A Spatial Equational Logic for the Applied π-Calculus

Étienne Lozes and Jules Villard

LSV, ENS Cachan, CNRS 61 av. du pdt Wilson, 94230 Cachan, France
{lozes,villard}@lsv.ens-cachan.fr

Abstract. Spatial logics have been proposed to reason locally and modularly on algebraic models of distributed systems. In this paper we define the spatial equational logic AπLwhose models are processes of the applied π-calculus. This extension of the π-calculus allows term manipulation and records communications as active substitutions in a frame, thus augmenting the underlying predefined equational theory. Our logic allows one to reason locally either on frames or on processes, thanks to static and dynamic spatial operators. We study the logical equivalences induced by various relevant fragments of AπL, and show in particular that the whole logic induces a coarser equivalence than structural congruence. We give characteristic formulae for some of these equivalences and for static equivalence. Going further into the exploration of AπL's expressivity, we also show that it can eliminate standard term quantification.

1 Introduction

Spatial logics. Spatial logics, partly inspired by pioneering ideas of resource logics [1,2], have been proposed to reason locally and modularly on algebraic models of distributed systems such as ambients [3] or π-calculus [4]. Two essential connectives in these logics are spatial conjunction and adjunct. The spatial conjunction $A * B$, which introduces local reasoning, is the cause of a very intensional discriminating power for spatial logics [5,6], as usually logical equivalence is structural congruence. When this connective is dropped, reasoning only with adjunct $-\!\!*$ yields extensional equivalences such as barb equivalence [7]. Though quite intuitive, these results are dependent on the nature of the process model they deal with, and should be treated with some care: the intensional equivalence might be much coarser than structural congruence, and logical equivalence does not have to be a congruence in the presence of adjunct.

A spatial equational logic. In this paper we investigate a spatial equational logic that extends the first order equational logic with spatial connectives. Term equality $M = N$ is defined by both global axioms from an equational theory $\mathcal{E}$ and local axioms from a frame. For example, the frame $\Phi = \{^{\mathrm{enc}(s,y)}/_{x}\} \,|\, \{^{\mathrm{pk}(n)}/_{y}\}$, that we will use as an example throughout this paper, augments the equational theory with the knowledge that x is an alias for a message encrypted with a public key, itself aliased by y. To reflect the private nature of s and n, these names will be hidden, and we will write $\nu n, s.\, \Phi$. Spatial conjunction $*$ may then split the frame into smaller pieces that may not share any secret. For instance, $\nu n, s.\, \Phi = (\nu s.\, \{^{\mathrm{enc}(s,y)}/_{x}\}) * (\nu n.\, \{^{\mathrm{pk}(n)}/_{y}\})$.

F. van Breugel and M. Chechik (Eds.): CONCUR 2008, LNCS 5201, pp. 387–401, 2008.

Our spatial equational logic, written AπL, naturally arises as the spatial logic of the applied π-calculus [8], an extension of π-calculus [9] where processes may communicate terms through channels. One peculiar aspect of applied π-calculus is that what is sent to the environment is kept as *active substitutions* that act as local aliases that extend the equational theory. For instance, $\nu n, s.\, \Phi$ might have been generated from a process $\nu n.\, \{^{\mathrm{pk}(n)}/_{y}\} \mid \nu s.\, \bar{a}\langle \mathrm{enc}(s, y)\rangle.P$ sending $\mathrm{enc}(s, y)$ to an environment listening on channel a, thus reducing to $\nu n, s.\, \Phi \mid P$.

Motivations. Cryptographic protocols are the most standard applications modeled with the applied π-calculus. Observation by a passive attacker may be modeled by a static observational equivalence, usually simply called static equivalence. This proved sound enough to express some complex cryptographic properties such as resistance to guessing attacks, through other observational equivalences related to static equivalence. Logical foundations for these notions of observations are however needed to understand their dependencies and provide more flexible specifications.

From a logical point of view, spatial logics can be considered as fragments of second order logics, hence have to be compared to first-order and second-order logics regarding expressiveness issues. This line of research has been actively followed [10,11,12], but is still open regarding first order equational logics.

Contributions. Our first contribution is the characterization of the logical equivalences of the static, static extensional, and dynamic fragments. Static fragments turn out to play similar roles as in the case of the π-calculus: static intensional equivalence is proved to coincide with structural congruence for frames, whereas static extensional logical equivalence coincides with static equivalence. This is, to our knowledge, the first logical characterization of static equivalence that is independent on the equational theory. All the constructions involved are rather simple compared to other logical characterizations of frame equivalence [13] for specific classes of equational theories. Moreover, characteristic formulae are derivable for both intensional and extensional equivalences.

Our second contribution deals with the logical equivalence for the dynamic intensional fragment. Surprisingly, we show that the logic cannot distinguish messages with similar information content. As a consequence, this equivalence is coarser than mere structural congruence. Moreover, we show that it is not a congruence, due to the possible introduction of noise in communications that the logic may not detect. This noticeably complicates the techniques to obtain an axiomatization of this equivalence. We point out some admissible axioms for logical equivalence and prove this axiomatization complete for the equational theory of finite trees.

Our third contribution is a quantifier elimination technique that shows that standard term quantifiers $\exists t.\, A$ can be mimicked by spatial connectives, which illustrates one more particular example of a spatial logic that is more expressive than first-order logic.

Structure of the paper. In Section 2, we collect all the necessary background on the applied π-calculus, and define our process compositions $*$ and $\natural$. Section 3 introduces AπL; Sections 4 and 5 present the characterizations of the logical equivalences for the static and dynamic fragments respectively. Section 6 establishes the quantifier elimination property.

2 Applied π-Calculus

2.1 Terms

The grammar of applied π-calculus processes relies on the definition of a set of *terms* along with an *equational theory*. This lets the user decide which cryptographic primitives the calculus will use for example. The set of terms is constructed using disjoint infinite sets $\mathcal{V}$ and $\mathcal{N}$ of respectively variables and names, and a finite *signature* Σ, which is a set of functions, each with its arity (constants have arity 0). Its grammar is as follows, $ar(f)$ being the arity of f:

$$M, N ::= x \in \mathcal{V} \mid a \in \mathcal{N} \mid f(M_1, \ldots, M_{ar(f)}) .$$

We will use the letters a, b, c, n, m, s to refer to elements of $\mathcal{N}$, x, y, z for elements of $\mathcal{V}$ and u, v, w for "meta-variables" that may belong either to $\mathcal{N}$ or $\mathcal{V}$. We will write M, N for terms.

These terms are equipped with an equivalence relation $\mathcal{E}$ called an equational theory on Σ, where membership of a pair (M, N) of terms is written $\mathcal{E} \vdash M = N$, or simply $M = N$ if $\mathcal{E}$ is clear from context. This relation must be closed under substitution of terms for variables or names ($M_1 = M_2$ implies $M_1[u{\leftarrow}N] = M_2[u{\leftarrow}N]$) and context application ($N_1 = N_2$ implies $M[x{\leftarrow}N_1] = M[x{\leftarrow}N_2]$). $fn(M)$ and $fv(M)$ are respectively the sets of free names and free variables of M, defined as usual, and $fnv(M) \triangleq fn(M) \cup fv(M)$.

2.2 Processes

Applied π-calculus extends the standard π-calculus with primitives for term manipulation, namely *active substitutions* and term communications. The grammar of processes is split into two levels: the *plain* processes which account for the dynamic part, and the *extended* ones, also simply referred to as "processes" which extend the former with a *static* part.Note that replication $!P^p$ is not part of our setting.

$P^p, Q^p, \ldots ::=$	plain processes	$P, Q, \ldots ::=$	(extended) processes
$\mathbf{0}$	null process	P^p	plain process
$P^p \mid Q^p$	composition	$\{^M/_x\}$	active substitution
$\nu a.\, P^p$	name restriction	$P \mid Q$	parallel composition
$a^{ch}(n).P^p$	name input	$\nu a.\, P$	name restriction
$\bar{a}^{ch}\langle n\rangle.P^p$	name output	$\nu x.\, P$	variable restriction
$a(x).P^p$	term input		
$\bar{a}\langle M\rangle.P^p$	term output		
$if\ M = N$ $then\ P^p\ else\ Q^p$	conditional		

Our grammar differs from the original one in that it allows communications of two kinds, that may not interfere: communications of names behave as in the standard π-calculus whereas communications of terms may interact with active substitutions and

conditionals. Names are thus allowed to serve both as channels through which communications may occur and as atoms on which to build terms, without the need to rely on a type system to ensure the validity of communications.

From now on, we will only consider extended processes whose active substitutions are cycle-free. Furthermore, we will always assume that there is at most one active substitution for each variable, and *exactly one* if the variable is restricted. A process following these constraints will be called well-formed.

The set of free names (resp. variables) of a process P is defined as usual and written $fn(P)$ (resp. $fv(P)$), with $fn(\{^M/_x\}) \triangleq fn(M)$ (resp. $fv(\{^M/_x\}) \triangleq \{x\} \cup fv(M)$), and with both restrictions and both inputs being binders. We write $fnv(P)$ for the set $fn(P) \cup fv(P)$.

Compositions of active substitutions of the form $\{^{M_1}/_{x_1}\} \mid \cdots \mid \{^{M_n}/_{x_n}\}$ will be written $\{^{\boldsymbol{M}}/_{\boldsymbol{x}}\}$, and referred to using σ, τ. Depending on the context, we will write $\boldsymbol{x}$ for both the vector $x_1, \ldots, x_n$ and the associated set $\{x_1, \ldots, x_n\}$, where $n = |\boldsymbol{x}|$. Trailing $\mathbf{0}$'s in processes will often be omitted, as well as null $else$ branches in conditionals.

2.3 Operational Semantics

The structural congruence relation $\equiv$ identifies processes that can be obtained one from another by mere rewriting. It is the smallest equivalence relation on well-formed extended processes that is stable by α-conversion on both names and variables and by application of contexts, and that satisfies the usual rules w.r.t. the AC properties of $\mid$ with neutral $\mathbf{0}$, and the scope extrusion of restrictions. For instance, we have that $\nu a.\,(\nu x.\,\{^a/_x\} \mid \bar{a}\langle b\rangle) \mid \bar{c}\langle y\rangle \;\equiv\; \nu x.\,\nu a.\,(\{^a/_x\} \mid \bar{a}\langle b\rangle \mid \bar{c}\langle y\rangle)$. In addition, it must satisfy the following rules:

$$\begin{array}{ll} \textsc{Alias} & \nu x.\,(\{^M/_x\} \mid P) \equiv P[x{\leftarrow}M] \\ \textsc{Subst} & \{^M/_x\} \mid P^p \equiv \{^M/_x\} \mid P^p[x{\leftarrow}M] \\ \textsc{Rewrite} & \{^M/_x\} \equiv \{^N/_x\} \text{ if } \mathcal{E} \vdash M = N \end{array}$$

Here, a context (resp. an evaluation context) is an extended process with a hole in place of a plain (resp. extended) process. This hole can be filled with any extended process provided the resulting extended process is well-formed. The original structural congruence [8] is closed by application of evaluation contexts instead of arbitrary contexts. This makes inductive characterization of processes up to $\equiv$ impossible (see Section 5).

Due to the REWRITE rule, two structurally congruent processes may not have the same set of free names or variables. Thus, we define the closures $\overline{fn}(P), \overline{fv}(P), \overline{fnv}(P)$ of these sets up to structural congruence, and the corresponding sets for terms. For instance, $\overline{fn}(P) \triangleq \bigcap_{Q \equiv P} fn(Q)$ and $\overline{fv}(M) \triangleq \bigcap_{N=M} fv(N)$.

It is worth mentioning that rules ALIAS and SUBST are not the ones from the original applied π-calculus, namely:

$$\begin{array}{ll} \textsc{Alias'} & \nu x.\,\{^M/_x\} \equiv \mathbf{0} \\ \textsc{Subst'} & \{^M/_x\} \mid P \equiv \{^M/_x\} \mid P[x{\leftarrow}M] \end{array}$$

With our rules, active substitutions may affect other active substitutions only if their domain is a restricted variable. They may only apply to plain processes otherwise. As

such, the rule ALIAS' is still valid in our setting, whereas SUBST' is restricted to plain processes only. This makes our quantifier elimination technique easier, as it dramatically limits the interferences between active substitutions. On the other hand, it does not change the behaviour of processes, as both the reduction rules and the static equivalence definitions, presented below, stay the same. Moreover, a process P can still always be rewritten into a set of restricted names $\boldsymbol{n}$, a public composition of active substitutions σ and a public plain process Q: $P \equiv \nu \boldsymbol{n}.\,(\sigma \mid Q)$.

Finally, internal reduction $\rightarrow$ is the smallest relation that satisfies the rules below and that is closed by structural congruence and by application of evaluation contexts.

$$\begin{array}{ll}
\textsc{Comm-T} & \bar{a}\langle x\rangle.P \mid a(x).Q \rightarrow P \mid Q \\
\textsc{Comm-C} & \bar{a}^{ch}\langle m\rangle.P \mid a^{ch}(n).Q \rightarrow P \mid Q[n{\leftarrow}m] \\
\textsc{Then} & \mathit{if}\ M = M\ \mathit{then}\ P\ \mathit{else}\ Q \rightarrow P \\
\textsc{Else} & \mathit{if}\ M = N\ \mathit{then}\ P\ \mathit{else}\ Q \rightarrow Q \\
 & \qquad (\text{when } \mathit{fv}(M,N) = \emptyset \text{ and } \mathcal{E} \nvdash M = N)
\end{array}$$

2.4 Frames

A *frame* is an extended process built up from active substitutions and the null process only. The *domain* $dom(\phi)$ of a frame ϕ is the set of variables upon which the active substitutions of ϕ act. The frame $\phi(P)$ of a process P is P in which every plain process embedded into P is set to $\mathbf{0}$. Similarly, the plain process $(P)^p$ associated with P is obtained by mapping every substitution over non-restricted variables to $\mathbf{0}$.

The following definitions are standard in the applied π-calculus:

- ϕ is *closed* when $\mathit{fv}(\phi) \subseteq dom(\phi)$;
- an evaluation context $C[\cdot]$ *closes* the frame ϕ when $C[\phi]$ is both well-formed and closed;
- two terms M and N are equal in the frame ϕ, written $\phi \vdash M = N$ when there exists a set of names $\boldsymbol{n}$ and a substitution σ (*i.e.* a public frame) such that $\phi \equiv \nu \boldsymbol{n}.\,\sigma$, $M\sigma = N\sigma$ and $\boldsymbol{n} \cap \mathit{fn}(M,N) = \emptyset$. Two terms are equal in the process P when they are equal in $\phi(P)$.

Definition 2.1 (Static equivalence). *Two closed frames ϕ and ψ are* statically equivalent*, written $\phi \approx_s \psi$, when $dom(\phi) = dom(\psi)$ and, for all terms M and N, $\phi \vdash M = N$ if and only if $\psi \vdash M = N$.*

Two processes are statically equivalent when their frames are.

As it is, static equivalence on non-closed frames does not lead to a congruence relation. For instance, with $\langle\,\cdot\,,\,\cdot\,\rangle$ being the pairing operation and π_1 the first projection, if we let $\phi = \nu n.\,\{^{\mathrm{dec}(\mathrm{enc}(\langle 1,n\rangle,1),y)}/_{x}\}$ and $\psi = \nu n.\,\{^{\mathrm{dec}(\mathrm{enc}(\langle 1,n\rangle,1),z)}/_{x}\}$ then $\phi \approx_s \psi$, but one can deduce $\pi_1(x) = 1$ from $\{^{1}/_{y}\} \mid \phi$ and not from $\{^{1}/_{y}\} \mid \psi$.

To overcome this issue, and later be able to write formulae characterizing static equivalence for both closed and non-closed frames, we introduce *strong* static equivalence $\approx_s^s$, which is the largest equivalence included in $\approx_s$ and closed by application of closing evaluation contexts. One can note that strong static equivalence is closed by application of arbitrary evaluation contexts (and not just closing ones), and that it coincides with static equivalence on closed frames.

2.5 Additional Operators

When splitting up a process P into a parallel composition of two subprocesses $P_1 \mid P_2$, two very different operations are performed, as both the dynamic and the static part of the process are split up into two extended processes. This does not match our intuition of local reasoning for processes. Indeed, consider a protocol $\nu n, n'.\, (\{^{\mathrm{ck}(n,n')}/_x\} \mid A(x,n) \mid B(x,n'))$ where Abelard and Héloïse share a compound key $\mathrm{ck}(n,n')$ generated from a nonce n of A and n' of B. Then a specification of the form $A \mid B$ would fail since this process is atomic. To overcome this situation we have broken down the parallel composition into two finer-grained operators: the first one $\mathbin{¦}$ keeps the same frame while splitting up the plain process and, conversely, the second one $*$ keeps the same plain process while splitting up the frame.

Definition 2.2 (Process and frame compositions). *Given frames ϕ, ϕ_1, ϕ_2, plain processes P, P_1, P_2 and names $\boldsymbol{n_1}, \boldsymbol{n_2}$ such that $\boldsymbol{n_1} \cap fn(\phi_2, P_2) = \boldsymbol{n_2} \cap fn(\phi_1, P_1) = \emptyset$, we let*

$$\begin{aligned} \nu \boldsymbol{n_1}.\, ((\nu \boldsymbol{n_2}.\, \phi) \mid P_1) \mathbin{¦} \nu \boldsymbol{n_2}.\, ((\nu \boldsymbol{n_1}.\, \phi) \mid P_2) &\triangleq \nu \boldsymbol{n_1 n_2}.\, (\phi \mid P_1 \mid P_2) \\ \nu \boldsymbol{n_1}.\, (\phi_1 \mid \nu \boldsymbol{n_2}.\, P) * \nu \boldsymbol{n_2}.\, (\phi_2 \mid \nu \boldsymbol{n_1}.\, P) &\triangleq \nu \boldsymbol{n_1 n_2}.\, (\phi_1 \mid \phi_2 \mid P)\,. \end{aligned}$$

In the following, for $\dagger$ in $\{\mid, *\}$, we write $P \leftrightarrow P_1 \dagger P_2$ if there are P',P'_1,P'_2 such that $P \equiv P'$, $P_1 \equiv P'_1$, $P_2 \equiv P'_2$ and $P' = P'_1 \dagger P'_2$.

Remark 2.3. Formally, $P \leftrightarrow P_1 * P_2$ is a ternary relation, and for some P_1, P_2, one may have $P \leftrightarrow P_1 * P_2$, $P' \leftrightarrow P_1 * P_2$ for some non congruent P, P'. Albeit not a composition law, $*$ projects as a composition on frames: $\phi(P * Q) \equiv \phi(P) \mid \phi(Q)$. Ternary relations also arise in the relational models of BI, or in context logics.

3 A Spatial Logic for the Applied π-Calculus

3.1 Syntax and Semantics

We assume an infinite set $\mathcal{TV}$ of *term variables*, distinct from $\mathcal{V}$, ranged over with $t, t', \ldots$, and we write U, V for terms that possibly contain these term variables. We call $\mathcal{L}_{\text{spat}}$ the set of formulae defined by the following grammar:

$$\begin{aligned} A, B ::= {} & U = V \mid \neg A \mid A \wedge B \mid \diamondsuit A \mid \exists t.\, A \mid \mathit{И} u.\, A \mid \mathsf{H} u.\, A \mid A \circledcirc u \mid \textcircled{c} u \\ & \mid \mathbf{0} \mid A \mathbin{¦} B \mid A \rhd B \mid \emptyset \mid A * B \mid A \mathrel{-\!\!*} B \end{aligned}$$

$U = V$ is the equality of terms w.r.t. the current frame, negation and conjunction are classical, and $\diamondsuit A$ is the strong reduction modality. $\exists t.\, A$ is term quantification and $\mathit{И} n.\, A$ and $\mathit{И} x.\, A$ are respectively the fresh name and variable quantifications. We use the same operator for both of these, as well as for the H, $\textcircled{c}$ and $\circledcirc$ operators, because our convention on namings lets us do so unambiguously. $\mathsf{H} u.\, A$ is the hidden name or variable quantification; $\textcircled{c} u$ means that u appears free in the process; $A \circledcirc u$ is hiding of name or variable u. $\mathbf{0}$ (resp. $\emptyset$) denotes the null plain process (resp. the empty frame) and $A \mathbin{¦} B$ (reps. $A * B$) plain process (resp. frame) composition. $A \rhd B$ (resp. $A \mathrel{-\!\!*} B$) is a guarantee operator and the adjunct of $A \mathbin{¦} B$ (resp. $A * B$).

We define two usual fragments of our logic: the extensional one, which lets one observe a process via its interactions with some (possibly constrained) environment, and the intensional one, which lets one explore the very structure of the process. We also distinguish between static operators, that only account for the frame, and dynamic ones, which account for the whole process. The four fragments are summed up by the table below, that defines which operators the formulae of each fragments may be composed of. The intensional fragment contains the extensional one and the dynamic one contains the static one; the static extensional fragment is thus common to all fragments, and the dynamic intensional one coincides with the whole logic.

	$Static$	$Dynamic$
$Extensional$	$=, \neg, \wedge, \exists, \mathsf{И}, \multimap, \oslash$	$\Diamond, \triangleright$
$Intensional$	$\mathsf{H}, *, \emptyset$	$ⓒ, \mathbf{0}, \mathbin{!}$

$$
\begin{array}{lll}
P, v \vDash U = V & \Leftrightarrow & P \vdash Uv = Vv \\
P, v \vDash \neg A & \Leftrightarrow & P, v \nvDash A \\
P, v \vDash A_1 \wedge A_2 & \Leftrightarrow & P, v \vDash A_1 \text{ and } P, v \vDash A_2 \\
P, v \vDash \Diamond A & \Leftrightarrow & \exists P'.\, P \to P' \text{ and } P', v \vDash A \\
P, v \vDash \exists t.\, A & \Leftrightarrow & \exists M.\, P, (v\{t \to M\}) \vDash A \\
P, v \vDash \mathsf{И} u.\, A & \Leftrightarrow & \exists u' \notin fnv(P, v, A).\, P, v \vDash A[u \leftarrow u'] \\
P, v \vDash \mathsf{H} u.\, A & \Leftrightarrow & \exists u' \notin fnv(P, v, A).\, \exists P'.\, P \equiv \nu u'.\, P' \text{ and } P', v \vDash A[u \leftarrow u'] \\
P, v \vDash A \oslash n & \Leftrightarrow & \nu n.\, P, v \vDash A \\
P, v \vDash A \oslash x & \Leftrightarrow & x \in dom(P) \text{ and } \nu x.\, P, v \vDash A \\
P, v \vDash ⓒ u & \Leftrightarrow & u \in \overline{fnv}(P) \\
P, v \vDash \mathbf{0} & \Leftrightarrow & (P)^p \equiv \mathbf{0} \\
P, v \vDash A_1 \mathbin{!} A_2 & \Leftrightarrow & \exists P_1, P_2.\, P \leftrightarrow P_1 \mathbin{!} P_2,\ P_1, v \vDash A_1 \text{ and } P_2, v \vDash A_2 \\
P, v \vDash A \triangleright B & \Leftrightarrow & \forall Q, R.\, (R \leftrightarrow P \mathbin{!} Q \text{ and } Q, v \vDash A) \text{ implies } R, v \vDash B \\
P, v \vDash \emptyset & \Leftrightarrow & \phi(P) \equiv \mathbf{0} \\
P, v \vDash A_1 * A_2 & \Leftrightarrow & \exists P_1, P_2.\, P \leftrightarrow P_1 * P_2,\ P_1, v \vDash A_1 \text{ and } P_2, v \vDash A_2 \\
P, v \vDash A \multimap B & \Leftrightarrow & \forall Q, R.\, (R \leftrightarrow P * Q \text{ and } Q, v \vDash A) \text{ implies } R, v \vDash B
\end{array}
$$

Fig. 1. Satisfaction relation

The operators' semantics, close to the one defined by Caires and Cardelli for the π-calculus [4], is given by a satisfaction relation described in Figure 1 whose judgements are of the form $P, v \vDash A$ between a process P, a spatial formula A and a valuation v. Valuations assign terms $\boldsymbol{M}$ to all the free variables $\boldsymbol{t}$ of the formula and are written $\{\boldsymbol{t} \to \boldsymbol{M}\}$ when $|\boldsymbol{t}| = |\boldsymbol{M}| = n$ and $v(t_i) = M_i$ for all $i \in \{1 \ldots n\}$. We write $v\{t \to M\}$ for the valuation v whose domain has been extended to t with $v(t) = M$. Finally, when A is closed and the valuation is empty, judgements are written $P \vDash A$. The following lemmas state that the logic behaves consistently w.r.t. structural congruence and that static operators may only state static properties indeed:

Lemma 3.1. *$P \vDash A$ and $P \equiv Q$ implies $Q \vDash A$.*

Lemma 3.2. *For every formula A of $\mathcal{L}^{stat}$, $P \vDash A$ iff $\phi(P) \vDash A$.*

Boolean operators are assumed to bind more tightly than compositions and adjunctions, which in turn bind more tightly than every other operator. Derived connectives $\forall t$, $\vee$, $\Leftrightarrow$ and $U \neq V$ are defined as usual, and so are the sets of free names and free variables of a formula A, written $fn(A)$ and $fv(A)$.

3.2 Derived Formulae

The formulae below will be useful in the following sections:

$$\top \triangleq \mathbf{0} \vee \neg\mathbf{0} \qquad \bot \triangleq \neg\top \qquad A[B] \triangleq (A \wedge \mathbf{0}) \mathbin{¦} ((B \wedge \emptyset) * \top)$$

$$A \blacktriangleright B \triangleq \neg(A \rhd \neg B) \qquad A \mathbin{—\!\circledast} B \triangleq \neg(A \mathbin{-\!\!*} \neg B) \qquad \mathbf{1} \triangleq \neg\mathbf{0} \wedge \neg(\neg\mathbf{0} \mathbin{¦} \neg\mathbf{0})$$

$$\mathbb{I} \triangleq \neg\emptyset \wedge \neg(\neg\emptyset * \neg\emptyset) \qquad \mathsf{public} \triangleq \neg \mathsf{H} n.\, ©n \qquad \mathsf{single} \triangleq \mathbf{1} \wedge \emptyset \wedge \mathsf{public}$$

Their meanings for a process P is that there must exist a process $Q \equiv P$ such that:

- $\top$: nothing is required; $\bot$ is always false;
- $A[B]$: $\phi(Q)$ verifies A and $(Q)^p$ verifies B;
- $A \blacktriangleright B$ (this is the dual of $\rhd$): there are Q', R such that $R \leftrightarrow Q \mathbin{¦} Q'$, $Q' \vDash A$ and $R \vDash B$, and similarly for $—\!\circledast$;
- $\mathbf{1}$ (resp. $\mathbb{I}$): $(Q)^p$ (resp. $\phi(Q)$) is not null and cannot be divided into two non-null processes;
- public: Q has no bound name: $\forall n, Q'.\, Q \equiv \nu n.\, Q' \Rightarrow n \notin \overline{fn}(Q')$;
- single: Q is guarded, either by a communication or by a conditional construct.

3.3 Cryptographic Examples

We now propose, on a very basic example, some possible avenues for using the spatial logic for the specification of some cryptographic properties. As usual, we interpret the frame as the history of past communications: restricted names are nonces or secrets, and each active substitution holds the content of an emitted message. Recall the frame $\nu n, s.\, \Phi$ of the introduction, modeling a situation where an encrypted secret s had been transmitted using a published public key $\mathrm{pk}(n)$ — we assume here the equational theory axiom $\mathrm{dec}(\mathrm{enc}(x, \mathrm{pk}(y)), \mathrm{sk}(y)) = x$ — and consider the frame $\nu n, s.\, \Phi \mid \phi = \nu n, s.\, \{^{\mathrm{enc}(s,y)}/_{x}\} \mid \{^{\mathrm{pk}(n)}/_{y}\} \mid \phi$.

Following the definition of the applied π-calculus, we will say that the secret s is deducible from this frame if the formula $\mathsf{leak} \triangleq \exists t.\, x = \mathrm{enc}(t, y)$ holds; for instance, choosing $\phi = \{^{\mathrm{sk}(n)}/_{z}\}$ would yield such a leak, with witness $t = \mathrm{dec}(x, z)$. The formula $\exists t. \forall t'.\, \mathsf{H} n.\, (t = \mathrm{pk}(n) \wedge t' \neq \mathrm{sk}(n))$ asserts that the published key is indeed public, whereas its associated private key is secret. An emitted message M represented by an active substitution $\{^{M}/_{z}\}$ in ϕ is part of the cause for a leak if the formula $(\neg\mathsf{leak}) \oslash z \wedge \mathsf{leak}$ holds.

One could also express static properties about authenticated sessions: in a protocol where each user is assigned a session identifier (here, a secret name) used in every subsequent communication, one may count the number of opened sessions with $*$-conjunctions

of the $\mathbb{I}$ formula. Indeed, if every other nonce used by the protocol within a session is generated from the session identifier, each subframe verifying $\mathbb{I}$ will correspond to a different session.

The dynamic part of the logic allows one to reason about the execution in isolation of some partners of a given protocol, or in a context which abides by some policy of the protocol: formulae $\mathsf{Client} \mid \mathsf{Server}$, or $\mathsf{Client} \triangleright \mathsf{Attack}$, would describe a protocol with a client and a server, or a server that might be attacked by a context following the specification of a genuine client.

4 Spatial Logic Applied to Frames

In this section, we establish that logical equivalences induced by the static fragments match static equivalences that have originally been proposed for the applied π-calculus: structural congruence for frames for the intensional fragment, and static equivalence for the extensional fragment.

4.1 Intensional Characterization

The formula $\mathsf{Subst}(x = M)$ below characterizes processes of the form $\{^M/_x\}$, for a given x and a given M. The other two will be useful for our quantifier elimination procedure in Section 6.

$$\mathsf{public\_frame} \triangleq \neg(\top * (\mathbb{I} \wedge \mathsf{H}x.\,\mathbb{I})) \qquad \mathsf{Subst}(x) \triangleq \mathsf{public\_frame} \wedge (\emptyset \oslash x)$$

$$\mathsf{Subst}(x = M) \triangleq \emptyset \oslash x \wedge x = M$$

Lemma 4.1. *For each process P, variable x and term M such that $\mathcal{E} \nvdash x = M$, we have:*

- $P \vDash \mathsf{public\_frame}$ *iff* $\phi(P) \equiv \{^{\boldsymbol{M}}/_{\boldsymbol{x}}\}$ *for some variables $\boldsymbol{x}$ and terms $\boldsymbol{M}$;*
- $P \vDash \mathsf{Subst}(x)$ *iff* $\phi(P) \equiv \{^N/_x\}$ *for some N;*
- $P \vDash \mathsf{Subst}(x = M)$ *iff* $\phi(P) \equiv \{^M/_x\}$.

Proof: First, observe that if $P \vDash \mathbb{I} \wedge \mathsf{H}x.\,\mathbb{I}$ then $P \vDash \mathbb{I}$, so either $\phi(P)$ is a single public active substitution or $\phi(P) \equiv \nu\boldsymbol{n}.\,\sigma$ where σ cannot be split into σ_1 and σ_2 that do not share names in $\boldsymbol{n}$. Moreover, $P \vDash \mathsf{H}x.\,\mathbb{I}$ so we can reveal a fresh variable to obtain a process whose frame still verifies $\mathbb{I}$. This is only possible if P's frame was of the latter form and if the active substitution created by this revelation makes use of some restricted name in $\boldsymbol{n}$. This illustrates one peculiar behaviour of variable revelation: it may reveal a substitution under the scope of an arbitrary number of name restrictions. Now, if $P \vDash \mathsf{public\_frame}$ then we cannot isolate a non-public subframe of P, so P's frame is public.

Reciprocally, if $\phi(P)$ is not public, then there is n such that $\phi \equiv (\nu n.\,\phi_1) * \phi_2$, $n \in \overline{fn}(\phi_1)$ and $\nu n.\,\phi_1 \vDash \mathbb{I}$. Then, for $x \notin dom(P)$, we have $\nu n.\,\phi_1 \equiv \nu x, n.\,(\phi_1 \,|\{^n/_x\})$ so $\nu n.\,\phi_1 \vDash \mathsf{H}x.\,\mathbb{I}$, hence the result.

The other two formulae are straightforward. Observe that the hide operator is used in conjunction with the $\emptyset$ predicate to state both $x \in dom(P)$ and $dom(P) \subseteq \{x\}$. $\square$

Once this basic block is defined, one can easily build up a formula capturing processes in a certain structural congruence class, as expressed by the following theorem:

Theorem 4.2 (Characteristic formulae for frames). *For all frames ϕ there exists a formula F_ϕ in $\mathcal{L}^{stat}_{int}$ such that for all extended processes P, $P \vDash \mathsf{F}_\phi$ if and only if $\phi(P) \equiv \phi$.*

For example, a characteristic formula for $\nu n, s.\Phi$ is $(\mathsf{H}s.\,\mathsf{Subst}(x = \mathrm{enc}(s, y))) * (\mathsf{H}n.\,\mathsf{Subst}(y = \mathrm{pk}(n)))$. Together with Lemma 3.1, this theorem gives a precise definition of logical equivalence induced by $\mathcal{L}^{\text{stat}}_{\text{int}}$ on frames:

Corollary 4.3 (Logical equivalence in $\mathcal{L}^{\text{stat}}_{\text{int}}$). *For all extended processes P and Q, P and Q satisfy the same formulae of $\mathcal{L}^{stat}_{int}$ if and only if $\phi(P) \equiv \phi(Q)$.*

4.2 Extensional Characterization

We will show in this section that logical equivalence for $\mathcal{L}^{\text{stat}}_{\text{ext}}$, or extensional equivalence, coincides with strong static equivalence and that, given a closed frame ϕ, one can construct a formula $\mathsf{F}^{\approx_s}_\phi$ characterizing the equivalence class of ϕ. The right-to-left inclusion is given by the following lemma:

Lemma 4.4. *$\phi \approx^s_s \psi$ and $\phi \vDash A$ implies $\psi \vDash A$.*

For the converse of the above lemma, let us first remark that one can characterize frames whose domains are $\boldsymbol{x}$ using the formula $\emptyset \oslash \boldsymbol{x}$. Then, one can define a characteristic formula $\mathsf{F}^{\approx_s}_\sigma$ for a public frame $\sigma = \{^M/_{\boldsymbol{x}}\}$ of size n:

$$\mathsf{F}^{\approx_s}_\sigma \triangleq \emptyset \oslash \boldsymbol{x} \wedge \bigwedge_{i=1}^{n} x_i = M_i\,.$$

Let ϕ be a closed frame $\nu \boldsymbol{n}.\,\sigma$ with σ a public frame, and consider the formula

$$\phi \;\mathsf{forces}\; U = V \triangleq \mathsf{F}^{\approx_s}_\sigma \mathrel{-\!\!*} \big((U = V) \oslash \boldsymbol{n}\big)\,.$$

Then $\emptyset, v \vDash \phi \;\mathsf{forces}\; U = V$ if and only if $\phi, v \vDash U = V$. Moreover, one can internalize an assumption $\emptyset \vDash A$ in the logic: a process P satisfies $(\emptyset \wedge \neg A) \mathrel{-\!\!*} \bot$ if and only if $\emptyset \vDash A$. We may then derive characteristic formulae for static equivalence on closed frames:

$$\mathsf{F}^{\approx_s}_\phi \triangleq \emptyset \oslash \boldsymbol{x} \wedge \forall t, t'.\,\big((\emptyset \wedge \neg\phi \;\mathsf{forces}\; t = t') \mathrel{-\!\!*} \bot\big) \Leftrightarrow t = t'\,.$$

Theorem 4.5 (Formulae for static equivalence). *For all closed frames ϕ, ψ, $\psi \vDash \mathsf{F}^{\approx_s}_\phi$ if and only if $\phi \approx_s \psi$.*

Using this theorem and the Lemma above, one concludes that two closed frames ϕ and ψ satisfy the same formulae of $\mathcal{L}^{\text{stat}}_{\text{ext}}$ if and only if $\phi \approx_s \psi$.

For the general case, let us consider two non-closed, logically equivalent frames ϕ_1 and ϕ_2. Then, for any closing evaluation context $C \equiv \nu \boldsymbol{n}.\,(\,\cdot \mid \sigma)$, they should both satisfy the formula $\mathsf{F}_\sigma \mathrel{-\!\!*} \mathsf{F}^{\approx_s}_{C[\phi_1]} \oslash \boldsymbol{n}$. Thus, for all closing evaluation contexts C, $C[\phi_2] \vDash \mathsf{F}^{\approx_s}_{C[\phi_1]}$ so $C[\phi_2] \approx_s C[\phi_1]$. This shows $\phi_1 \approx^s_s \phi_2$, so logical equivalence

on frames is indeed strong static equivalence on frames (and thus on processes, by Lemma 3.2).

5 Logical Characterization of Processes

In this section we study the logical equivalence induced by the dynamic intensional fragment.[1] More precisely, we write $P =_L Q$ if P, Q cannot be discriminated by $\mathcal{L}_{\text{spat}}$ formulae and look for a better understanding of $=_L$. We introduce a notion of intensional bisimulation that aims at characterizing $=_L$ by an Ehrenfeucht-Fraïssé game.

Definition 5.1 (Intensional bisimulation). *A relation $\mathcal{R}$ is an intensional bisimulation if $\mathcal{R}$ is symmetric and the following assertions hold for all $(P, Q) \in \mathcal{R}$:*

1. $\phi(P) \equiv \phi(Q)$
2. *if* $P^p \equiv \mathbf{0}$, *then* $Q^p \equiv \mathbf{0}$;
3. *if* $u \in \overline{fnv}(P)$, *then* $u \in \overline{fnv}(Q)$;
4. *if there is* P' *s.t.* $P \equiv \nu u.\, P'$, *then there is* Q' *s.t.* $Q \equiv \nu u.\, Q'$ *and* $P' \mathcal{R} Q'$;
5. *for* $\dagger \in \{*, |\}$, *for all* P_1, P_2, *if* $P \leftrightarrow P_1 \dagger P_2$, *then there are* Q_1, Q_2 *such that* $Q \leftrightarrow Q_1 \dagger Q_2$ *and* $P_i \mathcal{R} Q_i$;
6. *for* $\dagger \in \{*, |\}$, *for all* P_1, P', *if* $P_1 \leftrightarrow P \dagger P'$, *then there are* Q_1, Q' *such that* $Q_1 \leftrightarrow Q \dagger Q'$, $P' \mathcal{R} Q'$ *and* $P_1 \mathcal{R} Q_1$;
7. *if there is* P' *s.t.* $P \rightarrow P'$, *then there is* Q' *s.t.* $Q \rightarrow Q'$ *and* $P' \mathcal{R} Q'$;
8. $\nu u.\, P \mathcal{R} \nu u.\, Q$.

Let us stress the fact that the equivalents of conditions 6 and 8 do not occur in the original intensional bisimulation [14]. Fortunately, in that case the intensional bisimilarity was a congruence, and as a consequence, conditions 6 and 8 were admissible. Note moreover that conditions 6 and 8 do not entail that $\mathcal{R}$ is a congruence (even with $\dagger = |$).

Proposition 5.2. *Let $\mathcal{R}$ be an intensional bisimulation. Then $\mathcal{R} \subseteq =_L$.*

We now give two examples of intensional bisimulations that illustrate that logical equivalence is strictly coarser than structural congruence, and is not even a congruence in general. These bisimulations are based on the notion of shift functions. A unary function symbol f is called a shift function if there are unary function symbols $g_1, \ldots, g_n$ such that $\mathcal{E} \vdash f(\boldsymbol{g}(x)) = \boldsymbol{g}(f(x)) = x$. In the remainder, we assume some fixed shift function f — we will later on consider the case of the equational theory of trees, for which there is no such function. In cryptographic terms, M and $f(M)$ represent two different pieces of information that are deducible one from another by a linear deduction, which explains why they may be indistinguishable for some notion of observer matching our logic.

Let a be some fixed channel name, f a shift function and $\boldsymbol{g}$ such that $\mathcal{E} \vdash f(\boldsymbol{g}(x)) = \boldsymbol{g}(f(x)) = x$. We consider a transformation $\mathsf{shift}_a^f(P)$ that intuitively shifts all term communications of P on channel a using function f — this could be thought of as a reversible noise introduced globally on all communications over a.

[1] We also observed that dynamic extensional fragment does not characterize behavioral equivalence, but due to lack of space we will not develop this point.

Definition 5.2 (Shifted channels). *The transformation* $\mathsf{shift}_a^f(.)$ *is inductively defined as a morphism for all syntactic operators but term inputs and outputs on* a*, for which it is defined as follows:*

$$\mathsf{shift}_a^f(\bar{a}\langle M\rangle.P) \triangleq \bar{a}\langle f(M)\rangle.\mathsf{shift}_a^f(P)$$
$$\mathsf{shift}_a^f(a(x).P) \triangleq a(x).\mathsf{shift}_a^f(P[x\leftarrow \boldsymbol{g}(x)]).$$

Proposition 5.4. *The symmetric closure of* $\mathcal{R} = \{(P, \mathsf{shift}_f^a(P)), P \in \mathcal{P}\}$ *is an intensional bisimulation.*

Propositions 5.2 and 5.4 have some quite unexpected consequences on logical equivalence. First, it entails the following equivalences:

$$\bar{a}\langle 0\rangle =_L \bar{a}\langle f(0)\rangle \text{ and } \bar{a}\langle 0\rangle \mid \bar{b}\langle 0\rangle =_L \bar{a}\langle f(0)\rangle \mid \bar{b}\langle 0\rangle.$$

But the noise introduced on a channel should affect *all* of its communications, as it could otherwise be observed; in particular, it can be proved that:

$$\bar{a}\langle 0\rangle \mid \bar{a}\langle 0\rangle \neg_L \bar{a}\langle f(0)\rangle \mid \bar{a}\langle 0\rangle,$$

which shows that $=_L$ is not a congruence. Such a phenomenon was already observed for the spatial logic of CCS [15], where $=_L$ coincided with structural congruence up to injective renaming. Non-congruence makes the proof of the converse of Proposition 5.2 much harder than the Howe-like method used e.g. by Sangiorgi [14], even in the very simple case of CCS. Indeed, a global quantification over the shift function used for each channel should be expressed at the logical level, which calls on for a quantifiers elimination result; despite some progress in that direction (see next section), we did not succeed in using them to prove that $=_L$ is an intensional bisimulation.

Let now $\sim$ be the smallest equivalence on pairs of terms $M = N$ such that:

$$\begin{array}{ll} \textsc{Symmetry} & M = N \sim N = M \\ \textsc{Shift} & M = N \sim f(M) = f(N) \end{array}$$

where f is a shift function. Let moreover $\equiv'$ be the smallest congruence extending $\equiv$ with the following axiom:

$$\textsc{Test} \qquad \begin{array}{l} \mathit{if}\ test \\ \quad \mathit{then}\ P\ \mathit{else}\ Q \end{array} \equiv' \begin{array}{l} \mathit{if}\ test' \\ \quad \mathit{then}\ P\ \mathit{else}\ Q \end{array}$$

when $test \sim test'$. Then the following result can be established:

Proposition 5.5. $\equiv'$ *is an intensional bisimulation.*

As mentioned above, we did not succeed to prove any completeness result in the general case, but we managed to derive some widget formulae that are sufficiently expressive to characterize $\equiv'$ on, at least, the equational theory of finite trees. Due to lack of space, we skip the quite involved construction.

Theorem 5.6. *Let* $\mathcal{E}$ *be the theory of finite trees. Then for every process* P *there is a formula* $\mathsf{F}_P \in \mathcal{L}$ *such that for all processes* Q*,* $Q \models \mathsf{F}_P$ *if and only if* $Q \equiv' P$*. In particular,* $\equiv'$ *is the same as* $=_L$*.*

On π-calculus processes, *i.e.* processes that contain neither term communications nor conditionals, it can be shown using a similar technique that logical equivalence is structural congruence for all equational theories.

6 Elimination of Term Quantification

This section is devoted to the construction of a translation of any formula A of $\mathcal{L}_{\text{spat}}$ into a logically equivalent one that does not make use of term quantification, thus proving the following theorem:

Theorem 6.1 (Term quantification elimination). *For every closed formula $A \in \mathcal{L}$, there is a formula $[\![A]\!] \in \mathcal{L} \setminus \{\exists\}$ such that $A \Leftrightarrow [\![A]\!]$ is valid.*

$[\![A]\!]$ is defined by structural induction on the formula A. It leaves most of A's structure unchanged, while replacing every subformula $\exists t.\, A'$ with a formula of the form $\mathsf{H}x.\, [\![A']\!]_{\{t \to x\}}$. Hence, the H quantifier is in charge of picking a term M for the new active substitution $\{^M/_x\}$ it reveals, thus mimicking term quantification. Further occurrences of t in the formula will have to be replaced by x. This inductively builds up an *environment frame* placed alongside the actual process that records witnesses of term quantifications, but for which some maintenance work is needed during the translation of a formula. For instance, we will need to copy this environment on each side of a $*$ operator, and on the left-hand side of a $-\!\!*$. Moreover, to follow the semantics of $\exists t.\, A$, one has to make sure that this substitution does not use any hidden name of the process or any of the substitutions belonging to the environment frame.

To keep track of this, the translation will have to be of the form $[\![A]\!]_v$ where v is a valuation $\{\boldsymbol{t} \to \boldsymbol{x}\}$ that lets previously encountered term variables point to their corresponding variables in the domain of the environment frame. The translation thus starts with an empty valuation: $[\![A]\!] \triangleq [\![A]\!]_\emptyset$, and the valuation grows up each time a term quantification is encountered. We write e for the environment $\{\boldsymbol{x} \to \boldsymbol{M}\}$ corresponding to the environment frame $[\![e]\!] \triangleq \{^{\boldsymbol{M}}/_{\boldsymbol{x}}\}$. Moreover, we only consider environments e and translations $[\![A]\!]_v$ where $fv(A, \boldsymbol{M}) \cap \boldsymbol{x} = \emptyset$. Finally, when the domain of e matches the codomain of v, we write $e \circ v$ for the valuation $\{\boldsymbol{t} \to \boldsymbol{M}\}$.

We are now ready to give the inductive lemma we want to prove on $[\![A]\!]_v$:

Lemma 6.2 (Inductive hypothesis). *$P \vDash [\![A]\!]_v$ if and only if there exists Q and e such that $P \equiv Q \,|\, [\![e]\!]$, $fv(Q) \cap dom([\![e]\!]) = \emptyset$, and $Q, e \circ v \vDash A$.*

To meet the requirements of this lemma and make sure that P is indeed the composition of a process Q and an environment frame corresponding to e, we first define a formula Φ_v that will have to be verified at every step of the translation:

$$\Phi_{\{\boldsymbol{t} \to \boldsymbol{x}\}} \triangleq \bigwedge_{x \in \boldsymbol{x}} (\mathsf{Subst}(x) * \neg \copyright x)\,.$$

Lemma 6.3. *For all processes P and valuations v, $P \vDash \Phi_v$ if and only if there exists a process Q and an environment e such that $P \equiv Q \,|\, [\![e]\!]$ and $fv(Q) \cap dom([\![e]\!]) = \emptyset$.*

The translation of all the operators of the logic can be found in the companion technical report [16]. We will give here the proof sketches for the translations of $\exists t.\, A$ and $A_1 * A_2$. The actual translation of term quantification is as follows, where $x_{n+1} \notin fv(A, v)$:

$$[\![\exists t.\, A]\!]_v \triangleq \Phi_v \wedge \mathsf{H}x_{n+1}.\, [\![A]\!]_{v\{t \to x_{n+1}\}}\,.$$

It merely creates a fresh substitution, as the inductive hypothesis on $[\![A]\!]_{v\{t \to x_{n+1}\}}$ suffices to enforce the validity of the new environment frame.

The translation of frame composition needs the valuation frame to be copied in order for it to be present alongside both subprocesses. It is performed as follows:

$$[\![A_1 * A_2]\!]_v \triangleq \Phi_v \wedge \mathcal{U}\boldsymbol{x'}.\,(\Phi_{v'} \wedge \bigwedge_{x \in \boldsymbol{x}} \neg \copyright x \wedge \mathsf{Subst}(\boldsymbol{x'})) \mathrel{-\!\!\circledast} \begin{pmatrix} *_{i=1}^{n} (\mathsf{Subst}(x_i, x'_i) \wedge x'_i = x_i) * \top \\ \wedge [\![A_1]\!]_v * [\![A_2]\!]_{v'} \end{pmatrix}.$$

The idea is to add a new environment frame over fresh variables $\boldsymbol{x'}$. The left-hand side of $-\!\!\circledast$ ensures that this is a valid environment which does not make use of the variables of the previous environment. This is to avoid the possibility of creating active substitutions of the form $\{^{x}/_{x'}\}$ which would not make sense once we separate them from the first environment. The right-hand side makes sure that both environments are the same and distributes them over the interpretations of sub-formulae A_1 and A_2.

7 Conclusion

Related work. Spatial logics for process algebrae with explicit resources have been first studied by Pym [17]. The idea of distributing assertions about knowledge in space using spatial logics has been explored by Mardare [18]. More examples of applications of spatial connectives in cryptographic logics can be found in Kramer's thesis [19]. Hüttel *et al.* gave a logical characterization and characteristic formulae for static equivalence [13] for some classes of equational theories.

Extensions. One natural way to extend the logic could be to consider a weak, several steps version of the $\diamondsuit$ modality. We conjecture that this would allow us to handle the full applied π-calculus with replication, as in the case of ambients [6].

Future work. The decidability status of the logic depends on the considered equational theory, and is already limited by strong undecidability results for the first-order equational logic. At the time of this writing, we are investigating the decidability of the model-checking problem for (fragments of) our logic. A first positive result has been obtained for the static part of the logic without the magic wand operator or the ability to reveal variables, and for a very restricted class of equational theories [16]. A promising line of work would be to try to extend this result to common equational theories, and to allow the use of variable revelation in formulae.

Acknowledgments. We acknowledge, among others, Steve Kremer, Ralf Treinen and Simon Kramer for valuable discussions.

References

1. Reynolds, J.C.: Separation logic: A logic for shared mutable data structures. In: 17th IEEE Symposium on Logic in Computer Science (LICS 2002), pp. 55–74 (2002)
2. Ishtiaq, S., O'Hearn, P.W.: BI as an assertion language for mutable data structures. In: POPL 2001, pp. 14–26 (2001)

3. Gordon, A., Cardelli, L.: Anytime, anywhere: Modal logics for mobile ambients. In: Press, A. (ed.) POPL 2000, pp. 365–377 (2000)
4. Caires, L., Cardelli, L.: A spatial logic for concurrency (part I). Journal of Information and Computation 186(2) (2003)
5. Hirschkoff, D., Lozes, É., Sangiorgi, D.: Minimality results for spatial logics. In: Pandya, P.K., Radhakrishnan, J. (eds.) FSTTCS 2003. LNCS, vol. 2914, pp. 252–264. Springer, Heidelberg (2003)
6. Hirschkoff, D., Lozes, É., Sangiorgi, D.: On the expressiveness of the ambient logic. Logical Methods in Computer Science 2(2) (March 2006)
7. Hirschkoff, D.: An extensional spatial logic for mobile processes. In: CONCUR 2002. LNCS, vol. 3252. Springer, Heidelberg (2002)
8. Abadi, M., Fournet, C.: Mobile values, new names, and secure communication. In: POPL 2001, pp. 104–115 (2001)
9. Milner, R., Parrow, J., Walker, D.: A calculus of mobile processes, i. Inf. Comput. 100(1), 1–40 (1992)
10. Dawar, A., Gardner, P., Ghelli, G.: Expressiveness and complexity of graph logic. Inf. Comput. 205(3), 263–310 (2007)
11. Calcagno, C., Gardner, P., Hague, M.: From separation logic to first-order logic. In: Sassone, V. (ed.) FOSSACS 2005. LNCS, vol. 3441, pp. 395–409. Springer, Heidelberg (2005)
12. Kuncak, V., Rinard, M.: On spatial conjunction as second-order logic. Technical report, MIT CSAIL (October 2004)
13. Hüttel, H., Pedersen, M.D.: A logical characterisation of static equivalence. Electron. Notes Theor. Comput. Sci. 173, 139–157 (2007)
14. Sangiorgi, D.: Extensionality and intensionality of the ambient logics. In: POPL (2001)
15. Caires, L., Lozes, É.: Elimination of quantifiers and undecidability in spatial logics for concurrency. In: Gardner, P., Yoshida, N. (eds.) CONCUR 2004. LNCS, vol. 3170, pp. 240–257. Springer, Heidelberg (2004)
16. Villard, J., Lozes, É., Treinen, R.: A spatial equational logic for the applied pi-calculus. Research Report LSV-08-10, LSV, ENS Cachan, France, 44 pages (March 2008)
17. Pym, D., Tofts, C.: A Calculus and logic of resources and processes. Formal Aspects of Computing 18(4), 495–517 (2006)
18. Mardare, R.: Observing distributed computation. a dynamic-epistemic approach. In: Mossakowski, T., Montanari, U., Haveraaen, M. (eds.) CALCO 2007. LNCS, vol. 4624, pp. 379–393. Springer, Heidelberg (2007)
19. Kramer, S.: Logical Concepts in Cryptography. PhD thesis, École Polytechnique Fédérale de Lausanne (2007)