Partial Order Reduction for Verification of Spatial Properties of Pi-calculus Processes[*]

Reynald Affeldt[†] Naoki Kobayashi[‡]

Abstract

Mechanical tools have recently been developed that enable computer-aided verification of spatial properties of concurrent systems. To be practical, these tools are expected to deal with the state-space explosion problem. In order to alleviate this problem, we investigate partial order reduction techniques. The main problem is that spatial logics are very expressive and some spatial formulas actually prevent partial order reduction. In this paper, we focus on the spatial properties of structure and reduction (mainly the composition formula, the temporal modality, and the guarantee formula): we recast the issue of partial order reduction in terms of process calculi, identify problems with standard definitions of spatial formulas, introduce adequate restrictions, and propose fragments of spatial logics for which we show that partial order reduction holds. Technically, our approach relies on exploiting partially confluent communications and on identifying so-called invisible communications.

1 Introduction

Spatial logics [5–8] have been drawing much attention as specification languages for concurrent systems. They can express, among others, properties of structure of concurrent systems, for example whether or not a concurrent system is composed of two or more identifiable subsystems, or properties of restriction, for example whether or not a secret is hidden.

Recently, efforts have been made to construct tools for computer-aided verification of spatial-logic specifications of concurrent systems. Vieira and Caires have been developing a model checker for automatic verification of finite-control concurrent systems written in a nominal π-calculus and specified using a rich spatial logic [19]. The authors of the present paper have been developing a library for interactive verification of concurrent systems written in an applied version of the π-calculus using a restricted spatial logic [1]. Like all verification tools for concurrent systems, these tools must deal with the state-space explosion problem.

In this paper, we investigate the application of partial order reduction techniques to alleviate the state-space explosion problem for verification of spatial properties. The main issue is that spatial logics are very expressive, and it turns out that some spatial formulas prevent partial order reduction. As a first step towards a full support of spatial logics, we focus here on the spatial properties of structure and reduction (mainly the composition formula, the temporal modality, and the guarantee formula); we do not

[*] This article is a revised and extended version of the paper that appeared in the proceedings of EXPRESS 2004.

[†] Department of Computer Science, University of Tokyo, affeldt at yl.is.s.u-tokyo.ac.jp

[‡] Department of Computer and Mathematical Science, Tohoku University, koba at ecei.tohoku.ac.jp

consider the spatial properties of restriction (mainly, the revelation formula and the fresh quantifier). Concretely, we recast the issue of partial order reduction in terms of process calculi, identify problems with standard definitions of spatial formulas, introduce adequate restrictions, and propose fragments of spatial logics for which we show that partial order reduction holds. Technically, our approach relies on exploiting partially confluent communications and on identifying so-called invisible communications.

We briefly review the basic idea of partial order reduction. Let us consider some satisfaction relation $\models$ between the states of some reduction system and some set of formulas. The basic idea of partial order reduction is to exploit reductions $P \to P'$ such that, for some formula ϕ, $P \models \phi \Leftrightarrow P' \models \phi$. In such situations, in order to verify whether $P \models \phi$, one can choose to perform the reduction $P \to P'$ (even if there are other possible reductions) and check whether $P' \models \phi$. In this paper, our goal is to find appropriate conditions for the formula ϕ and the reduction $P \to P'$ in the case where ϕ is a formula of the spatial logic and $P \to P'$ is a reduction of some process calculus.

The issue of partial order reduction has already been addressed for usual temporal logics such as LTL and CTL* (for Kripke structures), but not for spatial logics. In particular, the existence of expressive spatial formulas makes this question difficult. In addition, it is non-trivial to find an appropriate syntactic condition for $P \to P'$ in the case of a process calculus. In usual model checkers like Spin, $P \to P'$ is just a transition caused by access to a local variable [14], but in process calculi, all the computations are communications.

Our contributions can be summarized as follows:

1. We identify a set of spatial formulas whose standard definitions prevents partial order reduction.

2. We recast the problem of partial order reduction in terms of process calculi. In particular, we introduce a syntactic notion of invisible communication.

3. We propose fragments of spatial logics such that invisibility and partial confluence or linearity of communications are sufficient criterions to enable sound partial order reduction.

Outline In Sect. 2, we introduce our target process calculus and the spatial formulas we deal with in this paper. In Sect. 3, we show informally with an example that the knowledge of partially confluent communications enables partial order reduction for verification of spatial properties. In Sect. 4, we discuss spatial formulas that, in their original form, prevent partial order reduction. In Sect. 5, we formally recast the problem of partial order reduction in terms of process calculi, including in particular a syntactic notion of invisible communication. In Sect. 6, we introduce the *TSL* logic, a fragment of spatial logics such that partially confluent and invisible communications enable partial order reduction. In Sections 7 and 8, we propose extensions of the *TSL* logic for which we prove that partial order reduction still holds.

2 Preliminaries: Target Process Calculus and Standard Spatial Logics

In this section, we introduce our target process calculus and the spatial formulas we deal with in this paper. As stated in the introduction, we focus on those spatial formulas that express properties of structure and reduction and omit the spatial formulas that express properties of restriction.

The final goal of our study is to develop partial order reduction methods for the π-calculus. In this paper, however, we focus on a fragment of the π-calculus where there is no name-passing. The motivation for ignoring name-passing is the clarity of presentation: it allows for more compact process expressions. This simplification has no impact on the validity of our results. Indeed, the central definitions of partial confluence and invisibility (next section) are valid in both settings, and proofs can readily be adapted to the setting with name-passing (as we did in [2]).

There are two syntactic entities in our target process calculus: names and processes. Processes use names to interact. In this paper, names are ranged over by $x, y, z, c, d, e, f, g, h$ and processes are ranged over by P, Q, R, T, U. The syntax of processes is given by the following grammar (we omit external choice and restrict replication to input processes):

$$P ::= \overline{c}.P \mid c.P \mid !c.P \mid (P|Q) \mid \nu x.P \mid 0$$

The output process $\overline{c}.P$ can send a signal along c (intuitively, a channel of communication) and then behave as P. The input process $c.P$ can receive some signal along c, and then behave as P. Parallel composition $P|Q$ makes it possible for processes to interact. The replicated input $!c.P$ behaves as infinitely many input processes in parallel. The restriction $\nu x.P$ indicates that the scope of the name x is restricted to P (the restriction binds closer than the composition). The process 0 represents termination (we omit trailing zeros; for instance, we write $\overline{c}$ instead of $\overline{c}.0$).

We now define the operational semantics of our target calculus. It relies on a binary relation called *structural congruence* that relates processes that only differ by spatial rearrangements. It is formally defined as the least congruence relation satisfying the following rules ($fn(P)$ is the set of free names in P):

$P \equiv P\|0$	zero	$\nu x.0 \equiv 0$	reszero
$P\|Q \equiv Q\|P$	comm	$\nu x.(P\|Q) \equiv P\|\nu x.Q \ (x \notin fn(P))$	extrusion
$P\|(Q\|R) \equiv (P\|Q)\|R$	assoc	$!c.P \equiv\ !c.P \mid c.P$	rep
$\nu x.\nu y.P \equiv \nu y.\nu x.P$	swap	$!c.P \equiv\ !c.P \mid !c.P$	rep2

The operational semantics of our target calculus is defined by the following reduction semantics:

$$\frac{}{\overline{c}.P|c.Q \rightarrow P|Q}\ \textsf{com}$$

$$\frac{P \rightarrow Q}{\nu x.P \rightarrow \nu x.Q}\ \textsf{res} \qquad \frac{P \rightarrow P'}{P|Q \rightarrow P'|Q}\ \textsf{par} \qquad \frac{Q \rightarrow Q' \quad P \equiv Q \quad P' \equiv Q'}{P \rightarrow P'}\ \textsf{struct}$$

As usual, the reflexive transitive closure is noted $\rightarrow^*$.

Spatial logics [5–8] are defined by a set of formulas and a satisfaction relation between processes and formulas. In this paper, we focus on the subset of spatial formulas whose syntax is given by the following grammar:

$$\phi ::= \top \mid \neg\phi \mid \phi_1 \vee \phi_2 \mid \overline{c}.\phi \mid c.\phi \mid \Diamond\phi \mid 0 \mid \phi_1|\phi_2 \mid \phi_1 \triangleright \phi_2$$

The semantics of spatial formulas is given by the satisfaction relation $\models$ defined as follows (we abbreviate the prefix $\nu x_1.\cdots.\nu x_n$ as $\nu x_{1,\ldots,n}$):

$$\begin{array}{lll}
P \models \top & \text{iff} & \text{always true} \\
P \models \neg\phi & \text{iff} & \text{not } P \models \phi \\
P \models \phi_1 \vee \phi_2 & \text{iff} & P \models \phi_1 \text{ or } P \models \phi_2 \\
P \models c.\phi & \text{iff} & \text{there exist } T, R, y_1, \ldots, y_n \text{ such that } P \equiv \nu y_{1,\ldots,n}.(c.T|R) \\
 & & \text{with } c \notin y_1, \ldots, y_n \text{ and } \nu y_{1,\ldots,n}.(T|R) \models \phi \\
P \models \overline{c}.\phi & \text{iff} & \text{there exist } T, R, y_1, \ldots, y_n \text{ such that } P \equiv \nu y_{1,\ldots,n}.(\overline{c}.U|R) \\
 & & \text{with } c \notin y_1, \ldots, y_n \text{ and } \nu y_{1,\ldots,n}.(U|R) \models \phi \\
P \models \Diamond\phi & \text{iff} & \text{there exists } P' \text{ such that } P \rightarrow P' \text{ and } P' \models \phi \\
P \models 0 & \text{iff} & P \equiv 0 \\
P \models \phi_1|\phi_2 & \text{iff} & \text{there exist } R_1, R_2, y_1, \ldots, y_n \text{ such that } P \equiv \nu y_{1,\ldots,n}.(R_1|R_2) \\
 & & \ \text{with } \nu y_{1,\ldots,n}.R_1 \models \phi_1 \text{ and } \nu y_{1,\ldots,n}.R_2 \models \phi_2 \\
P \models \phi_1 \triangleright \phi_2 & \text{iff} & R \models \phi_1 \text{ implies } P|R \models \phi_2 \text{ for any process } R
\end{array}$$

The zero formula (noted 0), the composition formula (noted $|$), and the guarantee formula (noted $\triangleright$) are peculiar to spatial logics. P satisfies $\phi_1|\phi_2$ if there exist R_1, R_2 such that P has the form $R_1|R_2$ with R_1 satisfying ϕ_1 and R_2 satisfying ϕ_2. P satisfies $\phi_1 \triangleright \phi_2$ if for any R satisfying ϕ_1, the composition of R and P satisfies ϕ_2. Formulas $\top$, $\neg$, $\vee$, input/output formulas, and the temporal modality ($\Diamond$) are similar to formulas in modal logics for concurrent processes (see for example [3]).

3 Motivating Example

We are interested in verifying processes against spatial formulas. In this section, we show informally how partially confluent communications may simplify such verifications.

Let us consider the following process:

$$P = \underbrace{\overline{c}.\overline{e}.f|c}_{\text{process } T} \mid \underbrace{\overline{d}.e.f|d}_{\text{process } R}$$

In the (sub)process T, the process $\overline{c}.\overline{e}.f$ is ready to send some signal along the name c and the process c waits for input along the name c. The process R is similar in structure. Both T and R share the names e and f: they use e to perform a hand-shake and will both seek access to some resource available along name f.

Let us assume that we want to verify that there is no race condition in the process P along the name f (this is actually wrong). Put formally, we want to verify that there is no execution such that the spatial formula $\phi = f.\top|f.\top$ is eventually true (a process satisfies $f.\top|f.\top$ if it consists of two processes satisfying $f.\top$; a process satisfies $f.\top$ if it consists of an input process that waits for some signal along f). The motivation for such a verification may be that the resource along f expects processes T and R to perform inputs in some predetermined order.

Naive verification of P leads to the exhaustive enumeration of all execution paths, and in general this approach is impractical because it leads to the state-space explosion problem.

In comparison, the knowledge of partially confluent communications enables efficient verification. Informally, a communication is partially confluent when it commutes with all other communications. Because the communication along c in our example is partially confluent, the state-space of P can be represented as follows (we represent communication with arrows and annotate them with the name used for communication):

$$\begin{tikzcd}
 & P = \overline{c}.\overline{e}.f|c|\overline{d}.e.f|d \arrow[dl, "c"'] \arrow[dr, "d"] & \\
P' = \overline{e}.f|\overline{d}.e.f|d \arrow[dr, "d"'] & & P'' = \overline{c}.\overline{e}.f|c|e.f \arrow[dl, "c"] \\
 & Q = \overline{e}.f|e.f \arrow[d, "e"] & \\
 & f|f &
\end{tikzcd}$$

Since both possible execution paths lead to the same state, it is intuitively obvious that the verification of P can be reduced to the verification of, say, P'. Although this is true for the verification of the formula ϕ, this is wrong for the verification of the formula $c.\top \wedge e.\top$ (where the conjunction has its usual meaning) because the latter actually holds of P''.

The state-space reduction described above is an example of partial order reduction. In the rest of this paper, we investigate under which conditions it is sound in presence of spatial formulas. We first discuss problems raised by standard spatial formulas and then introduce fragments of spatial logics that enable partial order reduction. More precisely, we prove that partial order reduction is possible for partially confluent and "invisible" communications (intuitively, the communication along c in the example above is invisible because the same spatial formulas hold for P and P', and P'' and Q).

4 Partial Order Reduction with Spatial Formulas: Discussion

For LTL and CTL*, partial order reduction is sound for the fragment without the "next" formula (see for instance [9]). So, a natural question is: Is partial order reduction sound for the standard spatial logic without the "next" formula? The answer is no: as discussed below, there are many other formulas of the spatial logic that prevent partial order reduction.

Problem with the Zero Formula Using the zero formula (noted 0, see Sect. 2) of spatial logics, it is possible to write formulas to count the number of non-zero subprocesses (this is observed for instance in [13]). For example, formulas below hold respectively for processes with one, two, or three non-zero subprocesses:

$$\begin{aligned}
&\mathbf{1} \overset{def}{=} \neg 0 \wedge \neg(\neg 0|\neg 0)\\
&\mathbf{2} \overset{def}{=} (\neg 0|\neg 0) \wedge \neg(\neg 0|\neg 0|\neg 0)\\
&\mathbf{3} \overset{def}{=} (\neg 0|\neg 0|\neg 0) \wedge \neg(\neg 0|\neg 0|\neg 0|\neg 0)
\end{aligned}$$

These formulas prevent partial order reduction. For instance, in the following example, the problem of verifying P cannot be reduced to the problem of verifying P' because it would let us conclude that **3** must be eventually true:

$$\begin{tikzcd}
& P = c|\overline{c}.\overline{e}|d|\overline{d} \not\models \mathbf{3} \arrow[dl, "c"'] \arrow[dr, "d"] & \\
P' = \overline{e}|d|\overline{d} \models \mathbf{3} \arrow[dr, "d"'] & & c|\overline{c}.\overline{e} \not\models \mathbf{3} \arrow[dl, "c"] \\
& \overline{e} \not\models \mathbf{3} &
\end{tikzcd}$$

Problem with the Input/Output Formulas There are alternative definitions for input/output formulas (see [4, 5], or [7, 8] where the formula for ambient locations can be compared with the output formula). Depending on these definitions, partial order reduction may not hold. For instance, let us consider the following alternative semantics for input/output formulas:

$$\begin{array}{lll}
P \models \overline{c}.\phi & \text{iff} & \text{there exist } Q, y_1, \ldots, y_n \text{ such that } P \equiv \nu y_{1,\ldots,n}.\overline{c}.Q \\
& & \quad \text{with } c \notin y_1, \ldots, y_n \text{ and } \nu y_{1,\ldots,n}.Q \models \phi \\
P \models c.\phi & \text{iff} & \text{there exist } Q, y_1, \ldots, y_n \text{ such that } P \equiv \nu y_{1,\ldots,n}.c.Q \\
& & \quad \text{with } c \notin y_1, \ldots, y_n \text{ and } \nu y_{1,\ldots,n}.Q \models \phi
\end{array}$$

These definitions are problematic because they can be used to implicitly test for the absence of actions. For example, they prevent partial order reduction for the following verification:

$$\begin{tikzcd}
& \overline{c}|c|\overline{d}|d \not\models \overline{d}.\top|d.\top \arrow[dl, "c"'] \arrow[dr, "d"] & \\
\overline{d}|d \models \overline{d}.\top|d.\top \arrow[dr, "d"'] & & \overline{c}|c \not\models \overline{d}.\top|d.\top \arrow[dl, "c"] \\
& 0 \not\models \overline{d}.\top|d.\top &
\end{tikzcd}$$

Observe that this problem does not occur with the definitions we gave in Sect. 2.

Problem with the Temporal Modality To compensate for the loss of expressiveness due to the removal of the "next" temporal modality, we can introduce its weak version (also defined in [13]):

$$P \models \Diamond\phi \text{ iff there exists } P' \text{ such that } P \rightarrow^* P' \text{ and } P' \models \phi$$

There is still a problem: mixed use of this temporal modality and the composition formula of spatial logics. For example, partial order reduction is not sound for the following process:

$$\begin{tikzcd}
d.((c.\overline{e}|\overline{c}) \mid (c.\overline{e}|\overline{c})) \mid \overline{d} \not\models \Diamond\overline{e}.\top|\Diamond\overline{e}.\top \arrow[d, "d"] \\
(c.\overline{e}|\overline{c}) \mid (c.\overline{e}|\overline{c}) \models \Diamond\overline{e}.\top|\Diamond\overline{e}.\top
\end{tikzcd}$$

The discussion so far is sufficient to define a first non-trivial fragment of spatial logics for which partial order reduction holds; this is what we will do in Sect. 6. Before that, in Sect. 5, we recast in terms of process calculi the conditions (partial confluence and invisibility) that are used as sufficient conditions for partial order reduction. In Sections

7 and 8, we discuss further issues regarding partial order reduction for spatial formulas, such as mixing of the composition formula and the temporal modality and the guarantee formula. The reason for delaying the discussion about other spatial formulas is that it will be better understood in the light of the *TSL* logic.

5 Invisible Communications and Partial Confluence

Our approach to partial order reduction is based on identifying partially confluent communications and on a syntactic definition of invisibility for communications. In this section, we formally define these notions.

5.1 Invisible Communications

Intuitively, a communication is *invisible* when it cannot be observed by the formulas of the logic at hand. In the case of Kripke structures and usual temporal logics, invisible transitions (rather than communications) are defined as those transitions that do not change the truth of atomic propositions, and therefore that do not change the truth of propositional formulas. In the case of the process calculi and spatial logics, names become the natural equivalent of atomic propositions. Our idea is to define invisible communications as the communications that do not change the truth of some subset of spatial formulas with a syntactic criterion on names.

For the purpose of defining invisible communications, we augment the reduction semantics of our target process calculus with labels. Let us first explain the intuition behind those labels. A labeled reduction is written $P \xrightarrow{l,S} Q$ where l is a name or a special label ϵ and S is a set of names. More precisely, l is the name used for the communication or the special label ϵ for an internal communication, and the names in S are the names that are "revealed" by the communication. For instance:

$$\overline{c}|c.(d.\overline{e}|\overline{c}) \xrightarrow{c,\{c,d\}} d.\overline{e}|\overline{c}$$

The corresponding reduction semantics is formally defined as follows:

$$\frac{\mathsf{guards}(P|Q) = S}{\overline{c}.P|c.Q \xrightarrow{\{c,S\}} P|Q}\ \mathsf{com}$$

$$\frac{P \xrightarrow{\alpha} Q}{\nu x.P \xrightarrow{\alpha\backslash x} \nu x.Q}\ \mathsf{res} \qquad \frac{P \xrightarrow{\alpha} P'}{P|Q \xrightarrow{\alpha} P'|Q}\ \mathsf{par} \qquad \frac{Q \xrightarrow{\alpha} Q' \quad P \equiv Q \quad P' \equiv Q'}{P \xrightarrow{\alpha} P'}\ \mathsf{struct}$$

where guards is defined inductively as follows:

$$\begin{array}{ll} \mathsf{guards}(\overline{c}.P) = \{c\} & \mathsf{guards}(P|Q) = \mathsf{guards}(P) \cup \mathsf{guards}(Q) \\ \mathsf{guards}(c.P) = \{c\} & \mathsf{guards}(\nu x.P) = \mathsf{guards}(P) - \{x\} \\ \mathsf{guards}(!c.P) = \{c\} & \mathsf{guards}(0) = \emptyset \end{array}$$

and $\alpha\backslash x$ is defined as follows:

$$\begin{array}{rcl} (y,S)\backslash x & = & \left\{ \begin{array}{l} (y, S - \{x\}) \text{ if } y \neq x \\ (\epsilon, S - \{x\}) \text{ if } y = x \end{array} \right. \\ (\epsilon,S)\backslash x & = & (\epsilon, S - \{x\}) \end{array}$$

In this paper, labels are ranged over by α. We write $P \to Q$ instead of $P \xrightarrow{\alpha} Q$ when the label α is not relevant, and $P \xrightarrow{c} Q$ instead of $P \xrightarrow{c,S} Q$ when the set S is not relevant.

We now define formally invisible communications:

Definition 1 (Invisible Communication). The communication $P \xrightarrow{c,S} P'$ is invisible w.r.t. the set of names N if $(\{c\} \cup S) \cap N = \emptyset$.

For example, the communication $c|\overline{c}.(d.\overline{e}|\overline{c}) \xrightarrow{c,\{c,d\}} d.\overline{e}|\overline{c}$ is invisible with respect to the set of names $\{e\}$ because $\{c,d\} \cap \{e\} = \emptyset$.

Using this definition, we can define spatial logics such that the starting and ending processes of invisible communications cannot be discriminated. For example, in Sect. 6.1, we will see that the starting and ending processes of the communication above cannot be discriminated by the formula $\overline{e}.\top$ because the communication is invisible w.r.t. $\{e\} = fn(\overline{e}.\top)$ ($fn(f)$ is the set of free names in f).

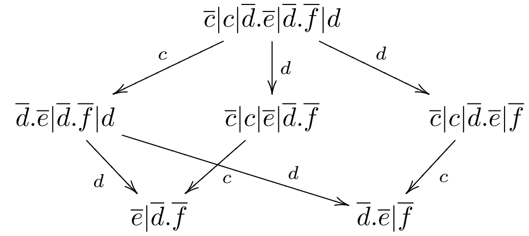
5.2 Partial Confluence

Intuitively, a communication is *partially confluent* when it commutes with any other communication. For example, this property is enjoyed by linearized names [16] and ω-receptive names [18]. More formally, we define partial confluence as follows:

Definition 2 (Partial Confluence). The set of *partially confluent communications* is the largest set S such that for any $(P, \alpha, Q) \in S$ we have:

- $P \xrightarrow{\alpha} Q$, and
- if $P \to P'$, then either:
  - $Q \equiv P'$, or
  - there exists Q' such that $Q \to Q'$ and $(P', \alpha, Q') \in S$.

For example, the communication along c in the process $P = \overline{c}|c|\overline{d}.\overline{e}|\overline{d}.\overline{f}|d$ is partially confluent because we have the following state-space:


$\overline{c}|c|\overline{d}.\overline{e}|\overline{d}.\overline{f}|d$
c
d
d
$\overline{d}.\overline{e}|\overline{d}.\overline{f}|d$
$\overline{c}|c|\overline{e}|\overline{d}.\overline{f}$
$\overline{c}|c|\overline{d}.\overline{e}|\overline{f}$
d
c
d
c
$\overline{e}|\overline{d}.\overline{f}$
$\overline{d}.\overline{e}|\overline{f}$

6 The *TSL* Logic: A Spatial Logic for Partial Order Reduction

As seen in Sect. 4, partial order reduction is not sound for the full spatial logic. In this section, we introduce the *TSL* logic, a restricted fragment of spatial logic for which partial order reduction holds.

6.1 The *TSL* Logic

The definition of the *TSL* logic takes into account the issues discussed in Sect. 4: there is no zero formula, the semantics of input/output formulas is defined appropriately, the usual temporal modality is replaced with its weak version (noted EF instead of $\Diamond$), arbitrary mixing of spatial and temporal formulas is prevented by distinguishing *state and temporal formulas.*

The set of formulas is a subset of the set of spatial formulas defined in Sect. 2.

Definition 3 (State Formulas). The syntax of *state formulas* is defined by the following grammar:

$$\phi ::= \top \mid \neg\phi \mid \phi_1 \vee \phi_2 \mid c.\phi \mid \bar{c}.\phi \mid (\phi_1|\phi_2)$$

The semantics of state formulas is defined by the satisfaction relation noted $\models$ defined as follows:

$$\begin{array}{lll}
P \models \top & \text{iff} & \text{always true} \\
P \models \neg\phi & \text{iff} & \text{not } P \models \phi \\
P \models \phi_1 \vee \phi_2 & \text{iff} & P \models \phi_1 \text{ or } P \models \phi_2 \\
P \models c.\phi & \text{iff} & \text{there exist } T, R, y_1, \ldots, y_n \text{ such that } P \equiv \nu y_{1,\ldots,n}.(c.T|R) \\
 & & \text{with } c \notin y_1, \ldots, y_n \text{ and } \nu y_{1,\ldots,n}.(T|R) \models \phi \\
P \models \bar{c}.\phi & \text{iff} & \text{there exist } T, R, y_1, \ldots, y_n \text{ such that } P \equiv \nu y_{1,\ldots,n}.(\bar{c}.U|R) \\
 & & \text{with } c \notin y_1, \ldots, y_n \text{ and } \nu y_{1,\ldots,n}.(U|R) \models \phi \\
P \models \phi_1|\phi_2 & \text{iff} & \text{there exist } R_1, R_2, y_1, \ldots, y_n \text{ such that } P \equiv \nu y_{1,\ldots,n}.(R_1|R_2) \\
 & & \text{with } \nu y_{1,\ldots,n}.R_1 \models \phi_1 \text{ and } \nu y_{1,\ldots,n}.R_2 \models \phi_2
\end{array}$$

Definition 4 (Temporal Formulas). The syntax of *temporal formulas* is defined by the following grammar:

$$f ::= \phi \mid \neg^t f \mid f_1 \vee^t f_2 \mid \text{EF } f \mid \text{AF } f$$

where ϕ ranges over the set of state formulas.

Their semantics is defined by the satisfaction relation noted $\models^t$ defined as follows. A *path* is a possibly infinite sequence of processes such that each process is obtained by a communication from the previous one. A path is *full* either if it is infinite, or if it is finite and the last process cannot be reduced. We write p_i for the ith process of a path p.

$$\begin{array}{lll}
P \models^t \phi & \text{iff} & P \models \phi \\
P \models^t \neg^t\phi & \text{iff} & \text{not } P \models^t \phi \\
P \models^t \phi_1 \vee^t \phi_2 & \text{iff} & P \models^t \phi_1 \text{ or } P \models^t \phi_2 \\
P \models^t \text{EF } f & \text{iff} & \text{there exists a path } p \text{ such that } p_1 = P \text{ and } p_k \models^t f \text{ for some } k \\
P \models^t \text{AF } f & \text{iff} & \text{for any full path } p \text{ such that } p_1 = P,\ p_k \models^t f \text{ for some } k
\end{array}$$

We define the *TSL* logic as the set of temporal formulas.

An important property of the *TSL* logic (and of spatial logics in general) is that the set of processes satisfying some formula is closed under structural congruence. We use this property silently throughout this paper.

We conclude this section with an important lemma stating that invisible communications cannot be observed by state formulas. Note that this lemma is not true for spatial logics in general but holds thanks to the restrictions we discussed in Sect. 4.

Lemma 1 (Invisible Communications cannot be Observed). Let ϕ be a state formula. If the communication $P \to P'$ is invisible w.r.t. $fn(\phi)$, then we have $P \models \phi \Leftrightarrow P' \models \phi$.

The proof of Lemma 1 relies on the following intermediate lemma. This intermediate lemma states that, if some state formula ϕ holds for some process, then we can remove without affecting validity input/output subprocesses whose input/output name is not a free name of ϕ (the proof can be found in appendix).

Lemma 2. Let c be a name and ϕ be a state formula such that $c \notin fn(\phi)$. For any $P, Q, y_1, \ldots, y_n$, we have $\nu y_{1,\ldots,n}.P \models \phi \Leftrightarrow \nu y_{1,\ldots,n}.(P|c.Q) \models \phi \Leftrightarrow \nu y_{1,\ldots,n}.(P|\overline{c}.Q) \models \phi$.

Thanks to this intermediate lemma, we can now prove Lemma 1:

Proof of Lemma 1. Let us assume that $P \xrightarrow{c,S} P'$. There exist $U, T, R, y_1, \ldots, y_n$ such that $P \equiv \nu y_{1,\ldots,n}.(\overline{c}.U|c.T|R)$ and $P' \equiv \nu y_{1,\ldots,n}.(U|T|R)$ with $\mathsf{guards}(U|T) = S$. Since $P \xrightarrow{c,S} P'$ is invisible w.r.t. $fn(\phi)$, $c \notin fn(\phi)$ and $S \cap fn(\phi) = \emptyset$. By Lemma 2, we have $\nu y_{1,\ldots,n}.(\overline{c}.U|c.T|R) \models \phi \Leftrightarrow \nu y_{1,\ldots,n}.R \models \phi$ and $\nu y_{1,\ldots,n}.R \models \phi \Leftrightarrow \nu y_{1,\ldots,n}.(U|T|R) \models \phi$. Therefore, $P \models \phi \Leftrightarrow P' \models \phi$. □

6.2 Partial Order Reduction

In this section, we prove that the knowledge of partially confluent and invisible communications enables partial order reduction for the *TSL* logic. We first state the corresponding theorem and illustrate with an example how it can be used to simplify reasoning about concurrent processes.

Theorem 1 (Partial Order Reduction for *TSL*). Let the communication $P \to Q$ be partially confluent and invisible w.r.t. the set of names N. Then, for any *TSL* formula f such that $fn(f) \subseteq N$, we have $P \models^t f \Leftrightarrow Q \models^t f$.

We show how to use this theorem on an example. Let us consider the following process:

$$P = \overline{c}|c|\overline{d}.\overline{e}.\overline{f}|d.e$$

Let us assume that we want to verify that P must eventually perform some output along the name f. Put formally, we want to verify whether $P \models^t \text{ AF } \overline{f}.\top$ holds.

Since the communication along name c is partially confluent and invisible with respect to the set of names $\{f\} = fn(\overline{f}.\top)$, by Theorem 1, this verification is equivalent to the verification of $P' \models^t \text{ AF } \overline{f}.\top$ with:

$$P' = \overline{d}.\overline{e}.\overline{f}|d.e$$

The latter verification is simpler because P' is deterministic. This simplification is better appreciated by examining the state-space of P, where we observe that the theorem allows us to restrict verification to the path $P, P', \ldots$:

$$\begin{tikzcd}
 & P = \overline{c}|c|\overline{d}.\overline{e}.\overline{f}|d.e \arrow[dl, "c"'] \arrow[dr, "d"] & & \\
P' = \overline{d}.\overline{e}.\overline{f}|d.e \arrow[dr, "d"] & & \overline{c}|c|\overline{e}.\overline{f}|e \arrow[dl, "c"'] \arrow[dr, "e"] & \\
 & \overline{e}.\overline{f}|e \arrow[dr, "e"'] & & \overline{c}|c|\overline{f} \arrow[dl, "c"] \\
 & & \overline{f} &
\end{tikzcd}$$

Similar reasoning can be applied to the example discussed in Sect. 3.

The rest of this section is dedicated to the proof of Theorem 1. We first introduce the notion of N-preserving bisimulation, where N is a set of names. We then show that if $P \to Q$ is partially confluent and invisible with respect to N, then P and Q are N-preserving bisimilar (Lemma 3). Finally, we show that N-preserving bisimilar processes satisfy the same temporal formulas f such that $fn(f) \subseteq N$ (Lemma 4). The theorem of partial order reduction for *TSL* (Theorem 1) is an immediate corollary of these lemmas.

Definition 5 (N-preserving Bisimulation). Let N be a set of names. A binary relation R on processes is an *N-preserving bisimulation* if whenever $(P, Q) \in R$:

- if $P \to P'$, then there exist $Q_1, \ldots, Q_n$ ($n \geq 1$) and i ($1 \leq i \leq n$) such that $Q = Q_1 \to \cdots \to Q_n$ and $(P, Q_1), \ldots, (P, Q_{i-1}), (P', Q_i), \ldots, (P', Q_n) \in R$,
- if $Q \to Q'$, then there exist $P_1, \ldots, P_n$ ($n \geq 1$) and i ($1 \leq i \leq n$) such that $P = P_1 \to \cdots \to P_n$ and $(P_1, Q), \ldots, (P_{i-1}, Q), (P_i, Q'), \ldots, (P_n, Q') \in R$, and
- for any state formula ϕ such that $fn(\phi) \subseteq N$, $P \models \phi \Leftrightarrow Q \models \phi$.

P and Q are N-bisimilar, written $P \approx_N Q$, if $(P, Q) \in R$ for some N-preserving bisimulation R.

Lemma 3. If $P \xrightarrow{\alpha} Q$ is an invisible communication w.r.t. a set N of names and if it is partially confluent, then $P \approx_N Q$.

Proof. Consider the relation $\mathcal{R} \overset{def}{=} \equiv \cup \{(P_1, Q_1) | P_1 \xrightarrow{\alpha} Q_1\}$. We show that $\mathcal{R}$ is an N-preserving bisimulation. Consider $(P, Q) \in \{(P_1, Q_1) | P_1 \xrightarrow{\alpha} Q_1\}$:

- Suppose that $P \to P'$. Since $P \xrightarrow{\alpha} Q$ is partially confluent, then either (1) $P' \equiv Q$, in which case we can take $Q_2 = Q$, and $(P, Q), (P', Q_2) \in \mathcal{R}$, or (2) there exists Q' such that $Q \to Q'$ and $P' \xrightarrow{\alpha} Q'$, in which case we can take $Q_2 = Q'$, and $(P, Q), (P', Q_2) \in \mathcal{R}$.
- Suppose that $Q \to Q'$. We have $P \xrightarrow{\alpha} Q \to Q'$. Therefore we can take $P_2 = Q$ and $P_3 = Q'$, and $(P, Q), (P_2, Q), (P_3, Q') \in \mathcal{R}$.
- Let ϕ be a state formula such that $fn(\phi) \subseteq N$. Since $P \xrightarrow{\alpha} Q$ is invisible with respect to N, by Lemma 1, we know that $P \models \phi \Leftrightarrow Q \models \phi$.

□

Lemma 4. If $P \approx_N Q$, then for any temporal formula f such that $fn(f) \subseteq N$, we have $P \models^t f \Leftrightarrow Q \models^t f$.

Proof. By induction on f:

- Case $f = \phi$: Given by Lemma 1.
- Cases $f = \neg^t f'$ and $f = f_1 \vee^t f_2$: Immediate.
- Case $f = \mathrm{EF}\ f'$:

  Case $\Rightarrow$. (The case $\Leftarrow$ is similar.) By assumption, there exists a path p such that $p_1 = P$ and there exists a natural k such that $p_k \models^t f'$. Since $P \approx_N Q$, there exists a path q such that $q_1 = Q$ and a natural j such that $p_k \approx_N q_j$. By the inductive hypothesis, $q_j \models^t f'$, which implies $Q \models^t \mathrm{EF}\ f'$.
- Case $f = \mathrm{AF}\ f'$:

  Case $\Rightarrow$. (The case $\Leftarrow$ is similar.) Let q be a full path such that $q_1 = Q$. Since $P \approx_N Q$, there exists a full path p such that $p_1 = P$ and such that for any natural i, there exists j_i such that $p_i \approx_N q_{j_i}$. Since $P \models^t \mathrm{AF}\ f'$, there exists a natural k such that $p_k \models^t f'$. By the inductive hypothesis, $q_{j_k} \models^t f'$. Thus, we have $Q \models^t \mathrm{AF}\ f'$.

□

Thanks to above lemmas, we can now prove Theorem 1:

Proof of Theorem 1. Obtained directly as a corollary of Lemma 4 by using Lemma 3. □

7 Extension: Mixing Spatial and Temporal Formulas

In the previous section, we have introduced the *TSL* logic, a fragment of spatial logic for which partial order reduction holds. Since arbitrary mixing of spatial and temporal formulas prevents partial order reduction, the *TSL* logic imposes a stratification between state and temporal formulas. In this section, we introduce (1) a formula called *guarded composition* that allows for temporal formulas to appear inside the composition formula, and (2) a fragment of spatial logic called *TSLmix* extended with the guarded composition for which partial order reduction still holds.

7.1 Guarded Composition and the *TSLmix* Logic

Intuitively, the guarded composition combines the semantics of the EF formula and of the composition formulas. Put formally:

Definition 6 (Guarded Composition). Let L be some logic with temporal formulas whose satisfaction relation is written $\models^t$. We call *guarded composition* the temporal formula noted $\mathrm{EF}(f_1|f_2)$ (where f_1, f_2 are temporal formulas) and whose semantics is defined as follows:

$$\begin{array}{lll} P \models^t \mathrm{EF}(f_1|f_2) & \text{iff} & \text{there exists a path } p \text{ such that } p_1 = P \text{ and} \\ & & \text{there exist } k, P_1, P_2, y_1, \ldots, y_n \text{ such that } p_k \equiv \nu y_{1,\ldots,n}.(P_1|P_2) \text{ with} \\ & & \nu y_{1,\ldots,n}.P_1 \models^t f_1 \text{ and } \nu y_{1,\ldots,n}.P_2 \models^t f_2 \end{array}$$

For example, we have $d.((c.\overline{e}|\overline{c})|(c.\overline{e}|\overline{c}))|\overline{d} \models^t \mathrm{EF}(\mathrm{AF}\ \overline{e}.\top|\mathrm{AF}\ \overline{e}.\top)$.

We now introduce the *TSLmix* logic, a negation-free fragment of spatial logic extended with the guarded composition formula.

Definition 7 (Mixed Formulas). The set of *mixed formulas* is given by the following grammar:

$$\phi \quad ::= \quad \top \mid \phi_1 \vee \phi_2 \mid c.\phi \mid \overline{c}.\phi \mid (\phi_1|\phi_2)$$

$$f \quad ::= \quad \phi \mid f_1 \vee^t f_2 \mid \mathrm{EF}\ f \mid \mathrm{AF}\ f \mid \mathrm{EF}(f_1|f_2)$$

The semantics of the guarded composition is given by Definition 6 and the semantics of other formulas is given by Definitions 3 and 4.

We define the *TSLmix* logic as the set of mixed formulas.

The restriction to a negation-free fragment is necessary for sound partial order reduction. This is illustrated by the following counter-example (we omit the trailing $\top$ in input formulas to facilitate reading):

$$\begin{tikzcd}
\overline{d}.f|d|c|\overline{c}.\overline{d}.h|\overline{d}.g|d \models^t \mathrm{EF}(\neg^t f \wedge^t \mathrm{EF}\ f \wedge^t \neg^t \mathrm{EF}\ h \mid \neg^t g \wedge^t \mathrm{EF}\ g \wedge^t \neg^t \mathrm{EF}\ h) \arrow[d, "c"] \\
\overline{d}.f|d|\overline{d}.h|\overline{d}.g|d \not\models^t \mathrm{EF}(\neg^t f \wedge^t \mathrm{EF}\ f \wedge^t \neg^t \mathrm{EF}\ h \mid \neg^t g \wedge^t \mathrm{EF}\ g \wedge^t \neg^t \mathrm{EF}\ h)
\end{tikzcd}$$

The starting process can be decomposed into the processes $\overline{d}.f|d|c$ and $\overline{c}.\overline{d}.h|\overline{d}.g|d$ that respectively satisfy $\neg^t f \wedge^t \mathrm{EF}\ f \wedge^t \neg^t \mathrm{EF}\ h$ and $\neg^t g \wedge^t \mathrm{EF}\ g \wedge^t \neg^t \mathrm{EF}\ h$. However, there is no way to decompose the ending process into processes that satisfy both parts of the formula.

Note that we can actually extend the *TSLmix* logic with the *TSL* formulas without compromising partial order reduction by requiring the negation-free condition only inside the guarded composition formula. In the following, we focus on mixed formulas for the sake of clarity.

7.2 Partial Order Reduction for the *TSLmix* Logic

In this section, we show that partial order reduction holds for the *TSLmix* logic in the case of linear and invisible communications (the general case for partially confluent communications is left for future work). Let us first recall the definition of linear communications [16].

Definition 8 (Linear Communications). Let c be a name and P be a process. Communications in P along c are *linear* if there is no case such that there exist $P_1, P_2, P_3, y_1, \ldots, y_n$ such that $P \rightarrow^* \nu y_{1,\ldots,n}.(c.P_1|c.P_2|P_3)$ or $P \rightarrow^* \nu y_{1,\ldots,n}.(\overline{c}.P_1|\overline{c}.P_2|P_3)$. (In other words, no race occurs on input/output along c.)

We now state the theorem of partial order reduction for the *TSLmix* logic:

Theorem 2 (Partial Order Reduction for *TSLmix*). Let the communication $P \rightarrow Q$ be linear and invisible w.r.t. the set of names N. Then, for any mixed formula f such that $fn(f) \subseteq N$, we have $P \models^t f \Leftrightarrow Q \models^t f$.

The rest of this section is dedicated to the proof of this theorem. Let us first explain informally the main idea of the proof. The difficulty resides in the case $P \models^t \mathrm{EF}(f_1|f_2) \Rightarrow Q \models^t \mathrm{EF}(f_1|f_2)$ when, along some path p such that $P \models^t \mathrm{EF}(f_1|f_2)$, we run into a satisfactory process p_k before the communication along c is executed. By assumption, p_k has the form $\overline{c}.U|c.T|R$ and, assuming $R \equiv R_1|R_2$, we might have $\overline{c}.U|R_1 \models^t f_1$ and $c.T|R_2 \models^t f_2$. In this situation, we do not have $U|R_1 \models^t f_1$ and $T|R_2 \models^t f_2$ in general (consider for example $\overline{c}.\overline{d}|\overline{d}.\overline{e}|d \models^t \mathrm{AF}\ \overline{e}.\top$). Fortunately, we can show that $U|R_1 \models^t \mathrm{EF}\ f_1$ and $T|R_2 \models^t \mathrm{EF}\ f_2$ to conclude for this case. To deal with this situation, we introduce the following two intermediate lemmas.

Lemma 5. Let c be a name and f be a mixed formula such that $c \notin fn(f)$. For any $U, R, y_1, \ldots, y_n$ such that $\nu y_{1,\ldots,n}.R \not\models^t \mathrm{EF}\ c.\top$ (resp. $\nu y_{1,\ldots,n}.R \not\models^t \mathrm{EF}\ \overline{c}.\top$), we have $\nu y_{1,\ldots,n}.(\overline{c}.U|R) \models^t f \Rightarrow \nu y_{1,\ldots,n}.R \models^t f$ (resp. $\nu y_{1,\ldots,n}.(c.U|R) \models^t f \Rightarrow \nu y_{1,\ldots,n}.R \models^t f$).

Lemma 6. Let f be a mixed formula. For any $P, Q, y_1, \ldots, y_n$, we have $\nu y_{1,\ldots,n}.P \models^t f \Rightarrow \nu y_{1,\ldots,n}.(P|Q) \models^t \mathrm{EF}\ f$.

Observe that Lemma 6 makes the negation-free condition necessary.

Thanks the intermediate lemmas above, we can now prove Theorem 2:

Proof of Theorem 2. By induction on f. Let us assume that $P \xrightarrow{c,S} Q$. We only show the case for the guarded composition.

- Case $f = \mathrm{EF}(f_1|f_2)$:
  - Case $\Leftarrow$: By assumption, there exists a path q such that $Q \models^t \mathrm{EF}(f_1|f_2)$ holds. Let us consider the path p such that $p_1 = P$ and $p_{i+1} = q_i$ for any natural i. By construction, the path p is such that $P \models^t \mathrm{EF}(f_1|f_2)$ holds.
  - Case $\Rightarrow$: By assumption, there exists a path p such that $p_1 = P$ and there exist $k, P_1, P_2, y_1, \ldots, y_{n_k}$ such that $p_k = \nu y_{1,\ldots,n_k}.(P_1|P_2)$ with $\nu y_{1,\ldots,n_k}.P_1 \models^t f_1$ and $\nu y_{1,\ldots,n_k}.P_2 \models^t f_2$. Since the communication along c is linear, there exist $U, T, R, y_1, \ldots, y_{n_1}$ such that $p_1 \equiv \nu y_{1,\ldots,n_1}.(\overline{c}.U|c.T|R)$ and, for any R' such that $R \rightarrow^* R'$, we have $c \notin \mathsf{guards}(R')$. There are two cases regarding the communication along c:
    - ∗ The communication along c is executed before p_k. In this situation, there exists l ($l < k$) such that, for any $i \leq l$, there exist $y_1, \ldots, y_{n_i}$ such that $p_i \equiv \nu y_{1,\ldots,n_i}.(\overline{c}.U|c.T|R_i)$ where R_i is a derivative of R (we assume

$R_1 = R$), and $p_{l+1} \equiv \nu y_{1,\ldots,n_l}.(U|T|R_l)$. Since the communication along c is linear, it is partially confluent, and thus we can construct a path q such that, for any $i \leq l$, $q_i = \nu y_{1,\ldots,n_i}.(U|T|R_i)$ and $q_i = p_{i+1}$ otherwise. The figure below depicts this situation (continuous arrows represent the path p and dashed arrows represent the path q):

$$\begin{tikzcd}
P \arrow[r, "*"] \arrow[d, dotted, "{c,S}"'] & p_l \equiv \nu y_{1,\ldots,n_l}.(\overline{c}.U|c.T|R_l) \arrow[d, "{c,S}"'] & & \\
Q \equiv q_1 \arrow[r, dashed, "*"] & q_l = p_{l+1} \equiv \nu y_{1,\ldots,n_l}.(U|T|R_l) \arrow[r, "*"] \arrow[r, dashed, "*"'] & q_{k-1} = p_k \arrow[r, "*"] \arrow[r, dashed, "*"'] & \cdots
\end{tikzcd}$$

By construction, we have $q_{k-1} = p_k$. Therefore, we have $Q \models^t \mathrm{EF}(f_1|f_2)$.

- ∗ The communication along c is not executed before p_k. In this situation, for any $i \leq k$, there exist $y_1, \ldots, y_{n_i}$ such that $p_i \equiv \nu y_{1,\ldots,n_i}.(\overline{c}.U|c.T|R_i)$ where R_i is a derivative of R (we assume $R_1 = R$). In particular, we have $p_k \equiv \nu y_{1,\ldots,n_k}.(P_1|P_2) \equiv \nu y_{1,\ldots,n_k}.(\overline{c}.U|c.T|R_k)$. There are several cases, depending on the position of the processes $\overline{c}.U$ and $c.T$ in P_1 or P_2:
  - · There exist R_{k_1}, R_{k_2} such that $R_k \equiv R_{k_1}|R_{k_2}$ and $P_1 \equiv \overline{c}.U|c.T|R_{k_1}$ and $P_2 \equiv R_{k_2}$ (the case with P_1 and P_2 exchanged is similar). By the inductive hypothesis, we have $\nu y_{1,\ldots,n_k}.(U|T|R_{k_1}) \models^t f_1$. Let us consider the path q such that, for any $i \leq k$, $q_i = \nu y_{1,\ldots,n_i}.(U|T|R_i)$. The figure below depicts both p and q paths:

    $$\begin{tikzcd}
    P \arrow[r, "*"] \arrow[d, dotted, "{c,S}"'] & p_k \equiv \nu y_{1,\ldots,n_k}.(\overline{c}.U|c.T|R_k) \arrow[r, "*"] \arrow[d, dotted, "{c,S}"'] & \cdots \\
    Q \equiv q_1 \arrow[r, dashed, "*"] & q_k = \nu y_{1,\ldots,n_k}.(U|T|R_k) \arrow[r, dashed, "*"] & \cdots
    \end{tikzcd}$$

    By construction, $q_k = \nu y_{1,\ldots,n_k}.(U|T|R_{k_1}|R_{k_2})$ with $\nu y_{1,\ldots,n_k}.(U|T|R_{k_1}) \models^t f_1$ and $\nu y_{1,\ldots,n_k}.R_{k_2} \equiv P_2 \models^t f_2$. Therefore, the path q is such that $Q \models^t \mathrm{EF}(f_1|f_2)$.
  - · There exist R_{k_1}, R_{k_2} such that $R_k \equiv R_{k_1}|R_{k_2}$ and $\nu y_{1,\ldots,n_k}.P_1 \equiv \nu y_{1,\ldots,n_k}.(\overline{c}.U|R_{k_1}) \models^t f_1$ and $\nu y_{1,\ldots,n_k}.P_2 \equiv \nu y_{1,\ldots,n_k}.(c.T|R_{k_2}) \models^t f_2$ (the case where the roles of P_1 and P_2 are exchanged is similar). Since the communication along c is linear, we can apply Lemma 5 to deduce $\nu y_{1,\ldots,n_k}.R_{k_1} \models^t f_1$ and $\nu y_{1,\ldots,n_k}.R_{k_2} \models^t f_2$. By Lemma 6, we deduce $\nu y_{1,\ldots,n_k}.(U|R_{k_1}) \models^t \mathrm{EF}\ f_1$ and $\nu y_{1,\ldots,n_k}.(T|R_{k_2}) \models^t \mathrm{EF}\ f_2$, and therefore $Q \models^t \mathrm{EF}(f_1|f_2)$.

□

The guarded composition formula is defined on the model of the EF formula. We did not consider the guarded composition formula defined on the model of the AF formula because it would prevent partial order reduction. Indeed, let us consider the temporal formula noted $\mathrm{AF}(f_1|f_2)$ defined as follows: $P \models^t \mathrm{AF}(f_1|f_2)$ iff for any full path p such that $p_1 = P$, there exist $k, P_1, P_2, y_1, \ldots, y_n$ such that $p_k \equiv \nu_{1,\ldots,n}.(P_1|P_2)$ with $\nu_{1,\ldots,n}.P_1 \models^t f_1$ and $\nu_{1,\ldots,n}.P_2 \models^t f_2$. Using this formula, it is possible to discriminate the starting and ending processes of a partially confluent and invisible communication, as illustrated by the following example:

$$\begin{tikzcd}
\overline{d}.\overline{e}|d|c|\overline{c}.\overline{d}|\overline{d}.\overline{e}|d \models^t \mathrm{AF}(\mathrm{AF}\ \overline{e}.\top|\mathrm{AF}\ \overline{e}.\top) \arrow[d, "c"] \\
\overline{d}.\overline{e}|d|\overline{d}|\overline{d}.\overline{e}|d \not\models^t \mathrm{AF}(\mathrm{AF}\ \overline{e}.\top|\mathrm{AF}\ \overline{e}.\top)
\end{tikzcd}$$

The starting process can be readily decomposed into the processes $\overline{d}.\overline{e}|d|c$ and $\overline{c}.\overline{d}|\overline{d}.\overline{e}|d$ that both satisfy AF $\overline{e}.\top$. In contrast, there is a full path from the ending process such that this decomposition is never possible (namely, the path $\overline{d}.\overline{e}|d|\overline{d}|\overline{d}.\overline{e}|d \rightarrow \overline{d}.\overline{e}|d|\overline{d}.\overline{e} \rightarrow \overline{e}|\overline{d}.\overline{e}$).

8 Extension: The Guarantee Formula

The guarantee formula is an important formula of spatial logics because it allows for contextual specifications. Unfortunately, its standard definition prevents partial order reduction. In this section, we introduce a restricted version of the guarantee formula and we show that we can extend the *TSL* logic with this guarantee formula without compromising partial order reduction.

8.1 Linear Guarantee and the *TSLgua* Logic

The standard definition of the guarantee formula (see Sect. 2) prevents partial order reduction. Indeed, using this formula, we can discriminate the starting and ending processes of partially confluent and invisible communications, as illustrated by the following example:

$$\begin{tikzcd}
c.f.\overline{d}|f.d.\overline{c}.\overline{e}|\overline{f}|\overline{c} \models^t \overline{d}.\top \triangleright \mathrm{EF}\ \overline{e}.\top \arrow[d, "c"] \\
f.\overline{d}|f.d.\overline{c}.\overline{e}|\overline{f} \not\models^t \overline{d}.\top \triangleright \mathrm{EF}\ \overline{e}.\top
\end{tikzcd}$$

When composed with, say, the process $\overline{d}$, the starting process satisfies EF $\overline{e}.\top$, whereas this is not true of the ending process. Intuitively, the problem is that the formula on the left-hand side of the guarantee formula characterizes processes that break the partial confluence property. In other words, the communication along c is partially confluent in the process $c.f.\overline{d}|f.d.\overline{c}.\overline{e}|\overline{f}|\overline{c}$ but it is not partially confluent anymore when this process is composed with the process $\overline{d}$.

The counter-example above motivates the definition of a restricted version of the guarantee formula:

Definition 9 (Linear Guarantee). Given a set of names S and a process P, the predicate $\mathsf{lin}(S, P)$ holds when, for any name $c \in S$, the communications along c in P are linear.

The *linear guarantee* formula, noted $f_1 \overset{S}{\triangleright} f_2$ where f_1, f_2 are temporal formulas and S is a set of names, is defined as follows:

$$P \models^t f_1 \overset{S}{\triangleright} f_2 \quad \text{iff} \quad P|R \models^t f_2 \text{ holds for any process } R \text{ such that } R \models^t f_1 \text{ and } \mathsf{lin}(S, P|R)$$

We define the TSLgua_S logic as the *TSL* logic extended with the linear guarantee $\overset{S}{\triangleright}$.

8.2 Partial Order Reduction for the *TSLgua* Logic

In this section, we show that partial order reduction holds for the *TSLgua* logic in the case of linear and invisible communications (the general case for partially confluent communications is left for future work). We first state the corresponding theorem, then illustrate its use with an example, and finally give an excerpt of its proof.

Theorem 3 (Partial Order Reduction for *TSLgua***).** Let the communication $P \xrightarrow{c} Q$ be linear and invisible w.r.t. the set of names N. Then, for any $f \in \mathit{TSLgua}_S$ such that $c \in S$ and $\mathit{fn}(f) \subseteq N$, we have $P \models^t f \Leftrightarrow Q \models^t f$.

We show how to use this theorem on an example. Let us consider the following process:

$$P = c.f.\overline{d}.\overline{f}.\overline{d}|f.d.\overline{e}|\overline{f}|\overline{c}$$

Let us assume that we want to verify that the composition of P with some process that seeks for input along the name d may eventually perform some output along the name e. Let us also assume that the name c is unknown outside P, so that the linearity of communications along c is preserved by composition. Put formally, we want to verify that $P \models^t d.\top \overset{\{c\}}{\triangleright} \mathrm{EF}\ \overline{e}.\top$ holds.

Since the communication along name c is partially confluent and invisible with respect to the set of names $\{d, e\} = \mathit{fn}(d.\top \overset{\{c\}}{\triangleright} \mathrm{EF}\ \overline{e}.\top)$, we can apply Theorem 3 and simplify the verification to $P' \models^t d.\top \overset{\{c\}}{\triangleright} \mathrm{EF}\ \overline{e}.\top$ with:

$$P' = f.\overline{d}.\overline{f}.\overline{d}|f.d.\overline{e}|\overline{f}$$

Proof of Theorem 3. By induction on f. We only show the case for the linear guarantee.

- Case $f = f_1 \overset{S}{\triangleright} f_2$:
  - Case $\Rightarrow$: Let R be some process such that $R \models^t f_1$ and $\mathsf{lin}(S, Q|R)$ holds. By assumption, no race occurs in $Q|R$ on input/output along names in S. Since $P \xrightarrow{c} Q$ is linear and $c \in S$, it is also true that no race occurs in $P|R$ on input/output along names in S. Since $P \models^t f_1 \overset{S}{\triangleright} f_2$, we have $P|R \models^t f_2$. By the inductive hypothesis, $Q|R \models^t f_2$. Thus, $Q \models^t f_1 \overset{S}{\triangleright} f_2$
  - Case $\Leftarrow$: Let R be some process such that $R \models^t f_1$ and $\mathsf{lin}(S, P|R)$ holds. Since $P \xrightarrow{S} Q$, we have $\mathsf{lin}(S, Q|R)$. Since $Q \models^t f_1 \overset{S}{\triangleright} f_2$, we have $Q|R \models^t f_2$. By the induction hypothesis, $P|R \models^t f_2$. Thus, $P \models^t f_1 \overset{S}{\triangleright} f_2$.

□

9 Conclusion

In this paper, we considered the issue of partial order reduction for the verification of spatial properties. First, we discussed spatial formulas whose standard definition for process calculi prevents partial order reduction. More precisely, we focused on spatial properties of structure and reduction (mainly the composition formula, the temporal modality, and the guarantee formula). Then, we recast the issue of partial order reduction in the terms of process calculi; in particular, we provided a syntactic definition of invisible communications. Finally, we defined three fragments of spatial logics (namely, the *TSL*, *TSLmix*, and *TSLgua* logics) for which we proved that the knowledge of partially confluent (or linear) and invisible communications enable partial order reduction.

Related Work Our work is directly related to partial order reduction techniques for model checking. Partial order reduction techniques have been defined for several temporal logics (LTL-X [9], CTL*-X [10], the weak modal mu-calculus [17]). The main originality of our work lies in the application to spatial formulas (that requires adequate restrictions) and the application to process calculi (that requires an appropriate definition of invisibility).

In the field of process calculi, confluence properties have long been recognized as an important feature regarding verification. In particular, Groote and Sellink study the relation between confluence and τ-inertness (the fact that some communications do not change the bisimulation class of labeled transition systems) and its application to verification of so-called linear processes [11]. Later, Groote and van de Pol generalize their approach to "partial confluence" (i.e., they do not require anymore all communications to be confluent) and study the reduction of labelled transition systems with respect to branching bisimulation [12]. The main difference with our work is that we are concerned with the reduction of state-spaces of processes (instead of abstract transition systems) with respect to spatial logics (instead of branching bisimulation).

The problem of finding partially confluent communications has been addressed several times in the literature on the π-calculus (linear types and linearized types [16], linear receptiveness and ω-receptiveness [18]). Our work can be seen as an application of this work.

Future Work In Sections 7 and 8, the proofs that partial order reduction holds for the *TSLmix* and *TSLgua* logics are limited to linear communications. We plan to investigate the validity of partial order reduction in the general case of partially confluent communications.

In order to augment the expressiveness of the *TSL*, *TSLmix*, and *TSLgua* logics, we plan to extend formulas with fairness conditions. This extension requires refinement of the definition of label and of the definition of partial confluence. These refinements call for special care in the definition of the operational semantics of the underlying process calculi (similarly to developments in [15]).

In this paper, we focused on the spatial properties of structure and reduction (mainly the composition formula, the temporal modality, and the guarantee formula). We plan to investigate the extension of our results to the spatial properties of restriction (mainly, the revelation formula and the fresh quantifier [5, 8]).

Our work was originally motivated by the construction of a library for interactive reasoning on concurrent programs [1] in the Coq proof assistant. At the time being, we use in this library some results from the present paper in the form of axioms. We plan to mechanically prove these axioms for sake of completeness of our library.

References

[1] R. Affeldt and N. Kobayashi. A Coq library for verification of concurrent programs. In *4th International Workshop on Logical Frameworks and Meta-Languages (LFM 2004), Cork, Ireland, July 5, 2004*, pages 66–83, Jul. 2004. Preliminary proceedings. Preliminary proceedings available at `http://cs-www.cs.yale.edu/homes/carsten/lfm04/`. Formal proceedings are to appear in *Electronic Notes in Theoretical Computer Science, Elsevier*. Coq documentation available at: `http://web.yl.is.s.u-tokyo.ac.jp/~affeldt/applpi/`.

[2] R. Affeldt and N. Kobayashi. Partial order reduction for verification of spatial properties of pi-calculus processes. In *11th International Workshop on Expressiveness in*

Concurrency (EXPRESS 2004), London, UK, August 30, 2004, pages 113–127, Aug. 2004. Preliminary proceedings. Formal proceedings are to appear in *Electronic Notes in Theoretical Computer Science, Elsevier*.

[3] R. M. Joachim Parrow David Walker. Modal logics for mobile processes. *Theoretical Computer Science*, 114(1):149–171, Jun. 1993.

[4] L. Caires. Behavioral and spatial observations in a logic for the pi-calculus. In *7th International Conference on Foundations of Software Science and Computation Structures (FOSSACS 2004), Barcelona, Spain, March 29–April 2, 2004*, volume 2987 of *Lecture Notes in Computer Science*. Springer, Mar. 2004.

[5] L. Caires and L. Cardelli. A spatial logic for concurrency (part I). In N. Kobayashi and B. C. Pierce, editors, *Theoretical Aspects of Computer Software (TACS 2001), Sendai, Japan*, number 2215 in Lecture Notes in Computer Science, pages 1–37. Springer, Oct. 2001.

[6] L. Caires and L. Cardelli. A spatial logic for concurrency (part II). In L. Brim, P. Jancar, M. Kretinsky, and A. Kucera, editors, *13th International Conference on Concurrency Theory (CONCUR 2002), Brno, Czech Republic*, number 2421 in Lecture Notes in Computer Science, pages 209–225. Springer, Aug. 2002.

[7] L. Cardelli and A. D. Gordon. Anytime, anywhere: modal logics for mobile ambients. In *27th ACM SIGPLAN-SIGACT symposium on principles of programming languages (POPL 2000), Boston, Massachusetts, USA, January 19–21, 2000*, pages 365–377. ACM Press, 2000.

[8] L. Cardelli and A. D. Gordon. Logical properties of name restriction. In *5th International Conference on Typed Lambda Calculi and Applications (TLCA 2001), Krakow, Poland, May 2–5, 2001*, volume 2044 of *Lecture Notes in Computer Science*, pages 46–60. Springer, May 2001.

[9] E. M. Clarke, O. Grumberg, and D. A. Peled. *Model Checking*. MIT Press, 2000.

[10] R. Gerth, R. Kuiper, D. Peled, and W. Penczek. A partial order approach to branching time logic model checking. *Information and Computation*, 150(2):132–152, 1999.

[11] J. F. Groote and M. Sellink. Confluence for process verification. In *6th International Conference on Concurrency Theory (CONCUR 1995), Philadelphia, PA, USA*, volume 962 of *Lecture Notes in Computer Science*, pages 204–218. Springer, Aug. 1995.

[12] J. F. Groote and J. van de Pol. State space reduction using partial tau-confluence. In *25th International Symposium on Mathematical Foundations of Computer Science (MFCS 2003), Bratislava, Slovak Republic*, volume 1893 of *Lecture Notes in Computer Science*, pages 383–393. Springer, Aug. 2000.

[13] D. Hirschkoff, E. Lozes, and D. Sangiorgi. Minimality results for the spatial logics. In *23rd Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2003), Mumbai, India*, volume 2914 of *Lecture Notes in Computer Science*, pages 252–264. Springer, Dec. 2003.

[14] G. J. Holzmann. *The SPIN Model Checker, Primer and Reference Manual*. Addison Wesley Professional, 2004.

[15] N. Kobayashi. A type system for lock-free processes. *Information and Computation*, 177(2):122–159, Sep. 2002.

[16] N. Kobayashi, B. C. Pierce, and D. N. Turner. Linearity and the Pi-Calculus. In *23rd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL 1996), St. Petersburg Beach, Florida, January 21–24, 1996*, pages 358–371. ACM Press, 1996.

[17] Y. S. Ramakrishna and S. A. Smolka. Partial-order reduction in the weak modal mu-calculus. In *8th International Conference on Concurrency Theory (CONCUR 1997), Warsaw, Poland*, volume 1243 of *Lecture Notes in Computer Science*, pages 5–24. Springer, Jul. 1997.

[18] D. Sangiorgi. The name discipline of uniform receptiveness (extended abstract). In P. Degano, R. Gorrieri, and A. Marchetti-Spaccamela, editors, *24th International Colloquium on Automata, Languages and Programming (ICALP 1997), Bologna, Italy, July 7-11, 1997*, volume 1256 of *Lecture Notes in Computer Science*, pages 303–313. Springer, Jul. 1997.

[19] H. Vieira and L. Caires. Spatial logic model checker user's guide (version 0.9). Technical report, Departamento de Informática, FCT/UNL, Dec. 2003. Revised March 2004.

A Proofs of Intermediate Lemmas

Proof of Lemma 2. By induction on ϕ. We only consider the equivalence $\nu y_{1,\dots,n}.P \models \phi \Leftrightarrow \nu y_{1,\dots,n}.(P|c.Q) \models \phi$ (the equivalence $\nu y_{1,\dots,n}.P \models \phi \Leftrightarrow \nu y_{1,\dots,n}.(P|\overline{c}.Q) \models \phi$ is similar).

- Cases $\phi = \top$, $\phi = \neg\phi'$, $\phi = \phi_1 \vee \phi_2$: Immediate.
- Case $\phi = d.\phi'$ (the case $\phi = \overline{d}.\phi'$ is similar):

  Case $\Rightarrow$: By assumption, there exist $T, R, z_1, \dots, z_m$ such that $\nu y_{1,\dots,n}.P \equiv \nu z_{1,\dots,m}.(d.T|R)$ with $d \notin z_1, \dots, z_m$ and $\nu z_{1,\dots,m}.(T|R) \models \phi'$. Since d is free in $\nu y_{1,\dots,n}.P$, we have $d \notin y_1, \dots, y_n$. By the properties of bound names, there exist $T', R', x_1, \dots, x_l$ such that $d \notin x_1, \dots, x_l$ and $\nu z_{1,\dots,m}.(T|R) \equiv \nu y_{1,\dots,n}.(\nu x_{1,\dots,l}.(T'|R'))$ and $P \equiv \nu x_{1,\dots,l}.(d.T'|R')$. By the inductive hypothesis, we have $\nu y_{1,\dots,n}.(\nu x_{1,\dots,l}.(T'|R')|c.Q) \models \phi'$. Therefore, we have $\nu y_{1,\dots,n}.(\nu x_{1,\dots,l}.(d.T'|R')|c.Q) \equiv \nu y_{1,\dots,n}.(P|c.Q) \models d.\phi'$.

  Case $\Leftarrow$: By assumption, there exist $T, R, z_1, \dots, z_m$ such that $\nu y_{1,\dots,n}.(P|c.Q) \equiv \nu z_{1,\dots,m}.(d.T|R)$ with $d \notin z_1, \dots, z_m$ and $\nu z_{1,\dots,m}.(T|R) \models \phi'$. Since d is free in $\nu y_{1,\dots,n}.(P|c.Q)$, we have $d \notin y_1, \dots, y_n$. Since $c \notin fn(\phi)$, we have $c \neq d$. Thus, by the properties of bound names, there exist $T', R', x_1, \dots, x_l$ such that $d \notin x_1, \dots, x_l$ and $\nu z_{1,\dots,m}.(T|R) \equiv \nu y_{1,\dots,n}.(\nu x_{1,\dots,l}.(T'|R')|c.Q)$ and $P \equiv \nu x_{1,\dots,l}.(d.T'|R')$. By the inductive hypothesis, we have $\nu y_{1,\dots,n}.(\nu x_{1,\dots,l}.(T'|R')) \models \phi'$. Therefore, we have $\nu y_{1,\dots,n}.(\nu x_{1,\dots,l}.(d.T'|R')) \equiv \nu y_{1,\dots,n}.P \models d.\phi'$
- Case $\phi = \phi_1|\phi_2$:

  Case $\Rightarrow$: By assumption, there exist $R_1, R_2, z_1, \dots, z_m$ such that $\nu y_{1,\dots,n}.P \equiv \nu z_{1,\dots,m}.(R_1|R_2)$ with $\nu z_{1,\dots,m}.R_1 \models \phi_1$ and $\nu z_{1,\dots,m}.R_2 \models \phi_2$. By the properties of bound names, there exist R'_1, R'_2 such that $\nu z_{1,\dots,m}.R_2 \equiv \nu y_{1,\dots,n}.R'_2$ and $\nu z_{1,\dots,m}.R_1 \equiv \nu y_{1,\dots,n}.R'_1$. By the inductive hypothesis, $\nu y_{1,\dots,n}.(R'_2|c.Q) \models \phi_2$. Thus, $\nu y_{1,\dots,n}.(P|c.Q) \equiv \nu y_{1,\dots,n}.(R'_1|R'_2|c.Q) \models \phi_1|\phi_2$.

  Case $\Leftarrow$: By assumption, there exist $R_1, R_2, z_1, \dots, z_m$ such that $\nu y_{1,\dots,n}.(P|c.Q) \equiv \nu z_{1,\dots,m}.(R_1|R_2)$ with $\nu z_{1,\dots,m}.R_1 \models \phi_1$ and $\nu z_{1,\dots,m}.R_2 \models \phi_2$. There are two cases. Let us consider the case where there exist, by the properties of bound names, R'_1, R'_2

such that $\nu z_{1,\ldots,m}.R_1 \equiv \nu y_{1,\ldots,n}.(R'_1|c.Q)$ and $\nu z_{1,\ldots,m}.R_2 \equiv \nu y_{1,\ldots,n}.R'_2$ (the case where the roles of R_1 and R_2 are exchanged is similar). By the inductive hypothesis, $\nu y_{1,\ldots,n}.R'_1 \models \phi_1$. Therefore, we have $\nu y_{1,\ldots,n}.P \equiv \nu y_{1,\ldots,n}.(R'_1|R'_2) \models \phi_1|\phi_2$.

□

Proof of Lemma 5. By induction on f. We only show the case for the output process (the case for the input process is similar).

- Cases $f = \top$, $f = \phi_1 \vee \phi_2$: Immediate.
- Case $f = \overline{d}.\phi'$ (the case $f = d.\phi'$ is similar): By assumption, there exist $U', R', z_1, \ldots, z_m$ such that $\nu y_{1,\ldots,n}.(\overline{c}.U|R) \equiv \nu z_{1,\ldots,m}.(\overline{d}.U'|R')$ with $d \notin z_1, \ldots, z_m$ and $\nu z_{1,\ldots,m}.(U'|R') \models \phi'$. Since d is free in $\nu y_{1,\ldots,n}.(\overline{c}.U|R)$, we have $d \notin y_1, \ldots, y_n$. Since $c \notin fn(f)$, we have $c \neq d$. Thus, there exist $U'', R'', x_1, \ldots, x_l$ such that $d \notin x_1, \ldots, x_l$ and $\nu z_{1,\ldots,m}.(U'|R') \equiv \nu y_{1,\ldots,n}.(\nu x_{1,\ldots,l}.(U''|R'')|\overline{c}.U)$ and $R \equiv \nu x_{1,\ldots,l}.(\overline{d}.U''|R'')$. By the inductive hypothesis, $\nu y_{1,\ldots,n}.(\nu x_{1,\ldots,l}.(U''|R'')) \models \phi'$. Therefore, we have $\nu y_{1,\ldots,n}.R \equiv \nu y_{1,\ldots,n}.(\nu x_{1,\ldots,l}.(\overline{d}.U''|R'')) \models \overline{d}.\phi'$.
- Case $f = f_1|f_2$: By assumption, there exist $P_1, P_2, z_1, \ldots, z_m$ such that $\nu y_{1,\ldots,n}.(\overline{c}.U|R) \equiv \nu z_{1,\ldots,m}.(P_1|P_2)$ with $\nu z_{1,\ldots,m}.P_1 \models \phi_1$ and $\nu z_{1,\ldots,m}.P_2 \models \phi_2$. There are two cases. There exist R_1, R_2 such that $R \equiv R_1|R_2$ and $\nu z_{1,\ldots,m}.P_1 \equiv \nu y_{1,\ldots,n}.(\overline{c}.U|R_1)$ (the other case where $\nu z_{1,\ldots,m}.P_2 \equiv \nu y_{1,\ldots,n}.(\overline{c}.U|R_2)$ is similar). By the inductive hypothesis, $\nu y_{1,\ldots,n}.R_1 \models \phi_1$. Therefore, we have $\nu y_{1,\ldots,n}.R \equiv \nu y_{1,\ldots,n}.(R_1|R_2) \models \phi_1|\phi_2$.
- Case $f = f_1 \vee^t f_2$: Immediate.
- Case $f = \mathrm{EF}(f_1|f_2)$ (the case $f = \mathrm{EF}\ f'$ is similar): By assumption, there exist a path p such that $p_1 = \nu y_{1,\ldots,n}.(\overline{c}.U|R)$ and there exist $k, P_1, P_2, y_1, \ldots, y_{n_k}$ such that $p_k \equiv \nu y_{1,\ldots,n_k}.(P_1|P_2)$ with $\nu y_{1,\ldots,n_k}.P_1 \models^t f_1$ and $\nu y_{1,\ldots,n_k}.P_2 \models^t f_2$ (we assume $n_1 = n$). Since $\nu y_{1,\ldots,n}.R \not\models^t \mathrm{EF}\ c.\top$, for any natural i, there exists R_i such that $p_i \equiv \nu y_{1,\ldots,n_i}.(\overline{c}.U|R_i)$ with R_i a derivative of R (we assume $R = R_1$). There are two cases. There exist R_{k_1}, R_{k_2} such that $R_k \equiv R_{k_1}|R_{k_2}$ and $P_1 \equiv \overline{c}.U|R_{k_1}$ (the other case where $P_2 \equiv \overline{c}.U|R_{k_2}$ is similar). By the inductive hypothesis, we have $\nu y_{1,\ldots,n_k}.R_{k_1} \models^t f_1$. Thus, there exists a path q such that $q_1 = \nu y_{1,\ldots,n}.R$ and $q_k \equiv \nu y_{1,\ldots,n_k}.(R_{k_1}|R_{k_2})$ with $\nu y_{1,\ldots,n_k}.R_{k_1} \models^t f_1$ and $\nu y_{1,\ldots,n_k}.R_{k_2} \models^t f_2$. Therefore, $R \models^t \mathrm{EF}(f_1|f_2)$.
- Case $f = \mathrm{AF}\ f'$: Let us consider some full path p such that $p_1 = \nu y_{1,\ldots,n}.R$ and, for any natural i, let us assume that there exist $R_i, y_1, \ldots, y_{n_i}$ such that $p_i \equiv \nu y_{1,\ldots,n_i}.R_i$ (we assume $R_1 = R$ and $n_1 = n$). Since $\nu y_{1,\ldots,n}.R \not\models \mathrm{EF}\ c.\top$, we can construct a full path q such that $q_i = \nu y_{1,\ldots,n_i}.(\overline{c}.U|R_i)$ for any natural i. By assumption, there exists a natural k such that $q_k = \nu y_{1,\ldots,n_k}.(\overline{c}.U|R_k) \models f'$. By the inductive hypothesis, we also have $p_k = \nu y_{1,\ldots,n_k}.R_k \models f'$. Therefore, we have $\nu y_{1,\ldots,n}.R \models^t \mathrm{AF}\ f'$.

□

Proof of Lemma 6. By induction on f.

- Case $f = \phi$: Given by Lemma 7 (see below).
- Case $f = f_1 \vee^t f_2$: Immediate.

- Case $f = \mathrm{EF}(f_1|f_2)$ (the case $f = \mathrm{EF}\ f'$ is similar): By assumption, there exists a path p such that $p_1 = \nu y_{1,\ldots,n}.P$ and there exist $k, P_1, P_2, y_1, \ldots, y_{n_k}$ such that $p_k \equiv \nu y_{1,\ldots,n_k}.(P_1|P_2)$ with $\nu y_{1,\ldots,n_k}.P_1 \models^t f_1$ and $\nu y_{1,\ldots,n_k}.P_2 \models^t f_2$ (we assume $n_1 = n$). By the inductive hypothesis, $\nu y_{1,\ldots,n_k}.(P_2|Q) \models^t \mathrm{EF}\ f_2$. Thus, $\nu y_{1,\ldots,n_k}.(P_1|P_2|Q) \models^t \mathrm{EF}(f_1|f_2)$. Therefore, we have $\nu y_{1,\ldots,n}.(P|Q) \models^t \mathrm{EF}(f_1|f_2)$.

- Case $f = \mathrm{AF}\ f'$: Let us consider some full path p such that $p_1 = \nu y_{1,\ldots,n}.P$. Let us assume that, for any natural i, there exist $P_i, y_1, \ldots, y_{n_i}$ such that $p_i \equiv \nu y_{1,\ldots,n_i}.P_i$ (we assume $n_1 = n$ and $P_1 = P$). By assumption, there is a natural k such that $p_k \models^t f'$. By the inductive hypothesis, $\nu y_{1,\ldots,n_k}.(P_k|Q) \models^t \mathrm{EF}\ f'$, which implies $\nu y_{1,\ldots,n_k}.(P_k|Q) \models^t \mathrm{EF}\ (\mathrm{AF}\ f')$. Let us consider the path q such that, for any natural i, $q_i \equiv \nu y_{1,\ldots,n_i}.(P_i|Q)$. The path q is such that $\nu y_{1,\ldots,n}.(P|Q) \models^t \mathrm{EF}\ (\mathrm{AF}\ f')$.

□

Lemma 7. Let ϕ be a state formula in *TSLmix*. For any $P, Q, y_1, \ldots, y_n$, we have $\nu y_{1,\ldots,n}.P \models \phi \Rightarrow \nu y_{1,\ldots,n}.(P|Q) \models \phi$.

Proof. By induction on ϕ.

- Cases $\phi = \top$, $\phi = \phi_1 \vee \phi_2$: Immediate.

- Case $\phi = d.\phi'$ (the case $\phi = \overline{d}.\phi'$ is similar): By assumption, there exist $T, R, z_1, \ldots, z_m$ such that $\nu y_{1,\ldots,n}.P \equiv \nu z_{1,\ldots,m}.(d.T|R)$ with $d \notin z_1, \ldots, z_m$ and $\nu z_{1,\ldots,m}.(T|R) \models \phi'$. Since d is free in $\nu y_{1,\ldots,n}.P$, we have $d \notin y_1, \ldots, y_n$. By the properties of bound names, there exist $T', R', x_1, \ldots, x_l$ such that $d \notin x_1, \ldots, x_l$ and $\nu z_{1,\ldots,m}.(T|R) \equiv \nu y_{1,\ldots,n}.(\nu x_{1,\ldots,l}.(T'|R'))$ and $P \equiv \nu x_{1,\ldots,l}.(d.T'|R')$. By the inductive hypothesis, $\nu y_{1,\ldots,n}.(\nu x_{1,\ldots,l}.(T'|R')|Q) \models \phi'$. Therefore, we have $\nu y_{1,\ldots,n}.(P|Q) \equiv \nu y_{1,\ldots,n}.(\nu x_{1,\ldots,l}.(d.T'|R')|Q) \models d.\phi'$.

- Case $\phi = \phi_1|\phi_2$: By assumption, there exist $P_1, P_2, z_1, \ldots, z_m$ such that $\nu y_{1,\ldots,n}.P \equiv \nu z_{1,\ldots,m}.(P_1|P_2)$ with $\nu z_{1,\ldots,m}.P_1 \models \phi_1$ and $\nu z_{1,\ldots,m}.P_2 \models \phi_2$. By the properties of bound names, there exist P_1', P_2' such that $\nu z_{1,\ldots,m}.P_1 \equiv \nu y_{1,\ldots,n}.P_1'$ and $\nu z_{1,\ldots,m}.P_2 \equiv \nu y_{1,\ldots,n}.P_2'$. By the inductive hypothesis, $\nu y_{1,\ldots,n}.(P_2'|Q) \models \phi_2$. Thus, $\nu y_{1,\ldots,n}.(P|Q) \equiv \nu y_{1,\ldots,n}.(P_1'|P_2'|Q) \models \phi_1|\phi_2$.

□