An Observational Model for Spatial Logics ⋆

Emilio Tuosto

Dipartimento di Informatica, Università di Pisa, `etuosto@di.unipi.it`

Hugo Torres Vieira

DI FCT, Universidade Nova de Lisboa, `htv@di.fct.unl.pt`

Abstract

Spatiality is an important aspect of distributed systems because their computations depend both on the dynamic behaviour *and* on the structure of their components. *Spatial logics* have been proposed as the formal device for expressing spatial properties of systems.

We define $\mathrm{CCS}_{||}$, a CCS-like calculus whose semantics allows one to observe spatial aspects of systems on the top of which we define models of the spatial logic. Our alternative definition of models is proved equivalent to the standard one. Furthermore, logical equivalence is characterized in terms of the bisimilarity of $\mathrm{CCS}_{||}$.

1 Introduction

In the last years there has been an increasing interest of many researchers in the investigation of the so-called *spatial properties* of systems, namely those properties that are tightly related to the structure of systems instead of on their dynamic behaviour [8,11,13]. A number of works has also been devoted to the exploration of spatial properties of data structures [9,10] and decidability properties of ad-hoc logics [18,14]. Spatiality is an important aspect of distributed systems, in fact, it is more and more evident that distributed computations depend both on the dynamic behaviour of their components *and* on the structure of the system that such components determine. This is even more central in wide area networks where the topology of systems might spread worldwide. For instance, a distributed file sharing application takes into account both the distribution of information and the bandwidth of the links used for downloading.

⋆ This work was funded by the EU project FET IST-2001-33100 Profundis.

Recently, *spatial logics* [6,8,7] have been proposed as the formal device for expressing spatial properties of systems. A typical formula of spatial logics is

$$A|B \tag{1}$$

which expresses the property of a system of being formed of two parts, one satisfying A and the other B.

Spatial logics express properties of distributed systems in a very elegant way since distributed systems hold a number of properties that are spatial in nature. To provide some intuition consider the property of unique handling that expresses that there is only one entity ready to receive messages on a determined channel, hence there is only one *part* of the system that can perform input actions on that channel. Consider also the property of exclusive resource access characterizing systems with resources known only to a single component. In similar terms secrecy can be interpreted as some information whose knowledge is confined to a part of a system. Notice that secrecy induces a notion of *spatial bound* on the processes that share the secret information.

Models of a spatial logic formula A are usually defined as the set of systems that hold the property expressed by A. In general, systems are expressed in a given process calculus equipped with an operational semantics, and the typical way of defining the entailment relation is by exploiting the underlying structural congruence. For instance, the models of formula (1) are those processes $P \equiv Q|R$ such that Q satisfies A and R satisfies B. Following the terminology of [25], this way of defining models can be called *fully intentional* since it relies on structural congruence (different approaches are discussed in Section 1.1).

In this work we consider the logic presented in [5] and interpret its formulae on the top of a simple variant of CCS [19]. A first contribution of this paper is an alternative definition of models which is not based on the structural congruence, but relies on the operational semantics of the calculus. The intuition is that, once a spatial logic and a process calculus with its observational semantics are fixed, we enrich the observations with *spatial observations*. Namely, we extend the semantics of processes so that structural information can be explicitly observed, yielding a *spatial semantics*. As a further result, we define models in terms of the spatial semantics and prove that the new definition is equivalent to the standard one. Spatial semantics naturally induces a bisimilarity, namely *spatial bisimilarity*. A third result is the characterization of the logical equivalence as spatial bisimilarity, indeed we prove that they coincide. Noteworthy, the definition of spatial bisimilarity is *fully extensional*, namely, it does not rely on structural congruence. Finally, we give an alternative characterization both of the logical equivalence and the bisimilarity by showing that they coincide with an *extended* structural congruence.

In this work we do not tackle implementation issues, however, our main motivation is to bring forth results that can be brought up to the development of algorithms, verification techniques and toolkits for checking logical equivalence. It is indeed of interest to develop tools and techniques for the verification of distributed systems against spatial logics specifications. One example of such a tool is the Spatial Logic Model Checker [27] which uses π-calculus to model the systems and the spatial logic of [5] to express the spatial properties. Even though model checking a given system against the formulae is extremely useful, it might also be important to check whether two systems are *logically equivalent*, namely, to check if they satisfy the same set of formulae. Within our approach this may hopefully be done by exploiting the spatial bisimilarity, in other words, checking logical equivalence reduces to checking spatial bisimilarity because they coincide.

We remark that characterizing logical equivalence with a fully extensional bisimilarity also has pragmatic impacts since we can exploit existing toolkits, e.g. [16,26], that rely on such behavioural equivalences. Adapting bisimilarity checking algorithms to logical equivalence checking is left for future work.

1.1 Related Work

Among the possible ways of formally defining concurrent and distributed systems we discuss those that are more closely related to our setting, namely frameworks that explicitly take into account spatial aspects of systems. We do not pretend to be exhaustive and leave out many approaches that do not have a strong connection with our work.

Process calculi and their operational semantics allow the syntax to be considered as a specification of the structure while the semantics as an abstraction from syntactic characteristics that focuses on the dynamic behaviour of the system. In this context there are basically two ways of retrieving information of the structure of a system. One exploits structural congruence in order to check whether a system fits or not some requirements [11,7]. The other dates back to the seminal work of [3] which introduced an observational semantics of CCS [19,20] where spatial information systems were deduced from the observable behaviour of systems.

Another example is the *tile model* [15,4] where observations of structural reconfigurations are neatly separated from those of their behaviour. Indeed, in the tile model a system evolves in two *dimensions*: the "vertical" dimension describes the dynamic evolution while the "horizontal" dimension details how the system reconfigures. However, the structural reconfiguration of a system only takes into account the changing of its interface. Noteworthy, all those approaches define observations that abstractly represent the structure of the system (following [25] they can be called extensional).

A different approach has been recently followed in [23] where concurrent systems are described in co-algebraic terms through the interplay of a pair of functors. The first functor takes care of representing the dynamic behaviour of systems, while the second yields the reconfiguration of their spatial structure. Basically, [23] distinguishes the two different kinds of observables and deals with them separately.

In [25] and [18] a spatial logic for (dialects of) Mobile Ambients [12] have been studied and the related logical equivalence has been contrasted with observational equivalences, e.g., the barbed congruence and the *intensional bisimilarity*. Basically, the intentional bisimulation requires the use of structural congruence in its definition when the observables of the calculus lack sufficient information on the spatial structure of the system. Moreover, [25] discusses the need for stuttering techniques to cope with recursion. Stuttering helps in contrasting intensional bisimilarity and barbed congruence. These approaches can be considered as hybrid with respect to [23] and [3].

Our approach is similar to [23], however, we uniformly handle spatial and behavioural observations, instead of dealing with them separately. Indeed, we expect that a co-algebraic definition of our spatial semantics relies on a co-algebra of a single functor rather than on a co-algebra of a pair of functors. Although the intent of both approaches consists of establishing a transition system that handles behaviour and structural observations, they mainly differ in the way of achieving the goal. Indeed, in our work the two kinds of observations, although somewhat different in nature, are handled in a uniform way (by equipping the transitions with structural inspections). As discussed in Section 6, this uniformity brings some benefits when considering implementation issues.

Despite the technical differences of the calculus and logic adopted, the approach that we followed has many similarities to [25,18], since our purpose is to give an co-inductive characterization of logical equivalence. However, we remark that the definition of spatial bisimilarity introduced here, is fully extensional while the bisimilarity of [25,18] exploits structural congruence. This is possible because we enrich the information carried by the observables which, in our case, suffices for characterizing spatial information of systems. A further remark is that we aim at applying existing verification techniques for checking logical equivalence, while in [25,18] the main motivation is to study expressiveness of equivalences.

Layout of the paper We define the basic calculus in Section 2 while the logic is reported in Section 3. Spatial semantics, *spatial bisimulation* and its properties are introduced in Section 4. *Observational models* of the logic are given in Section 5 together with the proofs that spatial bisimilarity, logical equivalence and extended structural congruence coincide. Final considerations are made in Section 6. We collect the detailed proofs of our results in Appendixes A and B.

2 Process Model

The purpose of this section is to introduce a calculus which provides the basis for defining models of the spatial logic. We introduce the $\mathrm{CCS}_{||}$ calculus, the *anchored CCS*, which basically smoothly extends CCS [19,20]. Different calculi could be used, however we prefer to stick to a simple CCS variant because our first purpose is to give a simple presentation of the main ideas of the paper.

We first give the definition of co-names, actions and processes.

Definition 1 [Co-names, actions and processes] Given an infinite set $\mathcal{N}$ of *names* (ranged over by a, $b \ldots n$, $m \ldots$) the sets $\bar{\mathcal{N}}$ of *co-names* and $\mathcal{A}$ of *actions* are defined by

$$\bar{\mathcal{N}} \triangleq \{\bar{n} \mid n \in \mathcal{N}\} \quad \text{(co-names)}$$
$$\mathcal{A} \triangleq \mathcal{N} \cup \bar{\mathcal{N}} \cup \{\tau\} \text{ (actions).}$$

We let α range over $\mathcal{A}$ and write $na(\alpha)$ be the set of names occurring in α.

The set $\mathcal{P}$ of *processes* is given by

$$P, Q ::= \mathbf{0} \mid \alpha.P \mid P|Q \mid (\boldsymbol{\nu} n)P \mid P||Q.$$

A $\mathrm{CCS}_{||}$ process is either empty, or a process prefixed by an action, or the parallel/anchor composition of two processes, or else a process where a name is restricted.

The syntax of $\mathrm{CCS}_{||}$ is very similar to that of CCS as presented in [20] aside from a few minor differences. First, the choice operator is missing in $\mathrm{CCS}_{||}$, however it can be easily added without any change in what follows. Second, $\mathrm{CCS}_{||}$ lacks recursion or iteration; there is no conceptual difficulty in considering recursive processes, however (differently from the choice operator) this would have made proofs more involved. Finally, the hiding operator is replaced by the restriction (in the spirit of [20]) and the *anchor* operator is introduced. The former is only a syntactic change and allows us to get also rid of the relabelling operator (which is replaced by name substitution), while the latter is necessary in order to model the parallel modality of the logic; this will be made more clear later.

Given a process P, the sets of *free* and *bound* names of P (denoted by $fn(P)$ and $bn(P)$, respectively) are defined as usual for the standard constructs and for the anchor in the following way.

$$fn(P||Q) = fn(P) \cup fn(Q), \quad bn(P||Q) = bn(P) \cup bn(Q).$$

Two processes are indistinguishable when one is obtained by α-renaming

$$\alpha.P \xrightarrow{\alpha} P \qquad (Act)$$

$$\frac{P \xrightarrow{\alpha} P'}{P|Q \xrightarrow{\alpha} P'|Q} \qquad (Par)$$

$$\frac{P \xrightarrow{\bar{a}} P' \quad Q \xrightarrow{a} Q'}{P|Q \xrightarrow{\tau} P'|Q'} \qquad (Comm)$$

$$\frac{P \xrightarrow{\alpha} Q}{(\boldsymbol{\nu} n)P \xrightarrow{\alpha} (\boldsymbol{\nu} n)Q}\, n \notin fn(\alpha) \qquad (Res)$$

$$\frac{P \equiv P' \xrightarrow{\lambda} Q' \equiv Q}{P \xrightarrow{\lambda} Q} \qquad (Cong)$$

Table 1
Behavioural semantics

bound names of the other; we write $P \equiv_\alpha Q$ when P and Q equivalent by α-renaming.

Definition 2 [Structural congruence] The *structural congruence* is denoted by $\equiv$ and is the least congruence (wrt to the operators of $\mathrm{CCS}_{||}$) relation on processes such that it includes the $\equiv_\alpha$, $(P, \mathbf{0}, |)$ is a commutative monoid and the following axioms hold.

$$(\boldsymbol{\nu} n)\mathbf{0} \equiv \mathbf{0} \qquad n \notin fn(P) \Rightarrow P|(\boldsymbol{\nu} n)Q \equiv (\boldsymbol{\nu} n)(P|Q) \qquad (\boldsymbol{\nu} n)(\boldsymbol{\nu} m)P \equiv (\boldsymbol{\nu} m)(\boldsymbol{\nu} n)P.$$

Let us remark that Definition 2 does not involve the anchor operator. As will be more clear later, $||$ is neither commutative, associative nor has a neutral element.

The behavioural semantics of $\mathrm{CCS}_{||}$ is given in terms of a labelled transition system.

Definition 3 [Behavioural semantics of $\mathrm{CCS}_{||}$] The semantics specifying the behaviour of the $\mathrm{CCS}_{||}$ processes is given by the relation $\longrightarrow \subseteq \mathcal{P} \times \mathcal{A} \times \mathcal{P}$ specified by the rules in Table 1.

For the moment, λ ranges over $\mathcal{A}$ (like α); later, we will extend the observables of $\mathrm{CCS}_{||}$ so that the structural congruence rule will have a wider application. Noteworthy, rules in Table 1 do not give any transition out of anchor processes. The reason is that anchor processes only have “structural” transitions and do not expose any “behavioural” observation. It is not possible to observe behaviour in the anchor since it consists in the observation of a spatial bound, somewhat analogous to restriction which also filters the possible behavioural transitions when they involve the restricted name.

3 Logic

In this section we present the syntax and semantics of the logic, very similarly to what can be found in [5]. The differences to note are that no recursive formulae are present here and that the interpretation of the parallel formula is extended to meet the process model as considered here. The syntax and semantics of the logic are presented in Definition 4, being the latter presented through the denotations of the formulae, i.e., the sets of processes that hold the formula ($[\![A]\!] \triangleq \{P \mid P \models A\}$).

Definition 4 [Spatial logic] Formulae of the spatial logic are defined by the following syntax:

$$
\begin{array}{rll}
A ::= & \mathbf{T} & \text{(True)} \\
\mid & \neg A & \text{(Negation)} \\
\mid & A \wedge B & \text{(Conjunction)} \\
\mid & \mathbf{0} & \text{(Void)} \\
\mid & A|B & \text{(Composition)} \\
\mid & <\alpha>.A & \text{(Action)} \\
\mid & n \circledR A & \text{(Revelation)} \\
\mid & \text{И} x.A & \text{(Fresh name)} \\
\mid & \exists x.A & \text{(Existential).}
\end{array}
$$

We let A, B, C range over formulae. The semantics of the logic is reported in Table 2.

Definition 4 uses names ($n, m \in \mathcal{N}$) and name variables ($x, y \in \mathcal{V}$) in formulae. The logical operators considered include propositional, spatial and temporal operators, freshness quantification, and first-order quantification.

Boolean connectives have the standard interpretation, spatial connectives are interpreted on the structure of the processes and temporal connectives are interpreted on the behaviour of processes. More precisely the spatial connectives have the following interpretations: $\mathbf{0}$ is satisfied by the empty process, $A|B$ is satisfied by processes that can be broken down into two components such that one satisfies A and the other satisfies B, either by considering the parallel composition or the anchorage of two processes, and $n \circledR A$ is satisfied by processes that hold a restriction n such that underneath the restriction the process holds A. The temporal operator present in the form of $<\alpha>.A$ is satisfied by processes that hold A after performing an action α. Finally the fresh name quantifier $\text{И} x.A$ is satisfied by processes that, for some name m fresh to both the process and the formula, hold $A\{x \leftarrow m\}$, and the existential quantifier $\exists x.A$ that denotes processes that, for some name m, hold $A\{x \leftarrow m\}$.

$$
\begin{aligned}
[\![\mathbf{T}]\!] &\triangleq \mathcal{P}\\
[\![\neg A]\!] &\triangleq \mathcal{P}\backslash[\![A]\!]\\
[\![A \wedge B]\!] &\triangleq [\![A]\!] \cap [\![B]\!]\\
[\![\mathbf{0}]\!] &\triangleq \{P \mid P \equiv \mathbf{0}\}\\
[\![A|B]\!] &\triangleq \{P \mid \exists Q, R\ .\ P \equiv Q|R \wedge Q \in [\![A]\!] \wedge R \in [\![B]\!]\}\\
&\quad \cup\{P||Q \mid P \in [\![A]\!] \wedge Q \in [\![B]\!]\}\\
[\![<\alpha>.A]\!] &\triangleq \{P \mid \exists Q\ .\ P \xrightarrow{\alpha} Q \wedge Q \in [\![A]\!]\}\\
[\![n®A]\!] &\triangleq \{P \mid \exists Q\ .\ P \equiv (\boldsymbol{\nu} n)Q \wedge Q \in [\![A]\!]\}\\
[\![\text{И}x.A]\!] &\triangleq \textstyle\bigcup_{n \notin fn(A)}([\![A\{x\leftarrow n\}]\!]\backslash\{P \mid n \in fn(P)\})\\
[\![\exists x.A]\!] &\triangleq \textstyle\bigcup_{n \in \mathcal{N}}[\![A\{x\leftarrow n\}]\!]
\end{aligned}
$$

Table 2
Syntax and Semantics of the logic

In both quantifiers the occurrence of x is binding with scope A.

In regard to the logic presented in [5] it is important to note that here the anchor composition of two processes is a model for the parallel modality of the logic, given that the processes are models of the inner formulae. Note that the separation of the processes and the left/right placement is given by the anchor, going along the lines of interpreting the anchor as a fixed spatial bound, where not even structural rearrangement can be considered.

A simple description of the anchor construct can be that it represents a snapshot of the system, in the sense that it captures a determined configuration and does not allow any reconfiguration or observation, either than the projection of the two parts of the system that the anchor is dividing. To further the intuition on the anchor let

$$P_{|} = a.\mathbf{0}|\bar{a}.\mathbf{0} \quad \text{and} \quad P_{||} = a.\mathbf{0}||\bar{a}.\mathbf{0},$$

then $P_{|} \models <\tau>.\mathbf{T}$ whilst $P_{||} \not\models <\tau>.\mathbf{T}$ and also $P_{|} \models <\bar{a}>.\mathbf{T}$ whilst $P_{||} \not\models <\bar{a}>.\mathbf{T}$ since behavior can be observed in a parallel composition but not in the anchor composition of processes. Moreover, we have that $P_{|} \models <\bar{a}>.\mathbf{T}|<a>.\mathbf{T}$ whilst $P_{||} \not\models <\bar{a}>.\mathbf{T}|<a>.\mathbf{T}$, since the parallel composition of processes is commutative and the anchor is not. For an example of an anchor process satisfying a formula note that $P_{||} \models <a>.\mathbf{T}|<\bar{a}>.\mathbf{T}$ since the inner components of the anchor satisfy the corresponding sub-formulae of the parallel modality. Lastly consider $\mathbf{0}||\mathbf{0} \not\models \mathbf{0}$ showing that even the anchor composition of void processes cannot be seen as the void process due to the spatial bound induced by the anchor.

This section is wrapped up by the definition of logical equivalence.

Definition 5 [Logical equivalence] Two processes P and Q are *logical equivalent*, written $P =_L Q$, iff they satisfy exactly the same set of formulae, namely iff, $P \models A \iff Q \models A$ holds for any formula A.

4 Observing the Structure

Our main purpose is to characterize the logical equivalence in terms of bisimilarity. Basically, this amounts to say that the observational semantics of $\mathrm{CCS}_{||}$ must contain some information that allow us to infer (part of) the structure of systems. There are mainly two different ways of obtaining such information. One possibility, in the line of [3,15], is to enrich behavioural observation with labels carrying information on the part of the system that fired a given action.

We adopt a different solution that completely separates behavioural and structural observations. The reason of our choice lies in the highly discriminative power of revelation and parallel modalities. On the one hand, the process $(\boldsymbol{\nu} n)\bar{n}$ is not logically equivalent to $\mathbf{0}$, according to the definition of model given in Section 3:

$$(\boldsymbol{\nu} n)\bar{n} \models n \circledR {<}\bar{n}{>}.\mathbf{0} \qquad \mathbf{0} \not\models n \circledR {<}\bar{n}{>}.\mathbf{0}.$$

On the contrary, a semantics that deduces the structure of the process from its behaviour cannot distinguish the two processes because neither of them expose any transition. Therefore, the bisimulation induced by such semantics cannot be the logical equivalence. On the other hand, interpretation of the parallel modality requires to precisely determine a subsystem *and* the corresponding remaining part. Probably, this can also be determined via an algebraic structure of the spatial/behavioural observations, however the resulting framework is in general particularly complex and involved.

Definition 6 [Spatial observables] The set Λ of *labels* is given by

$$\Lambda \triangleq \mathcal{A} \cup \Sigma_i \text{ where } \Sigma_i \triangleq \{\boldsymbol{\nu} n \mid n \in \mathcal{N}\} \cup \{\mathbf{0}, \phi, \rfloor, \lfloor\}.$$

The elements of Σ_i are the *spatial inspections*. Labels are ranged over by λ.

Definition 6 introduces the observables informing on the spatial structure of systems. Labels can either be the actions of Definition 1 or spatial inspections that yield information on the spatial structure of systems. A label $\boldsymbol{\nu} n$ will be used to observe restricted names, $\mathbf{0}$ says that the system is the process $\mathbf{0}$, ϕ is the observation induced by "freezing" a parallel composition of systems and, dually, $\rfloor$ and $\lfloor$ are the projections of components of systems.

Spatial observables are used to define the *spatial semantics* of $\mathrm{CCS}_{||}$.

$$
\begin{array}{llll}
(\mathit{Void}) & \mathbf{0} \xrightarrow{\mathbf{0}} \mathbf{0} & (\boldsymbol{\nu} n)P \xrightarrow{\boldsymbol{\nu} n} P & (\mathit{Reveal}) \\
(\mathit{Freeze}) & P|Q \xrightarrow{\phi} P||Q & P||Q \xrightarrow{\phi} P||Q & (\mathit{Anchor}) \\
(\mathit{Left}) & P||Q \xrightarrow{\rfloor} P & P||Q \xrightarrow{\lfloor} Q & (\mathit{Right})
\end{array}
$$

Table 3
Spatial semantics

Definition 7 [Spatial semantics of $\mathrm{CCS}_{||}$] The *spatial semantics of* $CCS_{||}$ is obtained by adding the axioms in Table 3 to the inference rules of Table 1.

Let us comment on axioms in Table 3. The first axiom states that the $\mathbf{0}$ process manifests a $\mathbf{0}$ transition. The axiom (*Reveal*) states that a process having n restricted has a transition labelled $\boldsymbol{\nu} n$ to a process where the restriction has been revealed. On the one hand, this is reminiscent of the open rule of the π-calculus where a restricted name can be extruded in a bound output transition. On the other hand, $\mathrm{CCS}_{||}$ does not have any rule that corresponds to the close rule of π-calculus; once a name has been opened, it cannot be closed. The third and fourth axiom state that the parallel or anchor composition of two processes P and Q can "freeze" into the corresponding anchor process $P||Q$, respectively. Since the anchor operator does not obey the monoidal laws, intuitively freezing a system correspond to avoid reconfiguring it via structural congruence and consider it as exactly composed of two parts, the left and right component. Indeed, the last two axioms basically give the semantics of the anchor operator, namely, $P||Q$ can only move to P or Q via a transition that determines the left or the right projection, respectively. Note that it is not possible to observe any transition derived from the inner components of an anchor, hence the anchor acts like a syntactic spatial bound that filters all transitions, except projections and freezing, which implies that a frozen system will only evolve into one of its components.

As said, $P||Q$ can be intuitively thought of as representing a system made of two parallel components that either projects to its first components or to the second. Hence, $||$ somehow represents parallel systems having limited "interleaving capacity". Also at a first glance, one might suppose that the anchor operator can be encoded with the synchronization, left- and right-merge operators introduced in [1,2]. We conjecture that this is not possible because all these operators are intimately related to the parallel operator (i.e., after a transition the arrival process still is a parallel process), while anchor processes evolve into one of their components only. We leave deeper comparisons for future work.

Now, we introduce the observational semantics of $\mathrm{CCS}_{||}$. We define the *spatial bisimilarity* simply by casting the classical definition to the case of behavioural *and* spatial observations. According to this definition, the basic properties of bisimulation relations hold for the spatial bisimulation as well.

Definition 8 [Spatial bisimulation for $\mathrm{CCS}_{||}$] A binary relation $B \subseteq \mathcal{P} \times \mathcal{P}$ is a bisimulation iff, whenever $(P, Q) \in B$ then

$$P \xrightarrow{\lambda} P' \Rightarrow \exists Q \xrightarrow{\lambda} Q'.(P', Q') \in B \tag{2}$$
$$Q \xrightarrow{\lambda} Q' \Rightarrow \exists P \xrightarrow{\lambda} P'.(P', Q') \in B. \tag{3}$$

Definition 8 is the usual bisimulation definition applied to the transition system of the spatial semantics of $\mathrm{CCS}_{||}$. In Appendix A we prove that the usual properties of bisimulations hold for spatial bisimulations (the proofs of these properties mimics those of the usual semantics and bisimulation of CCS [19]).

We call $\bigcup_{i \in I} B_i$ the *spatial bisimilarity* and denote it with $\sim$. By Proposition 3, $\sim$ is a bisimulation and contains any other bisimulation by definition.

Proposition 1 *The spatial bisimilarity is an equivalence relation.*

Proof. We prove that $\sim$ is reflexive, symmetric and transitive.
Reflexivity. The identity relation over processes is, by Proposition 2(2), a bisimulation; hence it is included in $\sim$.
Transitivity. Since, by Proposition 2(3), $\sim\sim$ is a bisimulation, we conclude $\sim\sim \subseteq \sim$ (by definition of $\sim$) which yields the transitivity of spatial bisimilarity.
Symmetry. By Propositions 2(1) and 3, $\sim^{-1}$ is a bisimulation and, by definition $\sim^{-1} \subseteq \sim$. ∎

Remark 9 The anchor operator is quite non-standard. Its operational semantics is vaguely similar to the semantics of the (internal) choice operator, but it does not hold the same algebraic properties: $||$ is neither commutative, associative, idempotent nor has a neutral element; even,

$$P||Q \not\sim Q||P, \quad P||(Q||R) \not\sim (P||Q)||R,$$
$$P||P \not\sim P \qquad\qquad P||\mathbf{0} \not\sim P$$

are not valid laws in $\mathrm{CCS}_{||}$. Moreover, $P||Q$ is *not* deadlocked, since it can evolve once a left or right projection is performed. Basically, $||$ only prevents a system to expose a behavioural transition *before* having committed to one of its components.

5 Characterizing Logical Equivalence

Having enriched the process model it is now possible to define the models of the logic over the transition system given in Definition 7, considering also a

$P \models_o \mathbf{T}$	always
$P \models_o \neg A$	$P \not\models_o A$
$P \models_o A \wedge B$	$P \models_o A \wedge P \models_o B$
$P \models_o \mathbf{0}$	$P \xrightarrow{\mathbf{0}} Q$
$P \models_o A \vert B$	$P \xrightarrow{\phi} P' \wedge P' \xrightarrow{\rfloor} Q \wedge P' \xrightarrow{\lfloor} R \wedge Q \models_o A \wedge R \models_o B$
$P \models_o n \circledR A$	$P \xrightarrow{\boldsymbol{\nu} n} Q \wedge Q \models_o A$
$P \models_o <\omega>.A$	$P \xrightarrow{\omega} Q \wedge Q \models_o A$
$P \models_o \text{И}x.A$	$\exists n \notin \mathit{fn}(A)\ .\ P \xrightarrow{\boldsymbol{\nu} n} Q \wedge P \models_o A\{x \leftarrow n\}$
$P \models_o \exists x.A$	$\exists n \in \mathcal{N}\ .\ P \models_o A\{x \leftarrow n\}$

Table 4
Satisfaction defined through observation

richer set of observations for the temporal operator, since now not only can we observe actions but also spatial inspections. We choose to add freezing and projections to the set of observables of the temporal modality.

Definition 10 [Temporal observables] The set Ω of *temporal observables* is given by

$$\Omega \triangleq \mathcal{A} \cup \{\phi, \rfloor, \lfloor\}.$$

We let ω range over Ω.

The syntax remains the same of that given in Definition 4 with the only difference that the (Action) modality $<\alpha>.A$ is now replaced by $<\omega>.A$. The new definition of the models of the logic is presented in Table 4. Some comments on the clauses of Table 4 follow. Propositional formulae are dealt with in the obvious way while the remaining cases exploit the spatial semantics of $\text{CCS}_{||}$. For instance, in order to satisfy the void formula, a process must expose a transition labelled with $\mathbf{0}$; the spatial semantics (Lemma 3 on page 20) guarantees that such processes are structurally congruent to the void process. A process satisfies a parallel formula $A|B$ when it can be frozen to a process that projects to two components that respectively hold A and B. In order to hold $n \circledR A$, a process must exhibit a $\boldsymbol{\nu} n$ transition that reaches a process holding A (Lemma 4 on page 20 states that if $P \xrightarrow{\boldsymbol{\nu} x} Q$ then $P \equiv (\boldsymbol{\nu} x)Q$) and similarly for the temporal modality; notice, however, that ω ranges over temporal observables, hence it can also be ϕ, $\rfloor$ or $\lfloor$. The formula $\text{И}x.A$ is satisfied by those process that hold $A\{x \leftarrow n\}$ for a name n not occurring neither in the process (hence, the process can reveal n) nor in A. The last case is trivial.

Remark 11 Interesting to note is that the addition of the projection and the freezing to the temporal observables allows for the rewriting of the parallel formula by means of the temporal modality. Indeed, the parallel modality can be written in the following way:

$$A|B \triangleq <\phi>.(<\rfloor>.A \wedge <\lfloor>.B).$$

This suggests that the logic as described in this section is interesting by itself and should not be seen only as an extension of previous models, since one of the most fundamental operators of the logic can be rewritten into simpler constructs.

Remark 12 The Definition 7 allows an automaton to be built out of a given process, being the transitions of the automaton labelled with observables of the spatial semantics. Interestingly, clauses in Table 4 provide a model checking algorithm that verifies spatial logic properties by visiting the automaton. Observe that formulae guide the algorithm in identifying the correct path.

We now show that the semantics of the logic defined on the spatial transitions, given in Table 4, and the usual one presented in Table 2 are equivalent, considering the extension of the temporal observables for both models. The detailed technical proofs of most of the results are reported in Appendix B, here we just give hints on the proofs and comments on the main lemmas and theorems.

Lemma 1 *If $P \models A$ and $P \equiv Q$ then $Q \models A$.*

Proof. By induction on the structure of A.

Theorem 1 $P \models A \iff P \models_o A.$

Since the models are shown to be equivalent we will not distinguish them further in the rest of the paper.

We now prove that logical equivalence and bisimilarity coincide. One practical consequence of this result is that it allows for the usage of minimization techniques on automata that characterize bisimilarity to characterize logical equivalence.

Lemma 2 *If $P \equiv Q$ then $P =_L Q$.*

Proof. Straightforward from Lemma 1 since for any A such that $P \models A$ we have that $Q \models A$ and conversely. ■

Theorem 2 *Two processes are logical equivalent iff they are bisimilar. $P =_L Q \iff P \sim Q$.*

Proof. The full proof is reported in Appendix B. Here we only give the interesting case of revelation.
($\Rightarrow$) We proceed by induction on the derivation of $P \xrightarrow{\lambda} P'$. Assuming

$P \xrightarrow{\nu n} P'$, then $P \models n\circledR\mathbf{T}$ and, since $P =_L Q$ by hypothesis, we conclude that that $Q \models n\circledR\mathbf{T}$ which, by definition, implies $Q \xrightarrow{\nu n} Q'$ and, therefore, $n \notin fn(P) \cup fn(Q)$.
Notice that the inference of the transition $Q \xrightarrow{\nu n} Q'$ subsumes n is chosen as the revealed name. The name n can either occurs free in Q' or not. Since $fn(Q')$ is finite, there is only a finite number of revelations falling into the first case (up to structural congruence). Moreover, if $n \notin fn(Q')$ then $Q \equiv Q'$, hence, the infinite number of such revelations can be identified by structural congruence. So, up to structural congruence, we have finitely many revelation transitions out of Q, say $Q \xrightarrow{\nu n} R_1, \ldots, Q \xrightarrow{\nu n} R_j$.
Let us now assume that $\forall i \in 1, \ldots, n \;.\; R_i \neq_L P'$, which gives us, considering Lemma 8, that $\exists A_1, \ldots, A_j \;.\; R_i \not\models A_i \wedge P' \models A_i$. Since $P' \models A_1 \wedge \ldots \wedge A_j$ we have that $P \models n\circledR(A_1 \wedge \ldots \wedge A_j)$ hence, since $P =_L Q$, we have that $Q \models n\circledR(A_1 \wedge \ldots \wedge A_j)$ which gives us that $Q \xrightarrow{\nu n} \bar{Q} \wedge \bar{Q} \models (A_1 \wedge \ldots \wedge A_j)$ but since $\exists j \in 1, \ldots, n \;.\; \bar{Q} \equiv R_j$ we have that, considering Lemma 1, $\bar{Q} \not\models A_j$ which is a contradiction. So $\exists i \in 1, \ldots, n \;.\; R_i =_L P'$ hence $Q \xrightarrow{\nu n} R_i \wedge R_i =_L P'$.

($\Leftarrow$) We show that for any A such that $P \models A$ then $Q \models A$ by induction on the structure of A (note that the converse is analogous). We consider only the revelation case and refer the interested reader to Appendix B for the detailed proof.
By hypothesis, $P \models n\circledR B$, hence there is $P \xrightarrow{\nu n} P'$ such that $P' \models B$. Since $P \sim Q$ (by hypothesis) we have that $Q \xrightarrow{\nu n} Q'$ and $Q' \sim P'$. By induction hypothesis we get that $Q' \models B$ which along with $Q \xrightarrow{\nu n} Q'$ gives us $Q \models n\circledR B$. That concludes the proof. ■

The case reported in the proof of Theorem 2 suggests how the verification techniques of Mihda discussed in Section 1 can be applied to $\mathrm{CCS}_{||}$. Partition refinement algorithms can deal only with finite automata. Even though we have not considered recursion or iteration, the transition systems processes (upon which the HD-automata are built) are not finite because they are infinitely branching. In fact, the use of structural congruence laws and the (*Cong*) rule allow infinite transitions out of a single process. However, the case reported in the previous proof shows how identifying the set of transitions with respect to structural congruence reports us to finite branching transition systems, moreover, this fact has been exploited also in all the other cases that encompass infinite branching.

Remark 13 Basically, the solution to the infinite branching problem of $\mathrm{CCS}_{||}$ is similar to that for the early semantics of π-calculus. Indeed, the peculiarity of early semantics lies in the input prefix rule $x(z).p \xrightarrow{xy} p\{y \leftarrow z\}$ implying that $x(z).p$ triggers an infinite number of transitions (one for each instantiated name y). In this case, one considers equivalent all those transitions that substitute z with a name "fresh" in p. Hence the "relevant" transitions are only those that either substitute a free name of p or a (single) fresh name. The HD-automata basically deal with the choice of these representative transitions.

In order to further characterize the logical equivalence and bisimilarity that have been described we introduce an extension to the structural congruence relation and prove that it coincides with bisimilarity and logical equivalence.

Definition 14 [Extended structural congruence] Let $\equiv_e$ denote the *extended structural congruence*, defined as the least congruence (wrt to the operators of $\mathrm{CCS}_{||}$) relation on processes generated by the axioms of structural congruence in Definition 2 and the following two axioms

$$P \equiv_e Q \implies P||R \equiv_e Q||R \qquad P \equiv_e Q \implies R||P \equiv_e R||Q$$

Basically, $\equiv_e$ is obtained by extending the usual structural congruence with the axioms expressing the congruence property with respect to anchor contexts.

The following theorem simply states that the extended structural congruence axiomatizes spatial bisimilarity.

Theorem 3 *Two processes are bisimilar iff they are extended structurally congruent.* $P \sim Q \iff P \equiv_e Q$.

The proof of Theorem 3 basically follows as the proof of Theorem 2 and is reported in Appendix B.

6 Concluding Remarks and Future Work

We have presented a spatial semantics based on a extension of a simple process model and a transition system equipped with structural and behavioural observations. Based on this semantics we define observational models of a spatial logic and characterize the logical equivalence by obtaining that the notions both of bisimilarity and of an extended structural congruence considered here coincide with the logical equivalence.

Regarding the process model, for the sake of simplicity, we experiment our ideas within a simple variant of CCS. Considering CCS instead of a name-passing calculus (e.g., π-calculus [21]) can appear a strong limitation. However, our approach can be smoothly adapted to name-passing calculi, therefore, we prefer to maintain the presentation as simple as possible rather than considering a more general setting. It is important to remark that introducing the anchor construct is of relevance, being its fixed configuration a syntactical facility that allows us to elegantly model the parallel modality of the logic. Not including the anchor and having that processes are considered up to structural congruence, it would be harder to observe the two parts that compose a system. Note that we can restrict to processes that initially do not contain the anchor, since the trick lies in the "freezing" of a system into an anchor, hence this approach is suitable for usual process calculi.

The transition system that yields the spatial semantics is plainly defined and tailored to the logic we presented. The notion of spatial bisimilarity presented is not a novelty with respect to intentional bisimilarity [25]. However, spatial bisimilarity is defined straightforwardly from the transition system, where in our case the intentionality lies. Verifying logical equivalence can be achieved by reusing existing tools for checking bisimulation due to the uniform handling of spatial and behavioural observables and the fully extensional definition of bisimilarity. Indeed, such tools check bisimilarities expressed as *ground* relations, namely bisimilarities where transitions are uninterpreted, and expect a single kind of observables. For instance, MIHDA [16,17] implements a partition refinement algorithm for *history dependent* automata [22,24] (HD-automata), which have been proposed as an operational model of history dependent calculi. Once a co-algebraic semantics and a mapping to HD-automata for a given calculus are defined, MIHDA checks the bisimulation by building the minimal realization of the systems and testing for their equality. Our framework can easily fit to MIHDA requirements which we leave as future work.

We expect that our approach naturally fits in the context of name passing calculi. We intend to lift our results to name passing calculi and we think that this will not involve a great deal of difficulties. Another future direction is to study the expressiveness of the logic presented here. More precisely, recalling the comments of Remark 11, the parallel modality can be encoded using freezing and projection. This suggests that our logic contains non-primitive modalities and it might be to use the more primitive as a basic platform for investigating weaker models of the logic.

Acknowledgements

Thanks to Luís Caires and Gianluigi Ferrari for helpful discussions and suggestions. Thanks also to Filippo Bonchi for related discussions.

References

[1] J. Bergstra and J. Klop. Process algebra for synchronous communication. *Information and Control*, 60(1-3):109–137, January/February/March 1984.

[2] J. Bergstra and J. Klop. Algebra of communicating processes with abstraction. *Theoretical Computuper Science*, 37:77–121, 1985.

[3] G. Boudol, I. Castellani, M. Hennessy, and A. Kiehn. Observing localities. *Theoretical Computer Science*, 114(1):31–61, June 1993.

[4] R. Bruni, F. Gadducci, and U. Montanari. Normal forms for algebras of connections. *Theoretical Computer Science*, 286(2):247–292, 2002.

[5] L. Caires. Behavioral and spatial properties in a logic for the pi-calculus. In I. Walukiwicz, editor, *Proc. of Foundations of Software Science and Computation Structures'2004*, Lecture Notes in Computer Science. Springer Verlag, 2004.

[6] L. Caires and L. Cardelli. A spatial logic for concurrency (Part I). In *TACAS*, Lecture Notes in Computer Science. Springer Verlag, 2001.

[7] L. Caires and L. Cardelli. A spatial logic for concurrency (Part II). In *CONCUR'02*, volume 2421 of *Lecture Notes in Computer Science*. Springer Verlag, 2002.

[8] L. Caires and L. Cardelli. A Spatial Logic for Concurrency (Part I). *Information and Computation*, 186(2):194–235, 2003.

[9] C. Calcagno, L. Cardelli, and A. Gordon. Deciding validity in a spatial logic for trees. In *Proceedings of the 2003 ACM SIGPLAN international workshop on Types in languages design and implementation*, pages 62–73. ACM Press, 2003.

[10] L. Cardelli, P. Gardner, and G. Ghelli. Manipulating Trees with Hidden Labels. In A. D. Gordon, editor, *Proceedings of the First International Conference on Foundations of Software Science and Computation Structures (FoSSaCS '03)*, Lecture Notes in Computer Science. Springer-Verlag, 2003.

[11] L. Cardelli and A. Gordon. Anytime, anywhere: Modal logics for mobile ambients. In *Proceedings of the 27th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POLP-00)*, pages 365–377, N.Y., January 2000. ACM Press.

[12] L. Cardelli and A. Gordon. Mobile ambients. *TCS: Theoretical Computer Science*, 240, 2000.

[13] W. Charatonik, S. Dal Zilio, A. D. Gordon, S. Mukhopadhyay, and J.-M. Talbot. The complexity of model checking mobile ambients. In F. Honsell and M. Miculan, editors, *Proceedings of the 4th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS 2001) (ETAPS 2001)*, volume 2030 of *LNCS*, pages 52–167. Springer, 2001.

[14] G. Conforti and G. Ghelli. Decidability of freshness, undecidability of revelation. In I. Walukiewicz, editor, *Proceedings of 7th FOSSACS Conference*, volume 2987 of *Lecture Notes in Computer Science*, pages 105–120, Barcellona (Spain), March/April 2004. Springer-Verlag.

[15] G. Ferrari and U. Montanari. Tile formats for located and mobile systems. *INFCTRL: Information and Computation (formerly Information and Control)*, 156, 2000.

[16] G. Ferrari, U. Montanari, and E. Tuosto. From co-algebraic specifications to implementation: The Mihda toolkit. In F. de Boer, M. Bonsangue, S. Graf, and W. de Roever, editors, *Second International Symposium on Formal Methods for Components and Objects*, volume 2852 of *Lecture Notes in Computer Science*. Springer-Verlag, November 2002.

[17] G. Ferrari, U. Montanari, and E. Tuosto. Coalgebraic minimisation of HD-automata for the π-calculus in a polymorphic λ-calculus. *Theoretical Computer Science*, 2004. To appear.

[18] D. Hirschkoff, E. Lozes, and D. Sangiorgi. Separability, Expressiveness and Decidability in the Ambient Logic. In *Third Annual Symposium on Logic in Computer Science*, Copenhagen, Denmark, 2002. IEEE Computer Society.

[19] R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.

[20] R. Milner. *Communicating and Mobile Systems: the π-calculus*. Cambridge University Press, 1999.

[21] R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes, I and II. *Information and Computation*, 100(1):1–40,41–77, September 1992.

[22] U. Montanari and M. Pistore. π-calculus, structured coalgebras, and minimal HD-automata. In M. Nielsen and B. Roman, editors, *MFCS: Symposium on Mathematical Foundations of Computer Science*, volume 1983 of *LNCS*. Springer Verlag, 2000. An extended version will be published on Theoretical Computer Science.

[23] L. Monteiro. A noninterleaving model of concurrency based on transition. In *Coalgebraic Methods in Computer Science*. Elsevier, 2004. To appear.

[24] M. Pistore. *History dependent automata*. PhD thesis, Computer Science Department, Università di Pisa, 1999.

[25] D. Sangiorgi. Extensionality and Intensionality of the Ambient Logics. In *28th Annual Symposium on Principles of Programming Languages*, pages 4–13. ACM, 2001.

[26] B. Victor and F. Moller. The Mobility Workbench — A Tool for the π-Calculus. In D. Dill, editor, *Proceedings of CAV '94*, volume 818 of *Lecture Notes in Computer Science*, pages 428–440. Springer-Verlag, 1994.

[27] H. Vieira and L. Caires. Spatial logic model checker user's guide. Technical Report TR-DI/FCT/UNL-03/2004, DI/FCT Universidade Nova de Lisboa, 2004.

A Appendix: Properties of Spatial Bisimulations

We prove some properties of spatial bisimulations.

Proposition 2 *Let B_1 and B_2 be two spatial bisimulations.*

- *1 B_1^{-1} is a bisimulation;*
- *2 $Id_{\mathcal{P}} = \{(P, P) \mid P \in \mathcal{P}\}$ is a bisimulation;*
- *3 $B_1 B_2$ is a bisimulation.*

Proof. We prove the three statements.
1. By definition, $(Q, P) \in {B_1}^{-1} \iff (P, Q) \in B_1$ and if $Q \xrightarrow{\lambda} Q'$ then there is a transition $P \xrightarrow{\lambda} P'$ such that $(P', Q') \in B_1$ (by the clause 3 of Definition 8). Then the first clause of the spatial bisimulation definition is satisfied because $(Q', P') \in B_1^{-1}$. We can prove the second clause in a similar way.
2. Trivial.
3. Observe that $(P, Q) \in B_1B_2 \iff$ there is a process R such that $(P, R) \in B_1$ and $(R, Q) \in B_2$. Now, assume that a transition $P \xrightarrow{\lambda} P'$ exists, then we can find two transitions, $R \xrightarrow{\lambda} R'$ and $Q \xrightarrow{\lambda} Q'$ such that $(P', R') \in B_1$ and $(R', Q') \in B_2$ (because B_1 and B_2 are bisimulations); then, by definition $(P', Q') \in B_1B_2$. (The proof that clause 8(3) holds for B_1B_2 is analogous.) ∎

Proposition 3 *Let B_i be a spatial bisimulation for any i in the index set I. Then $\bigcup_{i\in I} B_i$ is a spatial bisimulation.*

Proof. The conclusion easily follows by observing that $(P, Q) \in \bigcup_{i\in I} B_i$ iff there is $\bar{i} \in I$ such that $(P, Q) \in B_{\bar{i}}$ and, also, whenever $P \xrightarrow{\lambda} P'$ then there is $Q \xrightarrow{\lambda} Q'$ such that $(P', Q') \in B_{\bar{i}}$, since $B_{\bar{i}}$ is a spatial bisimulation. Hence, $(P', Q') \in \bigcup_{i\in I} B_i$ by definition. ∎

B Appendix: Logical Equivalence, Bisimilarity and Structural congruence

We first give the proof of Lemma 1 introduced on page 13.

Lemma 1 *If $P \models A$ and $P \equiv Q$ then $Q \models A$.*

Proof. By induction on the structure of A.

(**Case of T**) We immediately have that $Q \models A$.

(**Case of** $\neg A$) We have that $P \models \neg A$ and $P \equiv Q$. Assume $Q \models A$, then by induction hypothesis $P \models A$ which is a contradiction. Hence $Q \models \neg A$.

(**Case of** $A \wedge B$) We have that $P \models A \wedge B$ and $P \equiv Q$. From $P \models A \wedge B$ we have that $P \models A$ and $P \models B$ from which, by induction hypothesis, we obtain $Q \models A$ and $Q \models B$, hence $Q \models A \wedge B$.

(**Case of 0**) We have that $P \models \mathbf{0}$ and $P \equiv Q$. From $P \models \mathbf{0}$ we get that $P \equiv \mathbf{0}$ which along with $P \equiv Q$ gives us that $Q \equiv \mathbf{0}$ hence $Q \models \mathbf{0}$.

(**Case of** $A|B$) We have that $P \models A|B$ and $P \equiv Q$. From $P \models A|B$ we get that $\exists P_1, P_2 \,.\, (P \equiv P_1|P_2 \wedge P_1 \models A \wedge P_2 \models B) \vee (P \equiv P_1||P_2 \wedge P_1 \models A \wedge P_2 \models B)$

and since $P \equiv Q$ we have that $(Q \equiv P_1|P_2 \wedge P_1 \models A \wedge P_2 \models B) \vee (Q \equiv P_1||P_2 \wedge P_1 \models A \wedge P_2 \models B)$ hence $Q \models A|B$.

(**Case of** n®A) We have that $P \models n$®A and $P \equiv Q$. From $P \models n$®A we get that $P \equiv (\boldsymbol{\nu} n)P' \wedge P' \models A$. Since $P \equiv Q$ we have that $Q \equiv (\boldsymbol{\nu} n)P'$ hence $Q \models n$®A.

(**Case of** $<\omega>.A$) We have that $P \models <\omega>.A$ and $P \equiv Q$. From $P \models <\omega>.A$ we get that $P \xrightarrow{\omega} P' \wedge P' \models A$. Since $P \equiv Q$ we have that $Q \xrightarrow{\omega} P'$ (Cong) hence $Q \models <\omega>.A$.

(**Case of** $Ии x.A$) We have that $P \models Ии x.A$ and $P \equiv Q$. From $P \models Ии x.A$ we get that $P \models A\{x \leftarrow m\}$ for some m fresh to the formula and the process. Since $P \equiv Q$ we have that the processes have the same set of free names hence m is also fresh to Q. By induction hypothesis on $P \models A\{x \leftarrow m\}$ we obtain $Q \models A\{x \leftarrow m\}$ hence $Q \models Ии x.A$.

(**Case of** $\exists x.A$) We have that $P \models \exists x.A$ and $P \equiv Q$. From $P \models \exists x.A$ we get that, for some name m, $P \models A\{x \leftarrow m\}$ which by induction hypothesis gives us $Q \models A\{x \leftarrow m\}$, hence $Q \models \exists x.A$. ■

Lemma 3 *If* $P \xrightarrow{\mathbf{0}} Q$ *then* $P \equiv \mathbf{0}$.

Proof. By induction on the derivation of the label. There are only two ways to derive the $\mathbf{0}$ transition which are by the (Void) axiom, and in this case P is $\mathbf{0}$, or by the (Cong) inference rule, which tells us that $P \equiv P'$ and $P' \xrightarrow{\mathbf{0}} P''$, hence by induction hypothesis $P' \equiv \mathbf{0}$ so $P \equiv \mathbf{0}$. ■

Lemma 4 *If* $P \xrightarrow{\boldsymbol{\nu} x} Q$ *then* $P \equiv (\boldsymbol{\nu} x)Q$.

Proof. By induction on the derivation of the label. There are only two ways to derive the $\boldsymbol{\nu} x$ transition which are by the (Reveal) axiom, and in this case P is $(\boldsymbol{\nu} x)Q$, or by the (Cong) inference rule, which tells us that $P \equiv P'$ and $P' \xrightarrow{\boldsymbol{\nu} x} P''$ and $P'' \equiv Q$, hence by induction hypothesis $P' \equiv (\boldsymbol{\nu} x)P''$ so $P \equiv (\boldsymbol{\nu} x)Q$. ■

Lemma 5 *If* $Q \xrightarrow{\phi} Q_1||Q_2$ *then* $Q \equiv Q_1|Q_2 \vee Q \equiv Q_1||Q_2$.

Proof. By induction on the derivation of ϕ.

(**Case of** $Q_1||Q_2 \xrightarrow{\phi} Q_1||Q_2$) Q is $Q_1||Q_2$ hence the result is straightforward.

(**Case of** $Q_1|Q_2 \xrightarrow{\phi} Q_1||Q_2$) Q is $Q_1|Q_2$ hence the result is straightforward.

(**Case of** $Q \xrightarrow{\phi} Q_1||Q_2$) Derived from $Q \equiv Q' \xrightarrow{\phi} Q'' \equiv Q_1||Q_2$. By induction hypothesis we obtain that $Q' \equiv Q_1|Q_2 \vee Q' \equiv Q_1||Q_2$ and since $Q \equiv Q'$

we obtain $Q \equiv Q_1|Q_2 \vee Q \equiv Q_1||Q_2$. ∎

Lemma 6 *If* $P_1||P_2 \xrightarrow{\rfloor} P'$ *then* $P' \equiv P_1$ *and if* $P_1||P_2 \xrightarrow{\lfloor} P'$ *then* $P' \equiv P_2$.

Proof. Since the proofs are analogous let us prove $P_1||P_2 \xrightarrow{\rfloor} P' \implies P' \equiv P_1$ by induction on the derivation of the label. There are only two ways to derive the $\rfloor$ transition which are by the (Left) axiom, and in this case P' is P_1, or by the (Cong) inference rule, which tells us that $P_1||P_2 \equiv Q$, $Q' \equiv P'$ and $Q \xrightarrow{\rfloor} Q'$. By induction hypothesis we obtain $Q' \equiv P_1$ and since $Q' \equiv P'$ we obtain $P' \equiv P_1$.

Lemma 7 *If* $Q \xrightarrow{\rfloor} Q_1$ *and* $Q \xrightarrow{\lfloor} Q_2$ *then* $Q \equiv R_1||R_2 \wedge R_1 \equiv Q_1 \wedge R_2 \equiv Q_2$.

Proof. By inspecting the transition system one finds that only an anchor can project components hence $Q \equiv R_1||R_2$. From $Q \xrightarrow{\rfloor} Q_1$ and $Q \xrightarrow{\lfloor} Q_2$ and $Q \equiv R_1||R_2$ we get that $R_1||R_2 \xrightarrow{\rfloor} Q_1$ and $R_1||R_2 \xrightarrow{\lfloor} Q_2$ (Cong), which by Lemma 6 gives us that $R_1 \equiv Q_1$ and $R_2 \equiv Q_2$. ∎

We can now prove Theorem 1 reported on page 13.

Theorem 1 $P \models A \iff P \models_o A$.

Proof. We prove that for any process P and for all formulas A we have $P \models A$ iff $P \models_o A$. We start by proving that if $P \models A$ then $P \models_o A$ by induction on the structure of A.

(**Case of T**) We immediately have that $P \models_o \mathbf{T}$.

(**Case of** $\neg A$) We have that $P \models \neg A$, which gives us that $P \in \mathcal{P} \backslash [\![A]\!]$. Assume that $P \models_o A$ and prove by contradiction on induction in the structure of A.

(*Case of* $\mathbf{T}$) We immediately have $P \models \mathbf{T}$, hence a contradiction.

(*Case of* $\neg B$) We have that $P \models_o \neg B$ and that $P \models \neg\neg B$. From $P \models \neg\neg B$ we get $P \models B$ hence by induction hypothesis $P \models_o B$ which is a contradiction.

(*Case of* $B \wedge C$) We have that $P \models_o B \wedge C$ and $P \models \neg(B \wedge C)$. From $P \models_o B \wedge C$ we get that $P \models_o B$ and $P \models_o C$ which by induction hypothesis gives us $P \models B$ and $P \models C$, hence $P \models B \wedge C$ which is a contradiction.

(*Case of* $\mathbf{0}$) We have that $P \models_o \mathbf{0}$ and $P \models \neg\mathbf{0}$. From $P \models_o \mathbf{0}$ we get that $P \xrightarrow{\mathbf{0}} P'$ which, considering Lemma 3 gives us that $P \equiv \mathbf{0}$, hence $P \models \mathbf{0}$ which is a contradiction.

($Case\ of\ B|C$) We have that $P \models_o B|C$ and $P \models \neg B|C$. From $P \models_o B|C$ we get that $P \xrightarrow{\phi} P' \wedge P' \xrightarrow{\rfloor} Q \wedge P' \xrightarrow{\lfloor} R \wedge Q \models_o B \wedge R \models_o C$. By $P' \xrightarrow{\rfloor} Q$ and $P' \xrightarrow{\lfloor} R$ and from Lemma 7 we have that $P' \equiv \bar{Q}||\bar{R} \wedge \bar{Q} \equiv Q \wedge \bar{R} \equiv R$. Since $P \equiv P \xrightarrow{\phi} P' \equiv \bar{Q}||\bar{R}$ we have that $P \xrightarrow{\phi} \bar{Q}||\bar{R}$ (Cong). From $P \xrightarrow{\phi} \bar{Q}||\bar{R}$ and Lemma 5 we obtain $P \equiv \bar{Q}|\bar{R} \vee P \equiv \bar{Q}||\bar{R}$. By induction hypothesis on $Q \models_o B$ and $R \models_o C$ we have that $Q \models B$ and $R \models C$ so, considering Lemma 1, we have $\bar{Q} \models B$ and $\bar{R} \models C$ which along with either $P \equiv \bar{Q}|\bar{R}$ or $P \equiv \bar{Q}||\bar{R}$ gives us that $P \models B|C$ which is a contradiction.

($Case\ of\ n®B$) We have that $P \models_o n®B$ and $P \models \neg n®B$. From $P \models_o n®B$ we get that $P \xrightarrow{\boldsymbol{\nu} n} Q \wedge Q \models_o B$. From $P \xrightarrow{\boldsymbol{\nu} n} Q$ and Lemma 4 we obtain $P \equiv (\boldsymbol{\nu} n)Q$. By induction hypothesis on $Q \models_o B$ we get $Q \models B$. From $P \equiv (\boldsymbol{\nu} n)Q$ and $Q \models B$ we obtain $P \models n®B$ which is a contradiction.

($Case\ of\ {<}\omega{>}.B$) We have that $P \models_o {<}\omega{>}.B$ and $P \models \neg{<}\omega{>}.B$. From $P \models_o {<}\omega{>}.B$ we get that $P \xrightarrow{\omega} Q \wedge Q \models_o B$. By induction hypothesis we have that $Q \models B$ which along with $P \xrightarrow{\omega} Q$ gives us $P \models {<}\omega{>}.B$ which is a contradiction.

($Case\ of$ $\text{И}x.B$) We have that $P \models_o \text{И}x.B$ and $P \models \neg\text{И}x.B$. From $P \models_o \text{И}x.B$ we get that $P \xrightarrow{\boldsymbol{\nu} y} Q \wedge P \models_o B\{x\leftarrow y\}$ for some y fresh to the formula. From $P \xrightarrow{\boldsymbol{\nu} y} Q$ and Lemma 4 we obtain $P \equiv (\boldsymbol{\nu} y)Q$ that gives us that y is fresh to the process P. By induction hypothesis on $P \models_o B\{x\leftarrow y\}$ we obtain $P \models B\{x\leftarrow y\}$ which, since y is fresh to the formula and to the process gives us $P \models \text{И}x.B$ which is a contradiction.

($Case\ of\ \exists x.B$) We have that $P \models_o \exists x.B$ and $P \models \neg\exists x.B$. From $P \models_o \exists x.B$ we get that $P \models_o B\{x\leftarrow m\}$ for some name m. By induction hypothesis we obtain that $P \models B\{x\leftarrow m\}$ hence $P \models \exists x.B$ which is a contradiction.

Since we reached a contradiction in every case we have that $P \not\models_o A$ hence $P \models_o \neg A$.

(**Case of** $A \wedge B$) We have that $P \models A \wedge B$, which gives us that $P \models A$ and $P \models B$. By induction hypothesis we get that $P \models_o A$ and $P \models_o B$, hence $P \models_o A \wedge B$.

(**Case of 0**) We have that $P \models \mathbf{0}$, which gives us that $P \equiv \mathbf{0}$. Having $P \equiv \mathbf{0} \xrightarrow{\mathbf{0}} \mathbf{0} \equiv \mathbf{0}$ (Void) we get that $P \xrightarrow{\mathbf{0}} \mathbf{0}$ (Cong), hence $P \models_o \mathbf{0}$.

(**Case of** $A|B$) We have that $P \models A|B$, which gives us that $\exists Q, R \, . \, P \equiv Q|R \wedge Q \in [\![A]\!] \wedge R \in [\![B]\!]$ or $\exists Q, R \, . \, P \equiv Q||R \wedge Q \in [\![A]\!] \wedge R \in [\![B]\!]$.

In the first case we have that $P \equiv Q|R \xrightarrow{\phi} Q||R \equiv Q||R$ (Freeze) which gives us that $P \xrightarrow{\phi} Q||R$ (Cong). Having that $Q||R \xrightarrow{\rfloor} Q$ (Left) and $Q||R \xrightarrow{\lfloor} R$ (Right) and that $Q \models_o A$ and $R \models_o B$, obtained by induction hypothesis on

$Q \in [\![A]\!]$ and $R \in [\![B]\!]$, we get that $P \models_o A|B$.

In the second case we have that $P \equiv Q||R \xrightarrow{\phi} Q||R \equiv Q||R$ (Anchor) which gives us that $P \xrightarrow{\phi} Q||R$ (Cong) being the rest of the proof analogous to the previous case.

(**Case of** $n \circledR A$) We have that $P \models n \circledR A$ which gives us that $P \equiv (\boldsymbol{\nu} n)Q \wedge Q \in [\![A]\!]$. Having $P \equiv (\boldsymbol{\nu} n)Q \xrightarrow{\boldsymbol{\nu} n} Q \equiv Q$ (Reveal) we obtain that $P \xrightarrow{\boldsymbol{\nu} n} Q$ (Cong) and, since $Q \in [\![A]\!]$ by induction hypothesis we obtain that $Q \models_o A$, hence $P \models_o n \circledR A$.

(**Case of** $<\omega>.A$) We have that $P \models <\omega>.A$ which gives us that $\exists Q \,.\, P \xrightarrow{\omega} Q \wedge Q \in [\![A]\!]$. By induction hypothesis we have that $Q \models_o A$ which along with $P \xrightarrow{\omega} Q$ gives us $P \models_o <\omega>.A$.

(**Case of** $\text{И} x.A$) We have that $P \models \text{И} x.A$ which gives us that $P \models A\{x \leftarrow m\}$ for some name m fresh to both the process and the formula. Having that m is fresh to the formula and to the process we get that $P \equiv (\boldsymbol{\nu} m)P$ (obtained from $(\boldsymbol{\nu} n)\mathbf{0} \equiv \mathbf{0}$ and $n \notin fn(P) \Rightarrow P|(\boldsymbol{\nu} n)Q \equiv (\boldsymbol{\nu} n)(P|Q)$) and having that $P \equiv (\boldsymbol{\nu} m)P \xrightarrow{\boldsymbol{\nu} m} P \equiv P$ (Reveal) we get that $P \xrightarrow{\boldsymbol{\nu} m} P$ (Cong). By induction hypothesis on $P \models A\{x \leftarrow m\}$ we obtain $P \models_o A\{x \leftarrow m\}$, hence we have that $P \models_o \text{И} x.A$.

(**Case of** $\exists x.A$) We have that $P \models \exists x.A$ which gives us that $P \models A\{x \leftarrow m\}$ for some name m. By induction hypothesis we obtain that $P \models_o A\{x \leftarrow m\}$ hence $P \models_o \exists x.A$.

Now we prove that if $P \models_o A$ then $P \models A$ also by induction on the structure of A.

(**Case of** $\mathbf{T}$) We immediately have that $P \models \mathbf{T}$.

(**Case of** $\neg A$) We have that $P \models_o \neg A$, which gives us that $P \not\models_o A$. Assume $P \models A$ then, by the previous result, we have that $P \models_o A$ which is a contradiction, hence $P \not\models A$ so $P \models \neg A$.

(**Case of** $A \wedge B$) We have that $P \models_o A \wedge B$, which gives us that $P \models_o A$ and $P \models_o B$. By induction hypothesis we get that $P \models A$ and $P \models B$, hence $P \models A \wedge B$.

(**Case of** $\mathbf{0}$) We have that $P \models_o \mathbf{0}$, which gives us that $P \xrightarrow{\mathbf{0}} Q$. From Lemma 3 we have that $P \equiv \mathbf{0}$ which gives us that $P \models \mathbf{0}$.

(**Case of** $A|B$) We have that $P \models_o A|B$, which gives us that $P \xrightarrow{\phi} P' \wedge P' \xrightarrow{\rfloor} Q \wedge P' \xrightarrow{\lfloor} R \wedge Q \models_o A \wedge R \models_o B$. From $P' \xrightarrow{\rfloor} Q$ and $P' \xrightarrow{\lfloor} R$ and Lemma 7 we have that $P' \equiv \bar{Q}||\bar{R} \wedge \bar{Q} \equiv Q \wedge \bar{R} \equiv R$. Since $P \equiv P \xrightarrow{\phi} P' \equiv \bar{Q}||\bar{R}$ we have that $P \xrightarrow{\phi} \bar{Q}||\bar{R}$ (Cong). From $P \xrightarrow{\phi} \bar{Q}||\bar{R}$ and Lemma 5 we obtain

$P \equiv \bar{Q}|\bar{R} \vee P \equiv \bar{Q}||\bar{R}$. By induction hypothesis on $Q \models_o A$ and $R \models_o B$ we have that $Q \models A$ and $R \models B$ so, considering Lemma 1, we have $\bar{Q} \models B$ and $\bar{R} \models C$ which along with either $P \equiv \bar{Q}|\bar{R}$ or $P \equiv \bar{Q}||\bar{R}$ gives us that $P \models A|B$.

(**Case of** $n®A$) We have that $P \models_o n®A$ which gives us that $P \xrightarrow{\boldsymbol{\nu} n} Q \wedge Q \models_o A$. From $P \xrightarrow{\boldsymbol{\nu} n} Q$ and Lemma 4 we obtain $P \equiv (\boldsymbol{\nu} n)Q$. By induction hypothesis on $Q \models_o A$ we get $Q \models A$. From $P \equiv (\boldsymbol{\nu} n)Q$ and $Q \models A$ we obtain $P \models n®A$.

(**Case of** $<\omega>.A$) We have that $P \models_o <\omega>.A$ which gives us that $P \xrightarrow{\omega} Q \wedge Q \models_o A$. By induction hypothesis we have that $Q \models A$ which along with $P \xrightarrow{\omega} Q$ gives us $P \models <\omega>.A$.

(**Case of** $Иx.A$) We have that $P \models_o Иx.A$ which gives us that $P \xrightarrow{\boldsymbol{\nu} y} Q \wedge P \models_o A\{x \leftarrow y\}$ for some y fresh to the formula. From $P \xrightarrow{\boldsymbol{\nu} y} Q$ and Lemma 4 we obtain $P \equiv (\boldsymbol{\nu} y)Q$ that gives us that y is fresh to the process P. By induction hypothesis on $P \models_o A\{x \leftarrow y\}$ we obtain $P \models A\{x \leftarrow y\}$ which, since y is fresh to the formula and to the process gives us $P \models Иx.A$.

(**Case of** $\exists x.A$) We have that $P \models_o \exists x.A$ which gives us that $P \models_o A\{x \leftarrow m\}$ for some name m. By induction hypothesis we obtain that $P \models A\{x \leftarrow m\}$ hence $P \models \exists x.A$. ■

Lemma 8 *If $P \neq_L Q$ then $\exists A \,.\, P \models A \wedge Q \not\models A$.*

Proof. From $P \neq_L Q$ we have that $\exists B \,.\, (P \models B \wedge Q \not\models B) \vee (P \not\models B \wedge Q \models B)$, hence either $P \models B \wedge Q \not\models B$ or $P \not\models B \wedge Q \models B$ so $P \models B \wedge Q \not\models B$ or $P \models \neg B \wedge Q \not\models \neg B$ hence $\exists A \,.\, P \models A \wedge Q \not\models A$. ■

Lemma 9 *If $Q \xrightarrow{\phi} Q'$ then $Q' \equiv Q_1||Q_2$.*

Proof. By induction on the derivation of ϕ.

(**Case of** $Q_1||Q_2 \xrightarrow{\phi} Q_1||Q_2$) Q' is $Q_1||Q_2$ hence the result is straightforward.

(**Case of** $Q_1|Q_2 \xrightarrow{\phi} Q_1||Q_2$) Q' is $Q_1|Q_2$ hence the result is straightforward.

(**Case of** $Q \xrightarrow{\phi} Q'$) Derived from $Q \equiv P \xrightarrow{\phi} P' \equiv Q'$. By induction hypothesis we obtain that $P' \equiv Q_1||Q_2$ and since $P' \equiv Q'$ we obtain $Q' \equiv Q_1||Q_2$. ■

Lemma 10 *If $P_1||P_2 \xrightarrow{\phi} Q$ then $Q \equiv P_1||P_2$.*

Proof. By induction on the derivation of the label. There are only two ways to derive the ϕ transition for an anchor which are by the (Anchor) axiom, and in this case Q is $P_1||P_2$, or by the (Cong) inference rule, which tells us that $Q' \equiv Q$, $P' \equiv P_1||P_2$ and $P' \xrightarrow{\phi} Q'$. By induction hypothesis we obtain $Q' \equiv P_1||P_2$ and since $Q' \equiv Q$ we obtain $Q \equiv P_1||P_2$. ■

Lemma 11 *If $P_1 =_L Q_1$ and $P_2 =_L Q_2$ then $P_1||Q_1 =_L Q_1||Q_2$.*

Proof. Considering A such that $P_1||P_2 \models A$ we must prove that $Q_1||Q_2 \models A$, being the converse analogous. Proof by induction on the structure of A.

(**Case of T**) We immediately have that $Q_1||Q_2 \models \mathbf{T}$.

(**Case of** $\neg A$) We have that $P1||P_2 \models \neg A$ hence $P_1||P_2 \not\models \neg A$. Assume $Q_1||Q_2 \models A$ then, by induction hypothesis, we get $P_1||P_2 \models A$ which is a contradiction, hence $Q_1||Q_2 \not\models A$ so $Q_1||Q_2 \models \neg A$.

(**Case of** $A \wedge B$) We have that $P_1||P_2 \models A \wedge B$ so $P_1||P_2 \models A$ and $P_1||P_2 \models B$ which by induction hypothesis gives us $Q_1||Q_2 \models A$ and $Q_1||Q_2 \models B$ hence $Q_1||Q_2 \models A \wedge B$.

(**Case of 0**) Impossible since $P_1||P_2$ is never a model of $\mathbf{0}$.

(**Case of** $A|B$) We have that $P_1||P_2 \models A|B$ which gives us that $P_1||P_2 \xrightarrow{\phi} R \wedge R \xrightarrow{\rfloor} R_1 \wedge R \xrightarrow{\lfloor} R_2 \wedge R_1 \models A \wedge R_2 \models B$. Having $P_1||P_2 \xrightarrow{\phi} R$ and by Lemma 10 we get $R \equiv P_1||P_2$ which gives us $P_1||P_2 \xrightarrow{\rfloor} R_1$ and $P_1||P_2 \xrightarrow{\lfloor} R_2$ (Cong). Having $P_1||P_2 \xrightarrow{\rfloor} R_1$ and $P_1||P_2 \xrightarrow{\lfloor} R_2$ and considering Lemma 6 we get $P_1 \equiv R_1$ and $P_2 \equiv R_2$ hence, by Lemma 1, $P_1 \models A$ and $P_2 \models B$. Since $P_1 =_L Q_1$ and $P_2 =_L Q_2$ we have $Q_1 \models A$ and $Q_2 \models B$, and since $Q_1||Q_2 \xrightarrow{\phi} Q_1||Q_2 \wedge Q_1||Q_2 \xrightarrow{\rfloor} Q_1 \wedge Q_1||Q_2 \xrightarrow{\lfloor} Q_2$ we get that $Q_1||Q_2 \models A|B$.

(**Case of** $n®A$) We have that $P_1||P_2 \models n®A$ which gives us that $P_1||P_2 \xrightarrow{\boldsymbol{\nu} n} R \wedge R \models A$. Since the anchor construct does not support the scope extrusion of restrictions, the only possibility for the derivation of $P_1||P_2 \xrightarrow{\boldsymbol{\nu} n} R$ is $P_1||P_2 \equiv (\boldsymbol{\nu} n)(P_1||P_2)$ (derived from $n \notin fn(S) \implies S \equiv (\boldsymbol{\nu} n)S$) and $(\boldsymbol{\nu} n)(P_1||P_2) \xrightarrow{\boldsymbol{\nu} n} P_1||P_2$ and $P_1||P_2 \equiv R$. Since $P_1||P_2 \equiv R$ and $R \models A$, by Lemma 1, we obtain that $P_1||P_2 \models A$ which by induction hypothesis gives us $Q_1||Q_2 \models A$. Since $n \notin fn(P_1||P_2)$ we get that $n \notin fn(P_1)$ and $n \notin fn(P_2)$ which gives us that $P_1 \models n®\mathbf{T}$ and $P_2 \models n®\mathbf{T}$, and from $P_1 =_L Q_1$ and $P_2 =_L Q_2$, we obtain $Q_1 \models n®\mathbf{T}$ and $Q_2 \models n®\mathbf{T}$, hence $n \notin fn(Q_1||Q_2)$ so $Q_1||Q_2 \equiv (\boldsymbol{\nu} n)(Q_1||Q_2)$ and from $(\boldsymbol{\nu} n)(Q_1||Q_2) \xrightarrow{\boldsymbol{\nu} n} Q_1||Q_2$ we get that $Q_1||Q_2 \xrightarrow{\boldsymbol{\nu} n} Q_1||Q_2$ and since $Q_1||Q_2 \models A$ we have $Q_1||Q_2 \models n®A$.

(**Case of** $<\alpha>.A$) Impossible since $P_1||P_2$ is never a model of $<\alpha>.A$.

(**Case of** $<\phi>.A$) We have that $P_1||P_2 \models <\phi>.A$ that implies $P_1||P_2 \xrightarrow{\phi} R \wedge R \models A$. Having $P_1||P_2 \xrightarrow{\phi} R$ and considering Lemma 10 we get $R \equiv P_1||P_2$

and since $R \models A$ by Lemma 1 we obtain $P_1||P_2 \models A$. By induction hypothesis on $P_1||P_2 \models A$ we get $Q_1||Q_2 \models A$ and since $Q_1||Q_2 \xrightarrow{\phi} Q_1||Q_2$ we have $Q_1||Q_2 \models <\phi>.A$.

(**Case of** $<\rfloor>.A$) We have that $P_1||P_2 \models <\rfloor>.A$ that implies $P_1||P_2 \xrightarrow{\rfloor} R \wedge R \models A$. Having $P_1||P_2 \xrightarrow{\rfloor} R$ and considering Lemma 6 we get $R \equiv P_1$ and since $R \models A$ by Lemma 1 we obtain $P_1 \models A$. Having $P_1 =_L Q_1$ and $P_1 \models A$ we get $Q_1 \models A$ and since $Q_1||Q_2 \xrightarrow{\rfloor} Q_1$ we obtain $Q_1||Q_2 \models <\rfloor>.A$.

(**Case of** $<\lfloor>.A$) Analogous to the previous case.

(**Case of** $\text{И}x.A$) We have that $P_1||P_2 \models \text{И}x.A$ that implies for some name m fresh to the formula we have $P_1||P_2 \xrightarrow{\boldsymbol{\nu} m} R \wedge P_1||P_2 \models A\{x \leftarrow m\}$. From $P_1||P_2 \models A\{x \leftarrow m\}$ by induction hypothesis we get $Q_1||Q_2 \models A\{x \leftarrow m\}$ and from $P_1||P_2 \xrightarrow{\boldsymbol{\nu} m} R$ we obtain that $m \notin fn(P_1||P_2)$. Since $P_1 \models m®\mathbf{T} \wedge P_2 \models m®\mathbf{T}$ and $P_1 =_L Q_1$ and $P_2 =_L Q_2$ we have that $Q_1 \models m®\mathbf{T} \wedge Q_2 \models m®\mathbf{T}$, hence $m \notin fn(Q_1||Q_2)$ which gives us that $Q_1||Q_2 \xrightarrow{\boldsymbol{\nu} m} S$. From $Q_1||Q_2 \xrightarrow{\boldsymbol{\nu} m} S$ and $Q_1||Q_2 \models A\{x \leftarrow m\}$ we have that $Q_1||Q_2 \models \text{И}x.A$.

(**Case of** $\exists x.A$) We have that $P_1||P_2 \models \exists x.A$ which gives us that for some name m we have $P_1||P_2 \models A\{x \leftarrow m\}$ which by induction hypothesis gives us $Q_1||Q_2 \models A\{x \leftarrow m\}$ hence $Q_1||Q_2 \models \exists x.A$. ∎

Now we prove Theorem 2 reported on page 13.

Theorem 2 *Two processes are logical equivalent iff they are bisimilar.* $P =_L Q \iff P \sim Q$.

Proof. We first prove that $P =_L Q \implies P \sim Q$ by means of proving that $B_{=_L} = \{(P, Q) \mid P =_L Q\}$ is a bisimulation and hence is contained, by definition, in $\sim$.

We prove that if $P \xrightarrow{\lambda} P'$ then $Q \xrightarrow{\lambda} Q' \wedge P' B_{=_L} Q'$ by induction on the derivation of the label λ.

(**Case of** α) We have that $P =_L Q$ and $P \xrightarrow{\alpha} P'$. From $P \xrightarrow{\alpha} P'$ we have that $P \models <\alpha>.\mathbf{T}$ hence $Q \models <\alpha>.\mathbf{T}$, since the processes are logically equivalent. We also know that there is only a finite number of transitions up to structural congruence since the processes are finite branching, hence $\exists! Q_1, \ldots, Q_n \,.\, Q \xrightarrow{\alpha} Q_i \wedge 1 \leq i \leq n$, having $i \geq 1$ since $Q \models <\alpha>.\mathbf{T}$, such that $\forall R \,.\, Q \xrightarrow{\alpha} R \implies \exists j \in 1, \ldots, n \,.\, R \equiv Q_j$. Assuming that $\forall i \in 1, \ldots, n \; Q_i \neq_L P'$ then by Lemma 8 $\exists A_1, \ldots, A_n \,.\, (Q_i \not\models A_i \wedge P' \models A_i)$ from which we obtain $P \models <\alpha>.(A_1 \wedge \ldots \wedge A_n)$ which gives us $Q \models <\alpha>.(A_1 \wedge \ldots \wedge A_n)$ hence $\exists \bar{Q} \,.\, Q \xrightarrow{\alpha} \bar{Q} \wedge \bar{Q} \models (A_1 \wedge \ldots \wedge A_n)$. Since $\bar{Q} \equiv Q_i \wedge i \in 1, \ldots, n$, considering Lemma 1, we have that $(\bar{Q} \not\models A_i \wedge P' \models A_i)$ which is a contradiction since $\bar{Q} \models A_i$ hence $\exists i \,.\, Q_i =_L P'$ so $Q \xrightarrow{\alpha} Q_i \wedge Q_i =_L P'$.

(**Case of** $\rfloor$ **and** $\lfloor$) Analogous to the previous one.

(**Case of** $\mathbf{0}$) We have that $P =_L Q$ and $P \xrightarrow{\mathbf{0}} P'$. From $P \xrightarrow{\mathbf{0}} P'$ and Lemma 3 we get that $P \equiv \mathbf{0}$. Having $P \models \mathbf{0}$ and $P =_L Q$ we obtain $Q \models \mathbf{0}$, hence $Q \xrightarrow{\mathbf{0}} Q'$ which by Lemma 3 gives us $Q \equiv \mathbf{0}$, hence $Q \equiv P$. From $Q \equiv P$ and $P \xrightarrow{\mathbf{0}} P'$ we get that $Q \xrightarrow{\mathbf{0}} P'$ (Cong) hence $Q \xrightarrow{\mathbf{0}} P' \wedge P' =_L P'$.

(**Case of** ϕ) We have that $P =_L Q$ and $P \xrightarrow{\phi} P'$. From $P \xrightarrow{\phi} P'$ and Lemma 9 we have that $P' \equiv P_1||P_2$ which gives us that $P \xrightarrow{\phi} P_1||P_2$ (Cong). From $P \xrightarrow{\phi} P_1||P_2$ and Lemma 5 we have that $P \equiv P_1|P_2 \vee P \equiv P_1||P_2$.

If $P \equiv P_1||P_2$ then $P \models <\rfloor>.\mathbf{T}$, so $Q \models <\rfloor>.\mathbf{T}$ hence $Q \equiv Q_1||Q_2$ since only anchor processes can perform projections. Having $Q_1||Q_2 \xrightarrow{\phi} Q_1||Q_2$ (Anchor) we get that $Q \xrightarrow{\phi} Q$ (Cong). Since $P \equiv P_1||P2$ and $P' \equiv P_1||P_2$ we have that $P \equiv P'$ which gives us from Lemma 2 that $P =_L P'$ and since $P =_L Q$ we have that $Q =_L P'$, hence $Q \xrightarrow{\phi} Q \wedge Q =_L P'$.

If $P \equiv P_1|P_2$ from Lemma 2 we obtain $P_1|P_2 =_L P$ and since $P =_L Q$ we have that $P_1|P_2 =_L Q$. Since all processes are finite branching we have that Q has a finite number of relevant decompositions, i.e., a finite number of decompositions up to structural congruence of the components obtained, hence $\exists!(R'_1, R''_1), \ldots, (R'_n, R''_n)\ .\ Q \equiv R'_i|R''_i \wedge \forall \bar{R}', \bar{R}''\ .\ Q \equiv \bar{R}'|\bar{R}'' \implies \exists j \in 1, \ldots, n\ .\ \bar{R}' \equiv R'_j \wedge \bar{R}'' \equiv R''_j$. Let us now assume that $\forall i \in 1, \ldots, n\ .\ R'_i \neq_L P_1 \vee R''_i \neq_L P_2$, which, considering Lemma 8, implies $\exists A_1, \ldots, A_n\ .\ (R'_i \not\models A_i \wedge P_1 \models A_i) \vee (R''_i \not\models A_i \wedge P_2 \models A_i)$. Considering $A_0 \triangleq \mathbf{T}$ and $I_1 \triangleq \{j \mid j \in 0, \ldots, n \wedge P_1 \models A_j\}$ and $I_2 \triangleq \{k \mid k \in 0, \ldots, n \wedge P_2 \models A_k\}$ we have that $P_1|P_2 \models (\bigwedge_{i \in I_1} A_i)|(\bigwedge_{i \in I_2} A_i)$ and since $P_1|P_2 =_L Q$ we obtain $Q \models (\bigwedge_{i \in I_1} A_i)|(\bigwedge_{i \in I_2} A_i)$ so $\exists Q_1, Q_2\ .\ (Q \equiv Q_1|Q_2 \vee Q \equiv Q_1||Q_2) \wedge Q_1 \models (\bigwedge_{i \in I_1} A_i) \wedge Q_2 \models (\bigwedge_{i \in I_2} A_i)$ and since $P_1|P_2 =_L Q$ we have that $Q \not\equiv Q_1||Q_2$, hence $Q \equiv Q_1|Q_2$ which gives us that $\exists j \in 1, \ldots, n\ .\ Q_1 \equiv R'_j \wedge Q_2 \equiv R''_j$ hence, considering Lemma 1, gives us $R'_j \models (\bigwedge_{i \in I_1} A_i) \wedge R''_j \models (\bigwedge_{i \in I_2} A_i)$ which is a contradiction since $(j \in I_1 \implies R'_j \not\models A_j) \wedge (j \in I_2 \implies R''_j \not\models A_j)$, so we have that $\exists Q_1, Q_2\ .\ Q \equiv Q_1|Q_2 \wedge Q_1 =_L P_1 \wedge Q_2 =_L P_2$. Since $Q_1|Q_2 \xrightarrow{\phi} Q_1||Q_2$ (Freeze) and $Q \equiv Q_1|Q_2$ we obtain that $Q \xrightarrow{\phi} Q_1||Q_2$ (Cong). Since $P_1 =_L Q_1$ and $P_2 =_L Q_2$ from Lemma 11 we have that $P_1||P_2 =_L Q_1||Q_2$ and since $P' \equiv P_1||P_2$ from Lemma 2 we get that $P' =_L P_1||P_2$, so $P' =_L Q_1||Q_2$, hence $Q \xrightarrow{\phi} Q_1||Q_2 \wedge Q_1||Q_2 =_L P'$.

(**Case of** $\boldsymbol{\nu}n$) See the proof of Theorem 2 on page 13.

We now prove that if $P \sim Q \implies P =_L Q$ by showing that for any A such that $P \models A$ then $Q \models A$ by induction on the structure of A (note that the the converse is analogous).

(**Case of** $\mathbf{T}$) We immediately have $Q \models \mathbf{T}$.

(**Case of** $\neg A$) We have that $P \sim Q$ and $P \models \neg A$. From $P \models \neg A$ we obtain $P \not\models A$. Let us now assume $Q \models A$, then by induction hypothesis we get that $P \models A$ which is a contradiction hence $Q \not\models A$ so $Q \models \neg A$.

(**Case of** $A \wedge B$) We have that $P \sim Q$ and $P \models A \wedge B$. From $P \models A \wedge B$ we obtain $P \models A$ and $P \models B$ which by induction hypothesis gives us $Q \models A$ and $Q \models B$, hence $Q \models A \wedge B$.

(**Case of** $\mathbf{0}$) We have that $P \sim Q$ and $P \models \mathbf{0}$. From $P \models \mathbf{0}$ we have $P \xrightarrow{\mathbf{0}} P'$, which along with $P \sim Q$, gives us that $Q \xrightarrow{\mathbf{0}} Q'$, hence $Q \models \mathbf{0}$.

(**Case of** $A|B$) We have that $P \sim Q$ and $P \models A|B$. From $P \models A|B$ we get that $P \xrightarrow{\phi} P' \wedge P' \xrightarrow{\rfloor} P_1 \wedge P' \xrightarrow{\lfloor} P_2 \wedge P_1 \models A \wedge P_2 \models B$. From $P \xrightarrow{\phi} P' \wedge P' \xrightarrow{\rfloor} P_1 \wedge P' \xrightarrow{\lfloor} P_2$ and $P \sim Q$ we get that $Q \xrightarrow{\phi} Q' \wedge Q' \xrightarrow{\rfloor} Q_1 \wedge Q' \xrightarrow{\lfloor} Q_2$ and $P_1 \sim Q_1 \wedge P_2 \sim Q_2$ from which by induction hypothesis we get $Q_1 \models A \wedge Q_2 \models B$ hence $Q \models A|B$.

(**Case of** $n \circledR A$) See the proof of Theorem 2 on page 13.

(**Case of** $<\omega>.A$) We have that $P \sim Q$ and $P \models <\omega>.A$. From $P \models <\omega>.A$ we get that $P \xrightarrow{\omega} P' \wedge P' \models A$ which since $P \sim Q$ gives us $Q \xrightarrow{\omega} Q' \wedge P' \sim Q'$. By induction hypothesis we get that $Q' \models A$ which along with $Q \xrightarrow{\omega} Q'$ gives us $Q \models <\omega>.A$.

(**Case of** $\text{И}x.A$) We have that $P \sim Q$ and $P \models \text{И}x.A$. From $P \models \text{И}x.A$ we get that for some name m fresh to the formula $P \xrightarrow{\boldsymbol{\nu} m} P' \wedge P \models A\{x \leftarrow m\}$. Since $P \sim Q$ we have that $Q \xrightarrow{\boldsymbol{\nu} m} Q'$. By induction hypothesis on $P \models A\{x \leftarrow m\}$ we obtain $Q \models A\{x \leftarrow m\}$, hence $Q \models \text{И}x.A$.

(**Case of** $\exists x.A$) We have that $P \sim Q$ and $P \models \exists x.A$. From $P \models \exists x.A$ we get that for some name m we have $P \models A\{x \leftarrow ms\}$, which by induction hypothesis gives us $Q \models A\{x \leftarrow m\}$ hence $Q \models \exists x.A$. ∎

Now we prove Theorem 3 reported on page 15.

Theorem 3 *Two processes are bisimilar iff they are extended structurally congruent.* $P \sim Q \iff P \equiv_e Q$

Proof. We first prove that $P \sim Q \implies P \equiv_e Q$ by induction on the structure of P.

(**Case of** $\mathbf{0}$) We have that P is $\mathbf{0}$ hence $P \xrightarrow{\mathbf{0}} P'$ and since $P \sim Q$ we obtain $Q \xrightarrow{\mathbf{0}} Q'$, which by Lemma 3 gives us $Q \equiv \mathbf{0}$ hence $Q \equiv P$ so $Q \equiv_e P$.

(**Case of** $\alpha.P'$) Since $P \sim Q$ and considering Theorem 2 we have that $P =_L Q$. Since $P \models \neg(\neg\mathbf{0}|\neg\mathbf{0})$ and $P \models \text{И}x.x \circledR x \circledR T$ we have that $Q \models \neg(\neg\mathbf{0}|\neg\mathbf{0})$ and $Q \models \text{И}x.x \circledR x \circledR T$ which gives us that Q has only one component and has no restrictions at top level that occur in the process, hence it has the form of a prefix, and since $P \xrightarrow{\alpha} P'$ we have that $Q \xrightarrow{\alpha} Q'$, which along with the previous statement gives us that $Q \equiv \alpha.Q'$. By induction hypothesis on $P' \sim Q'$ we get that $P' \equiv_e Q'$ hence $P \equiv_e Q$.

(**Case of** $P_1||P_2$) Since $P \sim Q$, $P_1||P_2 \xrightarrow{\rfloor} P_1$ and $P_1||P_2 \xrightarrow{\lfloor} P_2$ we have that $Q \xrightarrow{\rfloor} Q_1$, $Q \xrightarrow{\lfloor} Q_2$, $P_1 \sim Q_1$ and $P_2 \sim Q_2$. By induction hypothesis on $P_1 \sim Q_1$ and $P_2 \sim Q_2$ we get that $P_1 \equiv_e Q_1$ and $P_2 \equiv_e Q_2$. Considering Lemma 7 we have that $Q \equiv R_1||R_2 \wedge R_1 \equiv Q_1 \wedge R_2 \equiv Q_2$, so $R_1 \equiv_e P_1$ and $R_2 \equiv_e P_2$ which gives us $P_1||P_2 \equiv_e R_1||R_2$ hence $P \equiv_e Q$.

(**Case of** $P_1|P_2$) Considering Theorem 2 we have that $P_1|P_2 =_L Q$. Since all processes are finite branching we have that Q has a finite number of relevant decompositions, i.e., a finite number of decompositions up to structural congruence of the components obtained, hence $\exists!(R'_1, R''_1), \ldots, (R'_n, R''_n) \,.\, Q \equiv R'_i|R''_i \wedge \forall \bar{R}', \bar{R}'' \,.\, Q \equiv \bar{R}'|\bar{R}'' \implies \exists j \in 1, \ldots, n \,.\, \bar{R}' \equiv R'_j \wedge \bar{R}'' \equiv R''_j$. Let us now assume that $\forall i \in 1, \ldots, n \,.\, R'_i \neq_L P_1 \vee R''_i \neq_L P_2$, which, considering Lemma 8, gives us that $\exists A_1, \ldots, A_n \,.\, (R'_i \not\models A_i \wedge P_1 \models A_i) \vee (R''_i \not\models A_i \wedge P_2 \models A_i)$. Considering $A_0 \triangleq \mathbf{T}$ and $I_1 \triangleq \{j \mid j \in 0, \ldots, n \wedge P_1 \models A_j\}$ and $I_2 \triangleq \{k \mid k \in 0, \ldots, n \wedge P_2 \models A_k\}$ we have that $P_1|P_2 \models (\bigwedge_{i \in I_1} A_i)|(\bigwedge_{i \in I_2} A_i)$ and since $P_1|P_2 =_L Q$ we obtain $Q \models (\bigwedge_{i \in I_1} A_i)|(\bigwedge_{i \in I_2} A_i)$ so $\exists Q_1, Q_2 \,.\, (Q \equiv Q_1|Q_2 \vee Q \equiv Q_1||Q_2) \wedge Q_1 \models (\bigwedge_{i \in I_1} A_i) \wedge Q_2 \models (\bigwedge_{i \in I_2} A_i)$ and since $P_1|P_2 =_L Q$ we have that $Q \not\equiv Q_1||Q_2$, hence $Q \equiv Q_1|Q_2$ which gives us that $\exists j \in 1, \ldots, n \,.\, Q_1 \equiv R'_j \wedge Q_2 \equiv R''_j$ hence, considering Lemma 1, we have that $R'_j \models (\bigwedge_{i \in I_1} A_i) \wedge R''_j \models (\bigwedge_{i \in I_2} A_i)$ which is a contradiction since $(j \in I_1 \implies R'_j \not\models A_j) \wedge (j \in I_2 \implies R''_j \not\models A_j)$, so we have that $\exists Q_1, Q_2 \,.\, Q \equiv Q_1|Q_2 \wedge Q_1 =_L P_1 \wedge Q_2 =_L P_2$. Considering Theorem 2 we have that $P_1 \sim Q_1$ and $P_2 \sim Q_2$ which by induction hypothesis gives us that $P_1 \equiv_e Q_1$ and $P_2 \equiv_e Q_2$ so $P_1|P_2 \equiv_e Q_1|Q_2$ hence $P \equiv_e Q$.

(**Case of** $(\boldsymbol{\nu} n)P'$) We have that $P \xrightarrow{\boldsymbol{\nu} n} P'$ which gives us $Q \xrightarrow{\boldsymbol{\nu} n} Q'$. From $P \xrightarrow{\boldsymbol{\nu} n} P'$ and $Q \xrightarrow{\boldsymbol{\nu} n} Q'$ we have that n is not a free name of both P and Q. Since all processes are finite we have that there is a finite number of revelations of a fresh name up to structural congruence, since the revelation can either pick up a restriction that occurs as a free name of the process within, and this set of restrictions is finite, or simply reveal a restriction that does not occur and in this case the processes obtained are all structurally congruent to one another, in fact they are structurally congruent to the initial one. So we have that $\exists! R_1, \ldots, R_n \,.\, Q \xrightarrow{\boldsymbol{\nu} n} R_i \wedge \forall \bar{R} \,.\, Q \xrightarrow{\boldsymbol{\nu} n} \bar{R} \implies \exists i \in 1, \ldots, n \,.\, \bar{R} \equiv R_i$. Let us now assume that $\forall i \in 1, \ldots, n \,.\, R_i \neq_L P'$, which gives us, considering Lemma 8, that $\exists A_1, \ldots, A_n \,.\, R_i \not\models A_i \wedge P' \models A_i$. Since $P' \models A_1 \wedge \ldots \wedge A_n$ we have that $P \models n® (A_1 \wedge \ldots \wedge A_n)$ hence, since $P =_L Q$, we have that $Q \models n® (A_1 \wedge \ldots \wedge A_n)$ which gives us that $Q \xrightarrow{\boldsymbol{\nu} n} \bar{Q} \wedge \bar{Q} \models (A_1 \wedge \ldots \wedge A_n)$ but since $\exists j \in 1, \ldots, n \,.\, \bar{Q} \equiv R_j$ we have that, considering Lemma 1, $\bar{Q} \not\models A_j$ which is a contradiction. So $\exists i \in 1, \ldots, n \,.\, R_i =_L P'$. From $R_i =_L P'$ and Theorem 2 we get that $R_i \sim P'$ which by induction hypothesis gives us

$R_i \equiv_e P'$ so $(\boldsymbol{\nu} n)R_i \equiv_e (\boldsymbol{\nu} n)P'$, and from $Q \xrightarrow{\boldsymbol{\nu} n} R_i$ and Lemma 4 we get that $Q \equiv (\boldsymbol{\nu} n)R_i$ hence $P \equiv_e Q$.

Considering Theorem 2 we now prove that $P \equiv_e Q \implies P \sim Q$ by obtaining that $P \equiv_e Q \implies P =_L Q$ proving that if $P \equiv_e Q$ for any A such that $P \models A$ then $Q \models A$ by induction on the structure of A (note that the converse is analogous).

(**Case of T**) We immediately have that $Q \models A$.

(**Case of** $\neg A$) We have that $P \models \neg A$ and $P \equiv_e Q$. Assume $Q \models A$, then by induction hypothesis $P \models A$ which is a contradiction. Hence $Q \models \neg A$.

(**Case of** $A \wedge B$) We have that $P \models A \wedge B$ and $P \equiv_e Q$. From $P \models A \wedge B$ we have that $P \models A$ and $P \models B$ from which, by induction hypothesis, we obtain $Q \models A$ and $Q \models B$, hence $Q \models A \wedge B$.

(**Case of 0**) We have that $P \models \mathbf{0}$ and $P \equiv_e Q$. If $P \equiv Q$, then from $P \models \mathbf{0}$ we get that $P \xrightarrow{\mathbf{0}} P'$ which along with $P \equiv Q$ gives us $Q \xrightarrow{\mathbf{0}} P'$ (Cong) hence $Q \models \mathbf{0}$. If $P \equiv_e Q \wedge P \not\equiv Q$ then $P \not\models \mathbf{0}$, because both P and Q contain the anchor construct, hence it is impossible.

(**Case of** $A|B$) We have that $P \models A|B$ and $P \equiv_e Q$. From $P \models A|B$ we get that $P \xrightarrow{\phi} P' \wedge P' \xrightarrow{\rfloor} P_1 \wedge P' \xrightarrow{\lfloor} P_2 \wedge P_1 \models A \wedge P_2 \models B$. Since $P \models A|B$ we have that P is either an anchor or a parallel composition of processes, and from $P \equiv_e Q$ we get that either $P \equiv P_1||P_2$ and $Q \equiv Q_1||Q_2$, or $P \equiv P_1|P_2$ and $Q \equiv Q_1|Q_2$, having in both cases $P_1 \models A$, $P_2 \models B$, $P_1 \equiv_e Q_1$ and $P_2 \equiv_e Q_2$, hence by induction hypothesis $Q_1 \models A$ and $Q_2 \models B$ which in both cases gives us that $Q \models A|B$.

(**Case of** $n\circledR A$) We have that $P \models n\circledR A$ and $P \equiv_e Q$. From $P \models n\circledR A$ we get that $P \xrightarrow{\boldsymbol{\nu} n} P' \wedge P' \models A$. From $P \xrightarrow{\boldsymbol{\nu} n} P'$ and Lemma 4 we get that $P \equiv (\boldsymbol{\nu} n)P'$ which regarding Definition 14 gives us $Q \equiv (\boldsymbol{\nu} n)Q'$, hence $Q \xrightarrow{\boldsymbol{\nu} n} Q'$, and $P' \equiv_e Q'$, which by induction hypothesis gives us $Q' \models A$, so $Q \models n\circledR A$.

(**Case of** $<\omega>.A$) We have that $P \models <\omega>.A$ and $P \equiv_e Q$. From $P \models <\omega>.A$ we get that $P \xrightarrow{\omega} P' \wedge P' \models A$. Having that the extended structural congruence does not interfere with the observables, since the new axioms only talk about structural rearrangement within anchors, we have that $Q \xrightarrow{\omega} Q'$ and $P' \equiv_e Q'$, which by induction hypothesis gives us that $Q' \models A$ hence $Q \models <\omega>.A$.

(**Case of** $Иx.A$) We have that $P \models Иx.A$ and $P \equiv_e Q$. From $P \models Иx.A$ we get that $P \xrightarrow{\boldsymbol{\nu} m} P' \wedge P \models A\{x \leftarrow m\}$ for some m fresh to the formula. From $P \models A\{x \leftarrow m\}$ by induction hypothesis we obtain $Q \models A\{x \leftarrow m\}$ and since $P \equiv_e Q$ implies that the processes have the same set of free names we have that $Q \xrightarrow{\nu m} Q'$ hence $Q \models \text{И} x.A$.

(**Case of** $\exists x.A$) We have that $P \models \exists x.A$ and $P \equiv_e Q$. From $P \models \exists x.A$ we get that, for some name m, $P \models A\{x \leftarrow m\}$ which by induction hypothesis gives us $Q \models A\{x \leftarrow m\}$, hence $Q \models \exists x.A$. $\blacksquare$