\begin{array}{llcll}
(\wedge) & (\text{И}n.\mathcal{A}_1) \wedge \mathcal{A}_2 & \rightsquigarrow & \text{И}n.(\mathcal{A}_1 \wedge \mathcal{A}_2) & (n \notin \mathrm{fn}(\mathcal{A}_2)) \\
(\neg) & \neg \text{И}n.\mathcal{A}_1 & \rightsquigarrow & \text{И}n.\neg\mathcal{A}_1 & \\
(|) & (\text{И}n.\mathcal{A}_1)|\mathcal{A}_2 & \rightsquigarrow & \text{И}n.(\mathcal{A}_1|\mathcal{A}_2) & (n \notin \mathrm{fn}(\mathcal{A}_2)) \\
(\triangleright L) & (\text{И}n.\mathcal{A}_1) \triangleright \mathcal{A}_2 & \rightsquigarrow & \text{И}n.\Big(\big(n\circledR\top \wedge \mathcal{A}_1\big)\triangleright\mathcal{A}_2\Big) & (n \notin \mathrm{fn}(\mathcal{A}_2)) \\
(\triangleright R) & \mathcal{A}_1 \triangleright (\text{И}n.\mathcal{A}_2) & \rightsquigarrow & \text{И}n.\Big(\big(n\circledR\top \wedge A_1\big)\triangleright\mathcal{A}_2\ ) & (n \notin \mathrm{fn}(\mathcal{A}_1)) \\
(Amb) & m[\text{И}n.\mathcal{A}] & \rightsquigarrow & \text{И}n.m[\mathcal{A}] & (m \neq n) \\
(@) & (\text{И}n.\mathcal{A})@m & \rightsquigarrow & \text{И}n.(\mathcal{A}@m) & (m \neq n) \\
(\circledR) & m\circledR\text{И}n.\mathcal{A} & \rightsquigarrow & \text{И}n.m\circledR\mathcal{A} & (m \neq n) \\
(\oslash) & (\text{И}n.\mathcal{A}) \oslash m & \rightsquigarrow & \text{И}n.(\mathcal{A} \oslash m) & (m \neq n)
\end{array}
$$

Fig. 4. Term rewriting system for prenexation.

connective without changing the semantic. Except for the $\triangleright$ connective, this turns out to be quite natural.

We present our algorithm as a rewriting system in Fig. 4. The essential result is then

Proposition 5.1 (*Correction of* $\rightsquigarrow$). *The term rewriting system* $\rightsquigarrow$ *defined by the rules of Fig. 4 preserves the semantics: for any* $\mathcal{A}, \mathcal{B} \in \mathrm{SAL}$, *if* $\mathcal{A}\rightsquigarrow\mathcal{B}$, *then* $\mathcal{A}\dashv\vdash\mathcal{B}$.

Proof (sketched). We only detail the proof for rule $(\triangleright L)$.

$$
\begin{array}{rl}
& P\vDash(\text{И}n.\mathcal{A}_1)\triangleright\mathcal{A}_2 \\
\Leftrightarrow & \forall\, Q, \forall n' \notin \mathrm{fn}(\mathcal{A}_1) \cup \mathrm{fn}(Q) \cdot Q\vDash\mathcal{A}_1(n \leftrightarrow n') \Rightarrow P|Q\vDash\mathcal{A}_2 \\
\Leftrightarrow & \forall\, Q, \forall n' \notin \mathrm{fn}(\mathcal{A}_1\triangleright\mathcal{A}_2) \cup \mathrm{fn}(P|Q) \cdot Q\vDash\mathcal{A}_1(n \leftrightarrow n') \Rightarrow P|Q\vDash\mathcal{A}_2 \\
\Leftrightarrow & \forall\, Q, \forall n' \notin \mathrm{fn}(\mathcal{A}_1\triangleright\mathcal{A}_2) \cup \mathrm{fn}(P|Q) \cdot Q\vDash\mathcal{A}_1(n \leftrightarrow n') \Rightarrow P|Q\vDash\mathcal{A}_2(n \leftrightarrow n') \\
\Leftrightarrow & \forall\, n' \notin \mathrm{fn}(\mathcal{A}_1\triangleright\mathcal{A}_2) \cup \mathrm{fn}(P), \\
& \forall\, Q \cdot n' \notin \mathrm{fn}(Q) \Rightarrow Q\vDash\mathcal{A}_1(n \leftrightarrow n') \Rightarrow P|Q\vDash\mathcal{A}_2(n \leftrightarrow n') \\
\Leftrightarrow & P\vDash\text{И}n.\big(\mathcal{A}_1 \wedge n\circledR\top\big)\triangleright\mathcal{A}_2. \qquad \square
\end{array}
$$

Remark 5.1. Some of the rules above (such as (Amb), $(\neg)$, and a variant of $(|L)$) have already been presented in [9], under the form of equalities. The same result is independently developed in [12].

We say that a formula $\mathcal{A}$ is *well-formed* if every variable bound by И is distinct from all other (bound and free) variables in $\mathcal{A}$. For such formulas, the side conditions in $\rightsquigarrow$ are always satisfied.

It is easy to see that $\rightsquigarrow$ defines a terminating rewriting system, and that the normal forms of well-formed formulas are formulas in prenex form. Confluence holds modulo permutation of consecutive И quantifiers.

Proposition 5.2 (*Prenex forms*). *For any formula $\mathcal{A}$, there are $\tilde{n}$, $\mathcal{A}'$ such that $\mathcal{A}\dashv\vdash\text{И}\tilde{n}.\mathcal{A}'$ and $\mathcal{A}'$ is quantifier free.*

This result directly implies the following extension of Theorem 4.4:

Theorem 5.3 (*Adjunct elimination*). *For any formula $\mathcal{A} \in \mathrm{SAL}$, there is a formula $[\mathcal{A}] \in \mathrm{SAL}_{\mathrm{int}}$ such that*

$$\mathcal{A}\dashv\vdash[\,\mathcal{A}\,].$$

Proof. There is $\mathcal{A}'$ quantifier free and $\tilde{n}$ such that $\mathcal{A}\dashv\vdash\text{И}\tilde{n}.\mathcal{A}'$ by Proposition 5.2. Then by Lemma 2.2 and Theorem 4.4, we may write

$$\mathcal{A}\dashv\vdash\text{И}\tilde{n}.\mathcal{A}'\dashv\vdash\text{И}\tilde{n}.[\mathcal{A}']\,. \qquad \square$$

Example 5.2. We show an example to illustrate how $\mathrm{SAL}_{\mathrm{int}}$ formulas can capture non-trivial properties expressed using the adjuncts. Let

$$\mathcal{A} ::= \Big(Hm'.m'[\top]\blacktriangleright\big(Hn_1.n_1[0]|Hn_2.n_2[Hn_3.n_3[0]]\big)\Big) \oslash m@m,$$

where $Hn.\mathcal{A}$ (H being the hidden name quantifier [1]) stands for $\text{И}n.n\circledR\mathcal{A}$. The prenex form of $\mathcal{A}$ is

$$\text{И}m', n_1, n_2, n_3.\Big((m'\circledR\top \wedge .m'\circledR m'[\top])\blacktriangleright\big(n_1\circledR n_1[0]|n_2\circledR n_2[n_3\circledR.n_3[0]]\big)\Big)\oslash m@m$$

Then $P \vDash \mathcal{A}$ iff there is Q such that

$$(\nu m)\,m[P]|(\nu m')\,m'[Q] \equiv (\nu n_1)(\nu n_2)(\nu n_3)\big(n_1[\mathbf{0}]|n_2[n_3[\mathbf{0}]]\big).$$

The only solutions of this equation are $P \equiv \mathbf{0}$ or $P \equiv (\nu n_3)n_3[\mathbf{0}]$. In other words, $\mathcal{A}$ is equivalent to $\mathcal{B} = 0 \vee Hn_3.n_3[0]$.

6. Adjuncts elimination and classical quantifiers

In this section, we consider a variant of SAL. Instead of fresh quantified formulas, we consider name quantification of the form $\forall x.\mathcal{A}$ and $\exists x.\mathcal{A}$ with the natural semantics:

$$P \vDash \forall x.\mathcal{A} \qquad \text{if} \qquad \forall n \in \mathcal{N}.P \vDash \mathcal{A}\{^n/_x\}.$$

Let us note $\mathrm{SAL}_{\mathrm{int}}{}^{\forall}$ the intensional fragment with classical quantification. We ask the question of adjuncts elimination for extensions of this logic. The undecidability result of Charatonik and Talbot [10] implies that there is no effective adjunct elimination for $\mathrm{SAL}_{\mathrm{int}}{}^{\forall} + \{\triangleright\}$. We establish now a more precise result:

Theorem 6.1 (*Expressiveness of adjuncts in* $\mathrm{SAL}_{\mathrm{int}}{}^{\forall}$). $\mathrm{SAL}_{\mathrm{int}}{}^{\forall}+\{\triangleright\}$, $\mathrm{SAL}_{\mathrm{int}}{}^{\forall}+\{@\}$ *and* $\mathrm{SAL}_{\mathrm{int}}{}^{\forall} + \{\oslash\}$ *are strictly more expressive than* $\mathrm{SAL}_{\mathrm{int}}{}^{\forall}$.

The proof of this theorem is based on the following observation. In any of the extensions we consider, it is possible to define a formula $\mathcal{A}$ such that

$$P \models \mathcal{A} \qquad \text{iff} \qquad \sharp\,\mathrm{fn}(P) \leqslant 1. \tag{1}$$

For the $\triangleright$ and @ connectives, we may first encode the formula $n = m$ as $\big(n[\top] \wedge \neg m[\top]\big) \triangleright \bot$ and $(n[\top])@m$. Then (1) is satisfied by the formula

$$\exists x.\, \forall y.\, \big(\neg y ® \top\big) \;\rightarrow\; x = y.$$

For the $\oslash$ connective, there is a direct formula satisfying (1):

$$\exists x.\, \big(\forall y. y ® \top\big) \oslash x.$$

We are now interested in proving that such a property cannot be expressed in $\mathrm{SAL_{int}}^{\forall}$. Our approach consists in studying the stability of $\models$ with respect to substitutions. We actually find some particular processes P for which $P \models \mathcal{A}$ is equivalent to $P \models \mathcal{A}\{^n/_m\}$. From this, we deduce processes P such that $P \models \mathcal{A}$ implies $P\{^n/_m\} \models \mathcal{A}$. This last result shows that, on certain conditions, a formula may not observe the action of equating two names in a process, which is contradictory with counting the number of free names.

We call *thread context* a context $\mathcal{C}$ of the form

$$\mathcal{C}[\,P\,] \equiv (\nu\tilde{n})\, n_1[\ldots n_k[\,P\,]\ldots]$$

with $\tilde{n} \subseteq \{n_1, \ldots, n_k\}$. We note $n(\mathcal{C}) \overset{\mathrm{def}}{=} \{n_1, \ldots, n_k\}$ and $d(\mathcal{C}) \overset{\mathrm{def}}{=} k$. For a formula $\mathcal{A}$, we note $d(\mathcal{A})$ the number of $n[.]$ connectives in $\mathcal{A}$.

Lemma 6.2. *Let $\mathcal{A}$ be a formula of* $\mathrm{SAL_{int}}^{\forall}$*, and $\mathcal{C}$ a thread context such that $d(\mathcal{C}) > d(\mathcal{A})$. Let n, m be two names such that $\{n, m\} \cap n(\mathcal{C}) = \emptyset$, and*

$$P \overset{\mathrm{def}}{=} \mathcal{C}\big[\,n[\mathbf{0}] | m[\mathbf{0}]\,\big].$$

Then $P \models \mathcal{A}$ iff $P \models \mathcal{A}\{^n/_m\}$.

Proof. By induction on the size of $\mathcal{A}$:

- the cases $\mathcal{A} = \mathcal{A}_1 \wedge \mathcal{A}_2$, $\mathcal{A} = \neg\mathcal{A}_1$, and $\mathcal{A} = \mathbf{0}$ are trivial.
- $\mathcal{A} = \mathcal{A}_1 | \mathcal{A}_2$. Assume first $P \models \mathcal{A}$. Since $d(\mathcal{C}) \geqslant 1$, we may assume by symmetry that $\mathbf{0} \models \mathcal{A}_2$ and $P \models \mathcal{A}_1$. Then $P \models \mathcal{A}_1\{^n/_m\}$ by induction, and $P \models \mathcal{A}\{^n/_m\}$. The other direction is proved similarly.
- $\mathcal{A} = a[\mathcal{A}_1]$. Assume first $P \models \mathcal{A}$. Then $\mathcal{C} \equiv a[\mathcal{C}']$ and $P' \overset{\mathrm{def}}{=} \mathcal{C}'[n[\mathbf{0}]|m[\mathbf{0}]] \models \mathcal{A}_1$. By induction $P' \models \mathcal{A}_1\{^n/_m\}$. Since $\{n, m\} \cap n(\mathcal{C})$, $a \neq m$, so $\mathcal{A}\{^n/_m\} = a[\mathcal{A}_1\{^n/_m\}]$, and $P \models \mathcal{A}\{^n/_m\}$. Assume now $P \models \mathcal{A}\{^n/_m\}$. Let $b = a\{^n/_m\}$. Then $\mathcal{C} \equiv b[\mathcal{C}']$ and $P' \overset{\mathrm{def}}{=} \mathcal{C}'[n[\mathbf{0}]|m[\mathbf{0}]] \models \mathcal{A}_1\{^n/_m\}$. Then $b \in n(\mathcal{C})$, so $b \notin \{m, n\}$, and $b = a$. By induction $P' \models \mathcal{A}_1$, so $P \models b[\mathcal{A}_1] = \mathcal{A}$.
- $\mathcal{A} = a ® \mathcal{A}_1$. Assume first $P \models \mathcal{A}$. Then $\mathcal{C} \equiv (\nu a)\mathcal{C}'$ and $P' \overset{\mathrm{def}}{=} \mathcal{C}'[n[\mathbf{0}]|m[\mathbf{0}]] \models \mathcal{A}_1$. Since n, m are free in P, $a \neq m$ and $a \neq n$. So $\{n, m\} \cap n(\mathcal{C}') = \emptyset$, and by induction, $P' \models \mathcal{A}_1\{^n/_m\}$. $\mathcal{A}\{^n/_m\} = a ® \mathcal{A}_1\{^n/_m\}$, and $P \models \mathcal{A}\{^n/_m\}$. The other direction is proved similarly.

- $\mathcal{A} = \forall x.\mathcal{A}_1$. Assume first $P \vDash \mathcal{A}$. Let take $a \in \mathcal{N}$. Then $P \vDash \mathcal{A}_1\{^a/_x\}$, and by induction $P \vDash \mathcal{A}_1\{^a/_x\}\{^n/_m\}$. For $a \neq m$, this is also $P \vDash \mathcal{A}_1\{^n/_m\}ax$. For $a = m$, this requires a bit more. Consider that $P \vDash \mathcal{A}_1\{^n/_x\}$. Then $P \vDash \mathcal{A}_1\{^n/_x\}\{^n/_m\}$ by induction. But $\mathcal{A}_1\{^n/_x\}\{^n/_m\} = \big(\mathcal{A}_1\{^n/_m\}\{^m/_x\}\big)\{^n/_m\}$, so by induction $P \vDash \mathcal{A}_1\{^n/_m\}\{^m/_x\}$. Hence $P \vDash \mathcal{A}_1\{^n/_m\}\{^a/_x\}$ for all a, that is $P \vDash \forall x.\mathcal{A}_1\{^n/_m\} = \mathcal{A}\{^n/_m\}$.
  Assume now that $P \vDash \mathcal{A}\{^n/_m\}$. Let take $a \in \mathcal{N}$. Then $P \vDash \mathcal{A}_1\{^n/_m\}\{^a/_x\}$. If $a \neq m$, this is $P \vDash \mathcal{A}_1\{^a/_x\}\{^n/_m\}$, so by induction $P \vDash \mathcal{A}_1\{^a/\}x$. For $a = m$, consider that $P \vDash \mathcal{A}_1\{^n/_m\}\{^n/_x\}$, that is $P \vDash \mathcal{A}_1\{^m/_x\}\{^n/_m\}$, so by induction $P \vDash \mathcal{A}_1\{^m/_x\}$. Hence $P \vDash \mathcal{A}_1\{^a/_x\}$ for all a, that is $P \vDash \mathcal{A}$. □

Lemma 6.3. *Let $\mathcal{A}$ be a formula of* $\mathrm{SAL_{int}}^{\forall}$, *and $\mathcal{C}$ a thread context such that $d(\mathcal{C}) > d(\mathcal{A})$. Let n, m be two names such that $\{n, m\} \cap n(\mathcal{C}) = \emptyset$, and moreover $m \notin \mathrm{fn}(\mathcal{A})$. Let*

$$P_1 \overset{\text{def}}{=} \mathcal{C}\big[\, n[\mathbf{0}] | m[\mathbf{0}] \,\big] \qquad \textit{and} \qquad P_2 \overset{\text{def}}{=} \mathcal{C}[\, n[\mathbf{0}] | n[\mathbf{0}] \,]$$

If $P_1 \vDash \mathcal{A}$, then $P_2 \vDash \mathcal{A}$.

Proof. By induction on the size of $\mathcal{A}$:

- the cases $\mathcal{A} = \mathcal{A}_1 \wedge \mathcal{A}_2$, $\mathcal{A} = \mathcal{A}_1 \vee \mathcal{A}_2$, $\mathcal{A} = \mathbf{0}$ and $\mathcal{A} = \neg\mathbf{0}$ are trivial.
- $\mathcal{A} = \mathcal{A}_1 | \mathcal{A}_2$. Since $d(\mathcal{C}) \geqslant 1$, we may assume by symmetry that $\mathbf{0} \vDash \mathcal{A}_2$ and $P_1 \vDash \mathcal{A}_1$. Then $P_2 \vDash \mathcal{A}_1$ by induction, and $P_2 \vDash \mathcal{A}$
- $\mathcal{A} = \mathcal{A}_1 || \mathcal{A}_2$. Since $d(\mathcal{C}) \geqslant 1$, $P_1 \vDash \mathcal{A}_1 \wedge \mathcal{A}_2$, $\mathbf{0} \vDash \mathcal{A}_1 \wedge \mathcal{A}_2$. By induction, $P_2 \vDash \mathcal{A}_1 \wedge \mathcal{A}_2$, that is $P_2 \vDash \mathcal{A}$
- $\mathcal{A} = a[\mathcal{A}_1]$. Then $\mathcal{C} \equiv a[\mathcal{C}']$ and $\mathcal{C}'[n[\mathbf{0}]|m[\mathbf{0}]] \vDash \mathcal{A}_1$. By induction $\mathcal{C}'[n[\mathbf{0}]|n[\mathbf{0}]] \vDash \mathcal{A}_1$, that is $P_2 \vDash \mathcal{A}$.
- $\mathcal{A} = \neg a[\mathcal{A}_1]$. Then either $\mathcal{C}$ is not of the form $n[\mathcal{C}']$, and $P_2 \vDash \neg a[\mathcal{A}_1]$, or $C \equiv n[\mathcal{C}']$ but $\mathcal{C}'[n[\mathbf{0}]|m[\mathbf{0}]] \vDash \neg\mathcal{A}_1$. Then by induction $\mathcal{C}'[n[\mathbf{0}]|n[\mathbf{0}]] \vDash \neg\mathcal{A}_1$, that is $P_2 \nvDash a[\mathcal{A}_1]$.
- $\mathcal{A} = a \circledR \mathcal{A}_1$. Then $\mathcal{C} \equiv (\nu a)\mathcal{C}'$ and $\mathcal{C}'[n[\mathbf{0}]|m[\mathbf{0}]] \vDash \mathcal{A}_1$. Since n, m are free in P, $a \notin \{m, n\}$, so $n(\mathcal{C}') \cap \{m, n\} = \emptyset$. Then by induction, $\mathcal{C}'[n[\mathbf{0}]|n[\mathbf{0}]] \vDash \mathcal{A}_1$, and $P_2 \vDash \mathcal{A}$.
- $\mathcal{A} = \neg a \circledR \mathcal{A}_1$. Assume first that a is free in P_1. Then $a \neq m$ since $m \notin \mathrm{fn}(\mathcal{A})$ by hypothesis. So a is also free in P_2 and $P_2 \vDash \mathcal{A}$. Assume now a is fresh for P_1 (and P_2). Let $\mathcal{C}'$ be such that $\mathcal{C} \equiv (\nu a)\mathcal{C}'$. Then $\mathcal{C}'[n[\mathbf{0}]|n[\mathbf{0}]] \nvDash \mathcal{A}_1$, otherwise $\mathcal{C}'[n[\mathbf{0}]|m[\mathbf{0}]] \vDash \mathcal{A}_1$ and $P \vDash \mathcal{A}$. So $P_2 \nvDash a \circledR \mathcal{A}_1$.
- $\mathcal{A} = \forall x.\mathcal{A}_1$. Let take $a \in \mathcal{N}$. Then $P_1 \vDash \mathcal{A}_1\{^a/_x\}$, and by induction $P_2 \vDash \mathcal{A}_1\{^a/_x\}$ for $a \neq m$. Let take some fresh m'. By equivariance, $P_1(m \leftrightarrow m') \vDash \forall x.\mathcal{A}_1$, so $P_1(m \leftrightarrow m') \vDash \mathcal{A}_1\{^m/_x\}$. Applying induction on P_1 and $\mathcal{A}_1\{^m/_x\}$ for m' instead of m, we have $P_2 \vDash \mathcal{A}_1\{^m/_x\}$. Hence $P \vDash \mathcal{A}_1\{^a/_x\}$ for all a, that is $P_2 \vDash \forall x.\mathcal{A}_1$.
- $\mathcal{A} = \exists x.\mathcal{A}_1$. Let $a \in \mathcal{N}$ be such that $P_1 \vDash \mathcal{A}_1\{^a/_x\}$. If $a \neq m$, then we may apply induction on $\mathcal{A}_1\{^a/_x\}$, and $P_2 \vDash \mathcal{A}_2\{^a/_x\}$, that is $P_2 \vDash \mathcal{A}$. Otherwise $P_1 \vDash \mathcal{A}_1\{^m/_x\}$. By Lemma 6.2, $P_1 \vDash \mathcal{A}_1\{^m/_x\}\{^n/_m\} = \mathcal{A}_1\{^n/_x\}\{^n/_m\}$, and again $P_1 \vDash \mathcal{A}_1\{^n/_x\}$. Then by induction, $P_2 \vDash \mathcal{A}_1\{^n/_x\}$, that is $P_2 \vDash \mathcal{A}$.

This last result implies the desired property about $\mathrm{SAL_{int}}^{\forall}$: □

Proposition 6.4. *There is no formula in* $\mathrm{SAL_{int}}^{\forall}$ *that satisfies* (1).

Proof. Let us assume by absurd we have some $\mathcal{A}$ such that

$$P \vDash \mathcal{A} \qquad \text{iff} \qquad \sharp\,\mathrm{fn}(P) \leqslant 1.$$

Then let $\mathcal{C}$ be the thread context of the form $(\nu a)a[\dots a[.]\dots]$, and $d(\mathcal{C}) = d(\mathcal{A}) + 1$. Let m, n be two fresh names. Then $\mathcal{C}[n[\mathbf{0}]|m[\mathbf{0}]] \vDash \neg\mathcal{A}$ by definition of $\mathcal{A}$, so by Lemma 6.3, $\mathcal{C}[n[\mathbf{0}]|n[\mathbf{0}]] \vDash \neg\mathcal{A}$. Moreover, by definition of $\mathcal{A}$, $\mathcal{C}[n[\mathbf{0}]|n[\mathbf{0}]] \vDash \mathcal{A}$, so the contradiction. □

7. Minimality of $\mathrm{SAL}_{\mathrm{int}}$

In this section, we show minimality w.r.t. expressive power of $\mathrm{SAL}_{\mathrm{int}}$.

Theorem 7.1 (*Minimality*). $\mathrm{SAL}_{\mathrm{int}}$ *is a minimal logic, that is all fragments of* $\mathrm{SAL}_{\mathrm{int}}$ *are less expressive*.

This result is the consequence of several technical lemmas for each connective. We may distinguish two forms of contribution to the expressiveness of the logic. We will say that a connective κ is *expressive* when there is a property expressed by a formula containing κ that cannot be expressed otherwise. As a consequence, this connective must belong to any minimal fragment. We will also say that a connective κ is *separative* when there exists two models P_1, P_2 and a formula containing κ satisfied by P_1 but not P_2, such that all κ-free formulas equally satisfy P_1 and P_2. Separative connectives are expressive as well, but in a deeper way: removing them, one reduces the separation power of the logic. For $\mathrm{SAL}_{\mathrm{int}}$, we will now establish the following classification:

- connectives $.|.$, $n\circledR.$, and $n[.]$ are separative,
- connectives 0, $\wedge$, $\neg$, И are expressive but not separative.

In particular, $\mathrm{SAL}_{\mathrm{int}}$ is minimal in terms of expressiveness, but as far as separation power is concerned, the minimal fragment is $\mathrm{SAL}_{\mathrm{int}} - \{\text{И}, \neg, \wedge, 0\}$, since for this fragment logical equivalence coincides with intensional bisimilarity.

Notice that we do not show that $\mathrm{SAL}_{\mathrm{int}}$ is the *unique* minimal fragment of SAL. This is far from being obvious.

Example 7.1. The fragment $\mathrm{SAL} - \{\wedge\}$ is surprisingly quite expressive, as the formula

$$\neg\text{И}n.n\circledR\neg n\circledR\big(\text{И}m_1.m_1\circledR\text{И}m_2.m_2\circledR m_1[m_2[0]]\big) \oslash n_1 \oslash n_2$$

shows. This formula is equivalent to $n_1[n_2[0]] \vee n_2[n_1[0]]$, and hence the proof of expressiveness of $\wedge$ (see below) must be carried out in a different way. We do not know the exact expressiveness of this fragment, one could think that it captures any finite set of processes. The interested reader may want to look for a formula for $n_1[0] \vee n_2[n_2[0]]$ in this fragment.

7.1. Separative connectives

We establish now that the connectives $.|.$, $n\circledR.$, and $n[.]$ are separative. Intuitively, $|$ carries the ability of $\mathrm{SAL}_{\mathrm{int}}$ to count, so without this connective it will not be possible to distinguish

$n[\mathbf{0}]|n[\mathbf{0}]$ from $n[\mathbf{0}]|n[\mathbf{0}]|n[\mathbf{0}]$; in the same way, $n[.]$ is necessary to separate $n_1[n_2[\mathbf{0}]]$ from $n_2[n_1[\mathbf{0}]]$, and n®. is the only way of specifying properties of hidden names, so it must be required to distinguish $(\nu n)n[\mathbf{0}]$ and $(\nu n)n[n[\mathbf{0}]]$.

Lemma 7.2. *If $\mathcal{A} \in \mathrm{SAL}_{\mathrm{int}} - \{|\}$, then $P_1 = n[\mathbf{0}]|n[\mathbf{0}] \vDash \mathcal{A}$ iff $P_2 = n[\mathbf{0}]|n[\mathbf{0}]|n[\mathbf{0}] \vDash \mathcal{A}$.*

Proof. By absurd, suppose there exists a formula $\mathcal{A}$ telling apart P_1 from P_2, take a minimal such $\mathcal{A}$, and reason by case analysis on $\mathcal{A}$.

- The cases $\mathcal{A} = \mathcal{A}_1 \wedge \mathcal{A}_2$, $\mathcal{A} = \neg\mathcal{A}_1$ and $\mathcal{A} = И m \mathcal{A}_1$ are straightforward.
- If $\mathcal{A} = 0$, then none of P_1, P_2 does satisfy $\mathcal{A}$.
- $\mathcal{A} = m®\mathcal{A}_1$: if $m = n$, then none of those processes do satisfy $\mathcal{A}$, otherwise the process satisfying $\mathcal{A}$ does satisfy $\mathcal{A}_1$, and $\mathcal{A}_1$ is a smaller separating formula.
- $\mathcal{A} = m[\mathcal{A}_1]$: none of the two processes do satisfy $\mathcal{A}$. □

Lemma 7.3. *If $\mathcal{A} \in \mathrm{SAL}_{\mathrm{int}} - \{n[.]\}$, then for any names n_1, n_2, we set $P_1 = n_1[n_2[\mathbf{0}]]$ and $P_2 = n_2[n_1[\mathbf{0}]]$. Then $P_1 \vDash \mathcal{A}$ iff $P_2 \vDash \mathcal{A}$.*

Proof. As above, by absurd and case analysis on a minimal $\mathcal{A}$:

- The cases $\mathcal{A} = \mathcal{A}_1 \wedge \mathcal{A}_2$, $\mathcal{A} = \neg\mathcal{A}_1$ and $\mathcal{A} = И m \mathcal{A}_1$ are straightforward.
- If $\mathcal{A} = 0$, then none of P_1, P_2 do satisfy $\mathcal{A}$.
- $\mathcal{A} = \mathcal{A}_1|\mathcal{A}_2$. We may assume by symmetry that $P_1 \vDash \mathcal{A}$. Also by symmetry, we may assume $P_1 \vDash \mathcal{A}_1$ and $\mathbf{0} \vDash \mathcal{A}_2$. If $P_2 \nvDash \mathcal{A}$, then $\mathcal{A}_1$ separates P_1 from P_2 and is a smaller formula: contradiction.
- $\mathcal{A} = m®\mathcal{A}_1$: if $m \in \{n_1, n_2\}$, then none of the two processes do satisfy $\mathcal{A}$, otherwise the process satisfying $\mathcal{A}$ also satisfies $\mathcal{A}_1$, and $\mathcal{A}_1$ is a smaller separating formula. □

Lemma 7.4. *Assume $\mathcal{A} \in \mathrm{SAL}_{\mathrm{int}} - \{n[.]\}$, We set $P_1 = (\nu n)n[n[\mathbf{0}]]$ and $P_2 = (\nu n)n[\mathbf{0}]$. Then $P_1 \vDash \mathcal{A}$ iff $P_2 \vDash \mathcal{A}$.*

Proof. Again, by absurd and case analysis on a minimal $\mathcal{A}$:

- The cases $\mathcal{A} = \mathcal{A}_1 \wedge \mathcal{A}_2$, $\mathcal{A} = \neg\mathcal{A}_1$ and $\mathcal{A} = И m \mathcal{A}_1$ are straightforward.
- If $\mathcal{A} = 0$, then none of P_1, P_2 do satisfy $\mathcal{A}$.
- $\mathcal{A} = \mathcal{A}_1|\mathcal{A}_2$. We may assume by symmetry that $P_1 \vDash \mathcal{A}$. Also by symmetry, we may assume $P_1 \vDash \mathcal{A}_1$ and $\mathbf{0} \vDash \mathcal{A}_2$. If $P_2 \nvDash \mathcal{A}$, then $\mathcal{A}_1$ separates P_1 from P_2 and is a smaller formula: contradiction.
- $\mathcal{A} = m[\mathcal{A}_1]$: none of P_1, P_2 do satisfy $\mathcal{A}$. □

7.2. Expressive connectives

We show that the connectives $\wedge, \neg, И, 0$ are expressive. Expressiveness proofs are more subtle than in the separability cases, since the loss of expressiveness is less sensitive. The scheme of the proof that the connective κ is expressive is to find a property (cardinality, stability by substitution, truncation, etc.) common to all set of models corresponding to any formula without κ, and a formula with κ whose set of models does not have this property.

7.2.1. $\wedge$ is expressive

By duality, $\wedge$ expresses disjunction; we will show that the intensional logic may not express the disjunction present in the formula $n_1[n_2[0]] \vee n_2[n_1[0]]$ without the $\wedge$ connective.

Remark 7.2. The $\wedge$ connective is probably the connective whose expressiveness is the most difficult to characterise. It would be even more difficult if one had to take into account adjuncts. As shown in Example 7.1, we may express the formula $n_1[n_2[0]] \vee n_2[n_1[0]]$ in $\mathrm{SAL} - \{\wedge\}$ using adjuncts.

We note $\mathcal{P}_2(\mathcal{N}) = \{\{n_1, n_2\} : n_1 \neq n_2\}$. We note $K_n = \{\{n, m\} : m \neq n\}$. We say that $K \subseteq \mathcal{P}_2(\mathcal{N})$ is cofinite if there is $N \subseteq \mathcal{N}$, N finite, such that for all $n_1, n_2 \notin N$, if $n_1 \neq n_2$ then $\{n_1, n_2\} \in K$. We may remark that K_1, K_2 are cofinite iff $K_1 \cap K_2$ is cofinite, and K is cofinite iff $K - K_n$ is cofinite.

Lemma 7.5. *Assume $\mathcal{A}$ is a formula of* $\mathrm{SAL_{int}} - \{\wedge\}$ *such that* $\mathbf{0} \not\models \mathcal{A}$. *We set*

$$K_{\mathcal{A}} \overset{\text{def}}{=} \left\{\{n_1, n_2\} : n_1 \neq n_2, n_1[n_2[\mathbf{0}]] \models \mathcal{A} \text{ and } n_2[n_1[\mathbf{0}]] \models \mathcal{A}\right\}.$$

Then either $K_{\mathcal{A}} = \emptyset$ or $K_{\mathcal{A}}$ is cofinite.

Proof. By induction on $\mathcal{A}$:
- $\mathcal{A} = \text{И}n.\mathcal{A}_1$. Then $\mathbf{0} \not\models \mathcal{A}_1$, and for any n_1, n_2 s.t. $n_1 \neq n$,$n_2 \neq n$ and $n_1 \neq n_2$, $\{n_1, n_2\} \in K_{\mathcal{A}_1}$ iff $\{n_1, n_2\} \in K_{\mathcal{A}_1}$. That is $K_{\mathcal{A}} - K_n = K_{\mathcal{A}_1} - K_n$.
- $\mathcal{A} = 0$: $\mathbf{0} \models \mathcal{A}$.
- $\mathcal{A} = \neg 0$: then $K_A = \mathcal{P}_2$.
- $\mathcal{A} = \mathcal{A}_1 | \mathcal{A}_2$: since $\mathbf{0} \not\models \mathcal{A}$, we may assume by symmetry that $\mathbf{0} \not\models \mathcal{A}_1$. If also $\mathbf{0} \not\models \mathcal{A}_2$, then $K_{\mathcal{A}} = \emptyset$. Otherwise, $K_{\mathcal{A}} = K_{\mathcal{A}_1}$.
- $\mathcal{A} = \mathcal{A}_1 || \mathcal{A}_2$: since $\mathbf{0} \not\models \mathcal{A}$, $\mathbf{0} \not\models \mathcal{A}_1$ and $\mathbf{0} \not\models \mathcal{A}_2$. then $K_{\mathcal{A}} = K_{\mathcal{A}_1} \cap K_{\mathcal{A}_2}$.
- $\mathcal{A} = n[\mathcal{A}_1]$: then $K_{\mathcal{A}} = \emptyset$.
- $\mathcal{A} = \neg n[\mathcal{A}_1]$: then $\mathcal{P}_2(\mathcal{N}) - K_n \subseteq K_{\mathcal{A}}$, so $K_{\mathcal{A}}$ is cofinite.
- $\mathcal{A} = n \circledR \mathcal{A}_1$: then $\mathbf{0} \not\models \mathcal{A}_1$, and $K_{\mathcal{A}} - K_n = K_{\mathcal{A}_1} - K_n$.
- $\mathcal{A} = \neg n \circledR \mathcal{A}_1$: then $\mathbf{0} \not\models \mathcal{A}_1$, and $K_{\mathcal{A}} - K_n = K_{\neg\mathcal{A}_1} - K_n$. □

Lemma 7.6. *Let n_1, n_2 be two distinct names. Then there is no formula $\mathcal{A} \in$* $\mathrm{SAL_{int}} - \{\wedge\}$ *equivalent to $n_1[n_2[0]] \vee n_2[n_1[0]]$.*

Proof. By absurd: if there is such a formula $\mathcal{A}$, then $\mathbf{0} \not\models \mathcal{A}$. Then by Lemma 7.5 $\sharp K_{\mathcal{A}} \neq 1$, and the contradiction. □

7.2.2. $\neg$ is expressive

$\neg$ enriches the expressive power in several ways; here we consider the property that the name n occurs free, expressed by $\neg n \circledR \top$, and show that negation is necessary to express it. To prove this, we remark that for a formula $\mathcal{A}$ without negation, there is a height h such that for all P, if $P \models \mathcal{A}$ then so does the truncation of P at height h, so we may find a contradiction by considering a process having an occurrence of n deep enough.

Definition 7.7. We define the truncation at height $h \in \mathbb{N}$ as $t_0(P) = \mathbf{0}$, and

$$t_h\big((\nu\tilde{n})(n_1[P_1]|\dots|n_r[P_r])\big) = (\nu\tilde{n})(n_1[t_{h-1}(P_1)]|\dots|n_r[t_{h-1}(P_r)]).$$

Note that $\mathrm{fn}(t_h(P)) \subseteq \mathrm{fn}(P)$.

Lemma 7.8. *If $\mathcal{A}$ is a formula without $\neg$, $s(\mathcal{A}) \leqslant h$ and $P \vDash \mathcal{A}$, then $t_h(P) \vDash \mathcal{A}$.*

Proof. By induction on $\mathcal{A}$:

- $\mathcal{A} = \mathcal{A}_1 \wedge \mathcal{A}_2$: then by induction $t_h(P) \vDash \mathcal{A}_1$, $t_h(P) \vDash \mathcal{A}_2$, so $t_h(P) \vDash \mathcal{A}_1 \wedge \mathcal{A}_2$.
- $\mathcal{A} = \text{И} n.\mathcal{A}_1$: then there is $n' \notin \mathrm{fn}(P)$ s.t. $P \vDash \mathcal{A}_1(n \leftrightarrow n')$. By induction $t_h(P) \vDash \mathcal{A}_1$ $(n \leftrightarrow n')$, $n' \notin \mathrm{fn}(t_h(P))$, so $t_h(P) \vDash \text{И} n.\mathcal{A}_1$.
- $\mathcal{A} = 0$: then $t_h(P) \equiv P \equiv \mathbf{0}$
- $\mathcal{A} = \mathcal{A}_1 | \mathcal{A}_2$: then $P \equiv P_1 | P_2$ with $P_\varepsilon \vDash \mathcal{A}_\varepsilon$, and by induction $t_h(P_\varepsilon) \vDash \mathcal{A}_\varepsilon$, so $t_h(P) \vDash \mathcal{A}$.
- $\mathcal{A} = n[\mathcal{A}_1]$: then $P \equiv n[P_1]$ and $P_1 \vDash \mathcal{A}_1$. By induction, $t_{h-1}(P_1) \vDash \mathcal{A}_1$, and so $t_h(P) \vDash \mathcal{A}$.
- $\mathcal{A} = n \circledR \mathcal{A}_1$: then $P \equiv (\nu n)P_1$ with $P_1 \vDash \mathcal{A}_1$. Then by induction $t_h(P_1) \vDash \mathcal{A}_1$, so $t_h(P) \vDash \mathcal{A}$. □

Lemma 7.9. *There is no formula $\mathcal{A} \in \mathrm{SAL_{int}} - \{\neg\}$ equivalent to $\neg n \circledR \bot$.*

Proof. Suppose $\mathcal{A}$ exists, and take $h = s(A)$. We note $P \equiv m[m[\dots m[\mathbf{0}]\dots]]$ and $Q \equiv m[m[\dots m[n[\mathbf{0}]]\dots]]$ a nesting of h ambients m, for some $m \neq n$. Then $Q \vDash \mathcal{A}$, $P \nvDash \mathcal{A}$, and $P \equiv t_h(Q)$, which contradicts Lemma 7.8. □

7.2.3. И is expressive

И is very useful to deal with an hidden name without making any hypothesis on the free names of processes (which revelation taken alone would do). Here we consider the property of having at least one hidden name, that is the model is congruent to $(\nu n)P'$ with $n \in \mathrm{fn}(P')$. This is expressed by the formula $\text{И} n.n \circledR \neg n \circledR \top$. For $N = \{n_1, \dots n_r\}$ we consider $P_N^n = n[n_1[\mathbf{0}]|\dots|n_r[\mathbf{0}]]$ for some $n \notin N$.

Lemma 7.10. *Assume some finite set of names N and a quantifier free formula $\mathcal{A}$ such that $\mathrm{fn}(\mathcal{A}) \subset N$, and $n \notin N$. Then*

$$P_N^n \vDash \mathcal{A} \qquad \text{iff} \qquad (\nu n)P_N^n \vDash \mathcal{A}$$

Proof. By induction on $\mathcal{A}$:

- the cases $\mathcal{A} = \mathcal{A}_1 \wedge \mathcal{A}_2$, and $\mathcal{A} = \neg \mathcal{A}_1$, are straightforward.
- if $\mathcal{A} = 0$: then none of the two processes satisfies $\mathcal{A}$.
- if $\mathcal{A} = \mathcal{A}_1 | \mathcal{A}_2$. Assume first that $P_N^n \vDash \mathcal{A}$. By symmetry, we may assume $P_N^n \vDash \mathcal{A}_1$ and $\mathbf{0} \vDash \mathcal{A}_2$. So $(\nu n)P_N^n \vDash \mathcal{A}_1$ by induction, and $(\nu n)P_N^n \vDash \mathcal{A}$. If we assume $(\nu n)P_N^n \vDash \mathcal{A}$, we may do the same reasoning.
- $\mathcal{A} = m[\mathcal{A}_1]$: none of P_N^n, $(\nu n)P_N^n$ does satisfy $\mathcal{A}$.
- $\mathcal{A} = m \circledR \mathcal{A}_1$: then $m \in \mathrm{fn}(\mathcal{A}) \subseteq N$, hence none of P_N^n, $(\nu n)P_N^n$ does satisfy $\mathcal{A}$.

Lemma 7.11. There is no formula $\mathcal{A} \in \mathrm{SAL}_{\mathrm{int}} - \{\text{И}\}$ equivalent to $\text{И}n.n\circledR n\circledR\bot$.

Proof. By absurd, let $\mathcal{A}$ be such a quantifier free formula, and $\{n_1, \ldots, n_r\} = \mathrm{fn}(\mathcal{A})$. Then $P_N^n \not\models \mathcal{A}$, so $(\nu n)P \not\models \mathcal{A}$, by Lemma 7.10, and the contradiction. □

7.2.4. 0 *is expressive*

Here we assume we take $\top$ instead of 0 as a primitive formula. Then 0 is not expressible. For this, we remark that for any $\mathcal{A}$ without 0 and for $n \notin \mathrm{fn}(A)$, $\mathbf{0} \models \mathcal{A}$ iff $n[\mathbf{0}] \models \mathcal{A}$.

Lemma 7.12. *Let* $\mathcal{A}$ *be a formula without* 0, *and* $n \notin \mathrm{fn}(A)$. *Then*

$$\mathbf{0} \models \mathcal{A} \quad \textit{iff} \;\; n[\mathbf{0}] \models \mathcal{A}$$

Proof. We reason by induction on $\mathcal{A}$

- $\mathcal{A} = \top$, $\mathcal{A} = \mathcal{A}_1 \wedge \mathcal{A}_2$, $\mathcal{A} = \neg\mathcal{A}_1$: straightforward.
- $\mathcal{A} = \text{И}m.\mathcal{A}_1$: We assume without loss of generality $m \neq n$. If $\mathbf{0} \models \text{И}m.\mathcal{A}_1$, then $\mathbf{0} \models \mathcal{A}_1$. $n[\mathbf{0}] \models \mathcal{A}_1$ by induction, so $n[\mathbf{0}] \models \text{И}n.\mathcal{A}_1$. Conversely, if $n[\mathbf{0}] \models \text{И}m.\mathcal{A}_1$, then $n[\mathbf{0}] \models \mathcal{A}_1$, so $\mathbf{0} \models \mathcal{A}_1$ by induction, and then $\mathbf{0} \models \text{И}n.\mathcal{A}_1$.
- if $\mathcal{A} = \mathcal{A}_1|\mathcal{A}_2$. Assume first that $\mathbf{0} \models \mathcal{A}_1|\mathcal{A}_2$. Then $\mathbf{0} \models \mathcal{A}_1 \wedge \mathcal{A}_2$, hence by induction $n[\mathbf{0}] \models \mathcal{A}_1$, and $n[\mathbf{0}] \models \mathcal{A}_1|\mathcal{A}_2$. If $\mathbf{0} \not\models \mathcal{A}_1|\mathcal{A}_2$, then we may assume by symmetry that $\mathbf{0} \not\models \mathcal{A}_1$. Assume by absurd that $n[\mathbf{0}] \models \mathcal{A}_1|\mathcal{A}_2$. Then $n[\mathbf{0}] \models \mathcal{A}_1$ and $\mathbf{0} \models \mathcal{A}_2$. By induction $\mathbf{0} \models \mathcal{A}_1$ and the contradiction.
- if $\mathcal{A} = m[\mathcal{A}_1]$. Then $m \neq n$ by hypothesis, and both $\mathbf{0} \not\models \mathcal{A}$ and $n[\mathbf{0}] \not\models \mathcal{A}$.
- if $\mathcal{A} = m\circledR\mathcal{A}_1$, $m \neq n$ by hypothesis. If $\mathbf{0} \models \mathcal{A}$, then $\mathbf{0} \models \mathcal{A}_1$, and by induction $n[\mathbf{0}] \models \mathcal{A}_1$ and $n[\mathbf{0}] \models \mathcal{A}$. Conversely, if $n[\mathbf{0}] \models \mathcal{A}$, then $n[\mathbf{0}] \models \mathcal{A}_1$, and $\mathbf{0} \models \mathcal{A}_1$ so $\mathbf{0} \models \mathcal{A}$ by induction. □

Lemma 7.13. *There is no formula* $\mathcal{A} \in \mathrm{SAL}_{\mathrm{int}} - \{0\}$ *equivalent to* 0.

Proof. By absurd, if $\mathcal{A}$ is such a formula an $n \notin \mathrm{fn}(\mathcal{A})$, then by Lemma 7.12, $n[\mathbf{0}] \models \mathcal{A}$ and the contradiction. □

8. SL and classical logic

In this section, we give a second illustration of our minimalisation method. We consider the assertion language presented in [4], referred as SL. SL holds spatial connectives $*$ and $-\!\!*$ similar to $|$ and $\triangleright$ in SAL, with a light but significant difference for $*$: the composition requires a compatibility condition $h \perp h'$ that is not always satisfied; in particular, it is not possible to compose two copies of the same structure ($h * h$). As a consequence, the expressiveness of $*$ is quite restricted and essentially express the separation of resources, which equality already expresses. For this reason, we can establish the elimination of both $*$ and $-\!\!*$. We define a classical fragment CL and prove it to be as expressive as SL (Fig. 5).

$$\boxed{\begin{array}{lll} e & ::= & x|\mathsf{nil}|- \\ P & ::= & (x \mapsto e_1, e_2)|x = y|\mathsf{emp}|\bot|P \Rightarrow P \\ & & |P * P|P \mathrel{-\!\!*} P \end{array}}$$

Fig. 5. Separation logic (SL).

8.1. Definitions

We assume a countable set Var of variables, ranged over with x, y, and a set Loc of locations such that Loc $\subseteq \mathbb{N}$. Expressions and assertions of SL are defined as in Fig. 5. We write $\mathsf{v}(P)$ for the set of variables occurring in P. Assertions express properties of memory states, modelled as a pair consisting of a store and a heap, as follows:

$$\begin{aligned} \text{Val} &\stackrel{\text{def}}{=} \text{Loc} \sqcup \{\mathsf{nil}\}, \\ \text{Store} &\stackrel{\text{def}}{=} \text{Var} \to \text{Val}, \\ \text{Heap} &\stackrel{\text{def}}{=} \text{Loc} \rightharpoonup_{\text{fin}} \text{Val}, \\ \text{State} &\stackrel{\text{def}}{=} \text{Stack} \times \text{Heap}, \end{aligned}$$

where $\rightharpoonup_{\text{fin}}$ stands for a partial function with finite domain. We range over stores with s, over heaps with h, and over states with σ. We note $\sigma_1 \bot \sigma_2$ for $s_1 = s_2$ and $dom(h_1) \cap dom(h_2) = \emptyset$, and, when this holds, $\sigma_1 * \sigma_2$ is the state defined by keeping the same store and by setting $h_1 * h_2(x) = h_1(x)$ or $h_2(x)$.

For a value v, we note $v \vDash_\sigma e$ if either $e = -$, or $v = e = \mathsf{nil}$, or $e = x$ and $v = s(x)$. We then note $(v_1, v_2) \vDash_\sigma (e_1, e_2)$ if $v_1 \vDash_\sigma e_1$ and $v_2 \vDash_\sigma e_2$. The condition for a state σ to match an assertion P, written $\sigma \vDash P$, is inductively defined as:

$$\begin{array}{lll} \sigma \vDash \bot & \text{never} & \\ \sigma \vDash (x \mapsto e_1, e_2) & \text{iff} & dom(h) = \{s(x)\} \text{and} \\ & & hs(x) \vDash_\sigma (e_1, e_2) \\ \sigma \vDash x = e_y & \text{iff} & s(x) = s(y) \\ \sigma \vDash \mathsf{emp} & \text{iff} & dom(h) = \emptyset \\ \sigma \vDash P_1 \Rightarrow P_2 & \text{iff} & \sigma \vDash P_1 \text{implies} \sigma \vDash P_2 \\ \sigma \vDash P_1 * P_2 & \text{iff} & \text{there exist} \sigma_1 \text{and} \sigma_2 \text{such that} \\ & & \sigma = \sigma_1 * \sigma_2;\ \sigma_1 \vDash P_1 \text{and} \sigma_2 \vDash P_2 \\ \sigma \vDash P_1 \mathrel{-\!\!*} P_2 & \text{iff} & \text{for all} \sigma_1 \text{such that} \sigma \bot \sigma_1, \\ & & \sigma_1 \vDash P_1 \text{implies} \sigma * \sigma_1 \vDash P_2. \end{array}$$

We may define as usual the connectives $\wedge, \vee, \top, \neg, \Leftrightarrow$ in the obvious way. We also introduce two *monotonic*[1] assertions (cf. Fig. 6). Any assertion of this form, or of the form $x = y$ will be said to be *atomic*. In the remainder, we actually take these as primitive, which ensure the encoding of $(x \mapsto e_1, e_2)$ and emp assertions through boolean combinations.[2]

[1] Or *intuitionistic*, using the terminology of Reynolds [18], that is assertions P such that $\sigma \vDash P$ implies $\sigma' \vDash P$ for all $\sigma' \geqslant \sigma$.

[2] On the contrary, it is not possible to encode $(x \hookrightarrow e_1, e_2)$ and $\mathsf{size} \geqslant n$ from $(x \mapsto e_1, e_2)$ and emp using only boolean combinations; this point is also discussed in conclusion.

monotonic assertion	encoding in SL	semantic
$(x \hookrightarrow e_1, e_2)$	$(x \mapsto e_1, e_2) * \top$	$s(x) \in dom(h)$and $hs(x) \models_\sigma (e_1, e_2)$
$\mathsf{size} \geqslant n$	$\underbrace{\neg\mathsf{emp} * \ldots * \neg\mathsf{emp}}_{n \text{ times}}$	$\sharp\, dom(h) \geqslant n$

Fig. 6. Monotonic assertions from SL.

$$\boxed{P \quad ::= P \Rightarrow P \,|\, \bot \,|\, (x \hookrightarrow e_1, e_2) \,|\, x = y \,|\, \mathsf{size} \geqslant n\,.}$$

Fig. 7. Classical fragment (CL) of SL.

We call *classical logic* (CL) the fragment of SL defined by the grammar of Fig. 7. We will note $w(P)$ for the maximal n such that $\mathsf{size} \geqslant n$ is a subassertion of P, and $\mathsf{v}(P)$ for the set of variables of P.

Our main result is the following:

Theorem 8.1. *CL is as expressive as SL, i.e. for all assertion P of SL, there exists a classical assertion P' of CL such that $\models P \Leftrightarrow P'$.*

At the same time, we also prove the following result: the monotonic (indeed atomic) fragment is as separative as the whole language, that is if two states satisfy the same monotonic assertions, then they satisfy the same assertions.

8.2. Proof of the translation

Our proof proceeds in the same way as for SAL: we define an intensional equivalence and prove that it has the precompactness and characteristic formula properties.

Let X be a finite set of variables, and w an integer. We say that two states σ and σ' are intensionally equivalent for X, w, written $\sigma \approx_{X,w} \sigma'$, if for all classical assertion P with $\mathsf{v}(P) \subseteq X$ and $w(P) \leqslant w$, $\sigma \models P$ iff $\sigma' \models P$.

Remarks. 1. This definition amounts to say that σ and σ' satisfy the same atomic classical assertions P with $\mathsf{v}(P) \subseteq X$ and $w(P) \leqslant w$.

2. Let us write $w(\sigma) = \sharp dom(h)$. Given three natural numbers a, b, w, we write $a =_w b$ if either $a = b$ or $a, b \geqslant w$. Then for any σ, σ' such that $\sigma \approx_{X,w} \sigma'$, $w(\sigma) =_w w(\sigma')$.

3. Equality assertions $x = y$ only depend on the store. We note $s =_X s'$ if these stores satisfy the same equality assertions with variables in X. Then for any σ, σ' such that $\sigma \approx_{X,w} \sigma'$, $s =_X s'$.

4. Let V be some set of values. We note $v =_V v'$ if either $v = v'$ or $\{v, v'\} \cap V = \emptyset$, and $(v_1, v_2) =_V (v_1', v_2')$ if $v_1 =_V v_1'$ and $v_2 =_V v_2'$. Then for any s, h, h' such that $(s, h) \approx_{X,w} (s, h')$, $dom(h) \cap s(X) = dom(h') \cap s(X)$ due to assertions $x \hookrightarrow -, -$, and for all $l \in s(X) \cap dom(h)$, $h(l) =_{s(X) \cup \{\mathsf{nil}\}} h'(l)$ due to assertions $x \hookrightarrow e_1, e_2$.

Let me say more about store equivalence. Consider a store s_0 and a state $\sigma = (s, h)$ such that $s_0 =_X s$. Then we may define a new state $\mathsf{shift}_{s_0,X}\sigma$ of store s_0 and heap h' defined such that

- $dom(h) = s_0\big(s^{-1}(dom(h)) \cap X\big) \cup B$ with B some arbitrary set of locations such that $\sharp dom(h) = \sharp dom(h')$ and $B \cap s_0(X) = \emptyset$.
- For all $l \in dom(h')$, if $l = s_0(x)$ and $hs(x) = (s(y), s(z))$ for some $x, y, z \in X$, $h' s_0(x)$ is set to be $(s_0(y), s_0(z))$, otherwise $h(l)$ is arbitrarily defined out of $s(X)$.

This is easy to check that σ and $\mathsf{shift}_{s_0,X}\sigma$ satisfy the same atomic assertions with variables in X. Moreover, this transformation is compositional, in the sense that $\mathsf{shift}_{s_0,X}(\sigma * \sigma') = \mathsf{shift}_{s_0,X}\sigma * \mathsf{shift}_{s_0,X}\sigma'$. This transformation is not completely deterministic, but assuming that every choice of a “fresh” value is made different at each time and at each call to $\mathsf{shift}_{,X}$, $\sigma \perp \tau$ will imply $\mathsf{shift}_{s_0,X}\sigma \perp \mathsf{shift}_{s_0,X}\tau$. We actually have the following stronger result:

Lemma 8.2. *For all assertions $P \in \mathrm{SL}$ with $\mathsf{v}(P) \subseteq X$, $\sigma \vDash P$ iff $\mathsf{shift}_{s_0,X}\sigma \vDash P$.*

The proof is straightforward by induction on the assertion P considering previous remarks.

We now recall the equivalence relation defined by Yang [15] for the decidability proof, and use it to derive the correction of $\approx_{X,w}$.

Definition 8.3 ($\sim_{s,n,X}$ *Hongseok Yang [15]*). Given a stack s, a natural number n and a set X of variables, $\sim_{s,n,X}$ is the relation between heaps such that $h \sim_{s,n,X} h'$ iff

1. $s(X) \cap dom(h) = s(X) \cap dom(h')$;
2. for all $l \in s(X) \cap dom(h)$, $h(l) =_{s(X)} h'(l)$;
3. $\sharp\big(dom(h) - s(X)\big) =_n \sharp\big(dom(h') - s(X)\big)$.

The first step of the correction proof is to factorize $\approx_{X,w}$ in $\sim_{s,n,X}$.

Lemma 8.4. *For any X, w, n such that $n + \sharp X \leqslant w$, for any $\sigma, \sigma', s, h, h'$ such that $\sigma = (s, h)$, $\sigma \approx_{X,w} \sigma'$, and $\mathsf{shift}_{s,X}\sigma' = (s, h')$, it holds that $h \sim_{s,n,X} h'$.*

Proof. By Lemma 8.2, $(s, h) \approx_{X,w} (s, h')$. Then conditions 1 and 2 in Definition 8.3 holds by Remark 4, so the proof follows from the verification of the condition 3 on the heap size.

Let us assume first that $\sharp\big(dom(h) - s(X)\big) < n$; then $\sharp dom(h) = k < n + \sharp X \leqslant w$, so $\sigma \vDash P = \mathsf{size}\geqslant k \wedge \neg \mathsf{size}\geqslant k + 1$, and $w(P) = k + 1 \leqslant w$. By definition of $\approx_{X,w}$, $\sigma' \vDash P$, so $\sharp dom(h') = k = \sharp dom(h)$. Moreover, $s(X) \cap dom(h) = s(X) \cap dom(h')$, so finally $\sharp\big(dom(h) - s(X)\big) = \sharp\big(dom(h') - s(X)\big)$.

Let us assume now that $\sharp\big(dom(h) - s(X)\big) \geqslant n$; and set $k = \min(\sharp dom(h), w)$, so that $\sigma \vDash \mathsf{size}\geqslant k$, and by definition of $\approx_{X,w}$, $\sigma' \vDash \mathsf{size}\geqslant k$. Moreover, $dom(h) \geqslant n + \sharp\big(dom(h) \cap s(X)\big)$, and $w \geqslant n + \sharp X \geqslant n + \sharp\big(dom(h) \cap s(X)\big)$, so finally $k \geqslant n + \sharp\big(dom(h) \cap s(X)\big)$. This gives $dom(h') \geqslant k \geqslant n + \sharp\big(dom(h') \cap s(X)\big)$ since $s(X) \cap dom(h) = s(X) \cap dom(h')$, i.e. $\sharp\big(dom(h') - s(X)\big) \geqslant n$.

$\sharp dom(h) \geqslant k \geqslant n + \sharp\big(dom(h) \cap s(X)\big)$, where $k = \min(\sharp dom(h), w)$. So $\sigma \vDash \mathsf{size}\geqslant k$, and by definition of $\approx_{X,w}$, $\sigma' \vDash \mathsf{size}\geqslant k$, so that finally $\sharp dom(h') \geqslant n + \sharp\big(dom(h') \cap s(X)\big)$. □

We now recall the correction result obtained by Yang and derive our correction from it. We first recall the notion of formula's size used by Yang:

$$\begin{array}{rlrlrl} |(e \mapsto e_1, e_2)| &= 1, & |e_1 = e_2| &= 0, & |\mathsf{emp}| &= 1, \\ |P \Rightarrow Q| &= \max(|P|, |Q|), & |\bot| &= 0, & & \\ |P * Q| &= |P| + |Q|, & |P -\!\!* Q| &= |Q|. & & \end{array}$$

Lemma 8.5. *Take s, h, h', n, X with $h \sim_{s,n,X} h'$. Then for all assertion $P \in \mathrm{SL}$ such that $\mathsf{v}(P) \subseteq X$ and $|P| \leqslant n$, $(s, h) \models P$ iff $(s, h') \models P$.*

The proof of this result is detailed in [15].

Corollary 8.6 (*Correction*). *Take σ, σ', w, X with $\sigma \approx_{X,w} \sigma'$. Then for all assertion $P \in \mathrm{SL}$ such that $\mathsf{v}(P) \subseteq X$ and $|P| + \sharp X \leqslant w$, $\sigma \models P$ iff $\sigma' \models P$.*

Proof. By Lemma 8.4, $h \approx_{s,n,X} h'$ with $\sigma = (s, h)$, $\mathsf{shift}_{s,X}\sigma' = (s, h')$, and $n = w - \sharp X$. Then $\sigma \models P$ implies $\mathsf{shift}_{s,X}\sigma' \models P$ by Lemma 8.5, which implies $\sigma' \models P$ by Lemma 8.2. □

We may now end the proof establishing the properties of precompactness and characteristic formula for $\approx_{X,w}$.

We write $\Phi_{X,w}$ for the set of atomic assertions P such that $\mathsf{v}(P) \subseteq X$ and $w(P) \leqslant w$. For X finite, $\Phi_{X,w}$ is finite as well. This has two important consequences:

Proposition 8.7 (*Precompactness*). *For all w and all finite X, $\approx_{X,w}$ has only finitely many classes.*

Proof. A class is represented by a subset $\Phi \subseteq \Phi_{X,w}$ of atomic assertions that are the ones satisfied by any state of the class. So there are less than $2^{\sharp \Phi_{X,w}}$ distinct classes. □

Proposition 8.8 (*Characteristic formula*). *For all states σ, for all X, w, there is a classical assertion $F_\sigma^{(X,w)}$ such that*

$$\forall \sigma'. \sigma' \models F_\sigma^{(X,w)} \quad \text{iff } \sigma \approx_{X,w} \sigma' .$$

Proof. Take

$$\bigwedge_{\sigma \models P,\, P \in \Phi_{X,w}} P \wedge \bigwedge_{\sigma \not\models P,\, P \in \Phi_{X,w}} \neg P . \qquad \square$$

We may now establish Theorem 8.1 noticing that any assertion P of SL is equivalent to the classical assertion:

$$\bigvee_{C \in \mathrm{State}_{/\approx_{X,w}},\, C \models P} F_C^{(X,n)} ,$$

where finiteness of this disjunction is ensured by Proposition 8.7.

9. Conclusion

We have established the adjuncts elimination property for SAL, a logic for trees with binders including the fresh quantifier И. This involves putting a formula in prenex form and then doing the transformation on the quantifier-free formula. The adjunct-free fragment $\mathrm{SAL}_{\mathrm{int}}$ then turns out to be a *minimal* logic.

We established the absence of adjunct elimination for the same logic where И is replaced by the usual ∀ quantifier, whichever adjunct is considered. This result, together with the difference w.r.t. decidability of model-checking on pure trees, illustrates the significant gap existing between the two forms of quantification.

Finally, we defined a classical fragment of the separation logic (SL), excluding both $*$ and $-\!\!*$, and proved it to be as expressive as the full SL. Our approach shows also that all the separative power of the logic lies in the monotonic fragment. When defining our classical fragment, we had to move from the assertions $x \mapsto e_1, e_2$ and emp to $x \hookrightarrow e_1, e_2$ and $\mathsf{size}\geqslant n$ in order to capture the $*$ connective; without that, it is probably possible to eliminate only the adjunct. Note that the assertion $(x \mapsto \mathsf{nil}, \mathsf{nil}) -\!\!* \mathsf{false}$ would be translated in CL as $x \hookrightarrow -, -$, which underlines the importance of the special expression $-$.

In relation to our study, some observations can be made regarding the difference between the И and the ∀/∃ quantification. The existence of prenex forms, the decidability of the model-checking on pure trees, the adjuncts elimination, are properties verified by the logic with the fresh quantifier, whereas they fail for the universal quantifier.

Yang proposed a clever counterexample to the elimination of $-\!\!*$ in a SL with quantifiers; this example seems of deeper meaning than the one presented in Section 6, but a better understanding of its implications is still lacking. In the same way, we do not know whether $*$ elimination remains true for the assertion language without $-\!\!*$ and with quantifiers.

The results we obtain for SAL and SL can be adapted to several other sub-structural logics. However, for the logics including the time modality $\diamond$ [8,1], adjuncts improve the expressiveness of the logic supporting an encoding of action modalities [19,13]. One could think to take them as primitives in the same spirit as for SL, and look for the adjunct elimination. However, even in the case of very elementary concurrent languages, this project is not realisable [2].

Acknowledgements

This work has been supported by the European FET—Global Computing project PROFUNDIS, and by the Action Incitative *Méthodes Formelles pour la Mobilité*-CNRS.

I would like to thank M.J. Gabbay for enlightening discussions about Nominal Sets theory. The anonymous referees, G. Ghelli, and H. Yang helped me significantly to improve the previous versions of this presentation. I also want to thank D. Sangiorgi, L. Monteiro, L. Caires, and D. Hirschkoff for their advice all along this work.

References

[1] L. Caires, L. Cardelli, A spatial logic for concurrency (Part I), in: Proc. TACS'01, Lecture Notes in Computer Science, Springer, Berlin, 2001.

[2] L. Caires, E. Lozes, Elimination of quantifiers and undecidability in spatial logics for concurrency, Proc. CONCUR'04, London, September 2004, pp. 240–257.

[3] C. Calcagno, L. Cardelli, A. Gordon, Deciding validity in a spatial logic for trees, in: Proc. TLDI'03, ACM Press, New York, 2003, pp. 62–73.

[4] C. Calcagno, H. Yang, P. O'Hearn, Computability and complexity results for a spatial assertion language for data structures, Proc. FSTTCS'01, Lecture Notes in Computer Science, vol. 2245, Springer, Berlin, 2001.

[5] L. Cardelli, P. Gardner, G. Ghelli, Manipulating trees with hidden labels, Foundations of Software Science and Computational Structures, Proc. Sixth Int. Conf. FOSSACS, Lecture Notes in Computer Science, vol. 2620, Springer, Berlin, 2003, pp. 216–232.

[6] L. Cardelli, G. Ghelli, A query language based on the ambient logic, Proc. ESOP'01, Lecture Notes in Computer Science, vol. 2028, Springer, Berlin, 2001, pp. 1–22.

[7] L. Cardelli, A. Gordon, Mobile Ambients, in: Proc. of FOSSACS'98, pp. 140–155.

[8] L. Cardelli, A. Gordon, Anytime anywhere modal logics for mobile ambients, in: Proc. POPL'00, ACM Press, New York, 2000, pp. 365–377.

[9] L. Cardelli, A. Gordon, Logical properties of name restriction, Proc. TLCA'01, Lecture Notes in Computer Science, vol. 2044, Springer, Berlin, 2001.

[10] W. Charatonik, J.-M. Talbot, The decidability of model checking mobile ambients, in: Proc. CSL'01 Lecture Notes in Computer Science, Springer, Berlin, 2001.

[11] M.J. Gabbay, A.M. Pitts, A new approach to abstract syntax involving binders, in: Proc. 14th Annu. Symp. on Logic in Computer Science, IEEE Computer Society Press, Washington, 1999, pp. 214–224.

[12] G. Ghelli, G. Conforti, Decidability of freshness, undecidability of revelation, in: Proc. FOSSACS'04, March 2004.

[13] D. Hirschkoff, E. Lozes, D. Sangiorgi, Separability expressiveness and decidability in the ambients logic, in: Proc. 17th IEEE Symp. on Logic in Computer Science, IEEE Computer Society, Silver Spring, MD, 2002, pp. 423–432.

[14] C.A.R. Hoare, An axiomatic basis for computer programming, Commun. ACM 12 (10) (1969) 576–580.

[15] Hongseok Yang, Local reasoning for stateful programs, Ph.D. Thesis, University of Illinois at Urbana Champaign, 2001.

[16] D. Lugiez, S. Dal-Zilio, C. Meyssonnier, A logic you can count on, in: Proc. POPL'04, 2004.

[17] J. Reynolds, Intuitionistic reasoning about shared mutable data structure, 2000.

[18] J. Reynolds, Separation logic: a logic for shared mutable data structures—invited paper, in: Proc. LICS'02, 2002.

[19] D. Sangiorgi, Extensionality and intensionality of the ambient logic, in: Proc. 28th POPL, ACM Press, New York, 2001, pp. 4–17.