Spatial Congruence for the Ambients is Decidable

Silvano Dal Zilio
Microsoft Research

May 2000

Technical Report
MSR-TR-2000-41

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

Spatial Congruence for the Ambients is Decidable

Silvano Dal Zilio
Microsoft Research

May 2000

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, the spatial congruence, is defined on the top of an unlabelled transition system, the reduction system. Reduction is used to represent a real step of evolution (in time), while spatial congruence is used to identify processes up to particular (spatial) rearrangements.

In this paper, we show that it is decidable to check whether two ambient calculus processes are spatially congruent, or not. Our proof is based on a natural and intuitive interpretation of ambient processes as edge-labelled finite-depth trees. This 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 interesting equivalence laws.

Contents

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 (Cardelli and Gordon 1998), 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, notions such as *mobile computations*, that is, computations that can dynamically change the place where they are executed and are continuously active before and after movement (like agents), the *sites* where these computations happen: processor, router, etc. the *mobility* of these sites, such as found with mobile computers (or even simply temporarily disconnected computers), or in the crossing of administrative boundary, like applets crossing a firewall.

In the study of process algebras, as in the mathematical branch of algebra from which they have borrowed their name, the notion of equivalence plays a central role. This paper reports a proof that spatial congruence, one of the simplest and most important equivalence between processes, is decidable. That is, the problem of checking whether two processes are spatially congruent, or not, is decidable.

Inspired by the chemical machine (Berry and Boudol 1992) and the "chemical" presentation of the π-calculus semantics (Milner 1992), the dynamic semantics of the ambient calculus is based on a *spatial congruence* relation, $\equiv$, 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. Intuitively, reduction is used to represent a real step of evolution (in time), while spatial congruence is used to identify processes up to particular (spatial) rearrangements.

The decidability result presented in this paper is important 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* (Cardelli and Gordon 1999a), 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.

To prove the decidability of spatial congruence, we use a natural and intuitive interpretation of ambient processes as edge-labelled finite-depth trees. This allows us to concentrate on the subtle interaction between two key operators of the ambient calculus, namely restriction, which accounts for the dynamic generation of new location names, and replication, which is used to encode recursion.

The result of our study is the definition of an algorithm to test spatial congruence and a definition of a normal form for processes that is useful in the proof of interesting equivalence laws.

The structure of the paper is as follows. In the next section, we introduce the syntax of the ambient calculus and define spatial congruence. In Section 3, we define the interpretation of processes as a certain kind of trees, called *spatial trees*, and study a very simple notion of equivalences between spatial trees. In Section 4 we relate spatial trees (and the notion of tree equivalence), to processes (and spatial congruence) and we define a notion of normal form for processes. Before concluding, we use our results to prove some interesting equivalence laws. Appendixes include proofs of propositions omitted from the main body of the paper.

2 The Ambient Calculus

The following tables summarize the syntax of (untyped, monadic) processes and the definition of spatial congruence.

The operators of the ambient calculus can be separated into two categories: the *spatial constructs*, restriction, void, composition, replication and ambient, which describe the "spatial configuration" of processes, and the *temporal constructs*, which describe their (possible) dynamic behaviours. As pointed out in (Cardelli and Gordon 2000), this is similar to the distinction between static and dynamic constructs made in CCS (Milner 1979).

Expressions, capabilities and processes:

$M ::=$	expression
n	name
$act\, M$	capability
ϵ	null path
$M.M'$	path composition
$act\, M ::=$	capability
$in\, M$	can enter into M
$out\, M$	can exit out of M
$open\, M$	can open M
$P, Q, R ::=$	process
$(\nu n)P$	restriction
$\mathbf{0}$	void
$P \mid Q$	composition
$!P$	replication
$M[P]$	ambient
$M.P$	capability action
$(n).P$	input action

$\langle M \rangle$	output action

In both an input, $(n).P$, and a restriction, $(\nu n)P$, the name n is bound with scope P. Let $fn(P)$ be the set of names that occur free in the process P. We identify processes up to consistent renaming of bound names.

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

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

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 (Cardelli and Gordon 1998).

2.1 Dynamic Semantics

The rules defining spatial congruence can 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 next category describes scoping rules for the restriction operator, $(\nu n)P$, used to model the dynamic generation of new ambient names.

Spatial congruence:

$P \equiv P$	(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)
$P \equiv Q \Rightarrow M[P] \equiv M[Q]$	(Struct Amb)
$P \equiv Q \Rightarrow M.P \equiv M.Q$	(Struct Action)
$P \equiv Q \Rightarrow (n).P \equiv (n).Q$	(Struct Input)
$P \mid Q \equiv Q \mid P$	(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)
$n \notin fn(P) \Rightarrow (\nu n)(P \mid Q) \equiv P \mid (\nu n)Q$	(Struct Res Par)
$n \neq m \Rightarrow (\nu n)m[P] \equiv m[(\nu n)P]$	(Struct Res Amb)
$(\nu n)\mathbf{0} \equiv \mathbf{0}$	(Struct Res Zero)
$n \notin fn(M) \Rightarrow (\nu n)M.P \equiv M.(\nu n)P$	(Struct Res Action)
$n \neq m \Rightarrow (\nu n)(m).P \equiv (m).(\nu n)P$	(Struct Res Input)
$\epsilon.P \equiv P$	(Struct ϵ)
$(M.M').P \equiv M.M'.P$	(Struct .)

Lemma 2.1 *For all processes P we have that $!P \equiv !P \mid !P$.*

Proof The relation $!P \equiv !P \mid !P$ can be derived as follows. By (Struct Repl Repl), $!P \equiv !!P$. By (Struct Repl Copy), (Struct Trans) and (Struct Repl), $!!P \equiv !(P \mid !P)$. By (Struct Repl Par), $!(P \mid !P) \equiv !P \mid !!P$. By (Struct Repl Repl), (Struct Symm) and (Struct Par), $!P \mid !!P \equiv !P \mid !P$. □

Almost every axiom in the definition of spatial congruence 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 (Milner 1992). As a matter of fact, these axioms are also missing in the seminal presentation of the ambient calculus (Cardelli and Gordon 1998), where the relation $\equiv$ is also called structural equivalence. These differences have motivated our change in terminology.

The rules added to spatial congruence are similar to those added to (π-calculus) structural congruence in (Engelfriet and Geselma 2000; Engelfriet and Geselma 1999), where the authors proved that the resulting equivalence is decidable. Another related work is (Hirschkoff 1999). In this paper, Hirschkoff independently proposed a similar extension and proved the decidability result using a more algorithmic approach.

2.2 Static Semantics

The definition of the calculus syntax imposes very few constraints on the set of admissible processes. For example, processes such as $(M.M')[P]$ or $out\,(out\,n).P$ are (syntactically) well formed. A usual method to avoid these pathological processes is to define a type system, as the one proposed in (Cardelli and Gordon

1999b). Since we are not interested in the operational behaviour of processes, we decide to impose simpler constraints, so simple that, in fact, they are not preserved by reduction.

Good processes:

(Good ϵ)
$$\frac{}{\epsilon \vdash ok}$$

(Good Act)
$$\frac{M \vdash ok}{act\, n.M \vdash ok}$$

(Good Zero)
$$\frac{}{\mathbf{0} \vdash ok}$$

(Good Res)
$$\frac{P \vdash ok}{(\nu n)P \vdash ok}$$

(Good Par)
$$\frac{P \vdash ok \qquad Q \vdash ok}{P \mid Q \vdash ok}$$

(Good Repl)
$$\frac{P \vdash ok}{!P \vdash ok}$$

(Good Amb)
$$\frac{P \vdash ok}{n[P] \vdash ok}$$

(Good Action)
$$\frac{M \vdash ok \qquad P \vdash ok}{M.P \vdash ok}$$

(Good Input)
$$\frac{P \vdash ok}{(n).P \vdash ok}$$

(Good Output 1)
$$\frac{}{\langle n \rangle \vdash ok}$$

(Good Output 2)
$$\frac{M \vdash ok}{\langle M \rangle \vdash ok}$$

The restriction to good processes is very simple and, despite the fact that it is not necessary to prove the decidability of spatial congruence, it greatly simplifies the definitions and the presentation of our results. In the remainder of this paper we restrict our study to good processes.

3 Spatial Trees

We define an interpretation of spatial processes as a certain kind of edge-labelled finite-depth trees, which we name *spatial trees*, following the definition given in (Cardelli and Gordon 2000) for a version of the ambient calculus without the restriction operator.

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.

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

A *multiplicity*, μ, is either 1 or ∞. A *cone*, C, is either the empty vector, written ϵ, an input: $\mu(n).T$, an output: $\mu\langle M\rangle$, a capability: $\mu act\, \eta.T$, or an edge: $\mu\eta[T]$ or $!X.T$, 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.

Spatial trees:

$\mu ::=$	multiplicity
$\quad 1$	$\quad$ single
$\quad \infty$	$\quad$ infinite
$C ::=$	cone
$\quad \epsilon$	$\quad$ empty vector
$\quad \mu(n).T$	$\quad$ input
$\quad \mu\langle M \rangle$	$\quad$ output of expression M
$\quad \mu\,act\,\eta.T$	$\quad$ capability
$\quad \mu\eta[T]$	$\quad$ edge tagged η
$\quad !X.T$	$\quad$ replicated edge with markers X
$S, T ::=$	spatial tree
$\quad C_1 + \cdots + C_k$	$\quad$ vector of cones

Cones are a special type of spatial trees. The cone $!X.T$ represents an infinite copy of the tree T such that, in each copy, the markers in X are replaced with fresh markers. In $!X.T$ the markers in X are bound with scope T. The cone $\mu(n).T$ represents an input action. In $\mu(n).T$ the name n is bound with scope T. Spatial trees are identified up to consistent renaming of bound names and markers.

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

$$fm(\epsilon) \triangleq \varnothing$$
$$fm(\mu\langle M \rangle) \triangleq fm(M)$$
$$fm(\mu\eta[T]) \triangleq \begin{cases} fm(T) \cup \{\eta\} & \text{if } \eta \text{ is a marker} \\ fm(T) & \text{otherwise} \end{cases}$$
$$fm(\mu\,act\,\eta.T) \triangleq \begin{cases} fm(T) \cup \{\eta\} & \text{if } \eta \text{ is a marker} \\ fm(T) & \text{otherwise} \end{cases}$$
$$fm(S + T) \triangleq fm(S) \cup fm(T)$$
$$fm(!X.T) \triangleq fm(T) \setminus X$$
$$fm(\mu(n).T) \triangleq fm(T)$$

We write $T\{n \leftarrow x\}$ for the capture-avoiding substitution of the marker x for all the free occurrences of the name n in T.

For convenience, we extend the replication constructor, $!X.T$, to the empty set of markers as follows:

$$!\varnothing.\epsilon \triangleq \epsilon$$
$$!\varnothing.\mu\langle M \rangle \triangleq \infty\langle M \rangle$$
$$!\varnothing.\mu(n).T \triangleq \infty(n).T$$

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

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

Proof An easy induction on the structure of 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 set of its connected sub-trees.

Connected parts, *conn*(T), of tree T:

For all trees $T = \sum_{i\in 1..p} C_i$ we can construct a graph as follows.

(1) Let $\mathcal{N}$ be the set of cones $\{C_1, \dots, C_p\}$.

(2) Let $\mathcal{G}$ be the graph with nodes in $\mathcal{N}$ and edges between nodes that have a common (free) marker.

(3) Compute the connected components of the graph $\mathcal{G}$, say $\mathcal{G}_1, \dots, \mathcal{G}_k$.

The connected parts of T, written $conn(T)$, is the set $\{T_1, \dots, T_k\}$ 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

(1) *If* $conn(T) = \{T_1, \dots, T_p\}$ *then for each* $j \in 1..p$ *the tree* T_j *is connected.*

(2) *If* $conn(T) = \{T_1, \dots, T_p\}$ *then* $T = T_1 + \cdots + T_p$.

(3) *If* $conn(T) = \{T_1, \dots, T_p\}$ *and* $fm(T_i) = \varnothing$ *then* T_i *is a cone.*

(4) *For all trees* T*, where* $T = \sum_{i\in 1..p} C_i$*, if* $fm(C_i) = \varnothing$ *then* C_i *is a connected part of* T.

3.1 Equality Between Spatial Trees

We define a reduction relation between trees, $X \vdash S \to T$. This reduction relation captures the essential intuitions of our model of edge-labelled trees, such has, for example: “empty sub-trees can be forgotten”, rule (Red Zero), or “two infinite copies of a sub-tree can be replaced by only one infinite copy of this same sub-tree”, rule (Red Add Edge). In the definition of reduction, $X \vdash S \to T$, the set X, the *effect* of the reduction, is used to represent markers that must not appear free in the result of a reduction.

Reduction: $X \vdash S \to T$

(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 Output)

$$\frac{}{\varnothing \vdash \infty\langle M\rangle + \mu\langle M\rangle \to \infty\langle M\rangle}$$

(Red Add Input)

$$\frac{}{\varnothing \vdash \infty(n).T + \mu(n).T \to \infty(n).T}$$

(Red Add Action)

$$\frac{}{\varnothing \vdash \infty act\,\eta.T + \mu act\,\eta.T \to \infty act\,\eta.T}$$

(Red Add Repl)

$$\frac{}{\varnothing \vdash !X.T + !X.T \to !X.T}$$

(Red Copy)

$$\frac{}{X \vdash !X.T + T \to !X.T}$$

(Red Sub)

$$\frac{X \vdash T \to S \qquad X \subseteq Y}{Y \vdash T \to S}$$

(Red Repl)

$$\frac{X \vdash T \to S \qquad (Y' = Y \cap fm(S))}{X \setminus Y \vdash !Y.T \to !Y'.S}$$

(Red η)

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

(Red $+$)

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

(Red Input)

$$\frac{X \vdash T \to S}{X \vdash \mu(n).T \to \mu(n).S}$$

(Red Action)

$$\frac{X \vdash T \to S}{X \vdash \mu act\,\eta.T \to \mu act\,\eta.S}$$

The rules for reductions can be separated in two categories. Rules that involves two cones: (Red Zero) to (Red Copy), of which only (Red Copy) and (Red Sub) can extend the effect, and structural rules: (Red Repl) to (Red Action), which states that $\to$ is compositional.

Proposition 3.3 *If $X \vdash S \to^* \sum_{i \in 1..p} C_i$ then there exist $S_1, \dots, S_p$ such that $S = \sum_{i \in 1..p} S_i$ and $X \vdash S_i \to^* C_i$ for each $i \in 1..p$.*

Proof An easy induction on the derivation of $X \vdash S \to^* \sum_{i\in 1..p} C_i$. $\square$

An equivalent of rules (Red Add Repl), (Red Copy) and (Red Repl), for the special case where the set X is empty, can be derived in our system.

Proposition 3.4

(1) $\varnothing \vdash\ !\varnothing.T +\ !\varnothing.T \to^*\ !\varnothing.T$.

(2) $\varnothing \vdash\ !\varnothing.T + T \to^*\ !\varnothing.T$.

(3) *If* $\varnothing \vdash T \to S$ *then* $\varnothing \vdash\ !\varnothing.T \to^*\ !\varnothing.S$.

It is worth mentioning that there are trees, T, such that $\varnothing \vdash\ !Y.T \to\ !\varnothing.S$. For example, if T is the tree $!Y.T' + T'$ then we get $\varnothing \vdash\ !Y.(!Y.T' + T') \to\ !\varnothing.!Y.T'$, which is equal to $!Y.T'$.

Equivalence between trees: $S \sim_X T$ and $S \approx T$

The relation $\sim_X$ is the smallest reflexive, symmetric and transitive relation such that if $X \vdash S \to T$ then $S \sim_X T$. 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 X such that $dom(\sigma_1) = fm(S)$ and $dom(\sigma_2) = fm(T)$ and $S\sigma_1 \sim_X T\sigma_2$.

The equivalence $\sim_X$ is a congruence and if $Y \subseteq X$ then $\sim_Y \subseteq \sim_X \subseteq \approx$. Basic properties of $\approx$ are:

Proposition 3.5 *The relation $\approx$ satisfies the congruence properties:*

(1) *If* $(fm(S) \cup fm(T)) \cap fm(R) = \varnothing$ *and* $S \approx T$ *then* $T + R \approx S + R$.

(2) *If* $S \approx T$ *then* $\mu n[S] \approx \mu n[T]$.

(3) *If* $S \approx T$ *then* $\mu act\, \eta.S \approx \mu act\, \eta.T$.

(4) *If* $S \approx T$ *then* $\mu(n).S \approx \mu(n).T$.

Next, we study the interaction between tree equivalence and substitution.

Lemma 3.6 *If* $X \vdash S \to T$ *then* $fm(T) \subseteq fm(S)$ *and* $fm(S) \setminus fm(T) \subseteq X$.

Proof See Appendix A.1. $\square$

Lemma 3.7 *If* $X \vdash S \to T$ *and* $x \notin X$ *then* $X \vdash S\{n{\leftarrow}x\} \to T\{n{\leftarrow}x\}$.

Proof An easy induction on the derivation of $X \vdash T \to S$. $\square$

In the previous lemma, the condition $x \notin X$ is fundamental. For example, we have $\{x\} \vdash\ !\{x\}.\langle x\rangle + \langle x\rangle + \langle n\rangle \to\ !\{x\}.\langle x\rangle + \langle n\rangle$, whereas the spatial tree $(!\{x\}.\langle x\rangle + \langle x\rangle + \langle n\rangle)\{n{\leftarrow}x\}$ does not reduce.

Corollary 3.8

(1) *If* $S \sim_X T$ *and* $x \notin X$ *then* $S\{n{\leftarrow}x\} \approx T\{n{\leftarrow}x\}$

(2) *If* $S \approx T$ *and* $x \notin fm(S) \cup fm(T)$ *then* $S\{n{\leftarrow}x\} \approx T\{n{\leftarrow}x\}$.

Next, we prove that the reduction relation on spatial trees is locally confluent.

Theorem 3.9 *If* $X_1 \vdash T \to T_1$ *and* $X_2 \vdash T \to T_2$ *then there exists a tree* S *such that* $X_1 \cup X_2 \vdash T_1 \to^* S$ *and* $X_1 \cup X_2 \vdash T_2 \to^* S$.

Proof See Appendix A.1. □

The reduction relation is *decreasing*, in the sense that the number of symbol is decreased along reductions. Therefore there can only be a finite number of reductions from any tree. Since the reduction relation is also confluent, we have the following property.

Theorem 3.10 *The relation* $\to$ *is strongly normalizing.*

Proof We define a weighting function for trees, $h(.)$, as follows:

$$
\begin{aligned}
&h(\epsilon) = 1\\
&h(\mu(n).T) = h(T) + 1\\
&h(\mu\langle M\rangle) = 1\\
&h(\mu act\, \eta.T) = h(T) + 1\\
&h(\mu\eta[T]) = h(T) + 1\\
&h(!X.T) = h(T) + 1\\
&h(S + T) = h(S) + h(T)
\end{aligned}
$$

Note that the function $h(.)$ is strictly positive. The theorem follows by showing that $X \vdash T \to S$ implies $h(T) > h(S)$. We proceed by induction on the derivation of $X \vdash T \to S$.

(Red Zero) Then $T = S + \epsilon$. Hence, $h(T) = h(S) + 1 > h(S)$.

(Red Add Edge) Then $T = \infty\eta[T'] + \mu\eta[T']$ and $S = \infty\eta[T']$. Hence, $h(T) = 2.(h(T') + 1) > h(S)$. Cases (Red Add Output), (Red Add Input), (Red Add Action), (Red Add Repl) and (Red Copy) are similar.

(Red Sub) Then $Y \vdash T \to S$ with $Y \subseteq X$. By induction hypothesis, $h(T) > h(S)$.

(Red η) Then $T = \mu\eta[T']$ and $S = \mu\eta(S')$ where $X \vdash T' \to S'$. By induction hypothesis, $h(T') > h(S')$. Hence, $h(T) = h(T') + 1 > h(S') + 1 = h(S)$. Cases (Red Input) and (Red Action) are similar.

(Red $+$) Then $T = T' + R$ and $S = S' + R$ where $X \vdash T' \to S'$. By induction hypothesis, $h(T') > h(S')$. Hence, $h(T) = h(T') + h(R) > h(S') + h(R) = h(S)$. $\square$

Based on the this result, we can define an algorithm to decide the equivalence of spatial trees, and therefore the equivalence of ambients processes. For instance, to decide if $S_1 \sim_X S_2$, you compute the normal form of S_1 and S_2, that is, the spatial trees S'_1, S'_2 such that $X \vdash S_i \to^* S'_i$ and S'_i is irreducible for each $i \in 1..2$. By Theorem 3.10, these trees exist and can be computed using a finite number of reductions. Then, you verify whether the normal forms are equal.

Theorem 3.11 *The equivalences $\sim_X$ and $\approx$ are decidable.*

Proof To decide if $S_1 \sim_X S_2$, you compute the normal form of S_1 and S_2, that is, the spatial trees S'_1, S'_2 such that $X \vdash S_i \to^* S'_i$ and S'_i is irreducible for each $i \in 1..2$. By Theorem 3.10, 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 up to α-equivalence of bound markers and associativity-commutativity of $+$. Since this is a decidable problem, we get that $\sim_X$ is decidable.

To decide if $S_1 \approx S_2$, you test whether $S_1\sigma_1 \sim_X S_2\sigma_2$ 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 $X \subseteq fm(S_1\sigma_1) \cup fm(S_2\sigma_2)$. 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 $\sim_X$ is decidable, we get that $\approx$ is decidable. $\square$

Using the property of strong normalization, 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 such that T' has the following form:

$$\underbrace{\sum_{i_1 \in I_1} \mu_{i_1} \eta_{i_1}[T_{i_1}] + \sum_{i_2 \in I_2} !X_{i_2}.T_{i_2}}_{\text{edges, with restriction and replication}} + \underbrace{\sum_{i_3 \in I_3} \mu_{i_3}\, act\, n_{i_3}.T_{i_3}}_{\text{actions}} + \underbrace{\sum_{i_4 \in I_4} \mu_{i_4} (n_{i_4}).T_{i_4}}_{\text{inputs}} + \underbrace{\sum_{i_5 \in I_5} \mu_{i_5} \langle M_{i_5} \rangle}_{\text{outputs}}$$

Where:

(1) I_1, I_2, I_3, I_4 and I_5 are finite and pairwise disjoint sets of indices.

(2) for all $i \in \bigcup_{j \in 1..5} I_j$, the trees T_j are 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 $\mu_i = \mu_j = 1$.

(4) for all $i, j \in I_2$, if $!X_i.T_i \sim_\varnothing !X_j.T_j$ then $i = j$.

3.2 Exponentiation of Spatial Trees

In this section, we define a new operation on trees, $exp(T)$, obtained as the outcome of replicating every connected part of T. This operation will prove useful in the interpretation of ambient processes.

Exponentiation, $exp(T)$, of a tree T:

The *exponentiation* of a tree T, written $exp(T)$, is the tree $!X_1.T_1 + \cdots + !X_p.T_p$ where $\{T_1, \ldots, T_p\} = conn(T)$ and $X_i = fm(T_i)$ for each $i \in 1..p$.

Next, we study the properties of $exp(T)$.

Proposition 3.12

(1) *If $fm(S) \cap fm(T) = \varnothing$ then $exp(S + T) \sim_\varnothing exp(S) + exp(T)$.*

(2) *The function $exp(.)$ is idempotent, that is, $exp(exp(T)) = exp(T)$.*

Lemma 3.13 *If $X \vdash S \to T$ then $X \vdash exp(S) \to^* exp(T)$.*

Proof See Appendix A.2. $\square$

Theorem 3.14 *If $S \approx T$ then $exp(S) \approx exp(T)$.*

Proof Assume $S \approx T$. By definition, there are two finite injective mappings, σ_1, σ_2 and a set X such that $S\sigma_1 \sim_X T\sigma_2$. By Theorem 3.9, there is a tree S' such that $X \vdash S\sigma_1 \to^* S'$ and $X \vdash T\sigma_2 \to^* S'$. By Lemma 3.13 several times, $X \vdash exp(S\sigma_1) \to^* exp(S')$ and $X \vdash exp(T\sigma_2) \to^* exp(S')$. Therefore, $exp(S\sigma_1) \sim_X exp(T\sigma_2)$, where $exp(T\sigma_2) = exp(T)$. Hence, $exp(S) \approx exp(T)$. $\square$

Proposition 3.15 *For all spatial trees T we have $exp(T) + T \approx exp(T)$.*

Proof The proposition follows by showing that $fm(T) \vdash exp(T) + T \to^* exp(T)$. This is proved using an easy induction on the derivation of $exp(T)$ and rule (Red Copy). $\square$

4 Relation Between Trees and Processes

We can now define the tree semantics of processes, that is, a mapping from (good) processes to spatial trees. This semantics extends a a similar definition given in (Cardelli and Gordon 1999a) for the calculus without the restriction operator.

Tree semantics (of good processes):

$[\![\mathbf{0}]\!] \triangleq \epsilon$	(Zero)
$[\![\langle M\rangle]\!] \triangleq 1\langle M\rangle$	(Output)
$[\![(n).P]\!] \triangleq 1(n).[\![P]\!]$	(Input)
$[\![act\,\eta.P]\!] \triangleq 1\,act\,\eta.[\![P]\!]$	(Action)
$[\![\epsilon.P]\!] \triangleq [\![P]\!]$	(Action ϵ)
$[\![(M.M').P]\!] \triangleq [\![M.(M'.P)]\!]$	(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)
$x \notin fm([\![P]\!]) \Rightarrow [\![(\nu n)P]\!] \triangleq [\![P]\!]\{n{\leftarrow}x\}$	(Res)

Next, we show that the axiomatisation of spatial congruence is sound.

Theorem 4.1 (Soundness) *If $P \equiv Q$ then $[\![P]\!] \approx [\![Q]\!]$.*

Proof See Appendix A.3. $\square$

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

Process semantics (of trees):

Let L be a set of fresh, pairwise distinct names with the cardinality of X.

$(\![\epsilon]\!) \triangleq \mathbf{0}$	(Empty)
$(\![1\langle M\rangle]\!) \triangleq \langle M\rangle$	(Output 1)
$(\![\infty\langle M\rangle]\!) \triangleq !\langle M\rangle$	(Output ∞)
$(\![1(n).T]\!) \triangleq (n).(\![T]\!)$	(Input 1)
$(\![\infty(n).T]\!) \triangleq !(n).(\![T]\!)$	(Input ∞)
$(\![1\,act\,n.T]\!) \triangleq act\,n.(\![T]\!)$	(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 ∞)
$(\![!X.T]\!) \triangleq !(\nu L)(\![T\{X{\leftarrow}L\}]\!)$	(Repl)
$(\![S+T]\!) \triangleq (\![S]\!) \mid (\![T]\!)$	(Sum)

Meaning, $mean(T)$, of a tree T:

The *meaning* of a tree T, written $mean(T)$, is the (good) 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.

Next, we prove that there is a simple relation between a process and the meaning of its interpretation.

Lemma 4.2 *For all processes P we have $mean([\![P]\!]) \equiv P$.*

Proof See Appendix A.3. $\Box$

Theorem 4.3 *If $S \approx T$ then $mean(S) \equiv mean(T)$.*

Proof Assume $S \approx T$. By definition, there are two finite injective mappings, σ_1, σ_2 and a set X such that $S\sigma_1 \sim_X T\sigma_2$. By Theorem 3.9, there is a tree S' such that $X \vdash S\sigma_1 \rightarrow^* S'$ and $X \vdash T\sigma_2 \rightarrow^* S'$. The property follows by Lemma A.11, proved in Appendix A.3, which states that if $X \vdash T \rightarrow S$ then $(\nu K)(\![T\sigma]\!) \equiv (\nu K)(\![S\sigma]\!)$, where σ is a bijection from $fm(T)$ to a set of fresh names and $K = \sigma(fm(T) \cap X)$. By Lemma A.11 several times, $(\nu K)(\![T\sigma_2\sigma']\!) \equiv (\nu K_1)(\![S'\sigma']\!)$ and $(\nu K)(\![S\sigma_1\sigma']\!) \equiv (\nu K)(\![S'\sigma']\!)$, where σ' is a bijection from $fm(S')$ to a set of fresh names and $K = \sigma(fm(S') \cap X)$. Hence, $(\nu K)(\![T\sigma_2\sigma']\!) \equiv (\nu K)(\![S\sigma_1\sigma']\!)$. By (Struct Res), $(\nu K')(\![T\sigma_2\sigma']\!) \equiv (\nu K')(\![S\sigma_1\sigma']\!)$, where $K' = \sigma(fm(S'))$. By α-renaming of markers in K', we get that $mean(S) = (\nu K')(\![S\sigma_1\sigma']\!)$ and $mean(T) = (\nu K')(\![T\sigma_2\sigma']\!)$. Hence, $mean(S) \equiv mean(T)$, as required.

Theorem 4.4 (Completeness) *If $[\![P]\!] \approx [\![Q]\!]$ then $P \equiv Q$.*

Proof Let P and Q be two processes such that $[\![P]\!] \approx [\![Q]\!]$. By Theorem 4.3, $mean([\![P]\!]) \equiv mean([\![Q]\!])$. By Lemma 4.2, $P \equiv mean([\![P]\!])$ and $Q \equiv mean([\![Q]\!])$. Hence, $P \equiv Q$. $\Box$

Theorem 4.5 (Decidability) *The relation $\equiv$ is decidable.*

Proof To decide whether $P \equiv Q$, you compute $[\![P]\!]$ and $[\![Q]\!]$ and you verify if they are equivalent. By Theorem. 3.11, this is a decidable problem. $\Box$

Using Theorem 4.4 and the definition of normal form for spatial trees given at the end of Section 3.1, we can define a normal form (up to spatial congruence) for ambient processes. This normal form is unique up to very simple (spatial) transformations such as commutativity-associativity of the parallel composition and the reordering of restrictions, as in rule (Struct Res Res) for instance. More precisely, for all processes P, there exists a process Q such that $P \equiv Q$ and Q is a process of the kind:

$$\underbrace{\prod_{i_1 \in I_1} n_{i_1}[Q_{i_1}] \mid \prod_{i_2 \in I_2} !n_{i_2}[Q_{i_2}] \mid \prod_{i_3 \in I_3} !(\nu L_{i_3})Q_{i_3} \mid}_{\text{ambients, with restriction and replication}}$$

$$\underbrace{\prod_{i_4 \in I_4} act\, n_{i_4}.Q_{i_4} \mid \prod_{i_5 \in I_5} !act\, n_{i_5}.Q_{i_5} \mid}_{\text{expressions}}$$

$$\underbrace{\prod_{i_6 \in I_6} (n_{i_6}).Q_{i_6} \mid \prod_{i_7 \in I_7} !(n_{i_7}).Q_{i_7} \mid \prod_{i_8 \in I_8} \langle M_{i_8} \rangle \mid \prod_{i_9 \in I_9} !\langle M_{i_9} \rangle}_{\text{input and output actions}}$$

Where:

(1) the set of indices $I_1, \ldots, I_9$ are finite and pairwise disjoint.

(2) for all $i \in \bigcup_{j \in 1..9} I_j$, 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 $i = j$.

5 Applications of Normal Form

In this section, we use our results on spatial congruence to prove interesting equivalence laws of ambient processes. For example, we can prove laws used in the extended presentation of the modal logic for ambients (Cardelli and Gordon 1999a, Section 2-10). We also prove Theorem 5.8, an interesting property that validates the distribution of restriction over parallel composition under certain hypothesis.

Theorem 5.1 *If $P \mid Q \equiv \mathbf{0}$ then $P \equiv \mathbf{0}$ and $Q \equiv \mathbf{0}$.*

Proof The proposition follows by showing that for any finite set of indices, I, if $X \vdash \sum_{i \in I} C_i \to^* \epsilon$ then $X \vdash C_i \to^* \epsilon$ for all $i \in I$. We proceed by induction on the derivation of $X \vdash \sum_{i \in I} C_i \to^* \epsilon$. The case $\sum_{i \in I} C_i = \epsilon$ is trivial.

(Red Zero) Then there exists $i \in I$ such that $C_i = \epsilon$ and $X \vdash \sum_{j \in I \setminus \{i\}} C_j \to^* \epsilon$. By induction hypothesis, $X \vdash C_j \to^* \epsilon$ for all $j \in I \setminus \{i\}$, as required.

(Red Add Edge) Then $I = \{i, j\}$ where $C_i = \infty\eta[T]$ and $C_j = \mu\eta[T]$. This contradicts the fact that there are no spatial trees, T, such that $X \vdash \mu\eta[T] \to^* \epsilon$. Cases (Red Add Output), (Red Add Input), (Red Add Repl), (Red Copy), (Red η), (Red Repl), (Red Input) and (Red Action) are similar.

(Red Sub) Then $Y \vdash \sum_{i \in I} C_i \to^* \epsilon$ where $Y \subseteq X$. By induction hypothesis, $Y \vdash C_i \to^* \epsilon$ for all $i \in I$. By (Red Sub), $X \vdash C_i \to^* \epsilon$ for all $i \in I$.

(Red +) Then there is a partition of I into two subsets, I_1 and I_2, such that $X \vdash \sum_{i \in I_1} C_i \to \sum_{i \in I_1'} C_i'$, with $X \vdash (\sum_{i \in I_1'} C_i') + (\sum_{i \in I_2} C_i) \to^* \epsilon$. By induction hypothesis, $X \vdash C_i \to^* \epsilon$ for all $i \in I_1' \cup I_2$. Hence, $X \vdash \sum_{i \in I_1} C_i \to \epsilon$. By induction hypothesis, $X \vdash C_i \to^* \epsilon$ for all $i \in I_1$, as required.

Now, assume $P \mid Q \equiv \mathbf{0}$. By Theorem 4.1, $[\![P]\!] + [\![Q]\!] \approx \epsilon$. Hence, there exists a set X such that $[\![P]\!] + [\![Q]\!] \sim_X \epsilon$. By Theorem 3.9, and since ϵ is an irreducible spatial trees, $X \vdash [\![P]\!] + [\![Q]\!] \to^* \epsilon$. Therefore, $X \vdash [\![P]\!] \to^* \epsilon$ and $X \vdash [\![Q]\!] \to^* \epsilon$. Hence, $[\![P]\!] \approx \epsilon$ and $[\![Q]\!] \approx \epsilon$. By Theorem 4.4, $P \equiv Q \equiv \mathbf{0}$. $\square$

Lemma 5.2 *If $X \vdash \mu\eta[T] \to S$ then there exist a tree S' and a set Y such that $Y \subseteq X$, $S = \mu\eta[S']$ and $Y \vdash T \to S'$ with $\eta \notin Y$.*

Proof By induction on the derivation of $X \vdash \mu\eta[T] \to S$. By inspection of the possible derivations, the last rule used can only be (Red Sub) or (Red η).

(Red Sub) We have $Y \vdash \mu\eta[T] \to S$ and $Y \subseteq X$ implies $X \vdash \mu\eta[T] \to S$. By induction hypothesis, there exists a tree S' and a set Y' such that $Y' \subseteq Y \subseteq X$, and $S = \mu\eta[S']$ and $Y' \vdash T \to S'$ and $\eta \notin Y'$, as required.

(Red η) Then there is S' such that $S = \mu\eta[S']$ and $X \vdash T \to S'$ and $\eta \notin X$. Let $Y = X$. We get that $Y \subseteq X$, and $Y \vdash T \to S'$ and $\eta \notin Y$, as required. $\square$

Lemma 5.3 *If $X \vdash !Y.T \to S$ then there exists a tree S' such that $S = !(Y \cap fm(S')).S'$ and $(X \cup Y) \vdash T \to S'$.*

Proof An easy induction on the derivation of $X \vdash !Y.T \to S$. The proof is similar to the one of Lemma 5.2. $\square$

Theorem 5.4 *For all processes P we have $n[P] \not\equiv \mathbf{0}$.*

Proof Assume $n[P] \equiv \mathbf{0}$. By Theorem 4.1, $[\![n[P]]\!] \approx \epsilon$. Hence, there exists a set X such that $1n[[\![P]\!]] \sim_X \epsilon$. By Theorem 3.9, and since ϵ is an irreducible spatial trees, $X \vdash 1n[[\![P]\!]] \to^* \epsilon$, which contradicts Lemma 5.2. Hence, $n[P] \not\equiv \mathbf{0}$. $\square$

Theorem 5.5 *If $n[P] \equiv Q \mid R$ then either $Q \equiv n[P]$ and $R \equiv \mathbf{0}$, or $Q \equiv \mathbf{0}$ and $R \equiv n[P]$.*

Proof The proposition follows by showing that for all finite sets of indices, I, if $X \vdash \sum_{i \in I} C_i \rightarrow^* 1n[S]$ then there exists $i \in I$ such that $X \vdash C_i \rightarrow^* 1n[S]$ and $X \vdash \sum_{j \in I \setminus \{i\}} C_j \rightarrow^* \epsilon$. We proceed by induction on the derivation of $X \vdash \sum_{i \in I} C_i \rightarrow^* 1n[S]$. The case $\sum_{i \in I} C_i = 1n[S]$ is trivial.

(Red Zero) Then there exists $j \in I$ such that $C_j = \epsilon$ and $X \vdash \sum_{i \in I \setminus \{j\}} C_i \rightarrow^* 1n[S]$. By induction hypothesis, there exists $i \in I \setminus \{j\}$ such that $X \vdash C_j \rightarrow^* 1n[S]$ and $X \vdash \sum_{k \in I \setminus \{i,j\}} C_k \rightarrow^* \epsilon$. Hence, $X \vdash \sum_{j \in I \setminus \{i\}} C_k \rightarrow^* \epsilon$, as required.

(Red Add Edge) Then $I = \{i, j\}$ where $C_i = \infty\eta[T]$ and $C_j = \mu\eta[T]$. This contradicts the fact that there are no spatial trees, T, such that $X \vdash \infty n[T] \rightarrow^* 1n[S]$.

(Red Add Output) Then $I = \{i, j\}$ where $C_i = \infty\langle M \rangle$ and $C_j = \mu\langle M \rangle$. This contradicts the fact that there are no actions, M, such that $X \vdash \infty\langle M \rangle \rightarrow^* 1n[S]$. Cases (Red Add Input), (Red Add Repl), (Red Copy), (Red Repl), (Red Input) and (Red Action) are similar.

(Red Sub) Trivial.

(Red η) Then $I = \{i\}$ and $X \vdash C_i \rightarrow^* 1n[S]$, as required.

(Red +) Then there is a partition of I into two subsets, I_1 and I_2, such that $X \vdash \sum_{i \in I_1} C_i \rightarrow \sum_{i \in I_1'} C_i'$, with $X \vdash (\sum_{i \in I_1'} C_i') + (\sum_{j \in I_2} C_j) \rightarrow^* 1n[S]$. By induction hypothesis, either (1) there is $i \in I_1'$ such that $X \vdash C_i \rightarrow^* 1n[S]$ and $X \vdash (\sum_{j \in I_1' \setminus \{i\}} C_j') + (\sum_{j \in I_2} C_j) \rightarrow^* \epsilon$, or (2) there is $i \in I_2$ such that $X \vdash C_i \rightarrow^* 1n[S]$ and $X \vdash (\sum_{j \in I_1'} C_j') + (\sum_{j \in I_2 \setminus \{i\}} C_j) \rightarrow^* \epsilon$. Assume we are in case (1). Hence, $X \vdash \sum_{i \in I_1} C_i \rightarrow^* 1n[S]$. By induction hypothesis, there exists an indices $i \in I_1$ such that $X \vdash C_i \rightarrow^* 1n[S]$ and $X \vdash \sum_{j \in I_1 \setminus \{i\}} C_j \rightarrow \epsilon$. By (Red +), $X \vdash \sum_{j \in (I_1 \cup I_2) \setminus \{i\}} C_j \rightarrow \epsilon$, as required. Assume we are in case (2). By Theorem 5.1, $X \vdash \sum_{i \in I_1} C_i \rightarrow \epsilon$ and, by (Red +), $X \vdash \sum_{j \in (I_1 \cup I_2) \setminus \{i\}} C_j \rightarrow \epsilon$, as required.

Assume $n[P] \equiv Q \mid R$. By Theorem 4.1, $[\![n[P]]\!] \approx [\![Q]\!] + [\![R]\!]$. Hence, there exists a set X and two finite injective mappings, σ_1, σ_2 such that $1n[[\![P]\!]]\sigma_1 \sim_X [\![Q\sigma_2]\!] + [\![R\sigma_2]\!]$. By Theorem 3.9, there exists a spatial tree S such that both $X \vdash 1n[[\![P]\!]]\sigma_1 \rightarrow^* S$ and $X \vdash [\![Q\sigma_2]\!] + [\![R\sigma_2]\!] \rightarrow^* S$. By Lemma 5.2, there exists a tree S' such that $S = 1n[S']$ and $X \vdash [\![Q\sigma_2]\!] + [\![R\sigma_2]\!] \rightarrow^* 1n[S']$ and $X \vdash 1n[[\![P]\!]]\sigma_1 \rightarrow^* 1n[S']$. Hence, $[\![n[P]]\!] \approx 1n[S']$. Therefore either (1) $X \vdash [\![Q\sigma_2]\!] \rightarrow^* 1n[S']$ and $X \vdash [\![R\sigma_2]\!] \rightarrow^* \epsilon$ or (2) $X \vdash [\![Q\sigma_2]\!] \rightarrow^* \epsilon$ and $X \vdash [\![R\sigma_2]\!] \rightarrow^* 1n[S']$. Assume we are in case (1). Hence, $[\![Q]\!] \approx 1n[S']$ and $[\![R]\!] \approx \epsilon$. By Theorem 4.4, $Q \equiv n[P]$ and $R \equiv \mathbf{0}$. Case (2) is symmetric. $\square$

Theorem 5.6 *If $m[P] \equiv n[Q]$ then $m = n$ and $P \equiv Q$.*

Proof Assume $m[P] \equiv n[Q]$. By Theorem 4.1, $[\![m[P]]\!] \approx [\![n[Q]]\!]$. Hence, there exists a set X and two finite injective mappings, σ_1, σ_2 such that $1m[[\![P]\!]\sigma_1] \sim_X 1n[[\![Q]\!]\sigma_2]$. By Theorem 3.9, there exists a spatial tree S such that both $X \vdash 1m[[\![P]\!]\sigma_1] \rightarrow^* S$ and $X \vdash 1n[[\![Q]\!]\sigma_2] \rightarrow^* S$. By Lemma 5.2, we have that $m = n$ and that there exists a tree S' such that $S = 1n[S']$ and $X \vdash [\![Q]\!]\sigma_2 \rightarrow^* S'$ and $X \vdash [\![P]\!]\sigma_1 \rightarrow^* S'$. Hence, $[\![P]\!] \approx S' \approx [\![Q]\!]$. By Theorem 4.4, $P \equiv Q$, as required. $\square$

Theorem 5.7 *If $(\nu n)P \equiv m[Q]$ then there exists R such that $P \equiv m[R]$ and $Q \equiv (\nu n)R$.*

Proof Assume $(\nu n)P \equiv m[Q]$. By Theorem 4.1, $[\![(\nu n)P]\!] \approx [\![m[Q]]\!]$. Therefore, for every fresh marker, x, we have $[\![P]\!]\{n{\leftarrow}x\} \approx 1m[[\![Q]\!]]$. By definition, there exist two finite injective mappings, σ_1, σ_2 and a set X such that $[\![P]\!]\sigma_1\{n{\leftarrow}z\} \sim_X 1m[[\![Q]\!]\sigma_2]$ where $z = \sigma_1(x)$. Let S be the normal form of $[\![P]\!]\sigma_1$. Therefore, $S \approx [\![P]\!]\sigma_1 \sim_Y 1m[[\![Q]\!]\sigma_2\{z{\leftarrow}n\}]$. By Proposition 3.3, it must be the case that $S = 1m[T]$ where $T \approx [\![Q]\!]\sigma_2\{z{\leftarrow}n\}$. Let R be the process $\mathit{mean}(T)$. Then, $[\![m[R]]\!] \approx S \approx [\![P]\!]$ and, by Theorem 4.4, $m[R] \equiv P$. Moreover, $[\![(\nu n)R]\!] \approx T\{n{\leftarrow}z\} \approx [\![Q]\!]$ and, by Theorem 4.4, $(\nu n)R \equiv Q$, as required. $\square$

Theorem 5.8 *If $P \mid Q \equiv (\nu n)R$ then there exist two processes, R_1, R_2, such that $R \equiv R_1 \mid R_2$, and $P \equiv (\nu n)R_1$, and $Q \equiv (\nu n)R_2$.*

Proof Assume $P \mid Q \equiv (\nu n)R$. By Theorem 4.1, $[\![P \mid Q]\!] \approx [\![(\nu n)R]\!]$. Therefore, for every fresh marker, x, we have $[\![P]\!]+[\![Q]\!] \approx [\![R]\!]\{n{\leftarrow}x\}$, where $\mathit{fm}([\![P]\!]) \cap \mathit{fm}([\![Q]\!]) = \varnothing$. By definition, there exist two finite injective mappings, σ_1, σ_2 and a set X such that $[\![P]\!]\sigma_1 + [\![Q]\!]\sigma_1 \sim_X [\![R]\!]\sigma_2\{n{\leftarrow}y\}$, where $y = \sigma_2(x)$.

Let S, T and O be the normal forms of $[\![P]\!]\sigma_1$, $[\![Q]\!]\sigma_1$ and $[\![R]\!]\sigma_2$, respectively. Hence, $S + T \sim_Y O\{n{\leftarrow}y\}$ for some set of markers Y such that $X \subseteq Y$ and with the side-condition $\mathit{fm}(S) \cap \mathit{fm}(T) = \varnothing$. Assume $\sum_{i \in 1..p} C_i$ is the (common) normal form of $S + T$ and $O\{n{\leftarrow}y\}$. By Proposition 3.3, and since S, T and O are in normal form, there is $S_1, \ldots, S_p, T_1, \ldots, T_p, O_1, \ldots, O_p$, each in normal form, 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) $S_i + T_i \sim_Y O_i\{n{\leftarrow}y\}$ for each $i \in 1..p$.

(3) $Y \vdash S_i + T_i \rightarrow^* C_i$ and $Y \vdash O_i\{n{\leftarrow}y\} \rightarrow^* C_i$ for each $i \in 1..p$.

The proof follows by constructing the spatial trees corresponding to the processes R_1, R_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 $O_i \sim_Y (S'_i + T'_i)$, and $S_i \sim_Y S'_i\{n{\leftarrow}y\}$, and $T_i \sim_Y T'_i\{n{\leftarrow}y\}$ 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.

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

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

An easy induction on the definition of $\sum_{i\in 1..p} C_i$ proves that:

$$\begin{array}{rcl} O & \sim_Y & \sum_{i\in 1..p}(S'_i + T'_i) \\ \sum_{i\in 1..p} S_i & \sim_Y & \sum_{i\in 1..p} S'_i\{n{\leftarrow}y\} \\ \sum_{i\in 1..p} T_i & \sim_Y & \sum_{i\in 1..p} T'_i\{n{\leftarrow}y\} \end{array}$$

Let R_1 and R_2 be the processes $\mathit{mean}(\sum_{i\in 1..p} S'_i)$ and $\mathit{mean}(\sum_{i\in 1..p} T'_i)$, respectively. Hence, $[\![R]\!] \approx O \sim_Y [\![R_1]\!]+[\![R_2]\!]$ and, by Theorem 4.4, $R \equiv R_1 \mid R_2$. Moreover, $[\![(\nu n)R_1]\!] \approx \sum_{i\in 1..p} S_i \approx [\![P]\!]$ and $[\![(\nu n)R_2]\!] \approx \sum_{i\in 1..p} T_i \approx [\![Q]\!]$. Therefore, by Theorem 4.4, $(\nu n)R_1 \equiv P$ and $(\nu n)R_2 \equiv Q$, as required. $\square$

The proof of the last result is surprisingly subtle and difficult, and uses different properties of normal forms and the interpretation of processes. Following the "constructive" approach taken in this paper, our proof method not only demonstrates the existence of a solution, but also describe an algorithm to compute it, that is, it defines an algorithm that computes the two processes R_1 and R_2. The following equivalences give an example of solutions to a non-trivial instance of Theorem 5.8:

$$\begin{aligned} (\nu n)\,\underbrace{(!(\nu n)n[\mathbf{0}] \mid n[\mathbf{0}])}_{R} &\equiv \underbrace{!(\nu n)n[\mathbf{0}]}_{P} \mid \underbrace{!(\nu n)n[\mathbf{0}]}_{Q} \\ &\equiv (\nu n)\,\underbrace{(!(\nu n)n[\mathbf{0}] \mid n[\mathbf{0}])}_{R_1} \mid (\nu n)\,\underbrace{!(\nu n)n[\mathbf{0}]}_{R_2} \end{aligned}$$

6 Summary

We proposed an algorithmic method to decide whether two ambient calculus processes are spatially congruent. This method is based on an intuitive interpretation of processes as edge-labelled trees (called spatial trees), and a strongly normalizing rewriting system. The results presented here extend previous results given in (Cardelli and Gordon 1999a) for a version of the calculus without restriction.

References

Berry, G. and G. Boudol (1992). The chemical abstract machine. *Theoretical Computer Science 96*, 217–248. 1

Cardelli, L. and A. D. Gordon (1998). Mobile ambients. In M. Nivat (Ed.), *Proc. of FoSSaCS '98 – Foundations of Software Science and Computational Structures*, Volume 1378 of *Lecture Notes in Computer Science*, pp. 140–155. Springer-Verlag. To appear in Theoretical Computer Science, special issue on Coordination, D. Le Métayer. 1, 2, 2.1

Cardelli, L. and A. D. Gordon (1999a, July). Anytime, anywhere: Modal logics for mobile ambients. Extended abstract of (Cardelli and Gordon 2000). 1, 4, 5, 6

Cardelli, L. and A. D. Gordon (1999b). Types for mobile ambients. In *Proc. of POPL '99 – 26th Annual ACM Symposium on Principles of Programming Languages*, pp. 79–92. 2.2

Cardelli, L. and A. D. Gordon (2000, January). Anytime, anywhere: Modal logics for mobile ambients. In *Proc. of POPL '00 – 27th Annual ACM Symposium on Principles of Programming Languages*. 2, 3, 6

Engelfriet, J. and T. Geselma (1999, January). Multisets and structural congruence of the pi-calculus with replication. *Theoretical Computer Science 211*(1-2), 311–337. 2.1

Engelfriet, J. and T. Geselma (2000, January). Structural congruence in the pi-calculus with potential replication. Technical Report 00-02, Leiden Institute of Advanced Computer Science. 2.1

Hirschkoff, D. (1999). *Mise en oeuvre de preuves de bisimulation*. Ph. D. thesis, École Nationale des Ponts et Chaussées. 2.1

Milner, R. (1979, October). Flow graphs and flow algebras. *Journal of the ACM 26*(4), 794–818. 2

Milner, R. (1992). Functions as processes. *Mathematical Structures in Computer Science 2*, 119–141. 1, 2.1

A Proofs

A.1 Equality Between Spatial Trees

In this appendix, we give the proof that were omitted in Section 3.1.

Lemma A.1 $\varnothing \vdash !\varnothing.T + !\varnothing.T \rightarrow^* !\varnothing.T$.

Proof By induction on the structure of T.

(Empty) Then $T = \epsilon$. By definition, $!\varnothing.\epsilon = \epsilon$ and, by (Red Zero), $\varnothing \vdash \epsilon + \epsilon \rightarrow \epsilon$.

(Output) Then $T = \mu\langle M\rangle$. By definition, $!\varnothing.\mu\langle M\rangle = \infty\langle M\rangle$ and, by (Red Add Output), $\varnothing \vdash \infty\langle M\rangle + \infty\langle M\rangle \rightarrow \infty\langle M\rangle$.

(Input) Then $T = \mu(n).T$. By definition, $!\varnothing.\mu(n).T = \infty(n).T$ and, by (Red Add Input), $\varnothing \vdash \infty(n).T + \infty(n).T \rightarrow \infty(n).T$.

(Action) Then $T = \mu act\,\eta.T$. By definition, $!\varnothing.\mu act\,\eta.T = \infty act\,\eta.T$ and, by (Red Add Action), $\varnothing \vdash \infty act\,\eta.T + \infty act\,\eta.T \rightarrow \infty act\,\eta.T$.

(Edge) Then $T = \mu\eta[T]$. By definition, $!\varnothing.\mu\eta[T] = \infty\eta[T]$ and, by (Red Add Edge), $\varnothing \vdash \infty\eta[T] + \infty\eta[T] \rightarrow \infty\eta[T]$.

(Repl) Then $T = !X.T$. By definition, $!\varnothing.!X.T = !X.T$ and, by (Red Repl), $\varnothing \vdash !X.T + !X.T \rightarrow !X.T$.

(Sum) Then $T = T_1 + T_2$. By definition, $!\varnothing.T = !\varnothing.T_1 + !\varnothing.T_2$. By induction hypothesis, $\varnothing \vdash !\varnothing.T_i + !\varnothing.T_i \rightarrow^* !\varnothing.T_i$ for each $i \in \{1, 2\}$. Therefore, by (Red +), $\varnothing \vdash (!\varnothing.T_1 + !\varnothing.T_1) + (!\varnothing.T_2 + !\varnothing.T_2) \rightarrow^* !\varnothing.T$. □

Lemma A.2 $\varnothing \vdash !\varnothing.T + T \rightarrow^* !\varnothing.T$.

Proof By induction on the structure of T.

(Empty) Then $T = \epsilon$. By definition, $!\varnothing.\epsilon = \epsilon$ and, by (Red Zero), $\varnothing \vdash \epsilon + \epsilon \rightarrow \epsilon$.

(Output) Then $T = \mu\langle M\rangle$. By definition, $!\varnothing.\mu\langle M\rangle = \infty\langle M\rangle$ and, by (Red Add Output), $\varnothing \vdash \infty\langle M\rangle + \mu\langle M\rangle \rightarrow \infty\langle M\rangle$.

(Input) Then $T = \mu(n).T$. By definition, $!\varnothing.\mu(n).T = \infty(n).T$ and, by (Red Add Input), $\varnothing \vdash \infty(n).T + \mu(n).T \rightarrow \infty(n).T$.

(Action) Then $T = \mu act\,\eta.T$. By definition, $!\varnothing.\mu act\,\eta.T = \infty act\,\eta.T$ and, by (Red Add Action), $\varnothing \vdash \infty act\,\eta.T + \mu act\,\eta.T \rightarrow \infty act\,\eta.T$.

(Edge) Then $T = \mu\eta[T]$. By definition, $!\varnothing.\mu\eta[T] = \infty\eta[T]$ and, by (Red Add Edge), $\varnothing \vdash \infty\eta[T] + \mu\eta[T] \rightarrow \infty\eta[T]$.

(Repl) Then $T = !X.T$. By definition, $!\varnothing.!X.T = !X.T$ and, by (Red Add Repl), $\varnothing \vdash !X.T + !X.T \to !X.T$.

(Sum) Then $T = T_1 + T_2$. By definition, $!\varnothing.T = !\varnothing.T_1 + !\varnothing.T_2$. By induction hypothesis, $\varnothing \vdash !\varnothing.T_i + T_i \to^* !\varnothing.T_i$ for each $i \in \{1, 2\}$. Therefore, by (Red $+$), $\varnothing \vdash (!\varnothing.T_1 + T_1) + (!\varnothing.T_2 + T_2) \to^* !\varnothing.T$. □

Lemma A.3 *If* $\varnothing \vdash T \to S$ *then* $\varnothing \vdash !\varnothing.T \to^* !\varnothing.S$.

Proof By induction on the derivation of $\varnothing \vdash T \to S$.

(Red Zero) Then $T = S + \epsilon$. By definition, $!\varnothing.T = !\varnothing.S + \epsilon$ and, by (Red Zero), $\varnothing \vdash !\varnothing.S + \epsilon \to !\varnothing.S$.

(Red Add Edge) Then $T = \infty\eta[T'] + \mu\eta[T']$ and $S = \infty\eta[T']$. By (Red Add Edge), $\varnothing \vdash \infty\eta[T'] + \infty\eta[T'] \to !\varnothing.S$. Cases (Red Add Output), (Red Add Input) and (Red Add Action) are similar.

(Red Add Repl) Then $T = !X.T' + !X.T'$ and $S = !X.T'$. We have $!\varnothing.T = T$ and $!\varnothing.S = S$. Trivial. Case (Red Repl) is similar.

(Red Sub) Then $Y \vdash T \to S$ with $Y \subseteq \varnothing$. Hence, $Y = \varnothing$. Trivial.

(Red Copy) Then $T = !X.T' + T'$. This case is impossible since it contradicts $X \neq \varnothing$.

(Red η) Then $T = \mu\eta[T']$ and $S = \mu\eta(S')$ where $\varnothing \vdash T' \to S'$. By (Red η), $\varnothing \vdash \infty\eta[T'] \to \infty\eta[S']$. Cases (Red Input) and (Red Action) are similar.

(Red $+$) Then $T = T' + R$ and $S = S' + R$ where $\varnothing \vdash T' \to S'$. By induction hypothesis, $\varnothing \vdash !\varnothing.T' \to !\varnothing.S'$. By (Red $+$), $\varnothing \vdash !\varnothing.T' + !\varnothing.R \to !\varnothing.S' + !\varnothing.R$. □

Proof of Lemma 3.6 *If* $X \vdash T \to S$ *then* $fm(S) \subseteq fm(T)$ *and* $fm(T) \setminus fm(S) \subseteq X$.

Proof By induction on the derivation of $X \vdash T \to S$. Note that the only reduction rule that decreases the set of free markers is (Red Copy). All the other rules preserve the set of free markers.

(Red Zero) Then $T = S + \epsilon$ and $X = \varnothing$. Hence, $fm(T) = fm(S)$, as required.

(Red Add Edge) Then $T = \infty\eta[T'] + \mu\eta[T']$ and $S = \infty\eta[T']$ and $X = \varnothing$. Hence, $fm(T) = fm(S)$, as required.

(Red Add Repl) Then $T = !X.T' + !X.T'$ and $S = !X.T'$ and $X = \varnothing$. Hence, $fm(T) = fm(S)$, as required.

(Red Sub) Then $Y \vdash T \to S$ where $Y \subseteq X$. By induction hypothesis, $fm(S) \subseteq fm(T)$ and $fm(T) \setminus fm(S) \subseteq Y \subseteq X$, as required.

(Red Copy) Then $T = !X.T' + T'$ and $S = !X.T'$. Therefore, $\mathit{fm}(S) \subseteq \mathit{fm}(T)$ and $\mathit{fm}(T) \setminus \mathit{fm}(S) = X$, as required.

(Red η) Then $T = \mu\eta[T']$ and $S = \mu\eta(S')$ where $X \vdash T' \to S'$ and $\eta \notin X$. By induction hypothesis, $\mathit{fm}(S') \subseteq \mathit{fm}(T')$ and $\mathit{fm}(T') \setminus \mathit{fm}(S') \subseteq X$. Hence, $\mathit{fm}(S) \subseteq \mathit{fm}(T)$ and $\mathit{fm}(S) \setminus \mathit{fm}(T) \subseteq X$, as required.

(Red $+$) Then $T = T' + R$ and $S = S' + R$ where $X \vdash T' \to S'$ and $\mathit{fm}(R) \cap X = \varnothing$. By induction hypothesis, $\mathit{fm}(S') \subseteq \mathit{fm}(T')$ and $\mathit{fm}(T') \setminus \mathit{fm}(S') \subseteq X$. Hence, $\mathit{fm}(S) \subseteq \mathit{fm}(T)$ and $\mathit{fm}(T) \setminus \mathit{fm}(S) \subseteq X$, as required.

(Red Repl) Then $T = !Y.T'$ and $S = !Y'.S'$ where $X \cup Y \vdash T' \to S'$ and $Y' = (Y \cap \mathit{fm}(S'))$. By induction hypothesis, $\mathit{fm}(S') \subseteq \mathit{fm}(T')$ and $\mathit{fm}(T') \setminus \mathit{fm}(S') \subseteq X \cup Y$. By definition, $\mathit{fm}(S) = \mathit{fm}(S') \setminus Y'$ and $\mathit{fm}(T) = \mathit{fm}(T') \setminus Y$. Hence, $\mathit{fm}(S) \subseteq \mathit{fm}(T)$ and $\mathit{fm}(T) \setminus \mathit{fm}(S) \subseteq X$, as required.

(Red Input) Then $T = \mu(n).T'$ and $S = \mu(n).S'$ where $X \vdash T' \to S'$. By induction hypothesis, $\mathit{fm}(S') \subseteq \mathit{fm}(T')$ and $\mathit{fm}(T') \setminus \mathit{fm}(S') \subseteq X$. Hence, $\mathit{fm}(S) \subseteq \mathit{fm}(T)$ and $\mathit{fm}(T) \setminus \mathit{fm}(S) \subseteq X$, as required.

(Red Action) Then $T = \mu\mathit{act}\, n.T'$ and $S = \mu\mathit{act}\, n.S'$ where $X \vdash T' \to S'$. By induction hypothesis, $\mathit{fm}(S') \subseteq \mathit{fm}(T')$ and $\mathit{fm}(T') \setminus \mathit{fm}(S') \subseteq X$. Hence, $\mathit{fm}(S) \subseteq \mathit{fm}(T)$ and $\mathit{fm}(T) \setminus \mathit{fm}(S) \subseteq X$, as required. $\square$

Next, we define the *condensed representation* of a tree, $T^\star$, which, intuitively, factors out all the possible reductions involving an empty cone.

Condensed tree: $T^\star$

$$\epsilon^\star \triangleq \epsilon \qquad (\mu\langle M\rangle)^\star \triangleq \mu\langle M\rangle$$
$$(\mu\eta[T])^\star \triangleq \mu\eta[T^\star] \qquad (!X.T)^\star \triangleq !(X \cap \mathit{fm}(T)).T^\star$$
$$(\mu\mathit{act}\, n.T)^\star \triangleq \mu\mathit{act}\, n.T^\star \qquad (\mu(n).T)^\star \triangleq \mu(n).(T^\star)$$
$$(S+T)^\star \triangleq \begin{cases} T^\star & \text{if } S^\star = \epsilon \\ S^\star & \text{if } T^\star = \epsilon \\ S^\star + T^\star & \text{otherwise} \end{cases}$$

We also define a new reduction relation on trees, $S \downarrow T$, that corresponds to the evaluation of the condensed representation of a spatial tree (see Lemma A.6). The notation $S \Downarrow T$ is short for the fact that there is a sequence of reductions $S_1 \downarrow S_2, \ldots, S_k \downarrow S_{k+1}$ where $S = S_1$ and $T = S_{k+1}$.

Located reduction: $S \downarrow T$

(Do Zero)
$$\frac{}{T + \epsilon \downarrow T}$$

(Do η)
$$\frac{T \downarrow S}{\mu\eta[T] \downarrow \mu\eta[S]}$$

(Do $+$)
$$\frac{T \downarrow S}{T + R \downarrow S + R}$$

(Do Repl)
$$\frac{T \downarrow S \quad (Y' = Y \cap \mathit{fm}(S))}{!Y.T \downarrow !Y'.S}$$

(Do Input)
$$\frac{T \downarrow S}{\mu(n).T \downarrow \mu(n).S}$$

(Do Action)
$$\frac{T \downarrow S}{\mu\mathit{act}\, n.T \downarrow \mu\mathit{act}\, n.S}$$

Basic properties of the condensed representation of a spatial tree are:

Lemma A.4 *If $T \downarrow S$ then $\varnothing \vdash T \to S$.*

Proof An easy induction on the derivation of $T \downarrow S$. $\square$

Lemma A.5 *If $T \downarrow S$ then $!\varnothing.T \downarrow !\varnothing.S$.*

Proof By induction on the derivation of $T \downarrow S$

(Do Zero) Then $T = S + \epsilon$. Therefore, $!\varnothing.T = !\varnothing.S + \epsilon$ and, by (Do Zero), $!\varnothing.S + \epsilon \downarrow !\varnothing.S$, as required.

(Do η) Then $T = \mu\eta[T']$ and $S = \mu\eta(S')$ where $T' \downarrow S'$. By (Do η), $\infty\eta[T'] \downarrow \infty\eta[S']$, as required.

(Do $+$) Then $T = T' + R$ and $S = S' + R$ where $X \vdash T' \to S'$. By induction hypothesis, $!\varnothing.T' \downarrow !\varnothing.S'$. Hence, by (Do $+$), $!\varnothing.T' + !\varnothing.R \downarrow !\varnothing.S' + !\varnothing.R$, as required.

(Do Repl) Then $T = !Y.T'$ and $S = !Y'.S'$ where $T' \downarrow S'$ and $Y' = (Y \cap \mathit{fm}(S'))$. We have $T = !\varnothing.T$ and $S = !\varnothing.S$. Trivial.

(Do Input) Then $T = \mu(n).T'$ and $S = \mu(n).S'$ where $T' \downarrow S'$. By (Do Input), $\infty(n).T' \downarrow \infty(n).S'$, as required.

(Do Action) Then $T = \mu \mathit{act}\, n.T'$ and $S = \mu \mathit{act}\, n.S'$ where $T' \downarrow S'$. By (Do Action), $\mu \mathit{act}\, n.T' \downarrow \mu \mathit{act}\, n.S'$, as required. $\square$

Basic properties of the condensed form are:

Lemma A.6 *For all spatial trees T we have $T \Downarrow T^\star$.*

Proof By induction the structure of T.

(Empty) Then $T = \epsilon$. We have $\epsilon^\star = \epsilon$. Trivial.

(Input) Then $T = \mu(n).S$. By induction hypothesis, $S \Downarrow S^\star$. By (Do Input), $T \Downarrow T^\star$, as required.

(Output) Then $T = \mu\langle M\rangle$. We have $(\mu\langle M\rangle)^\star = \mu\langle M\rangle$. Trivial.

(Capability) Then $T = \mu \mathit{act}\, \eta.S$. By induction hypothesis, $S \Downarrow S^\star$. By (Do Action), $\mu \mathit{act}\, \eta.S \downarrow \mu \mathit{act}\, n.S^\star$, as required.

(Edge) Then $T = \mu\eta[S]$. Hence, $T^\star = \mu\eta[S^\star]$. By induction hypothesis, $S \Downarrow S^\star$. By (Do η), $T \Downarrow T^\star$, as required.

(Repl) Then $T = !Y.S$. Hence, $T^\star = !Y'.S^\star$ where $Y' = (Y \cap \mathit{fm}(S))$. By induction hypothesis, $S \Downarrow S^\star$. By (Do Repl), $T \Downarrow !Y'.S^\star$, as required.

(Sum) Then $T = S_1 + S_2$. By induction hypothesis, $S_i \Downarrow S_i^\star$ for each $i \in \{1,2\}$. Assume $S_i^\star \neq \epsilon$ for each $i \in \{1,2\}$. Hence, $T^\star = S_1^\star + S_2^\star$ and, by (Do +), $T \Downarrow T^\star$. Assume $S_1^\star = \epsilon$. By (Do +) and (Do Zero), $T \Downarrow S_2^\star$. The case $S_2^\star = \epsilon$ is symmetric. □

Lemma A.7 *For all spatial trees T we have $\varnothing \vdash T \to^* T^\star$.*

Proof By Lemma A.6, $T \Downarrow T^\star$. Therefore, by Lemma A.4, $\varnothing \vdash T \to^* T^\star$. □

Lemma A.8 *For all spatial trees T we have $fm(T^\star) = fm(T)$.*

Proof An easy induction on the structure of T. □

Lemma A.9 *If $X \vdash T \to S$ then $X \vdash T^\star \to^* S^\star$.*

Proof By induction on the derivation of $X \vdash T \to S$.

(Red Zero) Then $T = S + \epsilon$ and $X = \varnothing$. Hence, $T^\star = S^\star$. Trivial.

(Red Sub) Then $Y \vdash T \to S$ with $Y \subseteq X$. By induction hypothesis, $Y \vdash T^\star \to^* S^\star$. By (Red Sub), $X \vdash T^\star \to^* S^\star$, as required. Cases (Red η), (Red +), (Red Repl), (Red Input) and (Red Action) are similar.

(Red Add Edge) Then $T = \infty\eta[T'] + \mu\eta[T']$ and $S = \infty\eta[T']$ and $X = \varnothing$. Hence, $T^\star = \infty\eta[T'^\star] + \mu\eta[T'^\star]$ and $S^\star = \infty\eta[T'^\star]$. By (Red Add Edge), $X \vdash T^\star \to S^\star$, as required.

(Red Add Repl) Then $T = !X.T' + !X.T'$ and $S = !X.T'$. Hence, $T^\star = !Z.T'^\star + !Z.T'^\star$ and $S^\star = !Z.T'^\star$, where $Z = X \cap fm(T')$. By (Red Add Repl), $X \vdash T^\star \to S^\star$, as required.

(Red Copy) Then $T = !X.T' + T'$ and $S = !X.T'$. There two possible cases:

(1) Assume $T'^\star \neq \epsilon$. Hence, $T^\star = !Z.T'^\star + T'^\star$ and $S^\star = !Z.T'^\star$, where $Z = X \cap fm(T')$. By (Red Copy), $Z \vdash T^\star \to S^\star$. By (Red Sub), $X \vdash T^\star \to S^\star$, as required.

(2) Otherwise. Hence, $T^\star = !Z.T'^\star = !Z.\epsilon$ and $S^\star = !Z.\epsilon$, where $Z = X \cap fm(T')$. By (Red Copy), $X \vdash T^\star \to S^\star$, as required.

(Red +) Then $T = T' + R$ and $S = S' + R$ where $X \vdash T' \to S'$. By induction hypothesis, $X \vdash T'^\star \to^* S'^\star$. There are three cases

(1) Assume $T'^\star = \epsilon$. Hence, $X \vdash \epsilon \to^* S'^\star$, that is, $S'^\star = \epsilon$. Therefore $T^\star = S^\star = R^\star$. Trivial.

(2) Assume $R^\star = \epsilon$. Therefore $T^\star = T'^\star$ and $S^\star = S'^\star$. Trivial.

(3) Otherwise, $T^\star = T'^\star + R^\star$ and $S^\star = S'^\star + R^\star$. By (Red +), $X \vdash T^\star \to S^\star$, as required. □

Lemma A.10 *If $X \vdash T + \epsilon \to S$ then $X \vdash T \to^* S^\star$.*

Proof Assume $X \vdash T + \epsilon \to S$. Hence, $T^\star = (T + \epsilon)^\star$ and, by Lemma A.9, $X \vdash (T + \epsilon)^\star \to^* S^\star$. By Lemma A.7, $\varnothing \vdash T \to^* T^\star$. Hence, by (Red Sub), $X \vdash T \to^* S^\star$. □

Proof of Theorem 3.9 *If $X_1 \vdash T \to T_1$ and $X_2 \vdash T \to T_2$ then there exists a tree S such that $X_1 \cup X_2 \vdash T_1 \to^* S$ and $X_1 \cup X_2 \vdash T_2 \to^* S$.*

Proof By induction on the derivation of $X_1 \vdash T \to T_1$.

(Red Sub) Then $Y_1 \vdash T \to T_1$ and $Y_1 \subseteq X_1$. By induction hypothesis, there exists a tree S such that $Y \vdash T_1 \to^* S$ and $Y \vdash T_2 \to^* S$, where $Y = y_1 \cup X_2$. Hence, $Y \subseteq X_1 \cup X_2$. By (Red Sub), $X_1 \cup X_2 \vdash T_1 \to^* S$ and $X_1 \cup X_2 \vdash T_2 \to^* S$, as required.

(Red Zero) Then $T = T_1 + \epsilon$ and $X_1 = \varnothing$. Hence, $X_2 \vdash T_1 + \epsilon \to T_2$ and $X_1 \cup X_2 = X_2$. Let S be the spatial tree $T_2^\star$. By Lemma A.10, $X_2 \vdash T_1 \to^* S$. By Lemma A.7, $\varnothing \vdash T_2 \to^* S$. By (Red Sub), $X_2 \vdash T_2 \to^* S$, as required.

(Red η) Then $T = \mu\eta[R]$. There exist two trees, R_1, R_2, such that $T_i = \mu\eta[R_i]$ and $X_i \vdash R \to R_i$ for each $i \in \{1, 2\}$. By induction hypothesis, there is a tree S' such that $X_1 \cup X_2 \vdash R_i \to^* S'$ for each $i \in \{1, 2\}$. Let $S = \mu\eta[S']$. By (Red η), $X_1 \cup X_2 \vdash T_i \to^* S$ for each $i \in \{1, 2\}$, as required. Cases (Red Repl) and (Red Input) are similar.

(Red Action) Then $T = \mu act\, \eta.R$ and $T_1 = \mu act\, \eta.R_1$ where $X_1 \vdash R \to R_1$. We proceed by induction on the derivation of $X_2 \vdash T \to T_2$.

- **(Red Add Edge)** This case is not possible since $\mu act\, \eta.R$ is not a vector of the kind $\mu_1\eta[T'] + \mu_2\eta[T'']$. Cases (Red Add Repl), (Red Copy), (Red η), (Red +), (Red Repl) and (Red Input) are similar.
- **(Red Sub) and (Red Zero)** These cases have been proved previously.
- **(Red Action)** There exists a spatial tree R_2, such that $T_2 = \mu act\, \eta.R_2$ and $X_2 \vdash R \to R_2$. By induction hypothesis, there is a tree S' such that $X_1 \cup X_2 \vdash R_i \to^* S'$ for each $i \in \{1, 2\}$. Let $S = \mu act\, \eta.S'$. By (Red Action), $X_1 \cup X_2 \vdash T_i \to^* S$ for each $i \in \{1, 2\}$, as required.

(Red Input) Similar to case (Red η).

(Red Repl) Similar to case (Red η).

(Red Add Edge) Then $T = \infty\eta[R] + \mu\eta[R]$ and $T_1 = \infty\eta[R]$ and $X_1 = \varnothing$. We proceed by case analysis on the derivation of $X_2 \vdash T \to T_2$.

(Red Add Repl) This case is not possible possible. Cases (Red Copy), (Red η), (Red Repl), (Red Input) and (Red Action) are similar.

(Red Add Edge) Then $T_1 = T_2$. Trivial.

(Red Sub) and (Red Zero) These cases have been proved previously.

(Red $+$) We have two possible cases. Either (1), we have $X_2 \vdash \mu\eta[R] \to R'$ and $T_2 = \infty\eta[R] + R'$, or (2), we have $X_2 \vdash \infty\eta[R] \to R'$ and $T_2 = R' + \mu\eta[R]$. In each cases we have the side condition (H1): $fm(\mu\eta[R]) \cap X_2 = \varnothing$. Assume we are in case (1). By Lemma 5.2, there exist a tree S' and a set Y, with $Y \subseteq X_2$, such that $R' = \mu\eta[S']$ and $Y \vdash R \to S'$ and $\eta \notin Y$. Let S be the spatial tree $\infty\eta[S']$. By (Red η) and (Red Sub), $X_2 \vdash \infty\eta[R] \to S$. Hence, $X_2 \vdash T_1 \to^* S$. By Lemma 3.6 and (H1), $fm(\infty\eta[S']) \cap X_2 = \varnothing$. Hence, by (Red η), (Red $+$) and (Red Sub), $X_2 \vdash R' + \infty\eta[R] \to R' + S$. By (Red Add Edge), $X_2 \vdash R' + S \to S$. Hence, $X_2 \vdash T_2 \to^* S$, as require. Case (2) is symmetric.

(Red Add Repl) Then $T = !Y.R + !Y.R$ and $T_1 = !Y.R$ and $X_1 = \varnothing$. We proceed by case analysis on the derivation of $X_2 \vdash T \to T_2$.

(Red Add Edge) This case is not possible. Cases (Red Copy), (Red η), (Red Repl), (Red Input) and (Red Action) are similar.

(Red Add Repl) Then $T_1 = T_2$. Trivial.

(Red Sub) and (Red Zero) These cases have been proved previously.

(Red $+$) Then $T_2 = S+!Y.R$ where $X_2 \vdash !Y.R \to S$ and (H1): $fm(!Y.R) \cap X_2 = \varnothing$. By Lemma 5.3, there exists a tree S' such that $S = !Z.S'$ and $X_2 \cup Y \vdash R \to S'$ where Z is the set $(Y \cap fm(S'))$. By (Red Repl), $X_2 \vdash !Y.R \to !Z.S'$. Hence, $X_2 \vdash T_1 \to^* S$. By Lemma 3.6 and (H1), $fm(!Z.S') \cap X_2 = \varnothing$. Therefore, by (Red Repl) and (Red $+$), $X_2 \vdash S+!Y.R \to S+!Z.S'$ and, by (Red Add Repl), $X_2 \vdash S+S \to S$. Hence, $X_2 \vdash T_2 \to^* S$, as required.

(Red Copy) Then $T = !X_1.R + R$ and $T_1 = !X_1.R$. We proceed by case analysis on the derivation of $X_2 \vdash T \to T_2$.

(Red Zero) This case has been proved previously.

(Red Add Edge) This case is not possible since there are no trees R' such that $!Y.R = \infty\eta[R']$. Cases (Red Add Repl), (Red η), (Red Repl), (Red Input) and (Red Action) are similar.

(Red Copy) Then $T_1 = T_2$. Trivial.

(Red Sub) This case has been proved previously.

(Red $+$) We have two possible cases. Either (1), we have $T_2 = S + R$ where $X_2 \vdash !X_1.R \to S$ and with the side condition that $fm(R) \cap X_2 = \varnothing$. Either (2), we have $T_2 = !X_1.R + S$ where $X_2 \vdash R \to S$ and with the side condition that $fm(!X_1.R) \cap X_2 = \varnothing$. Assume we are in case (1). By Lemma 5.3, there is a tree S' such that $S = !Z.S'$ and $X_2 \cup X_1 \vdash R \to S'$ where $Z = (X_1 \cap \mathit{fm}(R'))$. By Lemma 3.6, $\mathit{fm}(S) \cap X_2 = \varnothing$. By (Red +), $X_1 \vdash S + R \to S + S'$. By (Red Copy), $X_1 \vdash S + R' \to S$. Hence, $X_1 \cup X_2 \vdash T_2 \to^* S$. On the other part, by (Red Repl), $X_2 \vdash !X_1.R \to S$. Hence, $X_1 \cup X_2 \vdash T_1 \to^* S$, as required. Case (2) is symmetric.

(Red +) Then $T = R_1 + R_2$ and $T_1 = S_1 + R_2$ where $X_1 \vdash R_1 \to S_1$ and with the side condition that $\mathit{fm}(R_2) \cap X_1 = \varnothing$. We proceed by case analysis on the derivation of $X_2 \vdash T \to T_2$.

(Red Repl) This case is not possible since there are no trees R such that $R_1 + R_2 = !Y.R$. Cases (Red Repl), (Red Input) and (Red Action) are similar.

(Red Zero) This case has been proved previously. Cases (Red Add Edge), (Red Add Repl), (Red Sub) and (Red Copy) are similar.

(Red +) A possible case is such that $X_2 \vdash R_2 \to S_2$ and $T_2 = R_2 + S_2$, that is, the reductions come from two distinct subparts of T. Hence, by induction hypothesis and (Red +), we have $X_1 \cup X_2 \vdash T_i \to S_1 + S_2$ for each $i \in \{1, 2\}$, as required. Another cases involve "critical pairs". We denote S the part (that is, the sum of the cones) of T that are not involved in the reductions. By inspection of the structure of T we obtain the following cases.

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

(Red Add Edge)-(Red Add Edge) Then $T = \infty\eta[R] + \mu\eta[R] + \infty\eta[R] + S$ and $T_1 = T_2 = \infty\eta[R] + \infty\eta[R] + S$. Trivial. Case (Red Add Repl)-(Red Add Repl) is similar.

(Red Add Edge)-(Red Copy) Then $T = \mu_1\eta[R] + \mu_2\eta[R] + !X.\mu_2 \eta[R] + S$ and $T_1 = \infty\eta[R] + !X.\mu_2\eta[R] + S$ and $T_2 = \mu_1\eta[R] + !X.\mu_2\eta[R] + S$ where $\mu_1 = \infty$ or $\mu_2 = \infty$. This case is impossible since $\mathit{fm}(\eta(R)) \cap X \neq \varnothing$, which conflicts with the side-condition of rule (Red +).

(Red Add Repl)-(Red Copy) Then $T = R + !X.R + !X.R + S$ and $T_1 = R + !X.R + S$ and $T_2 = !X.R + !X.R + S$, where $X_1 = \varnothing$ and $X_2 = X$. By (Red Copy) and (Red +), $X \vdash T_1 \to !X.R + S$. By (Red Add Repl) and (Red +), $\varnothing \vdash T_2 \to !X.R + S$, as required.

(Red Copy)-(Red Copy) Then $T = R_1 + !X.R + R_2 + S$ and $T_1 = !X.R + R_2 + S$ and $T_2 = R_1 + !X.R + S$ and there exist two disjoint sets, X_1, X_2, such that $R_1\{X \leftarrow X_1\} = R_2\{X \leftarrow X_2\} = R$ and $(X_1 \cup X_2) \cap \mathit{fm}(S) = \varnothing$. By (Red Copy) and (Red +), $X_2 \vdash T_1 \to !X.R + S$ and $X_1 \vdash T_2 \to !X.R + S$, as required. $\square$

A.2 Exponentiation of Spatial Trees

Proof of Lemma 3.13 *If $X \vdash T \to S$ then $X \vdash \mathit{exp}(T) \to^* \mathit{exp}(S)$.*

Proof This lemma follows by showing that for all tree R such that $\mathit{fm}(R) \cap X = \varnothing$, if $X \vdash T \to S$ then $X \vdash \mathit{exp}(T+R) \to^* \mathit{exp}(S+R)$. We proceed by induction on the derivation of $X \vdash T \to S$.

(Red Zero) Then $T = S + \epsilon$ and $X = \varnothing$. In this case, $\mathit{exp}(T+R) = \mathit{exp}(S+R)$. Trivial.

(Red Add Edge) Then $T = \infty\eta[T'] + \mu\eta[T']$ and $S = \infty\eta[T']$ and $X = \varnothing$. Let R be a spatial tree. There are two possible cases:

(1) Either $\mathit{fm}(\mu\eta[T']) = \varnothing$. Then $T+R$ has at least two connected components and $\mathit{exp}(T+R) = \infty\eta[T'] + \infty\eta[T'] + \mathit{exp}(R)$. By (Red Add Repl), $\varnothing \vdash \mathit{exp}(T+R) \to \infty\eta[T'] + \mathit{exp}(R) = \mathit{exp}(S)$, as required.

(2) Otherwise, let $Y = \mathit{fm}(\mu\eta[T'])$. By inspection of the structure of R, there must exists a partition of R, say (R_1, R_2), such that $\mathit{fm}(R_1) \cap Y \neq \varnothing$ and $\mathit{fm}(R_2) \cap Y = \varnothing$. Therefore:

$$\mathit{exp}(T+R) = !Y.\infty\eta[T'] + \mu\eta[T'] + R_1 + \mathit{exp}(R_2)$$

By (Red Add Edge), (Red +) and (Red Repl):

$$\varnothing \vdash \mathit{exp}(T+R) \to !Y'.\infty\eta[T'] + R_1 + \mathit{exp}(R_2)$$

Where $Y' = Y \cap \mathit{fm}(\infty\eta[T'] + R_1)$. Since $\mathit{fm}(\mu\eta[T']) = \mathit{fm}(\infty\eta[T'])$, we have $Y' = Y$. Hence, $\varnothing \vdash \mathit{exp}(T+R) \to !Y.\infty\eta[T'] + R_1 + \mathit{exp}(R_2) = \mathit{exp}(S+R)$, as required.

Cases (Red Add Repl) and (Red Copy) are similar.

(Red Sub) Then $Y \vdash T \to S$ with $Y \subseteq X$. Let R be a tree such that $\mathit{fm}(R) \cap X = \varnothing$. Hence, $\mathit{fm}(R) \cap Y = \varnothing$. By induction hypothesis, $Y \vdash \mathit{exp}(T+R) \to^* \mathit{exp}(S+R)$. By (Red Sub), $X \vdash \mathit{exp}(T+R) \to^* \mathit{exp}(S+R)$, as required.

(Red η) $T = \mu\eta[T']$ and $S = \mu\eta[S']$ where $X \vdash T' \to S'$ and $\eta \notin X$. Let Z be the free markers of $\eta[T']$. Let R be a tree such that $\mathit{fm}(R) \cap X = \varnothing$. By inspection of the structure of R, there must exists a partition of R, say (R_1, R_2), such that $\mathit{fm}(R_1) \cap Z \neq \varnothing$ and $\mathit{fm}(R_2) \cap Z = \varnothing$. We have two different cases:

(1) Assume $Z \neq \varnothing$. Therefore, $\mathit{exp}(T+R) = !Z.\mu\eta[T'] + R_1 + \mathit{exp}(R_2)$. By Lemma A.9, $X \vdash T' \to S'$. By (Red Repl), (Red +) and (Red η), $X \setminus Z \vdash \mathit{exp}(T+R) \to^* !Z'.\mu\eta[S'] + R_1 + \mathit{exp}(R_2)$, where $Z' = Z \cap \mathit{fm}(\mu\eta[S'] + R_1)$. Therefore, $X \setminus Z \vdash \mathit{exp}(T+R) \to^* \mathit{exp}(S+R)$. By (Red Sub), $\mathit{exp}(T+R) \to^*_X \mathit{exp}(S+R)$, as required.

(2) Assume $Z = \varnothing$. Therefore, $\mathit{exp}(T+R) = \infty\eta[T'] + \mathit{exp}(R)$ and, by (Red η) and Lemma A.9, $X \vdash \mathit{exp}(T+R) \to \infty\eta[S'] + \mathit{exp}(R) = \mathit{exp}(S+R)$, as required.

Cases (Red Repl), (Red Input) and (Red Action) are similar.

(Red +) Then $T = T_1 + T_2$ and $S = S_1 + T_2$ where $X \vdash T_1 \rightarrow S_1$ and $(fm(T_2) \cap X = \varnothing)$. Let R be a tree such that $fm(R) \cap X = \varnothing$. Hence, $R+T_2$ is a tree such that $fm(R + T_2) \cap X = \varnothing$. By induction hypothesis, $X \vdash exp(T_1 + (T_2 + R)) \rightarrow^* exp(S_1 + (T_2 + R))$. By associativity of $+$, $X \vdash exp(T + R) = exp(T_1 + T_2 + R) \rightarrow^* exp(S_1 + T_2 + R) = exp(S + R)$, as required. $\square$

A.3 Relation Between Trees and Processes

Proof of Lemma 4.2 *For all processes P we have $mean([\![P]\!]) \equiv P$.*

Proof By induction on the structure of P.

(Zero) Then $P = \mathbf{0}$. Hence, $mean([\![\mathbf{0}]\!]) = mean(\epsilon) = \mathbf{0}$.

(Par) Then $P = P_1 \mid P_2$. By definition, $mean([\![P_1 \mid P_2]\!]) = mean([\![P_1]\!] + [\![P_2]\!])$, where $fm([\![P_1]\!]) \cap fm([\![P_2]\!]) = \varnothing$, and therefore, $mean([\![P_1]\!] + [\![P_2]\!]) = mean([\![P_1]\!]) \mid mean([\![P_2]\!])$. By induction hypothesis, $P_i \equiv mean([\![P_i]\!])$ for each $i \in \{1, 2\}$. By (Struct Par), $mean([\![P_1]\!]) \mid mean([\![P_2]\!]) \equiv P_1 \mid P_2$, as required.

(Amb) Then $P = m[P']$. Let σ be a bijection from $fm([\![P]\!])$ to a set of fresh names and K be the image of σ. Hence, we can assume that $m \notin K$. Therefore, $mean([\![m[P']]\!]) = mean(1m[[\![P']\!]]) = (\nu K)m[(\![[\![P']\!]\sigma]\!)]$.

By (Struct Res Amb), $mean([\![m[P']]\!]) \equiv m[mean([\![P']\!])]$. By induction hypothesis, $mean([\![P']\!]) \equiv P'$. By (Struct Amb), $m[mean([\![P']\!])] \equiv m[P']$. Hence, $mean([\![m[P']]\!]) \equiv m[P']$, as required.

(Res) Then $P = (\nu m)P'$. Let σ be a bijection from $fm([\![P]\!])$ to a set of fresh names and K be the image of σ. Hence, we can assume $m \notin K$. Let y be a fresh marker. Therefore:

$$\begin{aligned}
mean([\![(\nu m)P']\!]) &= (\nu K)(\![[\![(\nu m)P']\!]]\!)\sigma \\
&= (\nu K)(\nu m)(\![[\![P'\{m \leftarrow y\}]\!]]\!)\{y \leftarrow m\}\sigma \\
&= (\nu K)(\nu m)(\![[\![P']\!]]\!)\sigma
\end{aligned}$$

By induction hypothesis, $mean([\![P']\!]) \equiv P'$. By (Struct Res) and (Struct Res Res), $(\nu m)mean([\![P']\!]) \equiv (\nu m)P'$, as required.

(Repl) Then $P = !P'$. By induction hypothesis, $mean([\![P']\!]) \equiv P'$. Let $T = [\![P']\!]$ and $\{T_1, \ldots, T_p\}$ be the set $conn(T)$. Hence, $[\![P']\!] = \sum_{i \in 1..p} T_i$, and $[\![P]\!] = exp(T) = \sum_{i \in 1..p} !Y_i.T_i$, where $Y_i = fm(T_i)$. In particular, $fm([\![P]\!]) = \varnothing$. Hence, $mean([\![P]\!]) = \sum_{i \in 1..p} !(\nu K_i)(\![T_i]\!)\sigma_i$, where σ_i is a bijection from $fm(T_i)$ to fresh names and K_i is the image of σ_i. By (Struct

Trans) and (Struct Repl Par), $\mathit{mean}(\llbracket !P' \rrbracket) \equiv !\sum_{i \in 1..p}(\nu K_i)(\!\lbrack T_i \rbrack\!)\sigma_i$. By (Struct Trans), (Struct Repl) and (Struct Res Par):

$$\mathit{mean}(\llbracket !P' \rrbracket) \equiv !(\nu K_1, \ldots, K_p)(\!\lbrack \sum_{i \in 1..p} T_i \sigma_i \rbrack\!)$$

Hence, $\mathit{mean}(\llbracket !P' \rrbracket) \equiv !\mathit{mean}(\llbracket P' \rrbracket) \equiv P$, as required.

(Action) Then $P = M.P'$. The case follows by showing that for any actions M, M', we have (1) $\mathit{mean}(\llbracket M.P \rrbracket) \equiv M.P$ and (2) $\mathit{mean}(\llbracket (M.M').P \rrbracket) \equiv M.(M'.P)$. We proceed by induction on the structure of M. Let σ be a bijection from $\mathit{fm}(\llbracket P \rrbracket)$ to a set of fresh names and K be the image of σ.

(ϵ) For (1), we have $\mathit{mean}(\llbracket \epsilon.P' \rrbracket) = \mathit{mean}(1\epsilon.\llbracket P' \rrbracket) = (\nu K)(\!\lbrack \llbracket P' \rrbracket \sigma \rbrack\!)$. By induction hypothesis, $\mathit{mean}(\llbracket P' \rrbracket) \equiv P'$. By (Struct ϵ), $\mathit{mean}(\llbracket P' \rrbracket) \equiv \epsilon.P'$. Hence, $\mathit{mean}(\llbracket \epsilon.P' \rrbracket) \equiv \epsilon.P'$.

For (2), we have $\mathit{mean}(\llbracket (\epsilon.M').P \rrbracket) = (\nu K)(\!\lbrack \llbracket \epsilon.(M'.P') \rrbracket \sigma \rbrack\!)$. By induction hypothesis (1), $\mathit{mean}(\llbracket M'.P' \rrbracket) \equiv M'.P'$. By (Struct ϵ), $\mathit{mean}(\llbracket (\epsilon.M').P' \rrbracket) \equiv \epsilon.(M'.P')$, as required.

($\mathit{act}\, n$) For (1), we have:

$$\begin{aligned} \mathit{mean}(\llbracket \mathit{act}\, n.P' \rrbracket) &= \mathit{mean}(1\mathit{act}\, n.\llbracket P' \rrbracket) \\ &= (\nu K)\mathit{act}\, \sigma(n).(\!\lbrack \llbracket P' \rrbracket \sigma \rbrack\!) \end{aligned}$$

By (Struct Res Action), since n is not a marker, $\mathit{mean}(\llbracket \mathit{act}\, n.P' \rrbracket) = \mathit{act}\, n.(\nu K)\ (\!\lbrack \llbracket P' \rrbracket \sigma \rbrack\!)$. By induction hypothesis, $\mathit{mean}(\llbracket P' \rrbracket) \equiv P'$ and by (Struct Action), $\mathit{act}\, n.\mathit{mean}(\llbracket P' \rrbracket) \equiv \mathit{act}\, n.P'$. Hence, $\mathit{act}\, n.P'$ and $\mathit{mean}(\llbracket \mathit{act}\, n.P' \rrbracket)$ are spatially congruent, as required. Part (2) is similar.

($M_1.M_2$) For (1), we have, $\mathit{mean}(\llbracket (M_1.M_2).P' \rrbracket) = \mathit{mean}(\llbracket M_1.(M_2.P') \rrbracket)$. By induction hypothesis (2), $\mathit{mean}(\llbracket M_1.(M_2.P') \rrbracket) \equiv M_1.(M_2.P')$. By (Struct .), $M_1.(M_2.P') \equiv (M_1.M_2).P'$, as required.

For (2), we have:

$$\mathit{mean}(\llbracket ((M_1.M_2).M').P' \rrbracket) = \mathit{mean}(\llbracket M_1.((M_2.M').P') \rrbracket)$$

By induction hypothesis (2):

$$\mathit{mean}(\llbracket M_1.((M_2.M').P') \rrbracket) \equiv M_1.((M_2.M').P')$$

By (Struct .), $M_1.((M_2.M').P') \equiv (M_1.M_2).(M'.P')$, as required.

(Input) Then $P = (n).P'$. Let σ be a bijection from $\mathit{fm}(\llbracket P \rrbracket)$ to a set of fresh names and K be the image of σ. Hence, we can assume $n \notin K$. Therefore, $\mathit{mean}(\llbracket (n).P' \rrbracket) = \mathit{mean}(1(n).\llbracket P' \rrbracket) = (\nu K)1(n).(\!\lbrack \llbracket P' \rrbracket \sigma \rbrack\!)$. By (Struct Res Input), $\mathit{mean}(1(n).\llbracket P' \rrbracket) \equiv (n).(\nu K)(\!\lbrack \llbracket P' \rrbracket \sigma \rbrack\!)$. By induction hypothesis, $\mathit{mean}(\llbracket P' \rrbracket) \equiv P'$. By (Struct Input), $(n).\mathit{mean}(\llbracket P' \rrbracket) \equiv (n).P'$. Hence, $\mathit{mean}(\llbracket (n).P' \rrbracket) \equiv (n).P'$, as required.

(Output) Then $P = \langle M \rangle$. Hence, $fm(P) = \varnothing$ and therefore $mean([\![P]\!]) = mean(1\langle M \rangle) = P$. $\square$

Lemma A.11 *If $X \vdash T \to S$ then $(\nu K)(\![T\sigma]\!) \equiv (\nu K)(\![S\sigma]\!)$, where σ is a bijection from $fm(T)$ to a set of fresh names and $K = \sigma(fm(T) \cap X)$.*

Proof By induction on the derivation of $X \vdash T \to S$. Let σ be a bijection from $fm(T)$ to a set of fresh names and K be the image of $(fm(T) \cap X)$ by σ.

(Red Zero) Then $T = S + \epsilon$ and $X = \varnothing$. Hence, $K = \varnothing$ and $(\![T\sigma]\!) = (\![S\sigma]\!) \mid \mathbf{0}$. By (Struct Par Zero), $(\![T\sigma]\!) \equiv (\![S\sigma]\!)$, as required.

(Red Add Edge) Then $T = \infty\eta[R] + \mu\eta[R]$ and $S = \infty\eta[R]$ and $X = \varnothing$. Hence, $K = \varnothing$. For convenience, if η is a name we denote it by n. Otherwise, we denote η by $\sigma(\eta)$. There are two possible cases.

(1) Assume $\mu = \infty$. Hence, $(\![T\sigma]\!) = !n[(\![R\sigma]\!)] \mid !n[(\![R\sigma]\!)]$ and $(\![S\sigma]\!) = !n[(\![R\sigma]\!)]$. By Lemma 2.1, $(\![T\sigma]\!) \equiv (\![S\sigma]\!)$, as required.

(2) Assume $\mu = 1$. Hence, $(\![T\sigma]\!) = !n[(\![R\sigma]\!)] \mid n[(\![R\sigma]\!)]$ and $(\![S\sigma]\!) = !n[(\![R\sigma]\!)]$. By (Struct Repl Copy) and (Struct Symm), $(\![T\sigma]\!) \equiv (\![S\sigma]\!)$, as required.

Cases (Red Add Output), (Red Add Input) and (Red Add Action) are similar.

(Red Add Repl) Then $T = !X'.T' + !X'.T'$ and $S = !X'.T'$ and $X = \varnothing$. Hence, $K = \varnothing$ and $(\![T\sigma]\!) = !(\nu\sigma(X'))(\![T'\sigma]\!) \mid !(\nu\sigma(X'))(\![T'\sigma]\!)$ and $(\![S]\!) = !(\nu\sigma(X'))(\![T'\sigma]\!)$. By Lemma 2.1, $(\![T\sigma]\!) \equiv (\![S\sigma]\!)$, as required.

(Red Sub) we have $X' \vdash T \to S$ and $X' \subseteq X$. Let $K' = \sigma(fm(T) \cap L')$. Hence, $K' \subseteq K$. By induction hypothesis, $(\nu K')(\![T\sigma]\!) \equiv (\nu K')(\![S\sigma]\!)$. By (Struct Res), $(\nu K)(\![T\sigma]\!) \equiv (\nu K)(\![S\sigma]\!)$, as required.

(Red Copy) Then $T = !X.T' + T'$ and $S = !X.T$. Hence, $(\nu K)(\![T\sigma]\!) = (\nu K)(!(\nu K)(\![T'\sigma]\!) \mid (\![T'\sigma]\!))$. By (Struct Res Par), $(\nu K)(!(\nu K)(\![T'\sigma]\!) \mid (\![T'\sigma]\!)) \equiv !(\nu K)(\![T'\sigma]\!) \mid (\nu K)((\![T'\sigma]\!))$, and therefore, by (Struct Repl Copy), $(\nu K)(\![T\sigma]\!) = (\nu K)(\![S\sigma]\!)$.

(Red η) Then $T = \mu\eta[T']$ and $S = \mu\eta[S']$ where $X \vdash T' \to S'$ and $\eta \notin X$. For convenience, if η is a name we denote it by n. Otherwise, we denote it by $\sigma(\eta)$. We also assume that $\mu = 1$. The case $\mu = \infty$ is similar. Hence, $n \notin K$, and $(\nu K)(\![T\sigma]\!) = (\nu K)n[(\![T'\sigma]\!)]$. By (Struct Res Amb), $(\nu K)(\![T\sigma]\!) \equiv n[(\nu K)(\![T'\sigma]\!)]$. By induction hypothesis, $(\nu K)(\![T'\sigma]\!) = (\nu K)(\![S'\sigma]\!)$. Therefore, by (Struct Amb) and (Struct Res Amb), $(\nu K)(\![T\sigma]\!) \equiv (\nu K)n[(\![S'\sigma]\!)] = (\nu K)(\![S\sigma]\!)$, as required. Cases (Red +), (Red Repl), (Red Input) and (Red Action) are similar. $\square$

Proof of Theorem 4.1 *If $P \equiv Q$ then $[\![P]\!] \approx [\![Q]\!]$.*

Proof By induction on the derivation of $P \equiv Q$.

(Struct Refl) Trivial.

(Struct Symm) Then $Q \equiv P$. By induction hypothesis, $[\![Q]\!] \approx [\![P]\!]$, as required.

(Struct Trans) Then there exists R such that $P \equiv R$ and $R \equiv Q$. By induction hypothesis, $[\![P]\!] \approx [\![R]\!]$. Again, by induction hypothesis, $[\![R]\!] \approx [\![Q]\!]$. Hence, $[\![P]\!] \approx [\![R]\!]$.

(Struct Res) Then $P = (\nu n)P'$ and $Q \equiv (\nu n)Q'$ for some P', Q' with $P' \equiv Q'$. Hence, $[\![(\nu n)P']\!] = [\![P']\!]\{n \leftarrow x\}$ and $[\![(\nu n)Q']\!] = [\![Q']\!]\{n \leftarrow x\}$ for some fresh marker x. By induction hypothesis, $[\![P']\!] \approx [\![Q']\!]$. By Corollary 3.8, $[\![P]\!] \approx [\![Q]\!]$, as required.

(Struct Par) Then $P = P' \mid R$ and $Q = Q' \mid R$ with $P' \Rightarrow Q'$. Hence, $[\![P]\!] = [\![P']\!] + [\![R]\!]$ and $[\![Q]\!] = [\![Q']\!] + [\![R]\!]$, with the side condition that $\mathit{fm}([\![P']\!]) \cap \mathit{fm}([\![R]\!]) = \mathit{fm}([\![Q']\!]) \cap \mathit{fm}([\![R]\!]) = \varnothing$. By induction hypothesis, $[\![P']\!] \approx [\![Q']\!]$. By Proposition 3.5, $[\![P]\!] = [\![P']\!] + [\![R]\!] \approx [\![Q']\!] + [\![R]\!] = [\![Q]\!]$, as required.

(Struct Repl) Then $P = !P'$ and $Q = !Q'$ where $P' \equiv Q'$. Hence, $[\![P]\!] = \mathit{exp}([\![P']\!])$ and $[\![Q]\!] = \mathit{exp}([\![Q']\!])$. By induction hypothesis, $[\![P']\!] \approx [\![Q']\!]$. By Theorem 3.14, $\mathit{exp}([\![P']\!]) \approx \mathit{exp}([\![Q']\!])$, as required.

(Struct Amb) Then $P = n[P']$ and $Q = n[Q']$ where $P' \equiv Q'$. Hence, $[\![P]\!] = 1n[[\![P']\!]]$ and $[\![Q]\!] = 1n[[\![Q']\!]]$. By induction hypothesis, $[\![P']\!] \approx [\![Q']\!]$. By Proposition 3.5, $1n[[\![P']\!]] \approx 1n[[\![Q']\!]]$, as required.

(Struct Action) Then $P = M.P'$ and $Q = M.Q'$ where $P' \equiv Q'$. Hence, $[\![P]\!] = 1M.[\![P']\!]$ and $[\![Q]\!] = 1M.[\![Q']\!]$. By induction hypothesis, $[\![P']\!] \approx [\![Q']\!]$. By Proposition 3.5, $1M.[\![P']\!] \approx 1M.[\![Q']\!]$, as required.

(Struct Input) Then $P = (n).P'$ and $Q = (n).Q'$ where $P' \equiv Q'$. Hence, $[\![P]\!] = 1(n).[\![P']\!]$ and $[\![Q]\!] = 1(n).[\![Q']\!]$. By induction hypothesis, $[\![P']\!] \approx [\![Q']\!]$. By Proposition 3.5, $1(n).[\![P']\!] \approx 1(n).[\![Q']\!]$, as required.

(Struct Par Comm) Then $P = P_1 \mid P_2$ and $Q = P_2 \mid P_1$. For, we have $[\![P]\!] = [\![P_1]\!] + [\![P_2]\!]$ and, by commutativity of $+$, $[\![P]\!] = [\![P_2]\!] + [\![P_1]\!]$, as required.

(Struct Par Assoc) Then $P = (P_1 \mid P_2) \mid P_3$ and $Q = P_1 \mid (P_2 \mid P_3)$. For, we have $[\![P]\!] = ([\![P_1]\!] + [\![P_2]\!]) + [\![P_3]\!]$ and, by commutativity of $+$, $[\![P]\!] = [\![P_1]\!] + ([\![P_2]\!] + [\![P_3]\!])$, as required.

(Struct Par Zero) Then $P = Q \mid \mathbf{0}$. For, we have $[\![P]\!] = [\![Q]\!] + \epsilon \approx [\![Q]\!]$.

(Struct Repl Par) Then $P = !(P_1 \mid P_2)$ and $Q = !P_1 \mid !P_2$. Hence, $[\![P]\!] = \mathit{exp}([\![P_1]\!] + [\![P_2]\!])$, with the side condition that $\mathit{fm}([\![P_1]\!]) \cap \mathit{fm}([\![P_2]\!]) = \varnothing$. Moreover, for all tree T, $\mathit{fm}(\mathit{exp}(T)) = \varnothing$. Therefore, $[\![P]\!] = \mathit{exp}([\![P_1]\!]) + \mathit{exp}([\![P_2]\!]) = [\![!P_1 + !P_2]\!]$, as required.

(Struct Repl Zero) Then $P = !\mathbf{0}$ and $Q = \mathbf{0}$. Hence, $[\![P]\!] = \mathit{exp}([\![\mathbf{0}]\!]) = \epsilon$, as required.

(Struct Repl Copy) Then $P = !P'$ and $Q = !P' \mid P'$. By definition, $[\![Q]\!]$ is equal to $\mathit{exp}([\![P']\!]) + [\![P']\!]$. By Proposition 3.15, $[\![P]\!] = \mathit{exp}([\![P']\!]) \approx \mathit{exp}([\![P']\!]) + [\![P']\!]$, as required.

(Struct Repl Repl) Then $P = !!P'$ and $Q = !P'$. By definition, $[\![P]\!]$ is equal to $\mathit{exp}(\mathit{exp}([\![P']\!]))$ and $[\![Q]\!] = \mathit{exp}([\![P']\!])$. For all trees T we have $\mathit{exp}(\mathit{exp}(T)) = \mathit{exp}(T)$. Therefore, $[\![P]\!] \approx [\![Q]\!]$, as required.

(Struct Res Res) Then $P = (\nu n)(\nu m)P'$ and $Q = (\nu m)(\nu n)P'$. Since the spatial trees $[\![(\nu n)(\nu m)P]\!]$ and $[\![(\nu m)(\nu n)P]\!]$ are equal up to renaming of their free markers, we have $[\![P]\!] \approx [\![Q]\!]$, as required.

(Struct Res Par) Then $P = (\nu n)(P_1 \mid P_2)$ and $Q = P_1 \mid (\nu n)P_2$ where $n \notin \mathit{fn}(P_1)$. Hence, $[\![P]\!] = ([\![P_1]\!] + [\![P_2]\!])\{n{\leftarrow}x\}$ for some fresh marker x. By definition of the substitution function, $[\![P]\!] \approx [\![Q]\!]$, as required.

(Struct Res Amb) Then $P = (\nu n)m[P']$ and $Q = m[(\nu n)P']$ where $n \neq m$. Hence, $[\![P]\!] = (1m[[\![P']\!]])\{n{\leftarrow}x\}$ for some fresh marker x. By definition of the substitution function, we get that $[\![P]\!] \approx [\![Q]\!]$, as required.

(Struct Res Zero) Then $P = (\nu n)\mathbf{0}$ and $Q = \mathbf{0}$. Hence, $[\![P]\!] = \epsilon\{n{\leftarrow}x\} = \epsilon$. Therefore, $[\![P]\!] \approx [\![Q]\!]$, as required.

(Struct Res Action) Then $P = (\nu n)M.P'$ and $Q = M.(\nu n)P'$ where $n \notin \mathit{fn}(M)$. Hence, $[\![P]\!] = (1M.[\![P']\!])\{n{\leftarrow}x\} = [\![Q]\!]$, as required.

(Struct Res Input) Then $P = (\nu n)(m).P'$ and $Q = (m).(\nu n)P'$ where $n \neq m$. Hence, $[\![P]\!] = (1(m).[\![P']\!])\{n{\leftarrow}x\} = [\![Q]\!]$, as required.

(Struct ϵ) Then $P = \epsilon.Q$. By definition, $[\![P]\!] = [\![Q]\!]$. Trivial.

(Struct .) Then $P = (M.M').P'$ and $Q = M.(M'.P')$. By definition, $[\![P]\!] = [\![Q]\!]$. Trivial. □