Spatial Congruence for Ambients is Decidable

Silvano Dal Zilio

Microsoft Research

Abstract. The ambient calculus of Cardelli and Gordon is a process calculus for describing mobile computation where processes may reside within a hierarchy of locations, called ambients. The dynamic semantics of this calculus is presented in a chemical style that allows for a compact and simple formulation. In this semantics, an equivalence relation, called spatial congruence, is defined on the top of an unlabelled transition system.
We show that it is decidable to check whether two ambient calculus processes are spatially congruent or not. This result is based on a natural and intuitive interpretation of ambient processes as edge-labelled unordered trees, which allows us to concentrate on the subtle interaction between two key operators of the ambient calculus, namely restriction, that accounts for the dynamic generation of new location names, and replication, used to encode recursion. The result of our study is the definition of an algorithm to decide spatial congruence and a definition of a normal form for processes that is useful in the proof of important equivalence laws.

1 Introduction

Algebraic frameworks, of which process algebras are one of the most prominent examples, have proved to be a valuable mathematical tool to reason about the behaviour of distributed and communicating systems. Recently, Cardelli and Gordon have proposed a new process algebra, the ambient calculus [3], for describing systems with mobile computations.

In the ambient calculus, processes may reside within a hierarchy of locations, called *ambients*. Each location is a cluster of processes and sub-ambients that can move as a group.

Ambients provide an interesting abstraction that combines, within the same theoretical framework, three essential notions: *mobile computation*, *site* and *mobility*. Mobile computations are computations that can dynamically change the place where they are executed and are continuously active before and after movement, like it is the case with agents. Sites are the location where these computations happen, like processors or routers. Finally, mobility represents a modification in the sites topology that occurs, for instance, with mobile or temporarily disconnected computers, and in the crossing of administrative boundary, like applets crossing a firewall.

Proceedings of ASIAN'00 - 6th Asian Computing Science Conference, LNCS, pp. 1–16, 2000.

Inspired by Berry's and Boudol's Chemical Abstract Machine model [2] and Milner's "chemical" presentation of the π-calculus [13], the dynamic semantics of the ambient calculus is based on a *spatial congruence* relation, denoted $\equiv$, on which the reduction system is based. Spatial congruence identifies processes up to elementary spatial rearrangements and allows a simple and compact presentation of the reduction rules in which the sub-processes having to interact – the redexes in λ-calculus terminology – appear in contiguous position.

This paper reports a proof that spatial congruence, one of the simplest and most basic equivalence between processes, is decidable. That is, the problem of checking whether two processes are spatially congruent, or not, is decidable.

To prove the decidability of spatial congruence, we use a natural and intuitive interpretation of ambient processes as edge-labelled unordered trees. This allows us to concentrate on the subtle interaction between restriction and replication, two key operators of the ambient calculus. Roughly speaking, restriction accounts for the dynamic generation of fresh location names and replication is used to encode recursive behaviours.

The result of our study is twofold. First, we define an effective decision procedure to test spatial congruence. This procedure is based on basic algorithmic on trees and can be easily implemented. Second, we define a normal form for processes and a proof method that demonstrate useful in the proof of important equivalence laws.

The decidability result presented in this paper is useful in many respects. Since spatial congruence plays a central role in the definition of the operational semantics, any attempt to provide a mechanical proof of semantics-based properties will rely on a formal study of spatial congruence and an implementation of a test for equivalence of processes. Interesting examples of semantical properties include proof of equivalences or validity of program transformations.

Another application of our result is the study of the *modal logic for ambients* [4], where spatial congruence is used in the definition of the satisfaction relation. The decidability of spatial congruence is essential in the proof that model checking, for a particular subset of the logic, is decidable.

The outline of the remainder of this paper is as follows. Section 2 introduces the syntax of the ambient calculus and the definition of spatial congruence, and Section 3 defines an interpretation of processes as a certain kind of edge-labelled trees, called *spatial trees*. Section 4 studies a very simple notion of equivalence between spatial trees. We prove that this equivalence is decidable and we define a procedure to test the equivalence of spatial trees. This result relies on the existence of a computable normal form. In Section 5, we relate spatial trees to processes and tree equivalence to spatial congruence. Then, by transferring the results obtained on spatial trees, we prove the decidability of spatial congruence. Before concluding, we use our results to prove some interesting equivalence laws. Complete definition of the calculus and omitted proofs may be found in a long version of this paper [6].

2 The Ambient Calculus

The following tables summarize the syntax of processes and the definition of spatial congruence. For the sake of simplicity, we consider a minimal version of the untyped ambient calculus that includes only mobility primitives, as defined in [3], Section 2. In the extended version of this paper [6], we show that the results and algorithms presented here can be smoothly extended to the full ambient calculus.

The operators of the ambient calculus can be separated into two categories: *spatial constructs*, which describe the "spatial configuration" of processes, and *temporal constructs*, which describe their, possible, dynamic behaviours.

Spatial constructs are composed of restriction, void, composition and replication, which are commonly found in process calculi, and include an original constructor, $n[P]$, called an ambient. In the minimal ambient calculus, temporal constructs are only composed of actions, $act\, n.P$, where n is an *ambient name* and P is a process. In the full ambient calculus, temporal constructs also include the input and output operators defined in [3], which are missing from our presentation. As pointed out in [4], this separation is similar to the distinction between static and dynamic constructs made in CCS [12].

Capabilities and processes:

$act\, n ::=$	capability
$in\, n$	can enter n
$out\, n$	can exit n
$open\, n$	can open n
$P, Q, R ::=$	processes
$(\nu n)P$	restriction
$\mathbf{0}$	void
$P \mid Q$	composition
$!P$	replication
$n[P]$	ambient
$act\, n.P$	action

In a restriction, $(\nu n)P$, the name n is bound with scope P. The set of free names occurring in a process P, written $fn(P)$ is defined as follows, where restriction is the only binders. We identify processes up to consistent renaming of bound names.

Free names, $fn(P)$, of process P:

$fn((\nu n)P) \triangleq fn(P) \setminus \{n\}$	$fn(\mathbf{0}) \triangleq \varnothing$
$fn(P \mid Q) \triangleq fn(P) \cup fn(Q)$	$fn(!P) \triangleq fn(P)$
$fn(n[P]) \triangleq \{n\} \cup fn(P)$	$fn(act\, n.P) \triangleq \{n\} \cup fn(P)$

The rules defining spatial congruence can also be separated in different categories. The first two categories of rule state that it is an equivalence relation

and a congruence. The third category states that parallel composition is an associative and commutative operator with identity element $\mathbf{0}$. Another category specifies properties of replicated processes, $!P$, which acts like an infinite parallel composition of replicas of P. The last category describes scoping rules for the restriction operator, $(\nu n)P$, used to model the dynamic generation of new ambient names.

Spatial congruence: $P \equiv Q$

$P \equiv P$	(Struct Refl)
$Q \equiv P \Rightarrow P \equiv Q$	(Struct Symm)
$P \equiv Q, Q \equiv R \Rightarrow P \equiv R$	(Struct Trans)
$P \equiv Q \Rightarrow (\nu n)P \equiv (\nu n)Q$	(Struct Res)
$P \equiv Q \Rightarrow (P \mid R) \equiv (Q \mid R)$	(Struct Par)
$P \equiv Q \Rightarrow !P \equiv !Q$	(Struct Repl)
$P \equiv Q \Rightarrow n[P] \equiv n[Q]$	(Struct Amb)
$P \equiv Q \Rightarrow act\, n.P \equiv act\, n.Q$	(Struct Action)
$P \mid Q \equiv Q \mid P$	(Struct Par Comm)
$(P \mid Q) \mid R \equiv P \mid (Q \mid R)$	(Struct Par Assoc)
$P \mid \mathbf{0} \equiv P$	(Struct Par Zero)
$!(P \mid Q) \equiv !P \mid !Q$	(Struct Repl Par)
$!\mathbf{0} \equiv \mathbf{0}$	(Struct Repl Zero)
$!P \equiv P \mid !P$	(Struct Repl Copy)
$!P \equiv !!P$	(Struct Repl Repl)
$(\nu n)(\nu m)P \equiv (\nu m)(\nu n)P$	(Struct Res Res)
$(\nu n)\mathbf{0} \equiv \mathbf{0}$	(Struct Res Zero)
$n \notin fn(P) \Rightarrow (\nu n)(P \mid Q) \equiv P \mid (\nu n)Q$	(Struct Res Par)
$n \neq m \Rightarrow (\nu n)m[P] \equiv m[(\nu n)P]$	(Struct Res Amb)
$n \neq m \Rightarrow (\nu n)act\, m.P \equiv act\, m.(\nu n)P$	(Struct Res Action)

Almost every rule in the spatial congruence definition has an equivalent in the corresponding π-calculus equivalence, called *structural congruence*. The most significant differences lies in the axioms for replication, (Struct Repl Par) and (Struct Repl Repl), that are missing in the traditional definition of structural congruence [13]. As a matter of fact, these axioms are also missing in the seminal presentation of the ambient calculus [3], where the relation $\equiv$ is also called structural congruence. These differences have motivated our change in terminology.

Intuitively, the structural congruence relation of the π and ambient calculi should be decidable relations. But the author is not aware of any proof that structural congruence is decidable (or undecidable!) and these results seem very difficult to obtain.

The rules added to spatial congruence are similar to the rules proposed in [7,8] to extend the standard definition of structural congruence in the π-calculus. In

these papers, the authors proved that the resulting equivalence is decidable. Another related work is [10], where Hirschkoff independently proposed a similar extension to structural congruence and proved the decidability result using a more algorithmic approach. We go back to these results in Section 7, where we review related works.

Since the definition of the operational semantics is not needed in our study, we omit the definition of the reduction relation from the presentation. The reader interested in a thorough introduction to the ambient calculus is referred to [3].

3 Spatial Trees

We define an interpretation of spatial processes as a certain kind of edge-labelled unordered trees, which we name *spatial trees*. A spatial tree will represent the hierarchy defined by ambients nesting, using the traditional notion of hierarchy defined by sub-trees. In our intuition, *edges* stands for ambients and are tagged with an ambient name, *nesting* stands for ambient encapsulation and, following our analogy, parallel composition of processes naturally arises as trees sharing the same root. Since it is not possible to define a process containing an unbounded number of nested ambients, we will only consider finite-depth trees.

For convenience, and to avoid confusion, we use a distinct category of names, called *markers*, to model restricted ambient names. Markers are ranged over by $\mathsf{x}, \mathsf{y}, \ldots$ We use η to denote a name, n, or a marker, x. We use $K, L, \ldots$ to denote sets of names, and $\mathsf{X}, \mathsf{Y}, \mathsf{Z} \ldots$ for sets of markers.

A *multiplicity*, μ, is either 1 or ∞. A *cone*, C, is either the empty vector, written ϵ, an action: $\mu act\, \eta.T$, or an edge: $\mu\eta[T]$ or $!\mathsf{X}.T$, where T is a spatial tree and X is a non-empty set of markers.

A *spatial tree* is a finite vector of cones, $C_1 + \cdots + C_k$, also written $\sum_{i \in 1..k} C_i$. The $+$ operator is commutative and associative, with identity element ϵ; spatial trees are identified up to these equations.

Cones and spatial trees:

$\mu ::=$	multiplicity
$\quad 1$	$\quad$ single
$\quad \infty$	$\quad$ infinite
$C ::=$	cone
$\quad \epsilon$	$\quad$ empty vector
$\quad \mu act\, \eta.T$	$\quad$ action
$\quad \mu\eta[T]$	$\quad$ edge tagged η
$\quad !\mathsf{X}.T$	$\quad$ replicated edge with markers X
$S, T ::=$	spatial trees
$\quad C_1 + \cdots + C_k$	$\quad$ vector of cones

Cones are a special type of spatial trees. The cone $!\mathsf{X}.T$ represents an infinite copy of the tree T such that, in each copy, the elements of X are replaced with

fresh markers. In an edge, $!\mathsf{X}.T$, the markers in X are bound with scope T. Spatial trees are identified up to consistent renaming of bound markers.

Free markers, $fm(T)$, of tree T:

$fm(\epsilon) \triangleq \varnothing$
$fm(\mu n[T]) \triangleq fm(T)$
$fm(\mu \mathsf{x}[T]) \triangleq fm(T) \cup \{\mathsf{x}\}$
$fm(S + T) \triangleq fm(S) \cup fm(T)$
$fm(!\mathsf{X}.T) \triangleq fm(T) \setminus \mathsf{X}$
$fm(\mu act\, n.T) \triangleq fm(T)$
$fm(\mu act\, \mathsf{x}.T) \triangleq fm(T) \cup \{\mathsf{x}\}$

We write $T\{n{\leftarrow}\mathsf{x}\}$ for the capture-avoiding substitution of the marker x for each occurrences of the name n in the tree T. For convenience, we extend the replication constructor, $!\mathsf{X}.T$, to the empty set of markers as follows:

$!\varnothing.\epsilon \triangleq \epsilon$
$!\varnothing.\mu act\, \eta.T \triangleq \infty act\, \eta.T$
$!\varnothing.\mu \eta[T] \triangleq \infty \eta[T]$
$!\varnothing.!\mathsf{X}.T \triangleq !\mathsf{X}.T$
$!\varnothing.(S + T) \triangleq !\varnothing.S + !\varnothing.T$

Proposition 3.1. *We have* $!\varnothing.!\varnothing.T = !\varnothing.T$.

Since we have a notion of free and bound markers, we can define a notion of connected tree, that is, tree whose sub-trees share mutual markers.

Connected trees:

A tree $\sum_{i \in 1..p} C_i$ is connected if and only if there are no partitions of $1..p$ into two non-empty subsets, I, J, such that $fm(\sum_{i \in I} C_i) \cap fm(\sum_{i \in J} C_j) = \varnothing$.

Using this definition, we can compute for each tree the (unique) set of its connected sub-trees as follows. For all tree $T \triangleq \sum_{i \in 1..p} C_i$ we can construct a graph as follows.

(1) Let $\mathcal{N}$ be the set of cones $\{C_1, \ldots, C_p\}$.
(2) Let $\mathcal{G}$ be the graph with nodes in $\mathcal{N}$ and edges between nodes that have at least one common free marker.
(3) Compute the connected components of the graph $\mathcal{G}$, say $\mathcal{G}_1, \ldots, \mathcal{G}_k$.

The connected parts of T, written $conn(T)$, is the set $\{T_1, \ldots, T_k\}$ such that for all $i \in 1..k$ the spatial tree T_i is the vector of the cones included in $\mathcal{G}_i$. Basic properties of the connected components of a spatial tree are:

Proposition 3.2. *If $\{T_1, \ldots, T_p\}$ is the set of connected components, $conn(T)$, of a tree T then $T = T_1 + \cdots + T_p$, and for each $j \in 1..p$ the tree T_j is connected.*

4 Equality of Spatial Trees

We define a reduction relation between trees, $\mathsf{X} \vdash S \to T$, parameterised by a set of markers, X, called the *effect* of the reduction. This reduction relation captures the essential intuitions of the equivalence between edge-labelled trees and every rule in its definition corresponds to basic axioms of spatial congruence. For instance, rule (Red Zero) implies that "empty cones can be forgotten" and corresponds to rule (Struct Par Zero) of structural congruence. Likewise, rule (Red Add Edge) implies that "two infinite copies of a sub-tree can be replaced by only one copy" and corresponds to rule (Struct Repl Copy). We also define the equivalence induced by $\to$, in almost the same way the λ-calculus reduction relation induces β-equivalence.

In this section, we prove that every spatial tree can be factorised to an irreducible form, also called a *normal form*, which is not related to the reduction sequence used to compute it. Normal forms provide us with a unique representative for each tree and, more significantly, allow us to define a formal procedure to test the equivalence of trees, a key result in the proof of the decidability of spatial congruence.

Reduction: $\mathsf{X} \vdash S \to T$

(Red Zero)
$$\frac{}{\varnothing \vdash T + \epsilon \to T}$$

(Red Add Edge)
$$\frac{}{\varnothing \vdash \infty\eta[T] + \mu\eta[T] \to \infty\eta[T]}$$

(Red Add Action)
$$\frac{}{\varnothing \vdash \infty act\,\eta.T + \mu act\,\eta.T \to \infty act\,\eta.T}$$

(Red Add Repl)
$$\frac{}{\varnothing \vdash !\mathsf{X}.T + !\mathsf{X}.T \to !\mathsf{X}.T}$$

(Red Copy)
$$\frac{}{\mathsf{X} \vdash !\mathsf{X}.T + T \to !\mathsf{X}.T}$$

(Red Sub)
$$\frac{\mathsf{X} \vdash T \to S \qquad \mathsf{X} \subseteq \mathsf{Y}}{\mathsf{Y} \vdash T \to S}$$

(Red Repl)
$$\frac{\mathsf{X} \vdash T \to S \qquad (\mathsf{Z} = \mathsf{Y} \cap fm(S))}{\mathsf{X} \setminus \mathsf{Y} \vdash !\mathsf{Y}.T \to !\mathsf{Z}.S}$$

(Red η)
$$\frac{\mathsf{X} \vdash T \to S \qquad (\eta \notin \mathsf{X})}{\mathsf{X} \vdash \mu\eta[T] \to \mu\eta[S]}$$

(Red $+$)
$$\frac{\mathsf{X} \vdash T \to S \qquad (fm(R) \cap \mathsf{X} = \varnothing)}{\mathsf{X} \vdash T + R \to S + R}$$

(Red Action)
$$\frac{\mathsf{X} \vdash T \to S}{\mathsf{X} \vdash \mu act\,\eta.T \to \mu act\,\eta.S}$$

The rules for reduction can be separated in two categories. Rules (Red Zero) to (Red Copy) that involve two cones, or *critical pairs*, of which only (Red Copy)

can extend the effect. Rules (Red Repl) to (Red Action), the *structural rules*, which states that the relation $\rightarrow$ is compositional.

In a reduction, $\mathsf{X} \vdash S \rightarrow T$, the effect X records the markers that must not appear free in the result of the reduction.

We can derive an equivalent of rules (Red Add Repl), (Red Copy) and (Red Repl), for the special case where the set X is empty.

- $\varnothing \vdash !\varnothing.T + !\varnothing.T \rightarrow^* !\varnothing.T$.
- $\varnothing \vdash !\varnothing.T + T \rightarrow^* !\varnothing.T$.
- If $\varnothing \vdash T \rightarrow S$ then $\varnothing \vdash !\varnothing.T \rightarrow^* !\varnothing.S$.

Next, we define the equivalence relation on spatial trees induced by $\rightarrow$.

Equivalence relation between trees: $S \sim_{\mathsf{X}} T$ and $S \approx T$

The relation $\sim_{\mathsf{X}}$ is the smallest reflexive, symmetric and transitive relation such that if $\mathsf{X} \vdash S \rightarrow T$ then $S \sim_{\mathsf{X}} T$. The relation $\approx$ is such that $S \approx T$ if and only if there exist two finite injective mappings, σ_1, σ_2, and a set of markers X such that $dom(\sigma_1) = fm(S)$ and $dom(\sigma_2) = fm(T)$ and $S\sigma_1 \sim_{\mathsf{X}} T\sigma_2$.

From the structural rules of $\rightarrow$, it is trivial to show that $\sim_{\mathsf{X}}$ is a congruence and that if $\mathsf{Y} \subseteq \mathsf{X}$ then $\sim_{\mathsf{Y}} \subseteq \sim_{\mathsf{X}} \subseteq \approx$. Basic properties of $\approx$ are:

Proposition 4.1. *The relation $\approx$ satisfies the congruence properties, that is, if $(fm(S) \cup fm(T)) \cap fm(R) = \varnothing$ and $S \approx T$ then $S + R \approx T + R$; if $S \approx T$ then $\mu n[S] \approx \mu n[T]$; if $S \approx T$ then $\mu act\, \eta.S \approx \mu act\, \eta.T$.*

We prove that the reduction relation on spatial trees is locally confluent.

Lemma 4.2. *If $\mathsf{X}_1 \vdash T \rightarrow T_1$ and $\mathsf{X}_2 \vdash T \rightarrow T_2$ then there exists a tree S such that $\mathsf{X}_1 \cup \mathsf{X}_2 \vdash T_1 \rightarrow^* S$ and $\mathsf{X}_1 \cup \mathsf{X}_2 \vdash T_2 \rightarrow^* S$.*

Proof. By induction on the derivation of $\mathsf{X}_1 \vdash T \rightarrow T_1$. For the sake of brevity, we only consider the cases in which the two reductions originate from critical pairs that share a common cone. The complete proof can be found in [6].

In the particular case considered here, the tree T must be a composition, $R_1 + C + R_2 + T'$, where $\mathsf{X}_i \vdash R_i + C \rightarrow S_i$ and $T_i = S_i + R_j + T'$ for each $i \in \{1, 2\}$ and $i \neq j$. These must have been derived from (Red $+$) and therefore we have the side condition $(\star)$ $fm(R_2 + T') \cap \mathsf{X}_1 = fm(R_1 + T') \cap \mathsf{X}_2 = \varnothing$.

The proof follows by a case analysis on the rules used to derive the two reductions $\mathsf{X}_1 \vdash R_1 + C \rightarrow S_1$ and $\mathsf{X}_2 \vdash R_2 + C \rightarrow S_2$.

(Red Zero)-(Red Zero) Then $C = \epsilon$ and $T_1 = T_2 = R_1 + R_2 + S$. Trivial.

(Red Add Edge)-(Red Add Edge) Then $R_1 = \mu_1\eta[R]$, $R_2 = \mu_2\eta[R]$, $S_1 = S_2 = \infty\eta[R]$ and C is a cone $\mu\eta[R]$ for some multiplicities μ_1, μ_2, μ such that $\infty \in \{\mu, \mu_i\}$ for each $i \in \{1, 2\}$. Trivial. Case (Red Add Repl)-(Red Add Repl) is similar.

(Red Add Edge)-(Red Copy) Then C is an edge $\mu\eta[R]$ and it must be the case that $R_1 = \mu_1\eta[R]$ and $R_2 = !\mathsf{X}.\mu\eta[R]$, where μ or μ_1 is an infinite multiplicity and $\mathsf{X} \subseteq \mathsf{X}_2$. This case is impossible since (Red Copy) implies that $fm(\eta(R_1)) \cap \mathsf{X}_2 \neq \varnothing$, which conflicts with the side condition $(\star)$. Case (Red Copy)-(Red Add Edge) is similar.

(Red Add Repl)-(Red Copy) Then $T = R + !\mathsf{X}.R + !\mathsf{X}.R + T'$ and $S_1 = R + !\mathsf{X}.R$ and $S_2 = !\mathsf{X}.R + !\mathsf{X}.R$, where $\mathsf{X}_1 = \varnothing$ and $\mathsf{X}_2 = \mathsf{X}$. By (Red Copy) and (Red +), we have that $\mathsf{X} \vdash T_1 \to !\mathsf{X}.R + T'$. By (Red Add Repl) and (Red +), $\varnothing \vdash T_2 \to !\mathsf{X}.R + T'$, as required.

(Red Copy)-(Red Copy) Then $C = !\mathsf{X}.R$ for some tree R and sets of markers X, such that $R_1\{\mathsf{X} \leftarrow \mathsf{X}_1\} = R_2\{\mathsf{X} \leftarrow \mathsf{X}_2\} = R$ and $(\mathsf{X}_1 \cup \mathsf{X}_2) \cap fm(T') = \varnothing$. By (Red Copy) and (Red +), we get that $\mathsf{X}_2 \vdash T_1 \to !\mathsf{X}.R + T'$ and $\mathsf{X}_1 \vdash T_2 \to !\mathsf{X}.R + T'$, as required. □

Since it can be proved that the reduction relation is decreasing, in the sense that the number of symbols in the definition of a tree decreases after a reduction, there can only be a finite number of reductions from any tree and we have:

Theorem 4.3. *The relation $\to$ is strongly normalizing and confluent.*

We can define an algorithm to decide the equivalence of spatial trees based on this result. To decide if $S_1 \sim_{\mathsf{X}} S_2$, you compute the normal form of S_1 and S_2, that is, the spatial trees S_1', S_2' such that $\mathsf{X} \vdash S_i \to^* S_i'$ and S_i' is irreducible for each $i \in 1..2$. By Theorem 4.3, these trees exist and can be computed using a finite number of reductions. Then, you verify whether the normal forms are equal.

Theorem 4.4. *The equivalences $\sim_{\mathsf{X}}$ and $\approx$ are decidable.*

Proof. To decide if $S_1 \sim_{\mathsf{X}} S_2$, you compute the normal form of S_1 and S_2, that is, the spatial trees S_1', S_2' such that $\mathsf{X} \vdash S_i \to^* S_i'$ and S_i' is irreducible for each $i \in 1..2$. By Theorem 4.3, these trees exist and can be computed using a finite number of reductions. Then, you verify whether the normal forms are equal. This amount to test the equality of trees up to the renaming of bound markers and the associativity-commutativity of $+$. Since this is a decidable problem, we get that $\sim_{\mathsf{X}}$ is decidable.

To decide if $S_1 \approx S_2$, you test whether $S_1\sigma_1 \sim_{\mathsf{X}} S_2\sigma_2$ for each finite injective mapping σ_1, σ_2 and for each set X such that such that $dom(\sigma_1) = fm(S_1)$ and $dom(\sigma_2) = fm(S_2)$ and $\mathsf{X} \subseteq fm(S_1\sigma_1) \cup fm(S_2\sigma_2)$. It is sufficient to consider mappings σ_1, σ_2 that have their image in a fresh set of markers that has the cardinality of $fm(S_1) \cup fm(S_2)$. Since the sets $fm(S_2)$ and $fm(S_1)$ are finite, and since $\sim_{\mathsf{X}}$ is decidable, we get that $\approx$ is decidable. □

Using the strong normalization property, we can define a notion of *normal form* for trees. For all spatial trees T, there is a tree T' such that $T \approx T'$ and:

$$T' \triangleq \sum_{i_1 \in I_1} \mu_{i_1}\eta_{i_1}[T_{i_1}] + \sum_{i_2 \in I_2} !\mathsf{X}_{i_2}.T_{i_2} + \sum_{i_3 \in I_3} \mu_{i_3}\, act\, n_{i_3}.T_{i_3} \tag{4.1}$$

Where (1) I_1, I_2 and I_3 are finite and pairwise disjoint sets of indices; (2) for all indices $i \in \bigcup_{j\in 1..3} I_j$, the tree T_j is in normal form; (3) for all $i, j \in I_1$, if $\eta_i = \eta_j$ then $T_i \not\sim_\varnothing T_j$ or $\mu_i = \mu_j = 1$; and (4) for all $i, j \in I_2$, if $!\mathsf{X}_i.T_i \sim_\varnothing !\mathsf{X}_j.T_j$ then $i = j$.

It is worth mentioning that, contrary to a typical situation with normal forms found in other theoretical frameworks, the normal form given in (4.1) is syntactically smaller than the spatial trees associated with it.

5 Relation Between Trees and Processes

We define the tree semantics of processes, that is, a mapping from ambient processes to spatial trees, and we relate spatial congruence with the equivalence on spatial trees. Then, by transferring the decidability result obtained in the previous section, we infer the decidability of spatial congruence. This semantics extends a similar definition given in an extended version of [4] for a calculus without name restriction.

In the definition of the tree semantics of processes, we use a new operation on trees called *exponentiation*, $exp(T)$, obtained as the outcome of replicating every connected part of T. More formally, the exponentiation of a tree T, is the composition $!\mathsf{X}_1.T_1 + \cdots + !\mathsf{X}_p.T_p$ where $\{T_1, \ldots, T_p\} = conn(T)$ are the connected parts of T and $\mathsf{X}_i = fm(T_i)$ for each $i \in 1..p$.

Tree semantics:

$[\![\mathbf{0}]\!] \triangleq \epsilon$	(Zero)
$[\![act\,\eta.P]\!] \triangleq 1act\,\eta.[\![P]\!]$	(Action)
$[\![n[P]]\!] \triangleq 1n[[\![P]\!]]$	(Amb)
$[\![!P]\!] \triangleq exp([\![P]\!])$	(Repl)
$fm([\![P]\!]) \cap fm([\![Q]\!]) = \varnothing \Rightarrow [\![P \mid Q]\!] \triangleq [\![P]\!] + [\![Q]\!]$	(Par)
$\mathsf{x} \notin fm([\![P]\!]) \Rightarrow [\![(\nu n)P]\!] \triangleq [\![P]\!]\{n \leftarrow \mathsf{x}\}$	(Res)

In the same way tree composition, $S + T$, corresponds to parallel composition for processes, exponentiation is the analogue of replication, Furthermore, it is possible to prove properties of this derived operator corresponding to rules (Struct Repl), (Struct Repl Par), (Struct Repl Repl) and (Struct Repl Copy) respectively.

Proposition 5.1.

(1) *If* $S \approx T$ *then* $exp(S) \approx exp(T)$.
(2) *If* $fm(S) \cap fm(T) = \varnothing$ *then* $exp(S + T) \approx exp(S) + exp(T)$.
(3) *The function* $exp(.)$ *is idempotent:* $exp(exp(T)) = exp(T)$.
(4) *For all spatial trees* T *we have* $exp(T) + T \approx exp(T)$.

Using these properties, it is easy to prove that the axiomatisation of spatial congruence is sound.

Lemma 5.2. *If $P \equiv Q$ then $[\![P]\!] \approx [\![Q]\!]$.*

Next, we prove the completeness of our axiomatisation. We start by defining an inverse mapping from trees to processes.

Process semantics of trees:

$(\!\|\epsilon\|\!) \triangleq \mathbf{0}$	(Empty)
$(\!\|1act\, n.T\|\!) \triangleq act\, n.(\!\|T\|\!)$	(Action 1)
$(\!\|\infty act\, n.T\|\!) \triangleq !act\, n.(\!\|T\|\!)$	(Action ∞)
$(\!\|1n[T]\|\!) \triangleq n[(\!\|T\|\!)]$	(Edge 1)
$(\!\|\infty n[T]\|\!) \triangleq !n[(\!\|T\|\!)]$	(Edge ∞)
$(\!\|!\{\mathsf{x}_1, \ldots, \mathsf{x}_p\}.T\|\!) \triangleq !(\nu n_1)\ldots(\nu n_p)(\!\|T\{\mathsf{x}_1 \leftarrow n_1\}\ldots\{\mathsf{x}_p \leftarrow n_p\}\|\!)$	(Repl)
where $\{n_1, \ldots, n_p\}$ is a set of pairwise distinct names not free in $(\!\|T\|\!)$.	
$(\!\|S + T\|\!) \triangleq (\!\|S\|\!) \mid (\!\|T\|\!)$	(Sum)

The composition of the two interpretations $(\!|.|\!)$ and $[\![.]\!]$ differs from the identity over processes. For instance, we have $(\!|[\![(\nu u)\mathbf{0}]\!]|\!) = (\!|[\![\mathbf{0}]\!]|\!)$. Nonetheless, we can draw a simple relation between a process and the meaning of its interpretation. See Proposition 5.3 (2) below.

Let the *meaning* of a tree T, written $mean(T)$, be the process $(\nu K)(\!|T\sigma|\!)$, where σ is a bijection from $fm(T)$ to a set of fresh names and K is $\sigma(fm(T))$, the image of σ. Properties of $mean(.)$ are:

Proposition 5.3. (1) *If $S \approx T$ then $mean(S) \equiv mean(T)$ and* (2) *for all processes P we have $mean([\![P]\!]) \equiv P$.*

Let P and Q be two processes such that $[\![P]\!] \approx [\![Q]\!]$. By Proposition 5.3 (1), $mean([\![P]\!]) \equiv mean([\![Q]\!])$. By Proposition 5.3 (2), $P \equiv mean([\![P]\!])$ and $Q \equiv mean([\![Q]\!])$. Hence, by transitivity of spatial congruence, $P \equiv Q$. This proves that our interpretation of processes as spatial trees is *complete*, that is:

Lemma 5.4. *If $[\![P]\!] \approx [\![Q]\!]$ then $P \equiv Q$.*

Lemmas 5.2 and 5.4 state a full abstraction result between ambient processes and spatial trees with respect to the equivalences $\equiv$ and $\approx$ respectively. Therefore, every problem in the ambient calculus can be expressed in terms of problem on spatial trees. For instance, to decide whether $P \equiv Q$, a possible method is to compute $[\![P]\!]$ and $[\![Q]\!]$ and to verify if they are equivalent. By Theorem 4.4, this problem is decidable. It follows that:

Theorem 5.5. *The relation $\equiv$ is decidable.*

Using our interpretation of processes as spatial trees, we obtain another result for free. Indeed, through Lemma 5.4 and the normal form for spatial trees given in Section 4, we obtain a normal form for ambient processes that is unique up to very simple spatial transformations, that is, commutativity-associativity of

the parallel composition and the reordering of restrictions. If L is a finite set of names $\{n_1, \ldots, n_p\}$, we write $(\nu L)P$ for the process $(\nu n_1) \ldots (\nu n_p)P$. For all processes P, there is a process P' such that $P \equiv P'$ and:

$$P' \stackrel{\Delta}{=} (\nu L)(\prod_{i_1 \in I_1} n_{i_1}[Q_{i_1}] \mid \prod_{i_2 \in I_2} !n_{i_2}[Q_{i_2}] \mid \prod_{i_3 \in I_3} !(\nu L_{i_3})Q_{i_3} \\ \mid \prod_{i_4 \in I_4} act\, n_{i_4}.Q_{i_4} \mid \prod_{i_5 \in I_5} !act\, n_{i_5}.Q_{i_5}) \tag{5.1}$$

Where (1) the set of indices $I_1, \ldots, I_5$ are finite and pairwise disjoint; (2) for all $i \in \bigcup_{j \in 1..5} I_j$, the processes Q_j are in normal form; (3) for all $i \in I_1, j \in I_2$, if $n_i = n_j$ then $Q_i \not\equiv Q_j$; (4) for all $i, j \in I_3$, if $(\nu L_i)Q_i \equiv (\nu L_j)Q_j$ then $i = j$.

6 Applications

We can apply the results given in this paper to prove interesting equivalence laws like, for example, the one listed in Lemma 6.1 below. The laws examined in this section are particularly interesting because they are, at the same time, very useful in the formal study of the ambient calculus, and very difficult to prove directly, that is, for example, using an induction on derivations of the form $P \equiv Q$.

In the particular example of Lemma 6.1, we study three equivalence laws extracted from the presentation of Cardelli's and Gordon's modal logic for ambients [4], a logic used to describe properties of processes. These laws are essential to prove the soundness of several axioms of the logic.

An interesting fact is that we follow a similar proof technique in each case. We start by using the full abstraction result obtained in Section 5 to restate the problem in terms of equivalence between spatial trees, then, we prove the desired equivalence by exhibiting a property invariant by the reduction relation over trees.

Lemma 6.1.

(1) *If* $P \mid Q \equiv \mathbf{0}$ *then* $P \equiv \mathbf{0}$ *and* $Q \equiv \mathbf{0}$.
(2) *If* $n[P] \equiv Q \mid R$ *then either* $Q \equiv n[P]$ *and* $R \equiv \mathbf{0}$, *or* $Q \equiv \mathbf{0}$ *and* $R \equiv n[P]$.
(3) *If* $m[P] \equiv n[Q]$ *then* $m = n$ *and* $P \equiv Q$.

Proof. We only sketch the proof for case (1). Proofs for the other cases are similar and can be found in [6].

By the full abstraction result stated in Section 5, this problem is equivalent to prove that for every spatial trees, S, T, if $S + T \approx \epsilon$ then $S \approx \epsilon$ and $T \approx \epsilon$. By Theorem 4.3, since ϵ is an irreducible spatial tree with no free markers, this is also equivalent to prove that if $S + T \to^* \epsilon$ then $S \to^* \epsilon$ and $T \to^* \epsilon$.

The proposition follows by showing that for any finite set of cones, $(C_i)_{i \in I}$, if $\mathsf{X} \vdash \sum_{i \in I} C_i \to^* \epsilon$ then $\mathsf{X} \vdash C_i \to^* \epsilon$ for all $i \in I$. This can be proved by an easy induction on the derivation of $\mathsf{X} \vdash \sum_{i \in I} C_i \to^* \epsilon$.

Now, assume $P \mid Q \equiv \mathbf{0}$. By Lemma 5.2, $[\![P]\!] + [\![Q]\!] \approx \epsilon$. Hence, there exists a set X such that $[\![P]\!] + [\![Q]\!] \sim_{\mathsf{X}} \epsilon$. By Lemma 4.2, and since ϵ is an irreducible

spatial trees, we get that $\mathsf{X} \vdash [\![P]\!] + [\![Q]\!] \rightarrow^* \epsilon$, and therefore, $\mathsf{X} \vdash [\![P]\!] \rightarrow^* \epsilon$ and $\mathsf{X} \vdash [\![Q]\!] \rightarrow^* \epsilon$. Hence, $[\![P]\!] \approx \epsilon$ and $[\![Q]\!] \approx \epsilon$. By Lemma 5.4, $P \equiv Q \equiv \mathbf{0}$. □

Next, we prove three equivalence laws that validate the distribution of name restriction over void, ambient, and parallel composition. These laws play an essential role in the definition of axioms for an extension of the ambient modal logics with an operator for name restriction [5].

Lemma 6.2.

(1) *If* $(\nu n)P \equiv \mathbf{0}$ *then* $P \equiv \mathbf{0}$
(2) *If* $(\nu n)P \equiv m[Q]$ *then there exists* R *such that* $P \equiv m[R]$ *and* $Q \equiv (\nu n)R$.
(3) *If* $(\nu n)P \equiv Q \mid R$ *then there exist two processes,* P_1, P_2, *such that* $P \equiv P_1 \mid P_2$, *and* $Q \equiv (\nu n)P_1$, *and* $R \equiv (\nu n)P_2$.

Proof. Proof of (1) is similar to the proof of Lemma 6.1 (1) sketched above. In particular, we use the property that for any finite set of cones, $(C_i)_{i\in I}$, if $\mathsf{X} \vdash \sum_{i\in I} C_i \rightarrow^* \epsilon$ then $\mathsf{X} \vdash C_i \rightarrow^* \epsilon$ for all $i \in I$.

For (2), assume $(\nu n)P \equiv m[Q]$. By Lemma 5.2, $[\![(\nu n)P]\!] \approx [\![m[Q]]\!]$. Therefore, for every fresh marker, x, we have $[\![P]\!]\{n{\leftarrow}\mathsf{x}\} \approx 1m[[\![Q]\!]]$. By definition of $\approx$, there exist two finite injective mappings, σ_1, σ_2 and a set X such that $[\![P]\!]\sigma_1\{n{\leftarrow}\mathsf{y}\} \sim_{\mathsf{X}} 1m[[\![Q]\!]\sigma_2]$ where $\mathsf{y} = \sigma_1(\mathsf{x})$. Let S be the normal form of $[\![P]\!]\sigma_1$. Therefore, $S \approx [\![P]\!]\sigma_1 \sim_{\mathsf{Y}} 1m[[\![Q]\!]\sigma_2\{\mathsf{y}{\leftarrow}n\}]$. Since S is in normal form, it must be the case that $S = 1m[T]$ for some tree T such that $T \approx [\![Q]\!]\sigma_2\{\mathsf{y}{\leftarrow}n\}$. Let R be the process $mean(T)$. Then, $[\![m[R]]\!] \approx S \approx [\![P]\!]$ and, by Lemma 5.4, $m[R] \equiv P$. Moreover, $[\![(\nu n)R]\!] \approx T\{n{\leftarrow}\mathsf{y}\} \approx [\![Q]\!]$. By Lemma 5.4, $(\nu n)R \equiv Q$, as required.

For (3), Assume $(\nu n)P \equiv Q \mid R$. By Lemma 5.2, $[\![Q \mid R]\!] \approx [\![(\nu n)P]\!]$. Therefore, for every fresh marker, x, we have $[\![Q]\!] + [\![R]\!] \approx [\![P]\!]\{n{\leftarrow}\mathsf{x}\}$, where $fm([\![Q]\!]) \cap fm([\![R]\!]) = \varnothing$. By definition, there exist two finite injective mappings, σ_1, σ_2 and a set X such that $[\![Q]\!]\sigma_1 + [\![R]\!]\sigma_1 \sim_{\mathsf{X}} [\![P]\!]\sigma_2\{n{\leftarrow}\mathsf{y}\}$, where $\mathsf{y} = \sigma_2(\mathsf{x})$.

Let S, T and O be the normal forms of $[\![Q]\!]\sigma_1$, $[\![R]\!]\sigma_1$ and $[\![P]\!]\sigma_2$ respectively. Hence, $S + T \sim_{\mathsf{Y}} O\{n{\leftarrow}\mathsf{y}\}$ for some set of markers Y such that $\mathsf{X} \subseteq \mathsf{Y}$ and with the side condition: $fm(S) \cap fm(T) = \varnothing$. Assume $\sum_{i\in 1..p} C_i$ is the, common, normal form of $S + T$ and $O\{n{\leftarrow}\mathsf{y}\}$. Since S, T and O are normal forms, there exist three families of spatial trees in normal form, $(S_i)_{i\in 1..p}$, $(T_i)_{i\in 1..p}$, and $(O_i)_{i\in 1..p}$, such that:

(1) $S = \sum_{i\in 1..p} S_i$ and $T = \sum_{i\in 1..p} T_i$ and $O = \sum_{i\in 1..p} O_i$.
(2) $S_i + T_i \sim_{\mathsf{Y}} O_i\{n{\leftarrow}\mathsf{y}\}$ for each $i \in 1..p$.
(3) $\mathsf{Y} \vdash S_i + T_i \rightarrow^* C_i$ and $\mathsf{Y} \vdash O_i\{n{\leftarrow}\mathsf{y}\} \rightarrow^* C_i$ for each $i \in 1..p$.

The proof follows by constructing the spatial trees corresponding to the processes P_1, P_2. We proceed by defining two families of trees, $(S'_i)_{i\in 1..p}$ and $(T'_i)_{i\in 1..p}$, and proving that $O_i \sim_{\mathsf{Y}} (S'_i + T'_i)$, and $S_i \sim_{\mathsf{Y}} S'_i\{n{\leftarrow}\mathsf{y}\}$, and $T_i \sim_{\mathsf{Y}} T'_i\{n{\leftarrow}\mathsf{y}\}$ for each $i \in 1..p$. The trees S'_i and T'_i are defined by case analysis on the definition of C_i.

(Empty) Then $C_i = \epsilon$. Since S, T and O are in normal form, it must be the case that $S_i = T_i = O_i = \epsilon$. Let $S'_i = T'_i = \epsilon$. Trivial.

(Action) Then $C_i = \mu act\,\eta.S'$. Since O_i is in normal form, it must be the case that $O_i\{n{\leftarrow}\mathsf{y}\} = \mu act\,\eta.S'$. Let $S'_i = S_i\{\mathsf{y}{\leftarrow}n\}$ and $T'_i = T_i\{\mathsf{y}{\leftarrow}n\}$. Trivial. We follow the same definition for the cases where C_i is an edge.

(Repl) Then $C_i = !\mathsf{Y}'.T'$. Since O_i is in normal form, it must be the case that $O_i\{n{\leftarrow}\mathsf{y}\} = !\mathsf{Y}'.T'' + T'''$ and $T' \sim_{\mathsf{Y}\cup\mathsf{Y}'} T''$. Since S_i and T_i are in normal form and $fm(S_i) \cap fm(T_i) = \varnothing$, it must be the case that either (1) $S_i \sim_{\mathsf{Y}} C_i$ or (2) $T_i \sim_{\mathsf{Y}} C_i$. Assume we are in case (1). Let $S'_i = (S_i + T''')\{\mathsf{y}{\leftarrow}n\}$ and $T'_i = T_i\{\mathsf{y}{\leftarrow}n\}$. Then $S'_i\{n{\leftarrow}\mathsf{y}\} \sim_{\mathsf{Y}} C_i + T''' \sim_{\mathsf{Y}} C_i \sim_{\mathsf{Y}} S_i$, and $(S'_i + T'_i) = (S_i + T_i + T''')\{\mathsf{y}{\leftarrow}n\} \sim_{\mathsf{Y}} (C_i + T''')\{\mathsf{y}{\leftarrow}n\} \sim_{\mathsf{Y}} O_i$, as required.

An easy induction on the definition of $\sum_{i\in 1..p} C_i$ proves that $\sum_{i\in 1..p} S_i \sim_{\mathsf{Y}} \sum_{i\in 1..p} S'_i\{n{\leftarrow}\mathsf{y}\}$, and $\sum_{i\in 1..p} T_i \sim_{\mathsf{Y}} \sum_{i\in 1..p} T'_i\{n{\leftarrow}\mathsf{y}\}$, and $O \sim_{\mathsf{Y}} \sum_{i\in 1..p}(S'_i + T'_i)$. Let P_1 and P_2 be the processes $mean(\sum_{i\in 1..p} S'_i)$ and $mean(\sum_{i\in 1..p} T'_i)$ respectively. Hence, $[\![P]\!] \approx O \sim_{\mathsf{Y}} [\![P_1]\!] + [\![P_2]\!]$ and, by Lemma 5.4, $P \equiv P_1 \mid P_2$. Moreover, $[\![(\nu n)P_1]\!] \approx \sum_{i\in 1..p} S_i \approx [\![Q]\!]$ and $[\![(\nu n)P_2]\!] \approx \sum_{i\in 1..p} T_i \approx [\![R]\!]$. By Lemma 5.4, $(\nu n)P_1 \equiv Q$ and $(\nu n)P_2 \equiv R$, as required. □

Given three processes, P, Q and R, such that $(\nu n)P \equiv Q \mid R$, we define a *solution* of Lemma 6.2 (3) to be a couple (P_1, P_2) such that $P \equiv P_1 \mid P_2$, $Q \equiv (\nu n)P_1$, and $R \equiv (\nu n)P_2$. For example, the next equations give a solution to a non-trivial instance of (3) obtained by following the steps described in the proof of Lemma 6.2.

$$(\nu n)\underbrace{(!(\nu n)n[\mathbf{0}] \mid n[\mathbf{0}])}_{P} \equiv \underbrace{!(\nu n)n[\mathbf{0}]}_{Q} \mid \underbrace{!(\nu n)n[\mathbf{0}]}_{R}$$
$$\equiv (\nu n)\underbrace{(!(\nu n)n[\mathbf{0}] \mid n[\mathbf{0}])}_{P_1} \mid (\nu n)\underbrace{!(\nu n)n[\mathbf{0}]}_{P_2}$$

It is not clear how to prove Lemma 6.2 (3) without using spatial trees as an intermediate representation, and it is even less clear how to obtain solutions for this law. Therefore, it is interesting to note that, following the constructive approach taken in this paper, our proof not only demonstrates that there is always a solution, but also describes an algorithm to compute it.

7 Discussion

We propose an algorithmic method to decide whether two ambient processes are spatially congruent, or not. This method is based on an intuitive interpretation of processes as edge-labelled trees, and a strongly normalizing rewriting system.

The definitions and proof techniques defined in this paper can easily be transposed to other process calculi equipped with a chemical semantics, such as the π-calculus for instance, and natural candidates for comparison are [7] and [10]. Other examples of calculi amenable for the same study include the spi-calculus of Abadi and Gordon [1] and some process calculi of concurrent objects, like TyCo [14] and **concς** [9].

Our definition of spatial congruence is very similar to the definition of the π-calculus equivalence given in [7]. Hence, we obtain a new proof of decidability for structural congruence. A major difference with Engelfriet's and Geselma's work is that we propose a more direct approach, and define an algorithm to decide the equivalence of processes.

In the work of Hirschkoff [10], the decidability of structural congruence is proved using a rewriting system, as it is the case in this paper. There are two main differences with Hirschkoff's approach. First, we use an intermediate data structure, the spatial trees, which eliminate the need to explicitly manipulate the associative and commutative parallel composition operator. Second, we use an exponentiation function in the interpretation of processes. These two differences should result in a more efficient algorithm. Another distinguishing feature of our work is the definition of an effective technique for proving equivalence laws.

The results obtained in this paper are interesting because they lay the formal basis for the development of an algorithm to check spatial congruence. Such automatic tool for testing the equivalence of processes is a necessary component in machine-based verification of properties of the ambient calculus. A benefit of the algorithm obtained with our approach, which has been successfully implemented by Romain Kervarc and Daniel Hirschkoff [11], is that it is based on well-studied algorithmic over trees, such as associative-commutative tree unification.

Another interest of our study is given in Section 6, where we apply our theoretical framework to the proof of equivalence laws used in the definition of Cardelli's and Gordon's modal logic for mobile ambients [4,5].

References

1. Martìn Abadi and Andrew D. Gordon. A calculus for cryptographic protocols: the spi calculus. *Information and Computation*, 148:1–70, 1999.
2. Gérard Berry and Gérard Boudol. The chemical abstract machine. *Theoretical Computer Science*, 96:217–248, 1992.
3. Luca Cardelli and Andrew D. Gordon. Mobile ambients. In *Proceedings of FoSSaCS '98*, volume 1378 of *LNCS*, pp. 140–155, 1998.
4. Luca Cardelli and Andrew D. Gordon. Anytime, anywhere: Modal logics for mobile ambients. In *Proceedings of POPL '00*, pp. 365–377, 2000.
5. Luca Cardelli and Andrew D. Gordon. Logical properties of name restriction. unpublished notes, 2000.
6. Silvano Dal Zilio. Spatial congruence for the ambients is decidable. Technical Report MSR-TR-2000-41, Microsoft Research, May 2000.
7. J. Engelfriet and T. Geselma. Multisets and structural congruence of the pi-calculus with replication. *Theoretical Computer Science*, 211(1-2):311–337, Jan. 1999.
8. J. Engelfriet and T. Geselma. Structural congruence in the pi-calculus with potential replication. Technical Report 00-02, Leiden Institute of Advanced Computer Science, Jan. 2000.
9. Andrew D. Gordon and Paul D. Hankin. A concurrent object calculus: reduction and typing. In *Proceedings of HLCL '98*, Elsevier ENTCS, 1998.

10. Daniel Hirschkoff. *Mise en oeuvre de preuves de bisimulation*. PhD thesis, École Nationale des Ponts et Chaussées, 1999.
11. Daniel Hirschkoff and Romain Kervarc. Implementation of an algorithm to decide spatial congruence for ambients. LIP, École Normale Supérieure de Lyon, August 2000.
12. Robin Milner. Flow graphs and flow algebras. *Journal of the ACM*, 26(4):794–818, Oct. 1979.
13. Robin Milner. *Communicating and Mobile Systems: the Pi-Calculus*. Cambridge University Press, 2000.
14. Vasco T. Vasconcelos. Typed concurrent objects. In *Proceedings of ECOOP '94*, volume 821 of *LNCS*, pages 100–117, 1994.