A Spatial Logic for Concurrency
(Part I)

Luís Caires[1] and Luca Cardelli[2]

[1] Departamento de Informática FCT/UNL, Lisboa, Portugal
[2] Microsoft Research, Cambridge, UK

Abstract. We present a logic that can express properties of freshness, secrecy, structure, and behavior of concurrent systems. In addition to standard logical and temporal operators, our logic includes spatial operations corresponding to composition, local name restriction, and a primitive fresh name quantifier. Properties can also be defined by recursion; a central theme of this paper is then the combination of a logical notion of freshness with inductive and coinductive definitions of properties.

1 Introduction

We present a logic for describing the behavior and spatial structure of concurrent systems. Logics for concurrent systems are certainly not new [13, 9], but the intent to describe spatial properties seems to have arisen only recently. The spatial properties that we consider here are essentially of two kinds: whether a system is composed of two or more identifiable subsystems, and whether a system restricts the use of certain resources to certain subsystems. Previous work [5] has considered also whether a system is composed of named locations; in that case, the notion of spatial structure is particularly natural.

The initial motivation for studying these logics was to be able to specify systems that deal with fresh or secret resources such as keys, nonces, channels, and locations. In previous papers [5, 3], we have found that the spatial properties of process composition and of location structures are fairly manageable. Instead, the properties of restriction are much more challenging, and are closely related to the study of logical notions of freshness [11].

The main aim of this paper is to advance the study of restriction started in [6, 3] and to build a closer correspondence with treatments of freshness [11]. For simplicity, we use a basic process calculus (the asynchronous π-calculus) that includes composition and restriction. We omit locations in this paper because they are easy to add along the lines of [5], and are comparatively easier to handle than composition or restriction. It will become clear that our general approach is fairly insensitive to the details of specific process calculi, and is largely insensitive to the "dynamics" (reduction behavior) of specific calculi. Therefore, it can be easily adapted to other calculi, and perhaps even generalized to process frameworks [12].

A formula in our logic describes a property of a particular part of a concurrent system at a particular time; therefore it is modal in space as well as in time.

This dual modality can be captured by standard box and diamond operators, reflecting reachability in space and in time [5, 3]. As a further contribution of this paper, though, we wish to investigate a more general framework akin to the μ-calculus [15], where formulas can be recursive and can subsume box and diamond operators. We can then define new and interesting modalities, such as "under an arbitrary number of restrictions". The most technically challenging part of the paper is then the interaction of recursive formulas, already present in [3, 4], with a logical notion of freshness, already present in [6].

We now give a brief overview of the constructs of the logic, before moving on to the formal treatment. Let $\mathcal{P}$ be the set of (asynchronous π-calculus) processes. A *property* is a set of processes; a subset of $\mathcal{P}$. A closed formula denotes a property, namely, it denotes the collection of processes satisfying that formula.

• The collection of all properties (which is not quite the powerset of $\mathcal{P}$, as we shall discuss) has the structure of a Boolean Algebra under set inclusion, so we naturally get boolean connectives in the logic (we take $\mathtt{F}$, $A \wedge B$ and $A \Rightarrow B$ as primitive).

• The above propositional fragment is extended to predicate logic via a universal quantifier $\forall x.A$. This quantifier has standard properties, but the bound variable x ranges always over the countable set of channel names of the process calculus.

• The collection of all properties has also the structure of a quantale, induced by the parallel composition operator over processes. In the logic, this is reflected by the operators $A|B$ (the tensor, or parallel composition, of two properties), $\mathbf{0}$ (the unit of the tensor, or collection of void processes), and $A \triangleright B$ (the linear implication associated with the tensor). This last operator corresponds to context-system specifications, which are the concurrency theory equivalent of pre/post conditions.

• In much the same way that parallel process composition induces the quantale structure, process restriction induces a pair of operators $n®A$ and $A \oslash n$, called revelation and hiding, that give us a basis for describing restricted processes at the logical level.

• The notion of "fresh name" is introduced by a quantifier $\text{И}x.A$; this means that x denotes a name that is fresh with respect to the names used either in A or in processes satisfying A. This quantifier allows us to then derive a hidden name quantifier [6, 3]. $\text{И}x.A$ is defined along the lines of the freshness quantifier of Gabbay-Pitts [11]: the connections will be discussed. A similar $\text{И}x.A$ quantifier is studied in [6] (in absence of recursion), but is handled as a meta-level construct, and not as a proper formula that can be mixed with recursion.

• A logical operator $n\langle m\rangle$ allows us to assert that a message m is present over channel n, giving us some minimal power to observe the behavior of processes.

• A "next step" temporal operator, $\Diamond A$, allows us to talk about the behavior of a process after a single (unspecified) reduction step.

• Finally, a maximal fixpoint operator $\boldsymbol{\nu}X.A$ (where A is monotonic in X) introduces recursive formulas and, together with the other constructs, has the ability to define standard modal operators for temporal and spatial modalities, for instance $\Box A$ denoting that A holds anytime in the future, and $\boxtimes A$, meaning that A holds everywhere in space. A least fixpoint operator is definable.

The paper is organised as follows. We start with a concise review of the asynchronous π-calculus. In Section 3 we give a detailed presentation of the syntax of the spatial logic. In Section 4, we introduce the central notion of property set, we define satisfaction, and we proceed with the analysis of semantical aspects of the logic. In Section 5 we study an appropriate notion of logical validity. In Sections 6 and 7 we motivate and discuss fresh and hidden name quantification, and the recursive definition of properties.

2 Preliminaries on the asynchronous π-calculus

We review the syntax and operational semantics of the asynchronous π-calculus [1, 14], following the notations of [16]. We base our presentation on the use of transpositions (simple name replacements), which will become prominent later in the paper.

Definition 2.1 (Processes). *Given a countable set Λ of* names*, the set $\mathcal{P}$ of* processes *is given by the following abstract syntax*

$$\begin{array}{lll} m, n, p \in & \Lambda & \text{(Names)} \\ & & \\ P, Q, R ::= & & \\ & \mathbf{0} & \text{(Void)} \\ & P|Q & \text{(Par)} \\ & (\boldsymbol{\nu} n)P & \text{(Restriction)} \\ & m\langle n\rangle & \text{(Message)} \\ & m(n).P & \text{(Input)} \\ & !P & \text{(Replication)} \end{array}$$

We note by $na(P)$ the set of all names that textually occur in a process P (either bound or free).

Definition 2.2 (Free names in Processes). *For any process P, the set of* free names *of P, noted $fn(P)$, is inductively defined as follows.*

$$\begin{array}{l} fn(\mathbf{0}) \triangleq \mathbf{0} \\ fn(P|Q) \triangleq fn(P) \cup fn(Q) \\ fn(m\langle n\rangle) \triangleq \{m, n\} \\ fn((\boldsymbol{\nu} n)P) \triangleq fn(P) \setminus \{n\} \\ fn(m(n).P) \triangleq (fn(P) \setminus \{n\}) \cup \{m\} \\ fn(!P) \triangleq fn(P) \end{array}$$

In restriction $(\boldsymbol{\nu} n)P$ and input $m(n).P$, the distinguished occurrence of the name n is bound, with scope the process P. We will note by $bn(P)$ the set of names which occur bound in the process P, and by ©N the set $\{P \mid N \subseteq fn(P)\}$ of processes that contain all names in N free. If N is a finite set of names, and N'

is a any set of names, a substitution $\theta : N \to N'$ of *domain* N and *codomain* N' is a mapping assigning $\theta(n) \in N'$ to each $n \in N$, and n to each $n \notin N$. Thus, outside its domain, any substitution behaves like the identity. Given a substitution θ, we note by $\mathfrak{D}(\theta)$ its domain. The *image* of a substitution θ, noted $\mathfrak{I}(\theta)$, is the set $\{\theta(n) \mid n \in \mathfrak{D}(\theta)\}$. We will write $\{n\leftarrow m\}$ for the singleton substitution of domain $\{n\}$ that assigns m to n. If N is a set of names, we define $\theta(N) \triangleq \{\theta(n) \mid n \in N\}$.

Substitutions that just interchange a pair of names will play a special role in technical developments to follow. More precisely, the *transposition* of n and m, noted $\{n\leftrightarrow m\}$, denotes the substitution $\tau : \{m, n\} \to \{m, n\}$ such that $\tau(n) = m$ and $\tau(m) = n$. Note that $\{n\leftrightarrow m\} = \{m\leftrightarrow n\}$. Before defining safe substitution on processes, we first introduce transposition, and then define α-congruence in the style of [11].

Definition 2.3 (Transposition). *Given a process P and a transposition τ, we denote by $\tau{\cdot}P$ the process inductively defined as follows.*

$$\begin{array}{llll}
\tau{\cdot}\mathbf{0} & \triangleq \mathbf{0} & \tau{\cdot}((\boldsymbol{\nu} n)P) & \triangleq (\boldsymbol{\nu}\tau(n))\tau{\cdot}P \\
\tau{\cdot}m\langle n\rangle & \triangleq \tau(m)\langle\tau(n)\rangle & \tau{\cdot}(p(n).P) & \triangleq \tau(p)(\tau(n)).(\tau{\cdot}P) \\
\tau{\cdot}(P|Q) & \triangleq (\tau{\cdot}P)|(\tau{\cdot}Q) & \tau{\cdot}(!P) & \triangleq !\tau{\cdot}P
\end{array}$$

Proposition 2.4. *For all processes P and Q, and transpositions τ,*

1. $\tau{\cdot}\tau{\cdot}P = P$
2. $\{m\leftrightarrow n\}{\cdot}\tau{\cdot}P = \tau{\cdot}\{\tau(m)\leftrightarrow\tau(n)\}{\cdot}P$

Definition 2.5 (Congruence). *A binary relation $\cong$ on processes is a* congruence *whenever for all processes P, Q and R,*

$$\begin{array}{ll}
P \cong P & \text{(Cong Refl)} \\
P \cong Q \Rightarrow Q \cong P & \text{(Cong Symm)} \\
P \cong Q,\, Q \cong R \Rightarrow P \cong R & \text{(Cong Trans)} \\
P \cong Q \Rightarrow P|R \cong Q|R & \text{(Cong Parl)} \\
P \cong Q \Rightarrow R|P \cong P|Q & \text{(Cong Parr)} \\
P \cong Q \Rightarrow (\boldsymbol{\nu} n)P \cong (\boldsymbol{\nu} n)Q & \text{(Cong Res)} \\
P \cong Q \Rightarrow m(n).P \cong m(n).Q & \text{(Cong In)} \\
P \cong Q \Rightarrow !P \cong !Q & \text{(Cong Repl)}
\end{array}$$

In this paper we essentially make use of two congruences: α-congruence and structural congruence. As usual, *α-congruence* $\equiv_\alpha$ is the congruence that identifies processes up to the safe renaming of bound names.

Definition 2.6 (α-congruence). α-congruence*, noted $\equiv_\alpha$, is the least congruence on processes such that,*

$$\begin{array}{lll}
(\boldsymbol{\nu} n)P \equiv_\alpha (\boldsymbol{\nu} p)\{n\leftrightarrow p\}{\cdot}P & \textit{where } p \notin na(P) & \text{(Alpha Res)} \\
m(n).P \equiv_\alpha m(p).\{n\leftrightarrow p\}{\cdot}P & \textit{where } p \notin na(P) & \text{(Alpha In)}
\end{array}$$

Definition 2.7 (Safe substitution). *For any process P and substitution θ we denote by $\theta(P)$ the process inductively (and nondeterministically) defined as follows.*

$$
\begin{array}{lll}
\theta(\mathbf{0}) & \triangleq \mathbf{0} & \\
\theta(m\langle n\rangle) & \triangleq \theta(m)\langle\theta(n)\rangle & \\
\theta(P|Q) & \triangleq \theta(P)|\theta(Q) & \\
\theta((\boldsymbol{\nu} n)P) & \triangleq (\boldsymbol{\nu} p)\theta(P\{n\leftrightarrow p\}) & \textit{where } p \notin \mathfrak{D}(\theta) \cup \mathfrak{I}(\theta) \cup \mathit{fn}(P) \\
\theta(m(n).P) & \triangleq \theta(m)(p).\theta(P\{n\leftrightarrow p\}) & \textit{where } p \notin \mathfrak{D}(\theta) \cup \mathfrak{I}(\theta) \cup \mathit{fn}(P) \\
\theta(!P) & \triangleq !\theta(P) &
\end{array}
$$

We write $P\theta$ for $\theta(P)$ when θ has the form $\{n\leftarrow m\}$ or $\{n\leftrightarrow m\}$. We have

Lemma 2.8. *Let P be a process. Then*

$$
\begin{array}{ll}
\textit{1. } \tau\boldsymbol{\cdot} P \equiv_\alpha \tau(P) & \textit{where } \tau \textit{ is any transposition} \\
\textit{2. } P\{n\leftarrow p\} \equiv_\alpha \{n\leftrightarrow p\}\boldsymbol{\cdot} P & \textit{where } p \notin \mathit{fn}(P)
\end{array}
$$

From Lemma 2.8 the following alternative characterization of α-congruence immediately follows.

Proposition 2.9. *α-congruence is the least congruence on processes such that*

$$
\begin{array}{ll}
(\boldsymbol{\nu} n)P \equiv_\alpha (\boldsymbol{\nu} p)P\{n\leftarrow p\} & \textit{where } p \notin \mathit{fn}(P) \\
m(n).P \equiv_\alpha m(p).P\{n\leftarrow p\} & \textit{where } p \notin \mathit{fn}(P)
\end{array}
$$

As expected, safe substitution preserves α-congruence:

Proposition 2.10. *If $P \equiv_\alpha Q$ then $\theta(P) \equiv_\alpha \theta(Q)$.*

Definition 2.11 (Structural congruence). Structural congruence, *noted $\equiv$, is the least congruence on processes such that*

$$
\begin{array}{ll}
P \equiv_\alpha Q \Rightarrow P \equiv Q & \text{(Struct Alpha)} \\
P|\mathbf{0} \equiv P & \text{(Struct Par Void)} \\
P|Q \equiv Q|P & \text{(Struct Par Comm)} \\
P|(Q|R) \equiv (P|Q)|R & \text{(Struct Par Assoc)} \\
n \notin \mathit{fn}(P) \Rightarrow P|(\boldsymbol{\nu} n)Q \equiv (\boldsymbol{\nu} n)(P|Q) & \text{(Struct Res Par)} \\
n \neq p, n \neq m \Rightarrow (\boldsymbol{\nu} n)p(m).P \equiv p(m).(\boldsymbol{\nu} n)P & \text{(Struct Res Inp)} \\
(\boldsymbol{\nu} n)\mathbf{0} \equiv \mathbf{0} & \text{(Struct Res Void)} \\
(\boldsymbol{\nu} n)(\boldsymbol{\nu} m)P \equiv (\boldsymbol{\nu} m)(\boldsymbol{\nu} n)P & \text{(Struct Res Comm)} \\
!\mathbf{0} \equiv \mathbf{0} & \text{(Struct Repl Void)} \\
!P \equiv !P|P & \text{(Struct Repl Copy)} \\
!(P|Q) \equiv !P|!Q & \text{(Struct Repl Par)} \\
!!P \equiv !P & \text{(Struct Repl Repl)}
\end{array}
$$

Although the axiom (Struct Res Inp) is absent from standard presentations of π-calculi, the general consensus seems to be that such axiom is quite sensible as a structural congruence. (Struct Res Inp) is implicit in early work of Boudol on the chemical abstract machine, and is irrelevant as far as extensional properties of processes (eg. behavioral equivalence) are concerned. On the other hand, it has some convenient consequences in the setting of an intensional logic like ours. The axioms (Struct Repl Void/Par/Repl) were proposed by Engelfriet [10] in order to show the decidability of $\equiv$ in a version of the π-calculus.

Proposition 2.12 (Basic properties of $\equiv$). *For all processes P and Q,*

1. *If $P \equiv Q$ then $fn(P) = fn(Q)$.*
2. *If $n \notin fn(P)$ then $(\boldsymbol{\nu} n)P \equiv P$.*
3. *$P \equiv Q$ if and only if $\tau(P) \equiv \tau(Q)$.*
4. *For all substitutions θ, if $P \equiv Q$, then $\theta(P) \equiv \theta(Q)$.*

Proposition 2.13 (Inversion). *For all processes P and Q,*

1. *If $(\boldsymbol{\nu} n)P \equiv \mathbf{0}$ then $P \equiv \mathbf{0}$.*
2. *If $(\boldsymbol{\nu} n)P \equiv R|Q$ then there are R' and Q' such that either $R \equiv (\boldsymbol{\nu} n)R'$ and $Q \equiv Q'$ and $n \notin fn(Q)$, or $R \equiv R'$ and $Q \equiv (\boldsymbol{\nu} n)Q'$ and $n \notin fn(R)$.*
3. *If $(\boldsymbol{\nu} n)P \equiv (\boldsymbol{\nu} m)Q$ then either $P \equiv \{n \leftrightarrow m\} \cdot Q$ or there are P' and Q' such that $P \equiv (\boldsymbol{\nu} m)P'$, $Q \equiv (\boldsymbol{\nu} n)Q'$ and $P' \equiv Q'$.*

Versions of Proposition 2.13(1–2) for the Ambient Calculus have been proved in [8] and [7]. Proposition 2.13(3) has a simple proof based on results in [10], as suggested by J. Engelfriet.

The dynamics of processes is captured by reduction:

Definition 2.14 (Reduction). *Reduction is the least binary relation $\rightarrow$ on processes inductively defined as follows.*

$m\langle n\rangle \| m(p).P \rightarrow P\{p \leftarrow n\}$	(Red React)
$Q \rightarrow Q' \Rightarrow P \| Q \rightarrow P \| Q'$	(Red Par)
$P \rightarrow Q \Rightarrow (\boldsymbol{\nu} n)P \rightarrow (\boldsymbol{\nu} n)Q$	(Red Res)
$P \equiv P', P' \rightarrow Q', Q' \equiv Q \Rightarrow P \rightarrow Q$	(Red Struct)

Proposition 2.15 (Basic properties of $\rightarrow$). *For all processes P and Q,*

1. *If $P \rightarrow Q$ then $fn(Q) \subseteq fn(P)$.*
2. *For all substitutions θ, if $P \rightarrow Q$, then $\theta(P) \rightarrow \theta(Q)$.*
3. *If $P \rightarrow Q$ and $Q \equiv (\boldsymbol{\nu} n)Q'$ for some Q', then there are P' and Q'' such that $P \equiv (\boldsymbol{\nu} n)P'$, $P' \rightarrow Q''$ and $Q'' \equiv Q'$.*

Proposition 2.15(3) is a consequence of Proposition 2.13(3) and does not hold for versions of π-calculi where $\equiv$ does not satisfy (Struct Res Inp).

Remark 2.16. Reduction as defined in Definition 2.14 is "almost the same" as the usual one in the following sense. Let $\equiv_c$ be the the standard structural congruence of [16] restricted to the asynchronous π-calculus. Thus $\equiv_c \subseteq \equiv$. Likewise, let $\rightarrow_c$ be the standard reduction of the asynchronous π-calculus. It is clear that $\rightarrow_c \subseteq \rightarrow$. We also have that for all processes P and Q, if $P \rightarrow Q$ then there is a process R such that $P \rightarrow_c R$ and $Q \equiv R$.

3 Syntax of the Spatial Logic

Basic constructs of the spatial logic include logical, spatial and temporal operations, and a greatest fixpoint combinator supporting recursive (inductive and coinductive) definition of properties.

$$
\begin{array}{lll}
x, y, z & \in & \mathcal{V} \text{ (Name variables)} \\
X, Y, Z & \in & \mathcal{X} \text{ (Propositional variables)} \\
\eta & \in & \Lambda \cup \mathcal{V} \text{ (Names or name variables)}
\end{array}
$$

$$
\begin{array}{lll}
A, B, C ::= & & \\
 & \mathtt{F} & \text{(False)} \\
 & A \wedge B & \text{(Conjunction)} \\
 & A \Rightarrow B & \text{(Implication)} \\
 & \mathbf{0} & \text{(Void)} \\
 & A|B & \text{(Composition)} \\
 & A \triangleright B & \text{(Guarantee)} \\
 & \eta \circledR A & \text{(Revelation)} \\
 & A \oslash \eta & \text{(Hiding)} \\
 & \eta\langle\eta'\rangle & \text{(Message)} \\
 & \forall x.A & \text{(Universal quantification)} \\
 & \text{И} x.A & \text{(Fresh name quantification)} \\
 & \Diamond A & \text{(Next step)} \\
 & X & \text{(Propositional variable)} \\
 & \boldsymbol{\nu} X.A & \text{(Greatest fixpoint)}
\end{array}
$$

Fig. 1. Formulas.

Definition 3.1 (Formulas). *Given an infinite set $\mathcal{V}$ of name variables, and an infinite set $\mathcal{X}$ of propositional variables (mutually disjoint from Λ), formulas are defined as shown in Fig. 1.*

The meaning of these formulas was briefly discussed in the introduction; their semantics is given later in Definition 4.14. We highlight here some or the more unusual operators. The formula $\mathbf{0}$ is satisfied by any process in the structural congruence class of $\mathbf{0}$. The formula $A|B$ is satisfied by any process that can be decomposed into processes that satisfy respectively A and B. Guarantee is the logical adjunct of composition: $A \triangleright B$ is satisfied by those processes whose composition with any process satisfying A results in a process satisfying B. The formula $n \circledR A$ is satisfied by all processes congruent with some process $(\boldsymbol{\nu} n)P$, where P satisfies A; hiding is the logical adjunct of revelation. Message $m\langle n\rangle$ holds of processes structurally congruent to a message $m\langle n\rangle$. The formula $\text{И}x.A$ denotes fresh name quantification; a process satisfies $\text{И}x.A$ if for (some/all) fresh names n (fresh in the process and in the formula), it satisfies $A\{x \leftarrow n\}$. This quantifier exhibits the universal/existential ambivalence typical of freshness: a

C	$fn(C)$	$fv(C)$	$fpv(C)$
$\mathbf{F}$ $\mathbf{0}$	$\emptyset$	$\emptyset$	$\emptyset$
$A \wedge B$ $A \Rightarrow B$ $A \vert B$ $A \triangleright B$	$fn(A) \cup fn(B)$	$fv(A) \cup fv(B)$	$fpv(A) \cup fpv(B)$
$\eta\langle\eta'\rangle$	$\{\eta, \eta'\} \cap \Lambda$	$\{\eta, \eta'\} \cap \mathcal{V}$	$\emptyset$
$\eta®A$ $A \oslash \eta$	$fn(A) \cup (\{\eta\} \cap \Lambda)$	$fv(A) \cup (\{\eta\} \cap \mathcal{V})$	$fpv(A)$
$\forall x.A$ $Ix.A$	$fn(A)$	$fv(A) \setminus \{x\}$	$fpv(A)$
$\Diamond A$	$fn(A)$	$fv(A)$	$fpv(A)$
X	$\emptyset$	$\emptyset$	$\{X\}$
$\boldsymbol{\nu} X.A$	$fn(A)$	$fv(A)$	$fpv(A) \setminus \{X\}$

Fig. 2. Free names in formulas.

property holding of some fresh names should also hold of any other fresh name. As we shall see, combining the fresh name quantifier with revelation will enable us to define a hidden name quantifier, that is a quantifier over names that are locally restricted in the process at hand.

In formulas of the form $\forall x.A$, $Ix.A$ and $\boldsymbol{\nu} X.A$ the distinguished occurrences of x and X are binding, with scope the formula A. We define on formulas the relation $\equiv_\alpha$ of α-congruence in the standard way, that is, as the least congruence identifying formulas modulo safe renaming of bound (name and propositional) variables. We will consider formulas always modulo α-congruence.

Definition 3.2 (Free names and variables in formulas). *For any formula C, we introduce the following sets, inductively defined in Fig. 2.*

$fn(C)$	*free names in C*
$fv(C)$	*free name variables in C*
$fpv(C)$	*free propositional variables in C*

By $fnv(A)$ we mean the set $fv(A) \cup fn(A)$. A formula is *name-closed* if it has no free name variables, and *closed* if it has no free variables whatsoever.

Definition 3.3 (Negative and Positive Occurrences). *For any formula C, the set* $\mathrm{Neg}(C)$ *(resp.* $\mathrm{Pos}(C)$*) of the variables which occur negatively (resp. positively) in C are inductively defined in Fig. 3.*

We say that a propositional variable X is *positive* (resp. *negative*) in A if $X \in \mathrm{Pos}(A)$ (resp. $X \in \mathrm{Neg}(A)$). We also say that a formula A is *monotonic in X* (resp. *antimonotonic in X*) whenever $X \notin \mathrm{Neg}(A)$ (resp. $X \notin \mathrm{Pos}(A)$). Note that a variable X can be both positive and negative in a formula A. Moreover, if X is either positive or negative in A then $X \in fpv(A)$.

C	$\mathrm{Neg}(C)$	$\mathrm{Pos}(C)$
$\mathtt{F}$ $\mathbf{0}$ $\eta\langle\eta'\rangle$	$\emptyset$	$\emptyset$
$A \wedge B$ $A\|B$	$\mathrm{Neg}(A) \cup \mathrm{Neg}(B)$	$\mathrm{Pos}(A) \cup \mathrm{Pos}(B)$
$A \Rightarrow B$ $A \triangleright B$	$\mathrm{Pos}(A) \cup \mathrm{Neg}(B)$	$\mathrm{Neg}(A) \cup \mathrm{Pos}(B)$
$n®A$ $A \oslash n$ $\forall x.A$ $\text{И}x.A$ $\Diamond A$	$\mathrm{Neg}(A)$	$\mathrm{Pos}(A)$
X	$\emptyset$	$\{X\}$
$\boldsymbol{\nu}X.A$	$\mathrm{Neg}(A) \setminus \{X\}$	$\mathrm{Pos}(A) \setminus \{X\}$

Fig. 3. Negative and Positive occurrences.

A formula A is *well-formed* if for every sub-formula of A of the form $\boldsymbol{\nu}X.B$, B is well-formed and monotonic in X. In other words, if $\boldsymbol{\nu}X.B$ is well-formed, all free occurrences of X in B must occur just at positive positions. As usual, the well-formedness condition ensures monotonicity of the denotation mapping associated with fixpoint formulas. In what follows we will just consider well-formed formulas.

We extend the previously given notion of substitution to name variables and formulas as follows. When S is a finite set of either variables and names, and N is a set of names, $\theta : S \to N$ means that θ is a substitution assigning a name in N to each variable or name in S. If $\theta : S \to N$ is a substitution then θ_{-x} denotes the substitution of domain $S \setminus \{x\}$ and codomain N defined by $\theta_{-x}(y) = \theta(y)$, for all $y \in S \setminus \{x\}$.

Definition 3.4 (Safe substitution). *For any formula A and substitution θ we denote by $\theta(A)$ the formula inductively defined as follows.*

$$
\begin{array}{llll}
\theta(\mathtt{F}) & \triangleq \mathtt{F} & \theta(\eta®A) & \triangleq \theta(\eta)®\theta(A) \\
\theta(\mathbf{0}) & \triangleq \mathbf{0} & \theta(A\oslash\eta) & \triangleq \theta(A)\oslash\theta(\eta) \\
\theta(A \wedge B) & \triangleq \theta(A) \wedge \theta(B) & \theta(\text{И}x.A) & \triangleq \text{И}x.\theta_{-x}(A) \\
\theta(A \Rightarrow B) & \triangleq \theta(A) \Rightarrow \theta(B) & \theta(\forall x.A) & \triangleq \forall x.\theta_{-x}(A) \\
\theta(A|B) & \triangleq \theta(A)|\theta(B) & \theta(\Diamond A) & \triangleq \Diamond\theta(A) \\
\theta(A \triangleright B) & \triangleq \theta(A) \triangleright \theta(B) & \theta(X) & \triangleq X \\
\theta(m\langle n\rangle) & \triangleq \theta(m)\langle\theta(n)\rangle & \theta(\boldsymbol{\nu}X.A) & \triangleq \boldsymbol{\nu}X.\theta(A)
\end{array}
$$

When A and B are formulas, we denote by $A\{X \leftarrow B\}$ the capture avoiding substitution of all free occurrences of X in A by B, defined in the expected way. By $F\{-\}$ we denote a formula context with possibly multiple occurrences of the hole $-$. Then, whenever A is a formula, we denote by $F\{A\}$ the formula obtained by textually replacing every occurrence of the hole $-$ in the context $F\{-\}$ by A. Note that free (name or propositional) variables in A will be captured by binders present in $F\{-\}$; cf. the standard notion of context substitution.

4 Semantics

The semantics of formulas is defined by assigning to each formula A a set of processes $[\![A]\!]$, namely the set of all processes that satisfy the property denoted by A.

However, as we shall justify soon, not any set of processes can denote a property in a proper way. For instance, it is sensible to require $[\![A]\!]$ to be closed under structural congruence. That is, if a process P satisfy some property A, then any process Q such that $Q \equiv P$ must also satisfy A. Moreover, suppose we have $P \in [\![A]\!]$, $n \notin fn(A)$ but $n \in fn(P)$. Since $n \notin fn(A)$, the free occurrences of n in $fn(P)$ are *fresh* w.r.t. formula A. Now, the particular choice of the name n should not depend on A itself, since it is natural to consider that all fresh names w.r.t. A are to be treated uniformly (all alike). Therefore, it is natural to require that also $P\{n{\leftarrow}m\} \in [\![A]\!]$, where m is any other fresh name w.r.t. A and P, that is $m \notin fn(P) \cup fn(A)$.

Hence, we say that a set of processes Φ is supported by a set of names N if, for all m, n not in the support N, if P belongs to Φ then $P\{n{\leftrightarrow}m\}$ is also in Φ. We then take as properties only those sets of processes that have a finite support. Intuitively, the support of a property is the semantic counterpart of the set of free names of a formula; the least support of the denotation of a formula is included in the set of free names of the formula. Sets with infinite support could only correspond to formulas that have an infinite set of free names, and are therefore excluded.

Moreover, the notion of finite support seems crucial for the semantics of the fresh name quantifier, $\text{И}x.A$, and consequently for the semantics of the derived hidden name quantifier $\mathsf{H}x.A$. For example, the semantics of the spatial logics of [6, 5, 3] is given in terms of sets of processes that are closed only under structural congruence. If we try to extend that semantics to recursive formulas, we run into a problem: $\text{И}x.A$ is not a monotonic operator, and could not be used together with recursion. This discussion is continued in more detail in Section 4.3.

4.1 Property Sets

The above observations motivate the following notion of *property set.* A property set is a set of processes closed under structural congruence and finitely supported.

Definition 4.1 (Property Set). *Given a finite set of names N, a* property set *(Pset) supported by N is a set of processes Ψ such that*

1. *(closure under $\equiv$) If $P \in \Psi$ and $P \equiv Q$ then $Q \in \Psi$.*
2. *(finite support) For all $n, m \notin N$, if $P \in \Psi$ then $P\{n{\leftrightarrow}m\} \in \Psi$.*

In general, by "a Pset" we intend a Pset supported by some N; then, N is the support of the Pset. We use Ψ, Φ to range over property sets. The support N plays for some Pset Ψ a role similar to the one played by the set of free names of a given formula, and enables the definition of a notion of name freshness w.r.t. a possibly infinite set of processes. Given a finite set of names N, we will denote by $\mathbb{P}_N$ the set of all Psets supported by N, and by $\mathbb{P}_{-}$ the set of all Psets.

Lemma 4.2 (Operations on Psets). *For all finite $N \subseteq \Lambda$,*

1. *If $N \subseteq N'$ then $\mathbb{P}_N \subseteq \mathbb{P}_{N'}$.*
2. *(Bottom and Top) $\emptyset \in \mathbb{P}_N$ and $\mathcal{P} \in \mathbb{P}_N$.*
3. *(Meet and Join) If $S \subseteq \mathbb{P}_N$ then $\bigcap S \in \mathbb{P}_N$ and $\bigcup S \in \mathbb{P}_N$.*
4. *(Inverse) If $\Psi \in \mathbb{P}_N$ then $\overline{\Psi} = \mathcal{P} \setminus \Psi \in \mathbb{P}_N$.*

Note that Lemma 4.2(2-4) implies

Proposition 4.3 (Lattice). *For all finite $N \subseteq \Lambda$, we have*

1. *$\langle \mathbb{P}_N, \subseteq, \cup, \cap \rangle$ is a complete lattice.*
2. *$\langle \mathbb{P}_N, \cup, \cap, \overline{\cdot} \rangle$ is a Boolean algebra.*

Remark 4.4. Note that $\mathbb{P}_{-}$ is not closed under arbitrary unions. For instance, let $\langle m_1, m_2, \ldots \rangle$ be a linear ordering of Λ, let $P_0 \triangleq \mathbf{0}$ and for any $i > 0$, $P_i \triangleq m_i(n).P_{i-1}$. Then $\{P_i\}$ is finitely supported (with support $\{m_1, \ldots, m_i\}$) for any $i \geq 0$, but $\bigcup_{i \geq 0} \{P_i\}$ is not.

For any operator $f : \mathbb{P}_N \to \mathbb{P}_N$, we denote by $\mathsf{lfix}_{\mathbb{P}_N}(f)$ the least fixpoint of f and by $\mathsf{gfix}_{\mathbb{P}_N}(f)$ the greatest fixpoint of f.

Definition 4.5 (Tensor and Unit). *For every $\mathbb{P}_N$, define operations*

$$\otimes : \mathbb{P}_N \times \mathbb{P}_N \to \mathbb{P}_N \quad \mathbf{1} : \mathbb{P}_N$$

by letting, for all $\Psi, \Phi \in \mathbb{P}_N$

$$\begin{array}{ll} \Psi \otimes \Phi & \triangleq \{P \mid \textit{there are } Q \textit{ and } R \textit{ st. } P \equiv Q|R \textit{ and } Q \in \Psi \textit{ and } R \in \Phi\} \\ \mathbf{1} & \triangleq \{P \mid P \equiv \mathbf{0}\} \end{array}$$

In [5] it is shown that the set of all $\equiv$-closed subsets of $\mathcal{P}$ gives rise to a commutative quantale. The same result still holds for domains of Psets.

Proposition 4.6 (Quantale). *For all finite $N \subseteq \Lambda$, $\langle \mathbb{P}_N, \subseteq, \bigcup, \otimes, \mathbf{1} \rangle$ is a commutative quantale, that is:*

1. *$\langle \mathbb{P}_N, \subseteq, \bigcup \rangle$ is a complete join semilattice.*
2. *$\langle \mathbb{P}_N, \otimes, \mathbf{1} \rangle$ is a commutative monoid.*
3. *$\Phi \otimes \bigcup S = \bigcup \{\Phi \otimes \Psi \mid \Psi \in S\}$, for all $\Phi \in \mathbb{P}_N$ and $S \subseteq \mathbb{P}_N$.*

We now extend the application of transpositions (not of arbitrary substitutions!) to Psets as follows: if τ is a transposition and Ψ is a Pset, define $\tau(\Psi) \triangleq \{\tau(P) \mid P \in \Psi\}$.

Lemma 4.7 (Transposing Psets). *We have*

1. *For any process P and Pset Ψ, $P \in \tau(\Psi)$ if and only if $\tau(P) \in \Psi$.*
2. *$\Psi \in \mathbb{P}_N$ if and only if $\tau(\Psi) \in \mathbb{P}_{\tau(N)}$.*
3. *If $m, n \notin N$ and $\Psi \in \mathbb{P}_N$ then $\{m \leftrightarrow n\}(\Psi) = \Psi$.*

We can verify that a Pset Ψ supported by N is finitely supported by N in the precise sense of [11].

Definition 4.8 (Support). *Let N be a set of names. A Pset Ψ is* supported *by N whenever every permutation that fixes N also fixes Ψ. Moreover Ψ is* finitely supported *by N if supported by N and N is finite.*

A name permutation π over Λ is an injective name substitution such that $\mathfrak{D}(\pi) = \mathfrak{I}(\pi)$. Let S_Λ be the group of all name permutations; recall that any permutation can be expressed as a composition of transpositions. For any Pset Ψ, $\pi(\Psi) \in \mathbb{P}_-$, by Lemma 4.7(2). Hence $\mathbb{P}_-$ is an $S_\Lambda-$set.

Now, let $\Psi \in \mathbb{P}_N$. Pick any $\pi \in S_\Lambda$ and assume that π is not the identity permutation. This implies that there is some permutation π', such that $\pi'(m) = \pi(m)$ for all $m \in \Lambda$ and $\pi'(m) \neq m$, for all $m \in \mathfrak{D}(\pi')$. Assume that for all $n \in N$, $\pi(n) = n$. Then, for all $n \in N$, $\pi'(n) = n$. We can see that N is disjoint from $\mathfrak{D}(\pi') = \mathfrak{I}(\pi')$. Hence, π' can be written as a composition of transpositions $\tau_1 \cdots \tau_k$ such that $\tau_i = \{p_i \leftrightarrow q_i\}$ and $p_i, q_i \notin N$, for all $i = 1, \cdots, k$. Therefore $\pi'(\Psi) = \pi(\Psi) = \Psi$. This means that N (finitely) supports Ψ. We conclude that $\mathbb{P}_-$ is a *perm*(Λ)-set with the finite support property.

We also have

Proposition 4.9 (Least Support). *Let $\Phi \in \mathbb{P}_N$. Then*

1. *There is a least set of names $supp(\Phi)$ such that $\Phi \in \mathbb{P}_{supp(\Phi)}$.*
2. *For any transposition τ, if $supp(\Phi) = M$ then $supp(\tau(\Phi)) = \tau(M)$.*

Intuitively, the set of names $supp(\Phi)$ represents the set of "free names" of the Pset Φ (in the sense of Lemma 4.7(3)), hence $supp(-)$ is the semantic counterpart of the set $fn(-)$ of free names of a formula.

If a Pset Ψ is supported by N and $N \subseteq M$, Ψ is also supported by M. On the other hand, for any finite set of names S, we can always find a minimal Pset Φ containing Ψ and supported by S, given by

Definition 4.10 (Closure). *Let $\Phi \in \mathbb{P}_M$ and let N be a finite set of names. The* closure *of Φ on N is given by*

$$Cl(\Phi, N) \triangleq \bigcup_{\pi \in \Pi_N} \pi(\Phi)$$

where by Π_N we denote the the set of all permutations that fix N.

For any finite set of names N, $Cl(-, N)$ is a closure operator. We also have

Lemma 4.11. *If $\Phi \in \mathbb{P}_M$ and N is a finite set of names, $Cl(\Phi, N) \in \mathbb{P}_N$.*

4.2 Satisfaction

We define the semantics of a formula A by a Pset $[\![A]\!] \in \mathbb{P}_{fn(A)}$. Since A may contain free occurrences of propositional variables, its denotation depends on the denotation of such variables, which is given by a valuation.

Definition 4.12 (Valuation). *A valuation v is a mapping from a finite subset of $\mathcal{X}$, assigning to each propositional variable X in its domain $\mathfrak{D}(v)$ a Pset Ψ. Given a formula A, a valuation for A is any valuation v such that $fpv(A) \subseteq \mathfrak{D}(v)$.*

Thus, the role of valuations is to interpret free propositional variables occurring in the formula A. When v is a valuation, we will write $v[X \leftarrow \Psi]$ to denote the valuation of domain $\mathfrak{D}(v) \cup \{X\}$ that assigns Ψ to X, and behaves like v w.r.t. any other propositional variable $Z \neq X$. For any valuation v, we let

$$fn(v) \triangleq \bigcup \{supp(v(X)) \mid X \in \mathfrak{D}(v)\}$$

Taking into account the extra information yielded by a valuation, we now give a more refined characterization of the free names in a formula A as follows

Definition 4.13 (Free names under a valuation). *If A is a formula and v a valuation for A, we define the set $fn^v(A)$ of free names of A under v by*

$$\begin{array}{ll}
fn^v(\mathbf{0}) = fn^v(\mathtt{F}) & \triangleq \emptyset \\
fn^v(A \wedge B) = fn^v(A|B) = & \\
\quad fn^v(A \triangleright B) = fn^v(A \Rightarrow B) & \triangleq fn^v(A) \cup fn^v(B) \\
fn^v(\eta \circledR A) = fn^v(A \oslash \eta) & \triangleq (\{\eta\} \cap \Lambda) \cup fn^v(A) \\
fn^v(\forall x.A) = fn^v(\text{И} x.A) = \Diamond A & \triangleq fn^v(A) \\
fn^v(\boldsymbol{\nu} X.A) & \triangleq fn^{v[X \leftarrow \emptyset]}(A) \\
fn^v(X) & \triangleq supp(v(X))
\end{array}$$

Note that the definition of $fn^v(A)$ mimics the one given for $fn(A)$ except in the clauses for fixpoint and propositional variable. The set $fn^v(A)$ is used in an essential way in the definition of the semantics of the fresh name quantifier, where the quantification witness is tested for freshness w.r.t. the property denoted by the formula A, which may contain free occurrences of propositional variables.

Definition 4.14 (Denotation and Satisfaction). *The* denotation map *$[\![A]\!]_v$ is the function that assigns a Pset to each name-closed formula A and valuation (for A) v. It is inductively defined in Fig 4. We write $P \models_v A$ whenever $P \in [\![A]\!]_v$: this means that P* satisfies *formula A under valuation v.*

We now show that the denotation map is well-defined. In particular, we show that $[\![A]\!]_v$ is indeed a Pset (actually, $[\![A]\!]_v \in \mathbb{P}_{fn^v(A)}$) and that the semantics of the fixpoint operation is the intended one (that is, $\boldsymbol{\nu} X.B$ indeed denotes the greatest fixpoint). A more natural formulation of $[\![\boldsymbol{\nu} X.A]\!]_v$ will be given and justified in Proposition 4.27.

Since we are considering formulas up to α-congruence, we start by verifying that the denotation map is well-defined on the corresponding equivalence classes.

$$
\begin{array}{ll}
[\![\mathbf{F}]\!]_v & \triangleq \emptyset \\
[\![A \wedge B]\!]_v & \triangleq [\![A]\!]_v \cap [\![B]\!]_v \\
[\![A \Rightarrow B]\!]_v & \triangleq \{Q \mid \textit{if } Q \in [\![A]\!]_v \textit{ then } Q \in [\![B]\!]_v\} \\
[\![\mathbf{0}]\!]_v & \triangleq \mathbf{1} \\
[\![A|B]\!]_v & \triangleq [\![A]\!]_v \otimes [\![B]\!]_v \\
[\![A \triangleright B]\!]_v & \triangleq \{Q \mid \textit{if } P \in [\![A]\!]_v \textit{ then } P|Q \in [\![B]\!]_v\} \\
[\![n \circledR A]\!]_v & \triangleq \{Q \mid Q \equiv (\boldsymbol{\nu} n)P \textit{ and } P \in [\![A]\!]_v\} \\
[\![A \oslash n]\!]_v & \triangleq \{Q \mid (\boldsymbol{\nu} n)Q \in [\![A]\!]_v\} \\
[\![m\langle n\rangle]\!]_v & \triangleq \{Q \mid Q \equiv m\langle n\rangle\} \\
[\![\forall x.A]\!]_v & \triangleq \bigcap_{n \in \Lambda} [\![A\{x \leftarrow n\}]\!]_v \\
[\![\text{И} x.A]\!]_v & \triangleq \bigcup_{n \notin fn^v(A)} ([\![A\{x \leftarrow n\}]\!]_v \setminus \copyright\{n\}) \\
[\![\Diamond A]\!]_v & \triangleq \{P \mid P \to Q \textit{ and } Q \in [\![A]\!]_v\} \\
[\![X]\!]_v & \triangleq v(X) \\
[\![\boldsymbol{\nu} X.A]\!]_v & \triangleq \bigcup \{\Psi \in \mathbb{P}_{fn^v(\boldsymbol{\nu} X.A)} \mid \Psi \subseteq [\![A]\!]_{v[X \leftarrow \Psi]}\}
\end{array}
$$

Fig. 4. Denotation of formulas.

Lemma 4.15. *For all formulas A, B and valuations v for A and B, if $A \equiv_\alpha B$, then $[\![A]\!]_v = [\![B]\!]_v$.*

Note that assignments to propositional variables which do not occur free in the interpreted formula do not affect its denotation. Therefore, valuations can always be weakened and thinned whenever appropriate.

Remark 4.16. For any formula A, Pset Φ and valuation v for A, if $X \notin fpv(A)$ then $[\![A]\!]_v = [\![A]\!]_{v[X \leftarrow \Phi]}$.

We now extend the application of transpositions to valuations, this is done in the expected way: when v is a valuation, let $\tau(v)$ be the valuation with same domain as v and defined by $\tau(v)(X) \triangleq \tau(v(X))$, for all $X \in \mathfrak{D}(v)$.

Lemma 4.17. *For any formula A, valuation v and transposition τ*

1. *If $\tau = \{m \leftrightarrow n\}$ and $m, n \notin fn(v)$ then $[\![A]\!]_v = [\![A]\!]_{\tau(v)}$.*
2. *$\tau(fn^v(A)) = fn^{\tau(v)}(\tau(A))$.*
3. *Let $fn^v(\boldsymbol{\nu} X.A) \subseteq M$. Then $fn^v(\boldsymbol{\nu} X.A) \subseteq fn^{v[X \leftarrow \Psi]}(A) \subseteq M$, for all $\Psi \in \mathbb{P}_M$.*

We conclude that if $N = fn^v(\boldsymbol{\nu} X.A)$ then $fn^{v[X \leftarrow \Psi]}(A) = N$, for all $\Psi \in \mathbb{P}_N$.

Fundamental properties of the denotation mapping are stated in the following main theorem, from which all correctness properties of the semantics follow.

Theorem 4.18. *For all formulas A, appropriate valuations v, and Psets Ψ, Φ*

1. *$[\![A]\!]_v \in \mathbb{P}_{fn^v(A)}$.*
2. *For all transpositions τ, $\tau([\![A]\!]_v) = [\![\tau(A)]\!]_{\tau(v)}$.*
3. *If $X \notin \mathrm{Neg}(A)$ and $\Psi \subseteq \Phi$ then $[\![A]\!]_{v[X \leftarrow \Psi]} \subseteq [\![A]\!]_{v[X \leftarrow \Phi]}$.*
4. *If $X \notin \mathrm{Pos}(A)$ and $\Psi \subseteq \Phi$ then $[\![A]\!]_{v[X \leftarrow \Phi]} \subseteq [\![A]\!]_{v[X \leftarrow \Psi]}$.*

Proof. See Appendix.

Lemma 4.19. *For any formula A and valuation v for A we have*

$$supp([\![A]\!]_v) \subseteq fn^v(A)$$

Proof. By Theorem 4.18(1) $[\![A]\!]_v \in \mathbb{P}_{fn^v(A)}$; hence by Proposition 4.9 there is a least set $N = supp([\![A]\!]_v)$ such that $[\![A]\!]_v \in \mathbb{P}_N$. So $supp([\![A]\!]_v) \subseteq fn^v(A)$.

Remark 4.20. By inspection of the proof of Theorem 4.18 we can verify

– Assume $[\![A]\!]_v \in \mathbb{P}_N$ and $[\![B]\!]_v \in \mathbb{P}_M$. Then

$$\begin{array}{lll}
[\![\mathrm{F}]\!]_v \in \mathbb{P}_\emptyset & [\![A \Rightarrow B]\!]_v \in \mathbb{P}_{N\cup M} & [\![A \oslash n]\!]_v \in \mathbb{P}_{N\cup\{n\}} \\
[\![\mathbf{0}]\!]_v \in \mathbb{P}_\emptyset & [\![A|B]\!]_v \in \mathbb{P}_{N\cup M} & [\![\Diamond A]\!]_v \in \mathbb{P}_N \\
[\![p\langle q\rangle]\!]_v \in \mathbb{P}_{\{p,q\}} & [\![A \triangleright B]\!]_v \in \mathbb{P}_{N\cup M} & [\![X]\!]_v \in \mathbb{P}_{supp(v(X))} \\
[\![A \wedge B]\!]_v \in \mathbb{P}_{N\cup M} & [\![n \circledR A]\!]_v \in \mathbb{P}_{N\cup\{n\}} &
\end{array}$$

– If $[\![A\{x \leftarrow n\}]\!]_v \in \mathbb{P}_{N\cup\{n\}}$ for all $n \in \Lambda$, then $[\![\forall x.A]\!]_v \in \mathbb{P}_N$
– If $[\![A\{x \leftarrow n\}]\!]_v \in \mathbb{P}_{N\cup\{n\}}$ for all $n \notin fn^v(A)$ then $[\![Ин x.A]\!]_v \in \mathbb{P}_N$
– If $\Psi \in \mathbb{P}_N$ and $[\![A]\!]_{v[X\leftarrow\Psi]} \in \mathbb{P}_N$ then $[\![\boldsymbol{\nu} X.A]\!]_v \in \mathbb{P}_N$

Lemma 4.21. *Let A be a formula monotonic in X, v a valuation for $\boldsymbol{\nu} X.A$, and $fn^v(\boldsymbol{\nu} X.A) \subseteq M$. Then the function*

$$\boldsymbol{\lambda}\Psi.[\![A]\!]_{v[X\leftarrow\Psi]} : \mathbb{P}_M \rightarrow \mathbb{P}_M$$

is a monotonic operator over the complete lattice $\mathbb{P}_M$.

Proof. Let $\Psi \in \mathbb{P}_M$. By Lemma 4.17(3), we have $fn^{v[X\leftarrow\Psi]}(A) \subseteq M$. By Theorem 4.18(1) and Proposition 4.3(1), we conclude $[\![A]\!]_{v[X\leftarrow\Psi]} \in \mathbb{P}_M$. Monotonicity then follows from Theorem 4.18(3).

Lemma 4.22. *Let A be a formula monotonic in X and v a valuation for $\boldsymbol{\nu} X.A$. Then*

$$[\![\boldsymbol{\nu} X.A]\!]_v = \mathsf{gfix}_{\mathbb{P}_{fn^v(\boldsymbol{\nu} X.A)}}(\boldsymbol{\lambda}\Psi.[\![A]\!]_{v[X\leftarrow\Psi]})$$

Proof. Let $N = fn^v(\boldsymbol{\nu} X.A)$. By Lemma 4.21, the function $\boldsymbol{\lambda}\Psi.[\![A]\!]_{v[X\leftarrow\Psi]} : \mathbb{P}_N \rightarrow \mathbb{P}_N$ is a monotonic operator in the complete lattice $\mathbb{P}_N$. We conclude by Knaster-Tarski Lemma.

Lemma 4.23. *Let A be any formula, $v[X\leftarrow\Psi]$ a valuation for A and B, and B any formula in which X does not occur free. Then*

$$[\![A\{X\leftarrow B\}]\!]_v = [\![A]\!]_{v[X\leftarrow[\![B]\!]_v]}$$

Note that Lemmas 4.23 and 4.22 imply $[\![\boldsymbol{\nu} X.A]\!]_v = [\![A\{X\leftarrow\boldsymbol{\nu} X.A\}]\!]_v$.

The next lemma clarifies the relationship between free names in formulas and the supports of their denotations in fixpoint formulas. It essentially says that any pre-fixpoint of the mapping $\boldsymbol{\lambda}\Psi.[\![A]\!]_{v[X\leftarrow\Psi]}$ is contained in a larger pre-fixpoint of the same mapping with a smaller support, which is bounded from below by the set of names $fn^v(\boldsymbol{\nu} X.A)$. A consequence is that the greatest fixpoint of such mapping inhabits $\mathbb{P}_{fn^v(\boldsymbol{\nu} X.A)}$, cf. Lemma 4.26 below.

Lemma 4.24. *Let A be a formula and $v[X\leftarrow\Phi]$ a valuation for A such that $fn^v(\boldsymbol{\nu}X.A) \subseteq N \subseteq M$, $\Phi \subseteq [\![A]\!]_{v[X\leftarrow\Phi]}$ and $\Phi \in \mathbb{P}_M$.*
Then $Cl(\Phi, N) \subseteq [\![A]\!]_{v[X\leftarrow Cl(\Phi,N)]}$.

Proof. See Appendix.

Relying on Lemma 4.24 we can now provide a crisper (after-the-fact) justification for the choice of the particular support N in the semantic clause for the fixpoint operator. Indeed,

Lemma 4.25. *Let M be a finite set of names such that $fn^v(\boldsymbol{\nu}X.A) \subseteq M$. Then*

$$\mathsf{gfix}_{\mathbb{P}_M}(\boldsymbol{\lambda}\Psi.[\![A]\!]_{v[X\leftarrow\Psi]}) \subseteq [\![\boldsymbol{\nu}X.A]\!]_v$$

Proof. Assume $M \supseteq fn^v(\boldsymbol{\nu}X.A) = N$. Pick $P \in \mathsf{gfix}_{\mathbb{P}_M}(\boldsymbol{\lambda}\Psi.[\![A]\!]_{v[X\leftarrow\Psi]})$. Then $P \in \Phi$ for some $\Phi \in \mathbb{P}_M$ such that $\Phi \subseteq [\![A]\!]_{v[X\leftarrow\Phi]}$. By Lemma 4.24, we have $Cl(\Phi, N) \subseteq [\![A]\!]_{v[X\leftarrow Cl(\Phi,N)]}$. Hence $Cl(\Phi, N) \subseteq [\![\boldsymbol{\nu}X.A]\!]_v$. Since $\Phi \subseteq Cl(\Phi, N)$, we conclude $P \in [\![\boldsymbol{\nu}X.A]\!]_v$.

Therefore, on the one hand the fixpoint must be taken in the lattice $\mathbb{P}_M$, where the set M must necessarily include $fn^v(\boldsymbol{\nu}X.A)$. In fact, it is not hard to find a formula A for which the induced operator fails to be closed on $\mathbb{P}_M$ if $fn^v(\boldsymbol{\nu}X.A) \not\subseteq M$ (eg. $M = \emptyset$ and $A = p\langle p\rangle$). On the other hand, Lemma 4.25 states that nothing is lost by picking the set N instead of any bigger support M. More precisely, we have

Lemma 4.26. *Let $\boldsymbol{\nu}X.A$ be a formula and v a valuation for $\boldsymbol{\nu}X.A$. Suppose there is a $\subseteq$-greatest Pset Φ such that $\Phi = [\![A]\!]_{v[X\leftarrow\Phi]}$. Then $\Phi = [\![\boldsymbol{\nu}X.A]\!]_v$.*

Proof. Since Φ is a Pset, then $\Phi \in \mathbb{P}_M$ for some M. By Proposition 4.2(1), $\Phi \in \mathbb{P}_{M'}$ where $M' = M \cup fn^v(\boldsymbol{\nu}X.A)$. By assumption and Lemma 4.25, we have $\Phi \subseteq \{\Phi \in \mathbb{P}_{M'} \mid \Phi \subseteq [\![A]\!]_{v[X\leftarrow\Phi]}\} = \mathsf{gfix}_{\mathbb{P}_{M'}}(\boldsymbol{\lambda}\Phi.[\![A]\!]_{v[X\leftarrow\Phi]}) \subseteq [\![\boldsymbol{\nu}X.A]\!]_v$. Now, $[\![\boldsymbol{\nu}X.A]\!]_v = [\![A]\!]_{v[X\leftarrow[\![\boldsymbol{\nu}X.A]\!]_v]}$. Hence, $[\![\boldsymbol{\nu}X.A]\!]_v \subseteq \Phi$ by assumption. We conclude $\Phi = [\![\boldsymbol{\nu}X.A]\!]_v$.

In fact, even though $\mathbb{P}_-$ is not a complete lattice, the uniformity across renamings out of $\boldsymbol{\nu}X.A$ that exists among the pre-fixpoints of the denotation mapping (cf. Lemma 4.24) implies the following characterisation for the semantics of greatest fixpoint formulas.

Proposition 4.27. *We have $[\![\boldsymbol{\nu}X.A]\!]_v = \{\Psi \in \mathbb{P}_- \mid \Psi \subseteq [\![A]\!]_{v[X\leftarrow\Psi]}\}$.*

Proof. (Right to left). If $P \in \{\Psi \in \mathbb{P}_- \mid \Psi \subseteq [\![A]\!]_{v[X\leftarrow\Psi]}\}$ then there is $\Phi \in \mathbb{P}_-$ such that $P \in \Phi$ and $\Phi \subseteq [\![A]\!]_{v[X\leftarrow\Phi]}$. But $\Phi \in \mathbb{P}_M$ for some finite set of names M. Let $M' = fn^v(\boldsymbol{\nu}X.A)$. Then $\Phi \in \mathbb{P}_{M'\cup M}$. By Lemma 4.24 we have $Cl(\Phi, M') \subseteq [\![A]\!]_{v[X\leftarrow Cl(\Phi,M')]}$. Since $P \in Cl(\Phi, M') \in \mathbb{P}_{M'}$, we conclude $P \in [\![\boldsymbol{\nu}X.A]\!]_v$. (Left to right) Immediate since $\mathbb{P}_{fn^v(\boldsymbol{\nu}X.A)} \subset \mathbb{P}_-$.

Another consequence of the closure property stated in Theorem 4.18(2) is that the relation of satisfaction between processes and formulas is closed under fresh name renaming, that is, properties depend on the distinctions between names and not on their particular identity (a property called *equivariance* in [11]).

Lemma 4.28 (Fresh renaming). *Let P be a process and A a closed formula such that $P \models A$. If $m \notin fn(A) \cup fn(P)$ then $P\{n{\leftarrow}m\} \models A\{n{\leftarrow}m\}$.*

Proof. Since $m \notin fn(P) \cup fn(A)$, by Lemma 2.8(2) we have $P\{n{\leftarrow}m\} \equiv P\{n{\leftrightarrow}m\}$, and $A\{n{\leftarrow}m\} = A\{n{\leftrightarrow}m\}$. We conclude by Theorem 4.18(2).

It should be stressed that the use of transpositions, as suggested to us by A. Pitts, together with the notion of support, yields for Lemma 4.28 a proof that is much simpler than direct ones (eg. [2, 5]).

Basic Derived Connectives Some derived connectives of basic interest are defined as shown next.

$\neg A$	$\triangleq A \Rightarrow \mathtt{F}$	(Negation)
$\mathtt{T}$	$\triangleq \neg\mathtt{F}$	(True)
$A \vee B$	$\triangleq \neg A \Rightarrow B$	(Disjunction)
$A \Vert B$	$\triangleq \neg(\neg A \vert \neg B)$	(Decomposition)
$\exists x.A$	$\triangleq \neg\forall x.\neg A$	(Existential quantification)
$©\eta$	$\triangleq \neg\eta ® \mathtt{T}$	(Free name)
$\eta = \eta'$	$\triangleq \neg(©\eta \triangleright (©\eta \oslash \eta'))$	(Equality)
$\boxdot A$	$\triangleq \neg\Diamond\neg A$	(All next)

Standard operations of the classical predicate calculus, namely $\neg A$ (Negation), $\exists x.A$ (Existential quantification), $A \vee B$ (Disjunction) and $\mathtt{T}$ (True) are defined as expected. Another interesting connective is $A\Vert B$, the DeMorgan dual of composition $A|B$, which supports the definition of a form of spatial quantification. A process satisfies $A\Vert B$ if and only if every component of P w.r.t. composition, satisfies either A or B. We also have the modality $\boxdot$, which is the dual of $\Diamond$. A process satisfies $\boxdot A$ if and only if all processes to which it reduces in one step satisfy A. The free name predicate $©\eta$ holds of any process with a free occurrence of name η. Name equality is defined from general properties of restriction.

Proposition 4.29. *For every process P and names n, p we have*

1. *$P \in [\![©n]\!]_v$ if and only if $n \in fn(P)$.*
2. *$P \in [\![n = p]\!]_v$ if and only if $n = p$.*

Proof. 1. See [6]. 2. We verify that $P \in [\![©n \triangleright (©n \oslash p)]\!]_v$ if and only if $n \neq p$. Suppose $P \in [\![©n \triangleright (©n \oslash p)]\!]_v$. Then, for every Q such that $n \in fn(Q)$ we have $Q|P \in [\![©n \oslash p]\!]_v$. This implies $n \in fn((\boldsymbol{\nu} p)(Q|P))$, and thus $n \neq p$. Conversely, if $n \neq p$ and $n \in fn(Q)$ then $n \in fn((\boldsymbol{\nu} p)(Q|P))$ and $n \in fn(Q|P)$, for all P. Thus, for every Q such that $n \in fn(Q)$, we have $Q|P \in [\![©n \oslash p]\!]_v$ for all P.

4.3 Discussion

The semantics of the fresh quantifier $\text{И}x.A$ is based on finding fresh names outside of $\mathit{fn}^v(A) = \mathit{fn}^v(\text{И}x.A)$, and therefore outside the support of $[\![\text{И}x.A]\!]_v$ (by Lemma 4.19). The fact that names outside the support can be freely renamed (cf. Theorem 4.18(2) and Lemma 4.28) implies that any choice for a fresh name will work equally well.

It is instructive to see how freshness interacts with recursion. Consider the formula

$$\boldsymbol{\nu}Y.\text{И}x.n\langle x\rangle \triangleright \Diamond Y$$

By the fixpoint unfolding property, this formula must have the same meaning as

$$\text{И}x.n\langle x\rangle \triangleright \Diamond(\text{И}x'.n\langle x'\rangle \triangleright \Diamond \cdots)$$

Obviously, the $\mathit{fn}^-(-)$ of the original formula and of its expansion are the same. So, "at each iteration" we have to choose fresh names outside the same set of names, and there is an infinite supply of them. Moreover, at each iteration, the ©n part of the semantic definition of $\text{И}x.A$ subtracts those processes that use names that have been chosen in previous iterations. Further, since the fresh names used at each iteration can be freely renamed, they do not affect the support set, by Definition 4.1(2).

As already discussed, the notion of finite support and Pset seems crucial for the semantics of $\text{И}x.A$. In particular, without a notion of finite support (cf. Definition 4.1(2)), it seems natural to set $\mathit{fn}^v(X) \triangleq \{n \mid n \in \mathit{fn}(P) \text{ and } P \in v(X)\}$, since we must somehow take into account the names contributed by (a binding of) X. Then, consider the set $\Psi = \{p\langle p\rangle\}^{\equiv}$ (we note by $\{-\}^{\equiv}$ closure under structural congruence) and the formula $A = \text{И}x.(\neg x\langle x\rangle | X)$, with $\mathit{fn}^{[X\leftarrow\Psi]}(A) = \{p\}$. We can easily check that

$$q\langle q\rangle | p\langle p\rangle \in [\![A]\!]_{[X\leftarrow\Psi]} = \bigcup_{n\notin\{p\}} ([\![\neg n\langle n\rangle | X]\!]_{[X\leftarrow\Psi]} \setminus \copyright n)$$

Now consider $\Phi = \{r\langle r\rangle \mid r \in \Lambda\}^{\equiv}$ with $\mathit{fn}^{[X\leftarrow\Phi]}(A) = \Lambda$. So we have that $q\langle q\rangle | p\langle p\rangle \notin [\![A]\!]_{[X\leftarrow\Phi]} = \emptyset$. Hence, $\Psi \subseteq \Phi$, but $[\![A]\!]_{[X\leftarrow\Psi]} \not\subseteq [\![A]\!]_{[X\leftarrow\Phi]}$; a failure of monotonicity.

Instead, in our semantics $\text{И}x.A$ is a monotonic operator (cf. Theorem 4.18 (3,4)), the functional associated with a fixpoint formula $\boldsymbol{\nu}X.A$ is in fact a closed monotonic operator $\mathbb{P}_M \to \mathbb{P}_M$ (cf. Lemma 4.21). To take the fixpoint of the map $\boldsymbol{\lambda}\Phi.[\![A]\!]_{v[X\leftarrow\Phi]}$ we must pick a lattice $\mathbb{P}_M$ where M is some finite set of names. This is because the set of all Psets is not a complete lattice and we find it convenient to use the general result of Knaster-Tarski to solve the recursion. Then a question arises about what M should we pick.

Note that for each Pset Φ such that $\Phi \subseteq [\![A]\!]_{v[X\leftarrow\Phi]}$ there is a possibly different $\mathit{supp}([\![A]\!]_{v[X\leftarrow\Phi]})$, depending on $\mathit{supp}(\Phi)$. In principle, we could pick $M = \mathit{supp}([\![\boldsymbol{\nu}X.A]\!]_v)$, but $\mathit{supp}([\![\boldsymbol{\nu}X.A]\!]_v)$ is not known in advance. Now, by Lemma 4.19, $\mathit{fn}^v(\boldsymbol{\nu}X.A)$ always contains the support of $[\![\boldsymbol{\nu}X.A]\!]_v$, so $\mathit{fn}^v(\boldsymbol{\nu}X.A)$

can be used to approximate $\mathit{supp}([\![\boldsymbol{\nu} X.A]\!]_v)$. In fact, on the one hand we cannot pick M such that M does not contain $\mathit{fn}^v(\boldsymbol{\nu} X.A)$: as discussed above, there are formulas A for which $\boldsymbol{\lambda}\Phi.[\![A]\!]_{v[X\leftarrow\Phi]}$ is not a closed operator in $\mathbb{P}_M$. So M must contain $\mathit{fn}^v(\boldsymbol{\nu} X.A)$. On the other hand, Lemma 4.25 states that we can set exactly $M = \mathit{fn}^v(\boldsymbol{\nu} X.A)$.

Now, the uniform relation between the renamings out of M of the pre-fixpoints of $\boldsymbol{\lambda}\Phi.[\![A]\!]_{v[X\leftarrow\Phi]}$ made clear by Lemma 4.24 has as immediate consequence: although $\mathbb{P}_-$ is not closed under arbitrary unions, the union of all the pre-fixpoints of $\boldsymbol{\lambda}\Phi.[\![A]\!]_{v[X\leftarrow\Phi]}$ is indeed finitely supported (cf. Lemma 4.27). Hence, the semantic definition for $\boldsymbol{\nu} X.A$ in Fig. 4 just makes explicit that the support of the greatest fixpoint of the monotonic map $\boldsymbol{\lambda}\Phi.[\![A]\!]_{v[X\leftarrow\Phi]}$ is always to be found inside $\mathit{fn}^v(\boldsymbol{\nu} X.A)$.

5 Validity

We now introduce a notion of logical validity. A formula A is valid if all of its ground instances, under all valuations, is satisfied by all processes.

Definition 5.1 (Valid Formula). *A formula A is* valid *if for all substitutions θ with $\mathit{fv}(A) \subseteq \mathfrak{D}(\theta)$, for all valuations v such that $\mathit{fpv}(A) \subseteq \mathfrak{D}(v)$, we have $[\![\theta(A)]\!]_v = \mathcal{P}$.*

We will use the meta-level statement $\mathit{valid}(A)$ to assert validity of formula A. Logical validity satisfies the following general principles.

Proposition 5.2 (Instantiation). *Let $F\{-\}$ be any formula context. We have*

1. *For any η and formula A, $\mathit{valid}(A) \Rightarrow \mathit{valid}(A\{x\leftarrow\eta\})$.*
2. *For any formula A, $\mathit{valid}(F\{X\}) \Rightarrow \mathit{valid}(F\{A\})$*

Proof. 1. Assume $\mathit{valid}(A)$. Then for all substitutions θ where $\mathit{fv}(A) \subseteq \mathfrak{D}(\theta)$, for all valuations v such that $\mathit{fpv}(A) \subseteq \mathfrak{D}(v)$, we have $\mathcal{P} \subseteq [\![\theta(A)]\!]_v$. Let θ' be any substitution with $\mathfrak{D}(\theta') \subseteq \mathit{fv}(A\{x\leftarrow\eta\})$ and define $\sigma = \theta'_{-x} \circ \{x\leftarrow\theta'(\eta)\}$. Now, $\mathfrak{D}(\sigma) \subseteq \mathit{fv}(A)$. Thus $\mathcal{P} \subseteq [\![\sigma(A)]\!]_v$ for any appropriate valuation v. Since $\sigma(A) = \theta'(A\{x\leftarrow\eta\})$, we are done.

2. Similar to proof on Lemma 5.3 (induction in the size of $F\{-\}$).

Lemma 5.3 (Substitutivity). *Let $[\![\theta(A)]\!]_v = [\![\theta(B)]\!]_v$ for all substitutions θ and valuations v, and let $F\{-\}$ be a formula context. Then, for all substitutions σ and valuations w we have $[\![\sigma(F\{A\})]\!]_w = [\![\sigma(F\{B\})]\!]_w$.*

A direct consequence of substitutivity is

Proposition 5.4 (Replacement of equivalents). *Let $F\{-\}$ be any formula context. We have $\mathit{valid}(A \Leftrightarrow B) \Rightarrow \mathit{valid}(F\{A\} \Leftrightarrow F\{B\})$.*

Proof. Assume $\mathit{valid}(A \Leftrightarrow B)$. Then $[\![\theta(A)]\!]_v = [\![\theta(B)]\!]_v$ for any valuation v for $A \Leftrightarrow B$ and substitution θ. Let w be any valuation for $F\{A\} \Leftrightarrow F\{B\}$; we must show that $[\![\sigma(F\{A\})]\!]_w = [\![\sigma(F\{B\})]\!]_w$, for any substitution σ. But this follows directly from Lemma 5.3.

Proposition 5.5 (Coinduction). *Let the formula F be monotonic in X. For any formula A such that $X \notin fpv(A)$*

$$valid(A \Rightarrow F\{X \leftarrow A\}) \Rightarrow valid(A \Rightarrow \boldsymbol{\nu} X.F)$$

Proof. We assume $valid(A \Rightarrow F\{X \leftarrow A\})$, and prove $valid(A \Rightarrow \boldsymbol{\nu} X.F)$. To that end, we select any valuation v for $A \Rightarrow \boldsymbol{\nu} X.F$, any substitution θ and show $[\![\theta(A)]\!]_v \subseteq [\![\theta(\boldsymbol{\nu} X.F)]\!]_v$. We have

$$[\![\theta(A)]\!]_v \subseteq [\![\theta(F\{X \leftarrow A\}))]\!]_v = [\![\theta(F)\{X \leftarrow \theta(A)\}]\!]_v$$

By Lemma 4.23, $[\![\theta(F)\{X \leftarrow \theta(A)\}]\!]_v = [\![\theta(F)]\!]_{v[X \leftarrow [\![\theta(A)]\!]_v]}$.

Let $M = fn^v(\theta(F\{X \leftarrow A\})) \supseteq fn^v(\theta(\boldsymbol{\nu} X.F))$. By Lemma 4.21, the mapping

$$\mathfrak{G} \triangleq \boldsymbol{\lambda} \Psi.[\![\theta(F)]\!]_{v[X \leftarrow \Psi]} : \mathbb{P}_M \to \mathbb{P}_M$$

is a monotonic operator in the complete lattice $\mathbb{P}_M$. Moreover, we have $[\![\theta(A)]\!]_v \subseteq [\![\theta(F)]\!]_{v[X \leftarrow [\![\theta(A)]\!]_v]}$. Thus $[\![\theta(A)]\!]_v \subseteq \mathsf{gfix}_{\mathbb{P}_M}(\mathfrak{G})$. By Lemma 4.25, $\mathsf{gfix}_{\mathbb{P}_M}(\mathfrak{G}) \subseteq [\![\theta(\boldsymbol{\nu} X.F)]\!]_v$. Hence, $[\![\theta(A)]\!]_v \subseteq [\![\theta(\boldsymbol{\nu} X.F)]\!]_v$.

6 Fresh and Hidden Name Quantification

In this section the semantics of Section 4 is used to investigate basic properties of the fresh name quantifier and of the derived hidden name quantifier.

6.1 The Fresh Name Quantifier

As we have seen, freshness plays a central role in the spatial logic, but uses of the freshness quantifier $\mathsf{И} x.A$ can be rather subtle. Consider, as an example, the formula $\mathsf{И} x.x\langle m\rangle$, satisfied by any process P such that, for any n fresh in P and different from m, P satisfies $n\langle m\rangle$. But if P satisfies $n\langle m\rangle$, it must be congruent to $n\langle m\rangle$, and hence it must contain n. Therefore, n is not fresh in P, and the denotation of $\mathsf{И} x.x\langle m\rangle$ is empty. This shows that many simple uses of $\mathsf{И}$ are trivial, when the fresh name maps directly to a free name of the process.

There are, however, two basic ways of making good use of the fresh quantifier. The first way is to use $\mathsf{И}$ in conjunction with ®, so that the fresh name is used to reveal a restricted name of the process (then the fresh name does not map to a free name of the original process). In this situation, we definitely do not want the name used to reveal a restricted name to clash with some other name of the process. This is one of the reasons that motivates the use of $\backslash$©n in the semantics of $\mathsf{И} x.A$ (Fig. 4), to eliminate such a possibility. The combination of $\mathsf{И}$ and ® is discussed further in Section 6.2.

The second way is to use $\mathsf{И}$ in conjunction with $\triangleright$, so that the fresh name maps to a free name of the context, but not of the process. For example, consider the formula

$$\mathsf{И} x.\forall y.(x\langle y\rangle \triangleright \boxdot(x\langle y\rangle | \mathtt{T}))$$

This formula holds of all processes P that verify the following: if a message on a fresh channel x is composed in parallel with P, then no reduction from the resulting process consumes such a message.

Intuitively, we expect such a property to hold of every process. In fact, let P be any process, n some name not free in P, and m any name. Pick any process Q such that $Q \models_v n\langle m\rangle$. So, $Q \equiv n\langle m\rangle$. Now, we verify that if $Q|P \to R$, then $R \equiv n\langle m\rangle|P'$, where $P \to P'$, because $P \not\equiv n(q).R'|R''$. Thus $P \models_v n\langle m\rangle \triangleright \boxdot(n\langle m\rangle|\mathtt{T})$. Since m is arbitrary, $P \models_v \forall y.n\langle y\rangle \triangleright \boxdot(n\langle y\rangle|\mathtt{T})$. Since n is neither free in P nor belongs to $fn^v(\forall y.x\langle y\rangle \triangleright \boxdot(x\langle y\rangle|\mathtt{T}))$, we conclude $P \models_v \text{И}x.\forall y.x\langle y\rangle \triangleright \boxdot(x\langle y\rangle|\mathtt{T})$.

A fundamental consequence of closure of satisfaction under fresh renaming (Lemma 4.28) is the following characterisation of fresh name quantification, that makes clear the universal/existential ambivalence of freshness: if some property holds of a fresh name, it holds of all fresh names.

Proposition 6.1 (Gabbay-Pitts Property). *Let $\text{И}x.A$ be a name-closed formula, P a process, and v a valuation for $\text{И}x.A$. Then, the following statements are equivalent*

1. $P \models_v \text{И}x.A$.
2. *There is* $n \notin fn(P) \cup fn^v(A)$ *such that* $P \models_v A\{x\leftarrow n\}$.
3. *For all* $n \notin fn(P) \cup fn^v(A)$ *we have* $P \models_v A\{x\leftarrow n\}$.

Proof. ($1 \Rightarrow 2$) By definition. ($2 \Rightarrow 3$) By Remark 4.16, there is n such that $n \notin fn(P) \cup fn^{v_A}(A)$ and $P \models_{v_A} A\{x\leftarrow n\}$, where v_A is the restriction of v to the free propositional variables of A. Now, pick $m \notin fn(P) \cup fn^{v_A}(A)$, and let $\tau = \{m\leftrightarrow n\}$. By Theorem 4.18(2), we conclude $\tau(P) \models_{\tau(v_A)} \tau(A\{x\leftarrow n\})$, that is, $P \models_{\tau(v_A)} A\{x\leftarrow m\}$. Note that $m, n \notin fn(v_A)$; hence $P \models_{v_A} A\{x\leftarrow m\}$, by Lemma 4.17(1). Hence $P \models_v A\{x\leftarrow m\}$, by Remark 4.16. ($3 \Rightarrow 1$) Immediate.

A corollary of the previous proposition is

Proposition 6.2. *Let A be a name-closed formula and v a valuation for A and B. We have*

1. $[\![\forall x.A]\!]_v \subseteq [\![\text{И}x.A]\!]_v \subseteq [\![\exists x.A]\!]_v$
2. $[\![\text{И}x.(A \Rightarrow B)]\!]_v = [\![\text{И}x.A \Rightarrow \text{И}x.B]\!]_v$

Proof. 2. (Left to right) Assume $P \in [\![\text{И}x.(A \Rightarrow B)]\!]_v$ and $P \in [\![\text{И}x.A]\!]_v$. Then $P \in [\![A\{x\leftarrow n\}]\!]_v$ for some $n \notin fn(P) \cup fn^v(A)$, and $P \in [\![A\{x\leftarrow m\} \Rightarrow B\{x\leftarrow m\}]\!]_v$ for some $m \notin fn(P) \cup fn^v(A \Rightarrow B)$. By Proposition 6.1(3), for all $n \notin fn(P) \cup fn^v(A)$ we have $P \in [\![A\{x\leftarrow n\}]\!]_v$. In particular, $P \in [\![A\{x\leftarrow m\}]\!]_v$, thus $P \in [\![B\{x\leftarrow m\}]\!]_v$. We conclude $P \in [\![\text{И}x.B]\!]_v$. (Right to left) Assume $P \in [\![\text{И}x.A \Rightarrow \text{И}x.B]\!]_v$. Pick $m \notin fn(P) \cup fn^v(A \Rightarrow B)$ and assume $P \in [\![A\{x\leftarrow m\}]\!]_v$. Then $P \in [\![\text{И}x.A]\!]_v$, this implies $P \in [\![\text{И}x.B]\!]_v$. By Proposition 6.1(3), $P \in [\![B\{x\leftarrow m\}]\!]_v$. Hence $P \in [\![\text{И}x.(A \Rightarrow B)]\!]_v$.

We can verify that fresh quantification distributes over all boolean connectives, not only implication (cf. Proposition 6.2(2), it suffices to note that (trivially) $[\![\text{И}x.\mathtt{F}]\!]_v = [\![\mathtt{F}]\!]_v$). In the next lemma, we exemplify some other interesting properties of fresh name quantification.

Lemma 6.3 (Some properties of И). *We have*

1. $[\![\text{И}x.(A|B)]\!]_v = [\![\text{И}x.A|\text{И}x.B]\!]_v$
2. $[\![\text{И}x.\Diamond A]\!]_v = [\![\Diamond\text{И}x.A]\!]_v$

Proof. 1. (Right to left inclusion) Let $P \models_v \text{И}x.A|\text{И}x.B$. Then there are processes Q and R such that $P \equiv Q|R$ and $Q \models_v \text{И}x.A$ and $R \models_v \text{И}x.B$. Pick a name $n \notin fn(P) \cup fn^v(\text{И}x.(A|B))$. Then $n \notin fn(Q) \cup fn^v(\text{И}x.A)$ and $n \notin fn(R) \cup fn^v(\text{И}x.B)$. By Proposition 6.1(3) we have $Q \models_v A\{x{\leftarrow}n\}$ and $R \models_v B\{x{\leftarrow}n\}$. Then $P \models_v \text{И}x.(A|B)$ as claimed. (Left to right inclusion) Similar, use Proposition 6.1(2).

2. (Left to right inclusion) Let $P \models_v \text{И}x.\Diamond A$. Then there is a process Q, and a name n, fresh w.r.t. P and $fn^v(A)$ such that $P \rightarrow Q$ and $Q \models_v A\{x{\leftarrow}n\}$. But if n is fresh w.r.t P it is also fresh w.r.t. Q since $fn(Q) \subseteq fn(P)$. Hence $Q \models_v \text{И}x.A$ and $P \models_v \Diamond\text{И}x.A$. (Right to left inclusion) Let $P \models_v \Diamond\text{И}x.A$. Then there is Q and n fresh w.r.t. Q and $fn^v(A)$ such that $P \rightarrow Q$ and $Q \models_v A\{x{\leftarrow}n\}$. Pick any name m, fresh w.r.t. v and P (and thus fresh w.r.t. Q). Let $\tau = \{m{\leftrightarrow}n\}$, by Theorem 4.18(2) we have $Q \models_v A\{x{\leftarrow}m\}$, since $\tau(Q) = Q$, $\tau(v) = v$ and $\tau(A\{x{\leftarrow}n\}) = A\{x{\leftarrow}m\}$. Hence $P \models_v \Diamond A\{x{\leftarrow}m\}$, and then $P \models_v \text{И}x.\Diamond A$.

In [11] a И-quantifier for FM-set theory is defined, such that

$$\text{И}x.A \Leftrightarrow \{n \mid A\{x{\leftarrow}n\}\} \text{ is cofinite}$$

There is a quite close connection between this И-quantifier and ours, superficial differences being related to the fact that we are working in a modal logic. In our case, we have

Proposition 6.4. *$P \models_v \text{И}x.A$ if and only if $\{n \mid P \models_v A\{x{\leftarrow}n\}\}$ is cofinite.*

Proof. (Left to right) Pick $P \models_v \text{И}x.A$. Thus there is $n \notin fn^v(A) \cup fv(P)$ such that $P \models_v A\{x{\leftarrow}n\}$. By Gabbay-Pitts (Proposition 6.1(3)) we have that for all n if $n \notin fn^v(A) \cup fv(P)$ then $P \models_v A\{x{\leftarrow}n\}$. Hence $\{n \mid P \models_v A\{x{\leftarrow}n\}\}$ is cofinite. (Right to left) Assume $S = \{n \mid P \models_v A\{x{\leftarrow}n\}\}$ is cofinite. Then, there is a finite set $M(= \Lambda \setminus S)$ such that for all n, if $n \notin M$ then $P \models_v A\{x{\leftarrow}n\}$. Pick $m \notin fn^v(A) \cup fn(P) \cup M$. Then $P \models_v A\{x{\leftarrow}m\}$, hence $P \models_v \text{И}x.A$.

Now, let us define the following (meta-level) quantifier

$$\text{И}^*x.\mathcal{B} \triangleq \{n \in \Lambda \mid \mathcal{B}\{x{\leftarrow}n\}\} \text{ is cofinite}$$

where $\mathcal{B}$ is a meta-level statement of the (informal) theory of Psets of Section 4. Note that $\text{И}^*x.\mathcal{B}$ is defined exactly as the И-quantifier of Gabbay-Pitts. Then, we can read the statement of the previous proposition as:

$$P \in [\![\text{И}x.A]\!]_v \text{ if and only if } \text{И}^*n.P \in [\![A\{x{\leftarrow}n\}]\!]_v.$$

It is interesting to discuss alternative fresh quantifiers. Our semantics of $\text{И}x.A$ is such that $P \models_v \text{И}x.A$ holds if and only if there is a name n, fresh both in A and P, such that $P \models A\{x \leftarrow n\}$ (cf. Proposition 6.1). It is natural then to ask what happens if n only is required to be fresh in A. Let us define for this propose a different quantifier $\mathsf{F}x.A$ where

$$P \in [\![\mathsf{F}x.A]\!] \text{ if and only if } \exists n \notin fn(A) \text{ such that } P \in [\![A\{x \leftarrow n\}]\!]$$

One could then attempt to define $\text{И}x.A$ as $\mathsf{F}x.(A \wedge \neg \copyright x)$. Although $[\![\mathsf{F}x.A]\!]$ is a Pset, the main problems with $\mathsf{F}x.A$, with respect to $\text{И}x.A$, are a failure of monotonicity (Theorem 4.18 (3)), a failure of the substitutivity property (Lemma 5.3), and a failure of the Gabbay-Pitts property (Proposition 6.1) relating to a proper notion of "freshness".

For substitutivity, we have that $[\![n\langle n\rangle \vee \neg n\langle n\rangle]\!] = [\![\mathtt{T}]\!]$. So, we would expect that $[\![\mathsf{F}x.((n\langle n\rangle \vee \neg n\langle n\rangle) \wedge x\langle x\rangle)]\!] = [\![\mathsf{F}x.(\mathtt{T} \wedge x\langle x\rangle)]\!]$. But $n\langle n\rangle \in [\![\mathsf{F}x.(\mathtt{T} \wedge x\langle x\rangle)]\!]$, while $n\langle n\rangle \notin [\![\mathsf{F}x.((n\langle n\rangle \vee \neg n\langle n\rangle) \wedge x\langle x\rangle)]\!]$. So, $\mathsf{F}x.A$ is not a proper "compositional" logical operator. While, rather amazingly, $\text{И}x.A$ is.

For monotonicity, consider

$$\psi = \{q\langle q\rangle\}^{\equiv} \subseteq \{p\langle p\rangle, q\langle q\rangle\}^{\equiv} = \phi$$

Note that $\psi, \phi \in \mathbb{P}_{\{p,q\}}$, $fn^{v[X \leftarrow \psi]}(X|x\langle x\rangle) = \{q\}$ and $fn^{v[X \leftarrow \phi]}(X|x\langle x\rangle) = \{p, q\}$. On the one hand, $q\langle q\rangle|p\langle p\rangle \in [\![\mathsf{F}x.X|x\langle x\rangle]\!]_{v[X \leftarrow \psi]}$, because there is $n \notin \{q\}$ (namely p) such that $q\langle q\rangle|p\langle p\rangle \in [\![X|n\langle n\rangle]\!]_{v[X \leftarrow \psi]}$. On the other hand, we have $q\langle q\rangle|p\langle p\rangle \notin [\![\mathsf{F}x.(X|x\langle x\rangle)]\!]_{v[X \leftarrow \Phi]}$, because there is no n out of $\{p, q\}$ such that $q\langle q\rangle|p\langle p\rangle \in [\![X|n\langle n\rangle]\!]_{v[X \leftarrow \Phi]}$. So $[\![\mathsf{F}x.(X|x\langle x\rangle)]\!]_{v[X \leftarrow \psi]} \not\subseteq [\![\mathsf{F}x.(X|x\langle x\rangle)]\!]_{v[X \leftarrow \phi]}$. We conclude that $\mathsf{F}x.A$ cannot be used with recursive formulas.

For the Gabbay-Pitts property, consider whether $p\langle p\rangle \in [\![\mathsf{F}x.\neg x\langle x\rangle]\!]$. This means, by definition: there is a name n such that $p\langle p\rangle \in [\![\neg n\langle n\rangle]\!]$. This is true, take any $n \neq p$. If we had a Gabbay-Pitts property for $\mathsf{F}x.A$ we would obtain that for all names n, $p\langle p\rangle \in [\![n\langle n\rangle]\!]$. But this is false: take $n = p$. So, by the interpretation of the Gabbay-Pitts property, the candidate $\mathsf{F}x.A$ is not a proper "freshness" quantifier.

We may also consider an apartness connective (not a quantifier) $\eta\#A$ with semantics given by

$$P \in [\![n\#A]\!]_v \text{ if and only if } n \notin fn^v(A)$$

and then attempt to define fresh name quantification from $\copyright$ and $\exists$. Indeed, we can verify that $[\![\text{И}x.A]\!]_v = [\![\exists x.(\neg\copyright x \wedge x\#\exists x.A \wedge A)]\!]_v$. But substitutivity also fails for $x\#A$, eg. $\mathcal{P} = [\![n\#\mathtt{T}]\!]_v \neq [\![n\#(n\langle n\rangle \vee \neg n\langle n\rangle)]\!]_v = \emptyset$, hence it is not sensible to adopt $\eta\#A$ as a primitive formula.

6.2 The Hidden Name Quantifier

When combined with revelation, the fresh name quantifier gives rise to a natural operation of quantification over hidden (restricted) names in a process.

Intuitively, a name is revealed under a fresh identity, and then a property is asserted of the restricted process.

$$\mathsf{H}x.A \triangleq \mathsf{И}x.x\circledR A$$

A formula $\mathsf{H}x.A$ reads "there is a restricted name x such that A holds for the process under the restriction". From the above definition, we get the following direct semantic characterization of the name-closed formula $\mathsf{H}x.A$

$$[\![\mathsf{H}x.A]\!]_v = \{Q \mid Q \equiv (\boldsymbol{\nu}n)P \text{ and } n \notin \mathit{fn}(Q) \cup \mathit{fn}^v(A) \text{ and } P \in [\![A\{x\leftarrow n\}]\!]_v\}$$

The hidden name quantifier makes it possible to express properties of processes that depend on (or need to mention) some secret name. For a quite simple example, consider the closed formula $\exists y.\mathsf{H}x.(y\langle x\rangle|\mathtt{T})$.

This formula holds precisely of those processes which are ready to send a secret name over a public channel. As a further example, let A be some formula with a free occurrence of the name variable x, and consider

$$\mathit{Keeps}(A, x) \triangleq \mathsf{H}x.A \wedge (\mathtt{T} \triangleright \boxdot\mathsf{H}x.A|\mathtt{T})$$

A process that satisfies $\mathit{Keeps}(A, x)$ is always able to guarantee, until the next step, persistence of property A w.r.t. some secret x it owns, even when attacked by some other arbitrary process. Let K be the process

$$(\boldsymbol{\nu}m)(m\langle n\rangle|!a(p).m(q).m\langle p\rangle)$$

We have $\mathit{fn}(K) = \{a, n\}$. Now define $\mathit{Msg}(x, y) \triangleq (x\langle y\rangle|\mathtt{T})$. We can verify that K satisfies $\mathit{Keeps}(\exists y.\mathit{Msg}(x, y), x)$.

Lemma 6.5 (Some properties of H). *We have*

1. $[\![\mathsf{H}x.(A \wedge \neg\copyright x)]\!]_v = [\![\mathsf{И}x.A]\!]_v$
2. $[\![\mathsf{H}x.\Diamond A]\!]_v = [\![\Diamond\mathsf{H}x.A]\!]_v$

Proof. 1. (Left to right inclusion) Pick some process $P \in [\![\mathsf{H}x.(A \wedge \neg\copyright x)]\!]_v$. By the characterization given above, this means that $P \equiv (\boldsymbol{\nu}n)Q$ for some Q and n such that $n \notin \mathit{fn}(P) \cup \mathit{fn}^v(A)$ and $Q \in [\![A\{x\leftarrow n\} \wedge \copyright n]\!]_v$. But then, $n \notin \mathit{fn}(Q)$ and $Q \in [\![A\{x\leftarrow n\}]\!]_v$. We conclude $P \equiv (\boldsymbol{\nu}n)Q \equiv Q$, by Proposition 2.12(2). Therefore, $P \in [\![\mathsf{И}x.A]\!]_v$. In the other direction the proof is similar.

2. (Left to right inclusion) Pick some process $P \in [\![\mathsf{H}x.\Diamond A]\!]_v$. So, $P \equiv (\boldsymbol{\nu}n)Q$ for some Q and n such that $Q \to Q'$ and $Q' \in [\![A\{x\leftarrow n\}]\!]_v$, where n is fresh w.r.t. P and A. But then $P \to (\boldsymbol{\nu}n)Q'$. Since n is also fresh w.r.t. $(\boldsymbol{\nu}n)Q'$, we conclude $(\boldsymbol{\nu}n)Q' \in [\![\mathsf{H}x.A]\!]_v$. Hence $P \in [\![\Diamond\mathsf{H}x.A]\!]_v$. (Right to left inclusion) Take some process $P \in [\![\Diamond\mathsf{H}x.A]\!]_v$. Then there is Q such that $P \to Q$ and $Q \in [\![\mathsf{H}x.A]\!]_v$. Then $Q \equiv (\boldsymbol{\nu}n)R$ where $R \in [\![A\{x\leftarrow n\}]\!]_v$ and $n \notin \mathit{fn}(Q) \cup \mathit{fn}^v(A)$. Now, since $P \to Q$, by Proposition 2.15(3) there are P' and R' such that $P \equiv (\boldsymbol{\nu}n)P'$, $P' \to R'$ and $R' \equiv R$. This means that $P \in [\![\mathsf{H}x.\Diamond A]\!]$.

7 Recursive Definitions

The possibility of defining spatial, temporal and freshness properties by induction and coinduction is a major source of expressiveness of the spatial logic. For instance, by combining fixpoints formulas with hidden name quantification we can describe a "nonce generator", that is, a process that sends an unbounded number of fresh names on a channel:

$$\boldsymbol{\nu} X.\mathsf{H}x.m\langle x\rangle|X$$

We can verify that $!(\boldsymbol{\nu} n)m\langle n\rangle \models \boldsymbol{\nu} X.\mathsf{H}x.m\langle x\rangle|X$. Let $\Phi = \{P \mid P \equiv !(\boldsymbol{\nu} n)m\langle n\rangle\}$. We have that $\Phi \in \mathbb{P}_{\{m\}}$. It suffices to check that $\Phi \subseteq [\![\mathsf{H}x.m\langle x\rangle|X]\!]_{[X\leftarrow\Phi]}$. For this, take any $P \in \Phi$, so that we have $P \equiv !(\boldsymbol{\nu} n)m\langle n\rangle \equiv (\boldsymbol{\nu} n)m\langle n\rangle|!(\boldsymbol{\nu} n)m\langle n\rangle$. The left subprocess is in $[\![\mathsf{H}x.m\langle x\rangle]\!]_{[X\leftarrow\Phi]}$, and the right one is in $\Phi = [\![X]\!]_{[X\leftarrow\Phi]}$. More generally, we can verify that $P \models A$ implies $!P \models \boldsymbol{\nu} X.(A|X)$.

The least fixpoint operator is defined from the greatest fixpoint operator in the standard way. From these, the standard "always in the future" $\Box A$ and "eventually in the future" $\Diamond A$ modalities of (branching time) temporal logic are also defined.

$$\begin{array}{lll}
\boldsymbol{\mu} X.A & \triangleq \neg\boldsymbol{\nu} X.\neg A\{X\leftarrow\neg X\} & \text{(Least fixpoint)}\\
\Box A & \triangleq \boldsymbol{\nu} X.(A \wedge \boxdot X) & \text{(Always)}\\
\Diamond A & \triangleq \boldsymbol{\mu} X.(A \vee \diamondsuit X) & \text{(Eventually)}
\end{array}$$

Proposition 7.1. *We have*

1. $[\![\boldsymbol{\mu} X.A]\!]_v = \mathsf{lfix}_{\mathbb{P}_N}(\boldsymbol{\lambda}\Psi.[\![A]\!]_{v[X\leftarrow\Psi]})$ *where* $N = fn^v(\mu X.A)$
2. $P \in [\![\Box A]\!]_v$ *if and only if for all* Q *such that* $P \xrightarrow{*} Q$ *we have* $Q \in [\![A]\!]_v$
3. $P \in [\![\Diamond A]\!]_v$ *if and only if there is* Q *such that* $P \xrightarrow{*} Q$ *and* $Q \in [\![A]\!]_v$

Proposition 7.2 (Induction). *Let the formula F be monotonic in X. For any formula A such that $X \notin fpv(A)$*

$$valid(F\{X\leftarrow A\} \Rightarrow A) \Rightarrow valid(\boldsymbol{\mu} X.F \Rightarrow A)$$

By combining fixpoints with spatial and temporal connectives one can define many other interesting properties. For instance, let

$$\begin{array}{lll}
\textbf{\textit{group}}\ A & \triangleq \boldsymbol{\mu} X.(\mathbf{0} \vee A|X) & \text{(Group)}\\
\textbf{\textit{inside}}^{\exists}\ A & \triangleq \boldsymbol{\mu} X.(A \vee \mathsf{H}x.X) & \\
\textbf{\textit{inside}}^{\forall}\ A & \triangleq \neg\ \textbf{\textit{inside}}^{\exists}\ \neg A & \\
\diamondsuit A & \triangleq \textbf{\textit{inside}}^{\exists}(A|\mathtt{T}) & \text{(Somewhere)}\\
\boxtimes A & \triangleq \textbf{\textit{inside}}^{\forall}(A||\mathtt{F}) & \text{(Everywhere)}
\end{array}$$

The formula ***group*** A holds of any process P such that $P \equiv Q_1|\ldots|Q_k$ for some processes $Q_1,\ldots,Q_k$, where each Q_i satisfies A. The formula ***inside*** $^{\exists}A$ holds of any process P such that $P \equiv (\boldsymbol{\nu} n_1)\cdots(\boldsymbol{\nu} n_k)Q$ for some fresh names $n_1,\ldots,n_k$ and process Q, where Q satisfies A.

Lemma 7.3. *We have*

$$[\![\boldsymbol{\mu} X.(A \vee \mathsf{H}x.X)]\!]_v = \{(\boldsymbol{\nu} n_1)\cdots(\boldsymbol{\nu} n_k)Q \mid Q \in [\![A]\!]_v \textit{ and } n_i \notin fn^v(A)\}^{\equiv}$$

Proof. Let $\Psi \triangleq \{(\boldsymbol{\nu} n_1)\cdots(\boldsymbol{\nu} n_k)Q \mid Q \in [\![A]\!]_v$ and $n_i \notin fn^v(A)\}^{\equiv}$. Let $M = fn^v(A) = fn^v(\boldsymbol{inside}^{\,\exists} A)$; it is easy to check that $\Psi \in \mathbb{P}_M$. Ψ is closed under structural congruence by construction. Also, pick any process $P \in \Psi$, names $p, q \notin M$ and let $\tau = \{p \leftrightarrow q\}$. Then $P \equiv (\boldsymbol{\nu} n_1)\cdots(\boldsymbol{\nu} n_k)Q$ where $Q \in [\![A]\!]_v$. Hence $\tau(P) = (\boldsymbol{\nu}\tau(n_1))\cdots(\boldsymbol{\nu}\tau(n_k))\tau(Q)$, and we have $\tau(Q) \in [\![A]\!]_v$. We want to show that $\Psi = [\![\boldsymbol{\mu} X.(A \vee \mathsf{H}x.X)]\!]_v$.

We first prove $\Psi \subseteq [\![\boldsymbol{\mu} X.(A \vee \mathsf{H}x.X)]\!]_v$. To that end, we take any $\Phi \in \mathbb{P}_M$ such that $[\![A \vee \mathsf{H}x.X]\!]_{v[X \leftarrow \Phi]} \subseteq \Phi$ and verify $\Psi \subseteq \Phi$.

Pick $P \in \Psi$. Then $P \equiv (\boldsymbol{\nu} n_1)\cdots(\boldsymbol{\nu} n_k)Q$ and $Q \in [\![A]\!]_v$, for some $k \geq 0$. We show by induction on k that $P \in \Phi$.

If $k = 0$, we have $P \equiv Q \in [\![A]\!]_v$, and thus $P \in [\![A \vee \mathsf{H}x.X]\!]_{v[X \leftarrow \Phi]}$ by Remark 4.16, since X is not free in A. Hence $P \in \Phi$. If $k > 0$ we have $P \equiv (\boldsymbol{\nu} n_1)P'$ where $P' = (\boldsymbol{\nu} n_2)\cdots(\boldsymbol{\nu} n_k)Q \in \Psi$. By induction hypothesis, we have $P' \in \Phi$. Hence $P' \in [\![X]\!]_{v[X \leftarrow \Phi]}$. Thus $P \in [\![n_1 \circledR X]\!]_{v[X \leftarrow \Phi]}$. We have $n_1 \notin fn(P)$ and $n_1 \notin M = fn^{v[X \leftarrow \Phi]}(X) = fn^{v[X \leftarrow \Phi]}(\text{И}x.x \circledR X)$. Then $P \in [\![\mathsf{H}x.X]\!]_{v[X \leftarrow \Phi]}$, and so $P \in [\![A \vee \mathsf{H}x.X]\!]_{v[X \leftarrow \Phi]}$. We conclude $P \in \Phi$, also in this case. Hence $\Psi \subseteq [\![\boldsymbol{\mu} X.(A \vee \mathsf{H}x.X)]\!]_v$.

Finally, to show $[\![\boldsymbol{\mu} X.(A \vee \mathsf{H}x.X)]\!]_v \subseteq \Psi$, it suffices to verify the inclusion $[\![A \vee \mathsf{H}x.X]\!]_{v[X \leftarrow \Psi]} \subseteq \Psi$. If $P \in [\![A \vee \mathsf{H}x.X]\!]_{v[X \leftarrow \Psi]}$ then either $P \in [\![A]\!]_v$ and thus $P \in \Psi$, or $P \equiv (\boldsymbol{\nu} n)P'$ where $P' \in \Psi$ and $n \notin fn^v(A) \cup fn(P)$. In the last case, is also immediate that $P \in \Psi$. We conclude $\Psi = [\![\boldsymbol{\mu} X.(A \vee \mathsf{H}x.X)]\!]_v$.

A process P satisfies $\diamondsuit A$ if somewhere inside P, possibly under some restricted names, there is a component satisfying A. In a similar way, a process satisfies $\boldsymbol{inside}^{\forall}\ A$ if and only if for all names $n_1, \ldots, n_k$ fresh w.r.t. A, and processes Q such that $P \equiv (\boldsymbol{\nu} n_1)\cdots(\boldsymbol{\nu} n_k)Q$, Q satisfies A. A process P satisfies $\boxtimes A$ if and only if all components of P, regardless of the local name context in which they are placed, satisfy A. For example, a process P satisfies the closed formula $\boxtimes \neg\exists x.\exists y.x\langle y\rangle$ if and only if it contains no unguarded messages. Thus the formula $\boxtimes \neg\exists x.\exists y.x\langle y\rangle \Rightarrow \boxdot\mathtt{F}$, asserting that every process without unguarded messages is bound to inactivity, is logically valid.

For an example of a property making use of several of the spatial and temporal modalities consider

$$\begin{aligned}
&\mathit{Member}(x) \triangleq \copyright x \wedge \forall y.(y\langle x\rangle|\mathtt{T} \Rightarrow y = x)\\
&\mathit{Group} \triangleq \Box\mathsf{H}x.(\boldsymbol{group}\ \mathit{Member}(x))
\end{aligned}$$

A process satisfies *Group* if all of its future states will always be composed by a group of processes sharing a secret x, and such that each member of the group is unable to send x on a public channel.

8 Conclusions

We have investigated the satisfaction relation for a logic of concurrent processes that includes spatial operators, freshness quantifiers, and recursive formulas. In particular, we have shown how coinductively defined logical properties including freshness quantification can be given a semantics in terms of maximal fixpoints in a complete lattice of finitely supported sets of processes.

The logical rules arising from such a satisfaction relation will be investigated in Part II of this paper. Several interesting logical properties have already been discussed with respect to this model.

Some properties of the logic are very sensitive to the formulation of structural congruence (in fact, Sangiorgi has shown that the process equivalence induced by a similar logic is essentially structural congruence [17]). There is always a tension between embedding an expected process equivalence into the definition of structural congruence, or leaving it as a derived behavioral equivalence. Some desired logical properties have provided new insights into this delicate balance.

The general structure of our definitions can be easily adapted to various process calculi, and is also largely independent from the details of the operational semantics. Although the semantics considered here is based on unlabeled transition systems, it could be extended in a natural way to labeled transition systems, motivating then the introduction of Hennessy-Milner-like "labeled" modalities into the spatial logic. Nevertheless, some basic features of what a formula denotes, namely closure under structural congruence and finite support, should be expected to hold in all variations.

In conclusion, the general term of "spatial logic" has a fairly well defined, though informal, meaning. A spatial logic is a very intensional logic, that talks about fine details of process structure. This is what is required if we want to meaningfully describe the distribution of processes and the use of resources over a network.

Acknowledgments Andy Gordon contributed to the early stages of this paper. Thanks also to Giorgio Ghelli and Philippa Gardner for related discussions. Caires acknowledges the support of Microsoft Research for a visit to the Cambridge Lab during the time we worked on this paper.

References

1. G. Boudol. Asynchrony and the π-calculus (note). Rapport de Recherche 1702, INRIA Sofia-Antipolis, May 1992.
2. L. Caires. *A Model for Declarative Programming and Specification with Concurrency and Mobility.* PhD thesis, Departamento de Informática, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, 1999.
3. L. Caires and L. Monteiro. Verifiable and Executable Specifications of Concurrent Objects in $\mathcal{L}_\pi$. In C. Hankin, editor, *Programming Languages and Systems: Proceedings of the 7th European Symposium on Programming (ESOP 1998)*, number 1381 in Lecture Notes in Computer Science, pages 42–56. Springer-Verlag, 1998.

4. L. Cardelli and G. Ghelli. A Query Language Based on the Ambient Logic. In David Sands, editor, *Programming Languages and Systems: Proceedings of the 10th European Symposium on Programming (ESOP 2001)*, volume 2028 of *Lecture Notes in Computer Science*, pages 1–22. Springer-Verlag, 2001.
5. L. Cardelli and A. D. Gordon. Anytime, Anywhere. Modal Logics for Mobile Ambients. In *27th ACM Symposium on Principles of Programming Languages*, pages 365–377. ACM, 2000.
6. L. Cardelli and A. D. Gordon. Logical Properties of Name Restriction. In S. Abramsky, editor, *Typed Lambda Calculi and Applications*, number 2044 in Lecture Notes in Computer Science. Springer-Verlag, 2001.
7. W. Charatonik and J.-M. Talbot. The decidability of model checking mobile ambients. In *Proceedings of the 15th Annual Conference of the European Association for Computer Science Logic*, Lecture Notes in Computer Science. Springer-Verlag, 2001. To appear.
8. S. Dal-Zilio. Spatial Congruence for Ambients is Decidable. In *Proceedings of ASIAN'00 — 6th Asian Computing Science Conference*, number 1961 in Lecture Notes in Computer Science, pages 365–377. Springer-Verlag, 2000.
9. M. Dam. Relevance Logic and Concurrent Composition. In *Proceedings, Third Annual Symposium on Logic in Computer Science*, pages 178–185, Edinburgh, Scotland, 5–8 July 1988. IEEE Computer Society.
10. J. Engelfriet and Tj. Gelsema. A Multiset Semantics for the π-calculus with Replication. *Theoretical Computer Science*, (152):311–337, 1999.
11. M. J. Gabbay and A. M. Pitts. A New Approach to Abstract Syntax Involving Binders. In *14th Annual Symposium on Logic in Computer Science*, pages 214–224. IEEE Computer Society Press, Washington, 1999.
12. P. Gardner. From Process Calculi to Process Frameworks. In Catuscia Palamidessi, editor, *CONCUR 2000: Concurrency Theory (11th International Conference, University Park, PA, USA)*, volume 1877 of *Lecture Notes in Computer Science*, pages 69–88. Springer, August 2000.
13. M. Hennessy and R. Milner. Algebraic laws for Nondeterminism and Concurrency. *JACM*, 32(1):137–161, 1985.
14. K. Honda and M. Tokoro. On Asynchronous Communication Semantics. In M. Tokoro, O. Nierstrasz, and P. Wegner, editors, *Object-Based Concurrent Computing 1991*, number 612 in Lecture Notes in Computer Science, pages 21–51. Springer-Verlag, 1992.
15. D. Kozen. Results on the Propositional μ-Calculus. *TCS*, 27(3):333–354, 1983.
16. R. Milner. *Communicating and Mobile Systems: the π-calculus.* Cambridge University Press, 1999.
17. D. Sangiorgi. Extensionality and Intensionality of the Ambient Logics. In *28th Annual Symposium on Principles of Programming Languages*, pages 4–13. ACM, 2001.

Appendix (Proofs)

Lemma 4.2 For all finite $N \subseteq \Lambda$,

1. If $N \subseteq N'$ then $\mathbb{P}_N \subseteq \mathbb{P}_{N'}$.
2. (Bottom and Top) $\emptyset \in \mathbb{P}_N$ and $\mathcal{P} \in \mathbb{P}_N$.
3. (Meet and Join) If $S \subseteq \mathbb{P}_N$ then $\bigcap S \in \mathbb{P}_N$ and $\bigcup S \in \mathbb{P}_N$.
4. (Inverse) If $\Psi \in \mathbb{P}_N$ then $\overline{\Psi} = \mathcal{P} \setminus \Psi \in \mathbb{P}_N$.

Proof. 1. Pick $\Psi \in \mathbb{P}_N$. Now, Ψ is closed under $\equiv$; we must just verify that Ψ is supported by N'. Pick $m, n \notin N'$. Since $N \subseteq N'$, we have $m, n \notin N$. Therefore, since $\Psi \in \mathbb{P}_N$, for all $P \in \Psi$, we have $P\{m \leftrightarrow n\} \in \Psi$. Thus $\Psi \in \mathbb{P}_{N'}$.

2. Immediate.
3. Let $S \subseteq \mathbb{P}_N$.

   (a) Pick $P \in \bigcap S$. Then, for all $\Psi \in S$, we have $P \in \Psi$ and $\Psi \in \mathbb{P}_N$. Now, if $Q \equiv P$, then $Q \in \Psi$, for all $\Psi \in S$. Thus $Q \in \bigcap S$. Now let $m, n \notin N$. We have $P\{m \leftrightarrow n\} \in \Psi$, for all $\Psi \in S$. Hence $P\{m \leftrightarrow n\} \in \bigcap S$. We conclude $\bigcap S \in \mathbb{P}_N$.
4. Pick $P \in \bigcup S$. Then, there is $\Psi \in S$ such that $P \in \Psi$ and $\Psi \in \mathbb{P}_N$. Thus, if $Q \equiv P$, then $Q \in \Psi \subseteq \bigcup S$. Now let $m, n \notin N$. Then, $P\{m \leftrightarrow n\} \in \Psi \subseteq \bigcup S$. Hence $\bigcup S \in \mathbb{P}_N$.
5. Assume $\Psi \in \mathbb{P}_N$. Pick $P \in \overline{\Psi}$. Let $Q \equiv P$ and suppose $Q \notin \overline{\Psi}$. Then $Q \in \Psi$ and $P \in \Psi$, contradiction. Hence $Q \in \overline{\Psi}$. Likewise, pick $m, n \notin N$ and suppose $P\{m \leftrightarrow n\} \notin \overline{\Psi}$. Then $P\{m \leftrightarrow n\} \in \Psi$, and this implies $P \in \Psi$, a contradiction. Hence we must have $P\{m \leftrightarrow n\} \in \overline{\Psi}$. We conclude $\overline{\Psi} \in \mathbb{P}_N$.

Lemma 4.7 We have

1. For any process P and Pset Ψ, $P \in \tau(\Psi)$ if and only if $\tau(P) \in \Psi$.
2. $\Psi \in \mathbb{P}_N$ if and only if $\tau(\Psi) \in \mathbb{P}_{\tau(N)}$.
3. If $m, n \notin N$ and $\Psi \in \mathbb{P}_N$ then $\{m \leftrightarrow n\}(\Psi) = \Psi$.

Proof. 1. Let $P \in \tau(\Psi)$. Then there is $Q \in \Psi$ with $\tau(Q) = P$. Thus $Q \equiv \tau(P)$. Since Ψ is closed under $\equiv$, $\tau(P) \in \Psi$.

2. Assume $\Psi \in \mathbb{P}_N$. Pick $P \in \tau(\Psi)$. We have $P = \tau(Q)$ for some $Q \in \Psi$. Pick $R \equiv P$. Hence we have $\tau(R) \equiv Q$, and thus $\tau(R) \in \Psi$, by closure under $\equiv$ of Ψ. Therefore $R \in \tau(\Psi)$, and we conclude that $\tau(\Psi)$ is closed under $\equiv$. Now, pick $m, n \notin \tau(N)$, and let $\tau' = \{m \leftrightarrow n\}$. We have $\tau(m), \tau(n) \notin N$. Thus $\{\tau(m) \leftrightarrow \tau(n)\}(Q) \in \Psi$. This implies $\tau(\{\tau(m) \leftrightarrow \tau(n)\}(Q)) \in \tau(\Psi)$. But $\tau(\{\tau(m) \leftrightarrow \tau(n)\}(Q)) \equiv \{m \leftrightarrow n\}(\tau(Q)) \equiv \tau'(P)$, by Proposition 2.4(2). Hence $\tau'(P) \in \tau(\Psi)$.

   We conclude $\tau(\Psi) \in \mathbb{P}_{\tau(N)}$.
3. Assume $P \in \Psi \in \mathbb{P}_N$. Then $\{m \leftrightarrow n\}(P) \in \Psi$, and thus $P \in \{m \leftrightarrow m\}(\Psi)$. On the other hand, if $P \in \{m \leftrightarrow m\}(\Psi)$ then there is $Q \in \Psi$ such that $P = \{m \leftrightarrow m\}(Q)$. But $\{m \leftrightarrow m\}(Q) \in \Psi$, so $P \in \Psi$.

Lemma 4.11 If $\Phi \in \mathbb{P}_M$ and N is a finite set of names, $Cl(\Phi, N) \in \mathbb{P}_N$.

Proof. If $Q \in Cl(\Phi, N)$ then $Q \in \pi(\Phi)$ for some $\pi \in \Pi_N$. By Lemma 4.7(2), $\pi(\Phi)$ is a Pset, hence closed under structural congruence. Moreover, if we pick $n, m \notin N$ and let $\tau = \{n \leftrightarrow m\}$, then $\tau(Q) \in \tau(\pi(\Phi)) \subseteq Cl(\Phi, N)$. Hence $Cl(\Phi, N) \in \mathbb{P}_N$.

Proposition 4.6 For all finite $N \subseteq \Lambda$, $(\mathbb{P}_N, \subseteq, \bigcup, \otimes, \mathbf{1})$ is a commutative quantale.

Proof. We first verify that $(\mathbb{P}_N, \otimes, \mathbf{1})$ is a commutative monoid.

First, note that $\mathbf{1}$ is closed under $\equiv$ by definition. Moreover, for any $P \in \mathbf{1}$ and transposition τ, we have $\tau(P) \equiv \tau(\mathbf{0}) \equiv \mathbf{0}$. Hence, $\tau(P) \in \mathbf{1}$, for any transposition τ. We conclude that $\mathbf{1} \in \mathbb{P}_N$.

To verify that $\otimes$ is a binary operation on $\mathbb{P}_N$, pick any Ψ and Φ in $\mathbb{P}_N$, and some $P \in \Psi \otimes \Phi$. If $P' \equiv P$ then $P' \equiv Q|R$ where $Q \in \Psi$ and $R \in \Psi$. Hence $P' \in \Psi \otimes \Phi$. Moreover, if $m, n \notin N$ and $\tau = \{m \leftrightarrow n\}$, then $\tau(P) \equiv \tau(Q|R) \equiv \tau(Q)|\tau(R)$. Since $\tau(Q) \in \Psi$ and $\tau(R) \in \Phi$, we conclude $\tau(P) \in \Psi \otimes \Phi$. Now, from simple properties of structural congruence it follows that $\otimes$ is commutative, associative, and has unit $\mathbf{1}$.

Since $(\mathbb{P}_N, \subseteq, \cup, \cap)$ is a complete lattice, and thus closed under arbitrary joins, it remains to verify that $\otimes$ distributes over arbitrary joins, that is $\Psi \otimes \bigcup S = \bigcup\{\Psi \otimes \Phi \mid \Phi \in S\}$, for any $S \subseteq \mathbb{P}_N$. But this is an immediate consequence of the definition of $\otimes$.

Proposition 4.9 Let $\Phi \in \mathbb{P}_N$ then

1. There is a least set of names $supp(\Phi)$ such that $\Phi \in \mathbb{P}_{supp(\Phi)}$.
2. For any transposition τ, if $supp(\Phi) = M$ then $supp(\tau(\Phi)) = \tau(M)$.

Proof. (1) See [11].

(2) Let τ be a transposition, Φ be a Pset and assume $supp(\Phi) = M$. This means that $\Phi \in \mathbb{P}_M$ and for all N such that $\Phi \in \mathbb{P}_N$ we have $M \subseteq N$. To verify that $supp(\tau(\Phi)) = \tau(M)$ we need to show $\tau(\Phi) \in \mathbb{P}_{\tau(M)}$ (which holds by Lemma 4.7(2)) and that for all finite M' such $\tau(\Phi) \in \mathbb{P}_{M'}$ we have $\tau(M) \subseteq M'$. So, take a finite set of names M' and assume $\tau(\Phi) \in \mathbb{P}_{M'}$. Thus $\Phi \in \mathbb{P}_{\tau(M')}$, by Lemma 4.7(2). By assumption, we conclude $M \subseteq \tau(M')$. But then $\tau(M) \subseteq M'$ (for τ is an bijective mapping $\wp_{fin}(\Lambda) \to \wp_{fin}(\Lambda)$), and we are done.

Lemma 4.17 For any formula A, valuation v and transposition τ

1. If $\tau = \{m \leftrightarrow n\}$ and $m, n \notin fn(v)$ then $[\![A]\!]_v = [\![A]\!]_{\tau(v)}$.
2. $\tau(fn^v(A)) = fn^{\tau(v)}(\tau(A))$.
3. Let $fn^v(\boldsymbol{\nu} X.A) \subseteq M$.
   Then $fn^v(\boldsymbol{\nu} X.A) \subseteq fn^{v[X \leftarrow \Psi]}(A) \subseteq M$, for all $\Psi \in \mathbb{P}_M$.

Proof. (1) For any $X \in \mathfrak{D}(v)$ we have $m, n \notin supp(v(X))$ and $v(X) \in \mathbb{P}_{supp(v(X))}$. Thus $\tau(supp(X)) = supp(X)$ by Lemma 4.7(3). Hence $\tau(v) = v$.

(2) By induction on the size of the formula A. If A is $\mathbf{0}$ or $\mathtt{F}$ the result is immediate. If A is $B \square C$, for a binary connective $\square$ (eg. $A \wedge B$, $A|B$) we have

$$\tau(fn^v(B \square C)) = \tau(fn^v(B) \cup fn^v(C)) =$$
$$\tau(fn^v(B)) \cup \tau(fn^v(C)) = fn^{\tau(v)}(\tau(B)) \cup fn^{\tau(v)}(\tau(C)) = fn^{\tau(v)}(\tau(B \square C))$$

where the second equality results from τ being a bijection, and the third from the induction hypothesis. The cases for unary connectives (eg. $\Diamond B$), indexed connectives ($n \circledR B$ and $B \oslash n$) and quantifiers ($\forall x.B$, $\text{И}x.B$) are handled in a similar way. If A is a variable X then $\tau(\mathit{fn}^v(X)) = \tau(\mathit{supp}(v(X))) = \mathit{supp}(\tau(v(X))) = \mathit{fn}^{\tau(v)}(X) = \mathit{fn}^{\tau(v)}(\tau(X))$. If A is $\boldsymbol{\nu} X.B$ then

$$\tau(\mathit{fn}^v(\boldsymbol{\nu} X.B)) = \tau(\mathit{fn}^{v[X\leftarrow\emptyset]}(B)) = \mathit{fn}^{\tau(v)[X\leftarrow\emptyset]}(\tau(B)) = \mathit{fn}^{\tau(v)}(\tau(\boldsymbol{\nu} X.B))$$

(3) Let $N = \mathit{fn}^v(\boldsymbol{\nu} X.A) = \mathit{fn}^{v[X\leftarrow\emptyset]}(A) \subseteq M$. We have

$$\begin{aligned} N = \mathit{fn}(\boldsymbol{\nu} X.A) \cup \bigcup\{\mathit{supp}(v(Y)) \mid Y \in \mathit{fpv}(A)\} = \\ \mathit{fn}(A) \cup \bigcup\{\mathit{supp}(v(Y)) \mid Y \in \mathit{fpv}(A) \setminus \{X\}\} \end{aligned}$$

Now, for any $\Psi \in \mathbb{P}_M$ we have $\mathit{supp}(\Psi) \subseteq M$. Hence

$$\begin{aligned} N \subseteq \mathit{fn}(A) \cup \bigcup\{\mathit{supp}(v(Y)) \mid Y \in \mathit{fpv}(A) \setminus \{X\}\} \cup \mathit{supp}(\Psi) = \\ \mathit{fn}^{v[X\leftarrow\Psi]}(A) \subseteq N \cup M \subseteq M \end{aligned}$$

(since $N \subseteq M$).

Theorem 4.18 For any formula A and valuation v

1. $[\![A]\!]_v \in \mathbb{P}_{\mathit{fn}^v(A)}$.
2. For all transpositions τ, $\tau([\![A]\!]_v) = [\![\tau(A)]\!]_{\tau(v)}$.
3. If $X \notin \mathrm{Neg}(A)$ and $\Psi \subseteq \Phi$ then $[\![A]\!]_{v[X\leftarrow\Psi]} \subseteq [\![A]\!]_{v[X\leftarrow\Phi]}$.
4. If $X \notin \mathrm{Pos}(A)$ and $\Psi \subseteq \Phi$ then $[\![A]\!]_{v[X\leftarrow\Phi]} \subseteq [\![A]\!]_{v[X\leftarrow\Psi]}$.

Proof. The proof of (1—4) proceeds by mutual induction on the size of the formula A. Note that the size of a formula does not change when replacing names and variables for names. All induction hypotheses are interdependent in the cases for $\text{И}x.B$ and $\boldsymbol{\nu} X.B$. Instead of the equality presented in (2) we prove that $\tau([\![A]\!]_v) \subseteq [\![\tau(A)]\!]_{\tau(v)}$. From this fact, the equality immediately follows. Indeed, if $P \in [\![\tau(A)]\!]_{\tau(v)}$, we have $\tau(P) \in \tau([\![\tau(A)]\!]_{\tau(v)}) \subseteq [\![A]\!]_v$, and then $P \in \tau([\![A]\!]_v)$.

- Case $A = \mathtt{F}$
  (1) Clearly $[\![\mathtt{F}]\!]_v = \emptyset \in \mathcal{P}_{\mathit{fn}^v(\mathtt{F})}$.
  (2) Trivial, since $\tau([\![\mathtt{F}]\!]_v) = \tau(\emptyset) = \emptyset$.
  (3—4) Immediate.
- Case $A = B \wedge C$.
  (1) Then $[\![B \wedge C]\!]_v = [\![B]\!]_v \cap [\![C]\!]_v$.
  By induction hypothesis (1) and Lemma 4.2(1) both $[\![B]\!]_v$ and $[\![C]\!]_v$ belong to $\mathcal{P}_{\mathit{fn}^v(A)}$. We conclude by Lemma 4.2(3).
  (2) Let τ be a transposition. If $P \in \tau([\![A \wedge B]\!]_v)$ then there is P' such that $P = \tau(P')$ with $P' \in [\![A]\!]_v$ and $P' \in [\![B]\!]_v$. Then $P \in \tau([\![A]\!]_v)$ and $P \in \tau([\![B]\!]_v)$. By induction hypothesis (2), $P \in [\![\tau(A)]\!]_{\tau(v)}$ and $P \in [\![\tau(B)]\!]_{\tau(v)}$. Hence, $P \in [\![\tau(A \wedge B)]\!]_{\tau(v)}$.
  (3—4) Immediate.

- Case $A = B \Rightarrow C$.
(1) Pick $P \in [\![B \Rightarrow C]\!]_v$. Therefore, if $P \in [\![B]\!]_v$ then $P \in [\![C]\!]_v$. By induction hypothesis (1), $[\![B]\!]_v \in \mathbb{P}_{fn^v(B)}$ and $[\![C]\!]_v \in \mathbb{P}_{fn^v(C)}$.
Pick any $Q \equiv P$. Assume $Q \in [\![B]\!]_v$. Then $P \in [\![B]\!]_v$ and thus $P \in [\![C]\!]_v$. Hence $Q \in [\![C]\!]_v$, and we conclude $Q \in [\![B \Rightarrow C]\!]_v$.
Now pick $m, n \notin fn^v(B \Rightarrow C)$ and let $\tau = \{m \leftrightarrow n\}$. We have, $m, n \notin fn^v(B)$ and $m, n \notin fn^v(C)$. Assume $\tau(P) \in [\![B]\!]_v$. Thus $P \in [\![B]\!]_v$ by induction hypothesis (1). Then $P \in [\![C]\!]_v$ and by induction hypothesis (1) again $\tau(P) \in [\![C]\!]_v$. Hence $\tau(P) \in [\![B \Rightarrow C]\!]_v$.
(2) Let τ be a transposition. Pick $P \in \tau([\![B \Rightarrow C]\!]_v)$. Then, $\tau(P) \in [\![B]\!]_v$ implies $\tau(P) \in [\![C]\!]_v$. Assume $P \in [\![\tau(B)]\!]_{\tau(v)}$. By induction hypothesis (2), we conclude $\tau(P) \in [\![B]\!]_v$. Then, $\tau(P) \in [\![C]\!]_v$. By induction hypothesis (2) again, $P \in [\![\tau(C)]\!]_{\tau(v)}$. Therefore $P \in [\![\tau(B \Rightarrow C)]\!]_{\tau(v)}$.
(3) Then $X \notin \mathrm{Neg}(C)$ and $X \notin \mathrm{Pos}(B)$. Assume $P \in [\![B \Rightarrow C]\!]_{v[X \leftarrow \Psi]}$. This implies that if $P \in [\![B]\!]_{v[X \leftarrow \Psi]}$ then $P \in [\![C]\!]_{v[X \leftarrow \Psi]}$. Assume $P \in [\![B]\!]_{v[X \leftarrow \Phi]}$. Since $X \notin \mathrm{Pos}(B)$, by induction hypothesis (4), $P \in [\![B]\!]_{v[X \leftarrow \Psi]}$. Then $P \in [\![C]\!]_{v[X \leftarrow \Psi]}$. By induction hypothesis (3), $P \in [\![C]\!]_{v[X \leftarrow \Phi]}$. Hence $P \in [\![B \Rightarrow C]\!]_{v[X \leftarrow \Phi]}$.
(4) Symmetric to (3).
- Case $A = \mathbf{0}$.
(1) $[\![\mathbf{0}]\!]_v = \mathbf{1} \in \mathbb{P}_{fn^v(\mathbf{0})} = \mathbb{P}_\emptyset$ by Proposition 4.6.
(2) Let τ be a transposition. Pick $P \in \tau([\![\mathbf{0}]\!]_v)$. Then $\tau(P) \equiv \mathbf{0}$ and $P \equiv \mathbf{0}$. So $P \in [\![\tau(\mathbf{0})]\!]_{\tau(v)} = \mathbf{1}$.
(3—4) Immediate.
- Case $A = B|C$.
(1) By induction hypothesis (1) and Lemma 4.2(1) we have $[\![B]\!]_v \in \mathbb{P}_{fn^v(B)} \subseteq \mathbb{P}_{fn^v(B|C)}$. Likewise we conclude $[\![C]\!]_v \in \mathbb{P}_{fn^v(B|C)}$.
By Proposition 4.6, $[\![B|C]\!]_v = [\![B]\!]_v \otimes [\![C]\!]_v \in \mathbb{P}_{fn^v(B|C)}$.
(2) Let τ be a transposition. Pick $P \in \tau([\![B|C]\!]_v)$. Hence $\tau(P) \equiv Q'|Q''$ with $Q' \in [\![B]\!]_v$ and $Q'' \in [\![C]\!]_v$, for some Q' and Q''. Then $\tau(Q') \in \tau([\![B]\!]_v)$, and $\tau(Q') \in [\![\tau(B)]\!]_{\tau(v)}$, by induction hypothesis (2). Likewise, $\tau(Q'') \in [\![\tau(C)]\!]_{\tau(v)}$. Hence $\tau(Q'|Q'') \in [\![\tau(B|C)]\!]_{\tau(v)}$. To conclude, just note that $P \equiv \tau(Q'|Q'')$.
(3—4) Like (Case $A = B \wedge C$) above.
- Case $A = B \triangleright C$.
(1) Pick $P \in [\![B \triangleright C]\!]_v$. Then, for all $Q \in [\![B]\!]_v$, we have $Q|P \in [\![C]\!]_v$. Pick any $P' \equiv P$, and any $Q' \in [\![B]\!]_v$. We have $Q'|P' \equiv Q'|P$. By assumption, $Q'|P \in [\![C]\!]_v$, and, by induction hypothesis (1), $Q'|P' \in [\![C]\!]_v$. Hence $P' \in [\![B \triangleright C]\!]_v$.
Pick $m, n \notin fn^v(B \triangleright C)$ and let $\tau = \{m \leftrightarrow n\}$. Then $m, n \notin fn^v(B)$ and $m, n \notin fn^v(C)$. Now, pick any Q' such that $Q' \in [\![B]\!]_v$. By induction hypothesis (1), $\tau(Q') \in [\![B]\!]_v$. Therefore $\tau(Q')|P \in [\![C]\!]_v$. By induction hypothesis (1) and Proposition 2.12(3), $\tau(\tau(Q')|P) \equiv Q'|\tau(P) \in [\![C]\!]_v$. Hence $\tau(P) \in [\![B \triangleright C]\!]_v$.
(2) Let τ be a transposition. Pick $P \in \tau([\![B \triangleright C]\!]_v)$. Then there is $P' \equiv \tau(P)$ such that for all $Q \in [\![B]\!]_v$, we have $Q|P' \in [\![C]\!]_v$. Pick any $Q' \in [\![\tau(B)]\!]_{\tau(v)}$. Then $\tau(Q') \in \tau([\![\tau(B)]\!]_{\tau(v)})$. By induction hypothesis (2), $\tau(Q') \in [\![B]\!]_v$.

Then $\tau(Q')|\tau(P) \in [\![C]\!]_v$. So, $Q'|P \in \tau([\![C]\!]_v)$. By induction hypothesis (2), $Q'|P \in [\![\tau(C)]\!]_{\tau(v)}$. Hence $P \in [\![\tau(B \triangleright C)]\!]_{\tau(v)}$.
(3—4) Like (Case $A = B \Rightarrow C$) above.

- Case $A = q \circledR B$.
(1) Pick any $P \in [\![q \circledR B]\!]_v$. Then $P \equiv (\boldsymbol{\nu} q)Q$ and $Q \in [\![B]\!]_v$. Pick any $P' \equiv P$, we have $P' \equiv (\boldsymbol{\nu} q)Q$, and thus $P' \in [\![q \circledR B]\!]_v$. Pick $m, n \notin \mathit{fn}^v(q \circledR B)$ and let $\tau = \{m \leftrightarrow n\}$. Then $m, n \notin \mathit{fn}^v(B)$, $m \neq q$ and $n \neq q$. Hence $\tau(P) \equiv (\boldsymbol{\nu} q)\tau(Q)$, by Proposition 2.12(3). By induction hypothesis (1), we have $\tau(Q) \in [\![B]\!]_v$. So $\tau(P) \in [\![q \circledR B]\!]_v$.
(2) Let τ be a transposition. Pick $P \in \tau([\![q \circledR B]\!]_v)$. Then there is $P' = \tau(P)$ such that $P' \equiv (\boldsymbol{\nu} q)Q$ and $Q \in [\![B]\!]_v$. We have $\tau(P') \equiv (\boldsymbol{\nu} \tau(q))\tau(Q)$, by Lemma 2.8(1). By induction hypothesis (2), we have $\tau(Q) \in [\![\tau(B)]\!]_{\tau(v)}$. Hence $P = \tau(P') \in [\![\tau(q) \circledR \tau(B)]\!]_{\tau(v)} = [\![\tau(q \circledR B)]\!]_{\tau(v)}$.
(3) Then $X \notin \mathrm{Pos}(B)$. Assume $P \in [\![q \circledR B]\!]_{v[X \leftarrow \Psi]}$. Then $P \equiv (\boldsymbol{\nu} q)Q$ and $Q \in [\![B]\!]_{v[X \leftarrow \Psi]}$. By induction hypothesis (3), $Q \in [\![B]\!]_{v[X \leftarrow \Phi]}$. Hence $P \in [\![q \circledR B]\!]_{v[X \leftarrow \Phi]}$.
(4) Symmetrical.
- Case $A = B \oslash q$.
(1) Pick any $P \in [\![B \oslash q]\!]_v$. Then $(\boldsymbol{\nu} q)P \in [\![B]\!]_v$. Pick any $P' \equiv P$, we have $(\boldsymbol{\nu} q)P' \equiv (\boldsymbol{\nu} q)P$, and thus, by induction hypothesis (1), $(\boldsymbol{\nu} q)P' \in [\![B]\!]_v$. Therefore, $P' \in [\![B \oslash q]\!]_v$. Pick $m, n \notin \mathit{fn}^v(B \oslash q)$ and let $\tau = \{m \leftrightarrow n\}$. Then $m, n \notin \mathit{fn}^v(B)$, $m \neq q$ and $n \neq q$. By induction hypothesis (1), we have $\tau((\boldsymbol{\nu} q)P) \in [\![B]\!]_v$. Since $\tau((\boldsymbol{\nu} q)P) = (\boldsymbol{\nu} q)\tau(P)$, we have $\tau(P) \in [\![B \oslash q]\!]_v$.
(2) Let τ be a transposition. Pick $P \in \tau([\![B \oslash q]\!]_v)$. Then there is $P' = \tau(P)$ such that $(\boldsymbol{\nu} q)P' \in [\![B]\!]_v$. Then $\tau(P) \in [\![B \oslash q]\!]_v$, that is, $(\boldsymbol{\nu} q)\tau(P) \in [\![B]\!]_v$. Then $\tau((\boldsymbol{\nu} q)\tau(P) = (\boldsymbol{\nu} \tau(q))P \in \tau([\![B]\!]_v)$. By induction hypothesis (2), $(\boldsymbol{\nu} \tau(q))P \in [\![\tau(B)]\!]_{\tau(v)}$. Hence $P \in [\![\tau(B) \oslash \tau(q)]\!]_{\tau(v)} = [\![\tau(B \oslash q)]\!]_{\tau(v)}$.
(3—4) Like (Case $A = q \circledR B$) above.
- Case $A = p\langle q\rangle$.
(1) We have $[\![p\langle q\rangle]\!]_v = \{P \mid P \equiv p\langle q\rangle\}$, which is closed under $\equiv$ by definition. Additionally, for all $m, n \notin \{p, q\}$, we have that $\{n \leftrightarrow m\}P \equiv p\langle q\rangle$ for all $P \equiv p\langle q\rangle$.
(2) Let $\tau = \{m \leftrightarrow n\}$. If $P \in \tau([\![p\langle q\rangle]\!]_v)$ then $\tau(P) \equiv p\langle q\rangle$, and $P \equiv \tau(p\langle q\rangle)$. It is clear that $P \in [\![\tau(p\langle q\rangle)]\!]_{\tau(v)}$.
(3—4) Immediate.
- Case $A = \text{И}x.B$.
(1) Pick $P \in [\![\text{И}x.B]\!]_v$. Then, there is q such that $q \notin \mathit{fn}^v(B) \cup \mathit{fn}(P)$ and $P \in [\![B\{x \leftarrow q\}]\!]_v$. Pick $Q \equiv P$, by induction hypothesis (1) also $Q \in [\![B\{x \leftarrow q\}]\!]_v$, and $q \notin \mathit{fn}(Q)$ by Proposition 2.12(1). Thus $Q \in [\![\text{И}x.B]\!]_v$.
Now, pick $\tau = \{m \leftrightarrow n\}$ with $m, n \notin \mathit{fn}^v(\text{И}x.B) = \mathit{fn}^v(B)$. Pick any $p \notin \mathit{fn}^v(B) \cup \{m, n, q\} \cup \mathit{fn}(P) \cup \mathit{fn}(\tau(P))$ and let $\tau' = \{q \leftrightarrow p\}$. By induction hypothesis (2), we have $\tau'(P) \in [\![\tau'(B\{x \leftarrow q\})]\!]_{\tau'(v)}$, that is $P \in [\![B\{x \leftarrow p\}]\!]_v$, by Lemma 4.7(3). By induction hypothesis (1), we conclude $\tau(P) \in [\![B\{x \leftarrow p\}]\!]_v$. Thus, $\tau(P) \in [\![\text{И}x.B]\!]_v$, since $p \notin \mathit{fn}^v(B) \cup \mathit{fn}(\tau(P))$.
(2) Let $\tau = \{m \leftrightarrow n\}$. If $P \in \tau([\![\text{И}x.B]\!]_v)$ then $\tau(P) \in [\![\text{И}x.B]\!]_v$ and thus there is some $q \notin \mathit{fn}^v(B) \cup \mathit{fn}(\tau(P))$ such that $\tau(P) \in [\![B\{x \leftarrow q\}]\!]_v$.

Now, pick $p \notin \{m,n\} \cup fn(\tau(P)) \cup fn(P) \cup fn^v(B) \cup fn^{\tau(v)}(\tau(B))$ and let $\tau' = \{q \leftrightarrow p\}$. $\tau'(\tau(P)) = \tau(P) \in [\![\tau'(B\{x \leftarrow q\})]\!]_{\tau'(v)} = [\![B\{x \leftarrow p\}]\!]_{\tau'(v)}$.
By Remark 4.16 we have $[\![B\{x \leftarrow p\}]\!]_{\tau'(v)} = [\![B\{x \leftarrow p\}]\!]_{\tau'(w)}$, where w is the restriction of v to $fpv(B)$.
By Lemma 4.17(1) we have $[\![B\{x \leftarrow p\}]\!]_{\tau'(w)} = [\![B\{x \leftarrow p\}]\!]_w$, since $q, p \notin fn(w)$. So, $\tau(P) \in [\![B\{x \leftarrow p\}]\!]_w$.
By Remark 4.16, this implies $\tau(P) \in [\![B\{x \leftarrow p\}]\!]_v$.
But then $P \in [\![\tau(B)\{x \leftarrow p\}]\!]_{\tau(v)}$, by induction hypothesis (2). We conclude $P \in [\![\tau(\text{И}x.B)]\!]_{\tau(v)}$, since $p \notin fn^{\tau(v)}(\tau(B)) \cup fn(P)$.
(3) We have $X \notin \mathrm{Neg}(B\{x \leftarrow n\})$, for all $n \in \Lambda$. Let $P \in [\![\text{И}x.B]\!]_{v[X \leftarrow \Psi]}$ and w be the restriction of v to the free propositional variables of $\text{И}x.B$ (we assume that $X \in fpv(A)$ otherwise the result is immediate).
By Remark 4.16, $P \in [\![\text{И}x.B]\!]_{w[X \leftarrow \Psi]}$. Then, there is q such that $q \notin fn^{w[X \leftarrow \Psi]}(B) \cup fn(P)$ and $P \in [\![B\{x \leftarrow q\}]\!]_{w[X \leftarrow \Psi]}$.
Now, pick $p \notin fn^{w[X \leftarrow \Psi]}(B) \cup fn(P) \cup fn^{w[X \leftarrow \Phi]}(B)$ and define $\tau = \{q \leftrightarrow p\}$. Note that $p, q \notin supp(\Psi) \cup fn(w)$. By induction hypothesis (2), we have $\tau(P) = P \in \tau([\![B\{x \leftarrow q\}]\!]_{w[X \leftarrow \Psi]}) \subseteq [\![B\{x \leftarrow p\}]\!]_{w[X \leftarrow \Psi]}$, since $\tau(\Psi) = \Psi$ and $\tau(w) = w$.
By induction hypothesis (3), $P \in [\![B\{x \leftarrow p\}]\!]_{w[X \leftarrow \Phi]}$.
But p was chosen such that $p \notin fn^{w[X \leftarrow \Phi]}(A)$, so $P \in [\![\text{И}x.B]\!]_{w[X \leftarrow \Phi]}$. This implies $P \in [\![\text{И}x.B]\!]_{v[X \leftarrow \Phi]}$, by Remark 4.16.
(4) Symmetrical.

- Case $A = \forall x.B$.
(1) Pick any $P \in [\![\forall x.B]\!]_v$. Then, for all $q \in \Lambda$, $P \in [\![B\{x \leftarrow q\}]\!]_v$.
Pick any $Q \equiv P$. For all $q \in \Lambda$, by induction hypothesis (1), $Q \in [\![B\{x \leftarrow q\}]\!]_v$. Therefore, $Q \in [\![\forall x.B]\!]_v$.
Pick $m, n \notin fn^v(\forall x.B) = fn^v(B)$ and let $\tau = \{m \leftrightarrow n\}$. We need to show that for all $q \in \Lambda$, $\tau(P) \in [\![B\{x \leftarrow q\}]\!]_v$.
For all $q \notin \{m, n\}$, by induction hypothesis (1) we have $\tau(P) \in [\![B\{x \leftarrow q\}]\!]_v$.
For $q = m$, we have $P \in [\![B\{x \leftarrow m\}]\!]_v$. Then $\tau(P) \in [\![B\{x \leftarrow n\}]\!]_{\tau(v)}$, by induction hypothesis (2).
Let w be the restriction of v to $fpv(B)$. Note that $m, n \notin fn(w)$, for if (say) $n \in fn(w)$, then $n \in supp(w(X))$ for some $X \in fpv(B)$, and we would have $n \in fn^w(B) = fn^v(B)$, a contradiction.
By Remark 4.16, $\tau(P) \in [\![B\{x \leftarrow n\}]\!]_{\tau(w)}$.
Since we have $m, n \notin fn(w)$, by Lemma 4.17(1) we conclude $[\![B\{x \leftarrow n\}]\!]_{\tau(w)} = [\![B\{x \leftarrow n\}]\!]_w$. By Remark 4.16, $[\![B\{x \leftarrow n\}]\!]_w = [\![B\{x \leftarrow n\}]\!]_v$ and then $\tau(P) \in [\![B\{x \leftarrow n\}]\!]_v$.
For $q = n$, we conclude $\tau(P) \in [\![B\{x \leftarrow m\}]\!]_v$ in a similar way.
Then $\tau(P) \in [\![B\{x \leftarrow q\}]\!]_v$, for all $q \in \Lambda$; this implies $\tau(P) \in [\![\forall x.B]\!]_v$.
(2) Let τ be a transposition. Pick $P \in \tau([\![\forall x.B]\!]_v)$. Then $\tau(P) \in [\![B\{x \leftarrow q\}]\!]_v$, for all $q \in \Lambda$.
By induction hypothesis (2), for all $q \in \Lambda$, $P \in [\![\tau(B)\{x \leftarrow \tau(q)\}]\!]_{\tau(v)}$. Then $P \in [\![\tau(B)\{x \leftarrow q\}]\!]_{\tau(v)}$, for all $q \in \Lambda$, since τ is a bijection $\Lambda \to \Lambda$. Therefore, $P \in [\![\tau(\forall x.B)]\!]_{\tau(v)}$.

(3) We have $X \notin \mathrm{Neg}(B\{x\leftarrow n\})$, for all $n \in \Lambda$. Let $P \in [\![\forall x.B]\!]_{v[X\leftarrow\Psi]}$, that is, for all $n \in \Lambda$, $P \in [\![B\{x\leftarrow n\}]\!]_{v[X\leftarrow\Psi]}$. By induction hypothesis (3), for each $n \in \Lambda$, $P \in [\![B\{x\leftarrow n\}]\!]_{v[X\leftarrow\Phi]}$. Hence $P \in [\![\forall x.B]\!]_{v[X\leftarrow\Phi]}$.
(4) Symmetrical.

– Case $A = \Diamond B$.
(1) If $P \in [\![\Diamond B]\!]_v$ then there is R such that $P \to R$ and $R \in [\![B]\!]_v$. If $Q \equiv P$ then also $Q \to R$, so $Q \in [\![\Diamond B]\!]_v$. Now, pick $m, n \notin fn^v(\Diamond B) = fn^v(B)$, and let $\tau = \{m\leftrightarrow n\}$. By Proposition 2.15(2), $\tau(P) \to \tau(R)$. By induction hypothesis (1), $\tau(R) \in [\![B]\!]_v$. Hence $\tau(P) \in [\![\Diamond B]\!]_v$.
(2) Let τ be a transposition. If $P \in \tau([\![\Diamond B]\!]_v)$ then $\tau(P) \in [\![\Diamond B]\!]_v$. Thus there is Q such that $\tau(P) \to Q$ and $Q \in [\![B]\!]_v$. By Proposition 2.15(2), and induction hypothesis (2), $P \to \tau(Q)$ and $\tau(Q) \in [\![\tau(B)]\!]_{\tau(v)}$. Hence $P \in [\![\tau(\Diamond B)]\!]_{\tau(v)}$.
(3—4) Immediate, by the induction hypothesis.

– Case $A = Z$.
(1) We have $[\![Z]\!]_v = v(Z)$. Since $fn^v(Z) = supp(v(Z))$, we have that $[\![Z]\!]_v \in \mathbb{P}_{fn^v(Z)}$.
(2) Let τ be a transposition. If $P \in \tau([\![Z]\!]_v)$ then $\tau(P) \in v(Z)$. Therefore, $P \in \tau(v(Z)) = \tau(v)(Z) = [\![Z]\!]_{\tau(v)}$.
(3) The case $Z \neq X$ is trivial. If $Z = X$, the assumption yields $[\![X]\!]_{v[X\leftarrow\Psi]} = \Psi \subseteq \Phi = [\![X]\!]_{v[X\leftarrow\Phi]}$.
(4) $X \notin \mathrm{Pos}(Z)$ implies $X \neq Z$, and we conclude.

– Case $A = \boldsymbol{\nu} Z.B$.
Let $M = fn^v(\boldsymbol{\nu} Z.B)$.
(1) We have $[\![\boldsymbol{\nu} Z.B]\!]_v = \bigcup\{\Psi \in \mathbb{P}_M \mid \Psi \subseteq [\![B]\!]_{v[Z\leftarrow\Psi]}\}$. Pick $\Psi \in \mathbb{P}_M$. Then $v[Z\leftarrow\Psi]$ is a valuation. By induction hypothesis (1), $[\![B]\!]_{v[Z\leftarrow\Psi]} \in \mathbb{P}_M$, since $fn^{v[Z\leftarrow\Psi]}(B) = M$ by Lemma 4.17(3). Hence

$$\mathfrak{G} \triangleq \boldsymbol{\lambda}\Psi.[\![B]\!]_{v[Z\leftarrow\Psi]}$$

is a mapping $\mathbb{P}_M \to \mathbb{P}_M$. By induction hypothesis (3), $\mathfrak{G}$ is monotonic. By Knaster-Tarski Lemma, it admits a greatest fixpoint $\mathcal{G} \in \mathbb{P}_M$ and $[\![\boldsymbol{\nu} Z.B]\!]_v = \mathcal{G}$.
(2) Let τ be a transposition.
We must show $\tau([\![\boldsymbol{\nu} Z.B]\!]_v) \subseteq [\![\boldsymbol{\nu} Z.\tau(B)]\!]_{\tau(v)}$. We claim that the right-hand side of this inclusion is the greatest fixpoint of the mapping

$$\mathfrak{T} \triangleq \boldsymbol{\lambda}\Phi.[\![\tau(B)]\!]_{\tau(v)[Z\leftarrow\Phi]}$$

To verify the claim, we start by showing that $\mathfrak{T}$ is a monotonic operator on $\mathbb{P}_{\tau(M)}$. Pick any $\Phi \in \mathbb{P}_{\tau(M)}$. By Lemma 4.7(2), we have $\tau(\Phi) \in \mathbb{P}_M$. As in subcase (1) above, we conclude that $v[Z\leftarrow\tau(\Phi)]$ is a valuation and that $[\![B]\!]_{v[Z\leftarrow\tau(\Phi)]} \in \mathbb{P}_M$. By induction hypothesis (2) $\tau([\![B]\!]_{v[Z\leftarrow\tau(\Phi)]}) \subseteq [\![\tau(B)]\!]_{\tau(v)[Z\leftarrow\Phi]}$, and $[\![\tau(B)]\!]_{\tau(v)[Z\leftarrow\Phi]} \in \mathbb{P}_{fn^{\tau(v)[Z\leftarrow\Phi]}(B)}$, by induction hypothesis (1). By Lemma 4.17(2—3), $fn^{\tau(v)[Z\leftarrow\Phi]}(\tau(B)) = \tau(M)$.

So $[\![\tau(B)]\!]_{\tau(v)[Z\leftarrow\Phi]} \in \mathbb{P}_{\tau(M)}$ and we conclude that $\mathfrak{T}$ is indeed a mapping $\mathbb{P}_{\tau(M)} \to \mathbb{P}_{\tau(M)}$. Moreover, such operator is monotonic by induction hypothesis (3). Thus, by Knaster-Tarski Lemma, it admits a greatest fixpoint $\mathcal{T} \in \mathbb{P}_{\tau(M)}$ where $\mathcal{T}$ coincides with $[\![\boldsymbol{\nu} Z.\tau(B)]\!]_{\tau(v)}$, since by Lemma 4.17(2) $\tau(M) = fn^{\tau(v)}(\boldsymbol{\nu} Z.\tau(B))$. We have thus proved the claim.
To conclude, it remains to show $\tau(\mathcal{G}) \subseteq \mathcal{T}$ (we now use $\mathcal{G}$ and $\mathfrak{G}$, defined in subcase (1) above). Pick any $Q \in \tau(\mathcal{G}) = \tau(\mathfrak{G}(\mathcal{G})) = \tau([\![B]\!]_{v[Z\leftarrow\mathcal{G}]})$. By induction hypothesis (2), $Q \in [\![\tau(B)]\!]_{\tau(v[Z\mapsto\mathcal{G}])} = [\![\tau(B)]\!]_{\tau(v)[Z\mapsto\tau(\mathcal{G})])} = \mathfrak{T}(\tau(\mathcal{G}))$.
Hence $\tau(\mathcal{G}) \subseteq \mathfrak{T}(\tau(\mathcal{G}))$, that is $\tau(\mathcal{G})$ is a pre-fixpoint of $\mathfrak{T}$. Therefore, we conclude $\tau(\mathcal{G}) \subseteq \mathcal{T}$, by Knaster-Tarski Lemma.
(3) We have $Z \notin \mathrm{Neg}(B)$. By Lemma 4.15, we may assume $Z \neq X$. We need to show (†)

$$\bigcup\{\varphi \in \mathbb{P}_M \mid \varphi \subseteq [\![B]\!]_{v[X\leftarrow\Psi][Z\leftarrow\varphi]}\} \subseteq$$

$$\bigcup\{\varphi \in \mathbb{P}_{M'} \mid \varphi \subseteq [\![B]\!]_{v[X\leftarrow\Phi][Z\leftarrow\varphi]}\}$$

where $M = fn^{v[X\leftarrow\Psi]}(\boldsymbol{\nu} Z.B)$ and $M' = fn^{v[X\leftarrow\Phi]}(\boldsymbol{\nu} Z.B)$. To that end, pick any process P in the left-hand side of the inclusion (†). Then, there is $\varphi \in \mathbb{P}_M$ such that $P \in \varphi$ and $\varphi \subseteq [\![B]\!]_{v[X\leftarrow\Psi][Z\leftarrow\varphi]}$.
Now, define $\varphi^* \triangleq Cl(\varphi, M')$. We have $\varphi^* \in \mathbb{P}_{M'}$ by Lemma 4.11; we also have $\varphi \subseteq \varphi^*$, hence $P \in \varphi^*$. By induction hypothesis (3) (twice), we have $\varphi \subseteq [\![B]\!]_{v[X\leftarrow\Phi][Z\leftarrow\varphi^*]}$, since $Z \notin \mathrm{Neg}(B)$.
It remains to show $\varphi^* \subseteq [\![B]\!]_{v[X\leftarrow\Phi][Z\leftarrow\varphi^*]}$. Let w be the restriction of v to the free propositional variables of B. Then $[\![B]\!]_{v[X\leftarrow\Phi][Z\leftarrow\varphi^*]} = [\![B]\!]_{w[X\leftarrow\Phi][Z\leftarrow\varphi^*]}$ by Remark 4.16(1) (we assume $X \in fpv(B)$, for otherwise the result would be immediate).
Pick $Q \in \varphi^*$. By Definition 4.10 $Q = \pi(Q')$ for some $Q' \in \varphi$ and $\pi \in \Pi_{M'}$. Thus $\pi(Q') \in \pi([\![B]\!]_{w[X\leftarrow\Phi][Z\leftarrow\varphi^*]})$. By induction hypothesis (2), we have $\pi(Q') = Q \in [\![B]\!]_{w[X\leftarrow\Phi][Z\leftarrow\varphi^*]}$, since $\pi(B) = B$, $\pi(w) = w$, $\pi(\Phi) = \Phi$ (because $M' = fn^{w[X\leftarrow\Phi]}(\nu Z.B)$), and also $\pi(\varphi^*) = \varphi^*$.
Since by Remark 4.16 $[\![B]\!]_{v[X\leftarrow\Phi][Z\leftarrow\varphi^*]} = [\![B]\!]_{w[X\leftarrow\Phi][Z\leftarrow\varphi^*]}$, we conclude $\varphi^* \subseteq [\![B]\!]_{v[X\leftarrow\Phi][Z\leftarrow\varphi^*]}$. Hence φ^* is included on the right-hand side of (†). So P belongs to the right-hand side of (†) and we are done.
(4) Symmetrical.

Lemma 4.24 Let A be a formula and $v[X\leftarrow\Phi]$ a valuation for A such that $fn^v(\boldsymbol{\nu} X.A) \subseteq N \subseteq M$, $\Phi \subseteq [\![A]\!]_{v[X\leftarrow\Phi]}$ and $\Phi \in \mathbb{P}_M$.
Then $Cl(\Phi, N) \subseteq [\![A]\!]_{v[X\leftarrow Cl(\Phi,N)]}$.

Proof. By definition, we have $Cl(\Phi, N) \triangleq \bigcup_{\pi\in\Pi_N} \pi(\Phi)$ where Π_N is the set of all permutations that fix N. Note that any permutation in Π_N can be written as a composition of a finite number of transpositions $\{p\leftrightarrow q\}$ such that $p, q \notin N$.

We have $\Phi \subseteq Cl(\Phi, N)$ and $Cl(\Phi, N) \in \mathbb{P}_N$. Therefore, by Theorem 4.18(3), $[\![A]\!]_{v[X\leftarrow\Phi]} \subseteq [\![A]\!]_{v[X\leftarrow Cl(\Phi,N)]}$. We now prove $Cl(\Phi, N) \subseteq [\![A]\!]_{v[X\leftarrow Cl(\Phi,N)]}$.

Pick $P \in Cl(\Phi, N)$. Then $P = \pi(P')$ for some $P' \in \Phi$ and $\pi \in \Pi_N$. Let w be the restriction of v to the free propositional variables of $\boldsymbol{\nu} X.A$. By Remark 4.16

we have $\Phi \subseteq [\![A]\!]_{w[X\leftarrow Cl(\Phi,N)]}$. Note that for all $\pi \in \Pi_N$ we have $\pi(w) = w$ (since $fn^w(\boldsymbol{\nu}X.A) \subseteq N$), $\pi(A) = A$, and $\pi(Cl(\Phi,N)) = Cl(\Phi,N)$. Hence $P' \in [\![A]\!]_{w[X\leftarrow Cl(\Phi,N)]}$. But then, by Theorem 4.18(2) and Remark 4.16

$$\pi(P') = P \in \pi([\![A]\!]_{w[X\leftarrow Cl(\Phi,N)]}) =$$
$$[\![A]\!]_{\pi(w)[X\leftarrow \pi(Cl(\Phi^N,N))]} = [\![A]\!]_{w[X\leftarrow Cl(\Phi,N)]} = [\![A]\!]_{v[X\leftarrow Cl(\Phi,N)]}$$

We conclude $Cl(\Phi,N) \subseteq [\![A]\!]_{v[X\leftarrow Cl(\Phi,N)]}$.

Lemma 5.3 Let $[\![\theta(A)]\!]_v = [\![\theta(B)]\!]_v$ for all substitutions θ and valuations v, and let $F\{-\}$ be a context. Then, for all substitutions σ and valuations w we have $[\![\sigma(F\{A\})]\!]_w = [\![\sigma(F\{B\})]\!]_w$.

Proof. By induction on the size of the context $F\{-\}$.

- Case $F\{-\} = G\{-\} \Rightarrow H\{-\}$.
  By definition, $[\![\sigma(F\{A\})]\!]_w$ is the set of all processes Q such that if $Q \in [\![\sigma(G\{A\})]\!]_w$ then $Q \in [\![\sigma(H\{A\})]\!]_w$. By induction hypothesis, $[\![\sigma(G\{A\})]\!]_w = [\![\sigma(G\{B\})]\!]_w$ and $[\![\sigma(H\{A\})]\!]_w = [\![\sigma(H\{B\})]\!]_w$.
  Hence $[\![\sigma(F\{A\})]\!]_v = [\![\sigma(F\{B\})]\!]_v$.
- Cases $F\{-\} = G\{-\} \wedge H\{-\}$, $F\{-\} = G\{-\}|H\{-\}$, $F\{-\} = G\{-\} \triangleright H\{-\}$, $F\{-\} = \boxdot G\{-\}$, $F\{-\} = n\circledR G\{-\}$, and $F\{-\} = n\oslash G\{-\}$. By induction hypothesis, as above.
- Case $F\{-\} = \forall x.G\{-\}$. Assume $x \notin \mathfrak{D}(\sigma)$.
  We have $[\![\sigma(F\{A\})]\!]_w = \bigcap_{n\in\Lambda} [\![\sigma(G\{A\})\{x\leftarrow n\}]\!]_w$. By induction hypothesis, $[\![\sigma(G\{A\})\{x\leftarrow n\}]\!]_w = [\![\sigma(G\{B\})\{x\leftarrow n\}]\!]_w$, for all n.
  Therefore $\bigcap_{n\in\Lambda} [\![\sigma(G\{A\})\{x\leftarrow n\}]\!]_w = \bigcap_{n\in\Lambda} [\![\sigma(G\{B\})\{x\leftarrow n\}]\!]_w$.
  But $\bigcap_{n\in\Lambda} [\![\sigma(G\{B\})\{x\leftarrow n\}]\!]_w = [\![\sigma(F\{B\})]\!]_w$.
- Case $F\{-\} = \text{И}x.G\{-\}$. Assume $x \notin \mathfrak{D}(\sigma)$.
  (Left to right inclusion) Pick $P \in [\![\sigma(\text{И}x.G\{A\})]\!]_w$. Then there is $n \notin fn^w(\sigma(G\{A\})) \cup fn(P)$ such that $P \in [\![\sigma(G\{A\})\{x\leftarrow n\}]\!]_w$.
  Let $m \notin fn^w(\sigma(G\{A\})) \cup fn^w(\sigma(G\{B\})) \cup fn(P)$, let $\tau = \{m\leftrightarrow n\}$. Let u be the restriction of w to the free propositional variables of $G\{A\}$, we have $[\![\sigma(G\{A\})\{x\leftarrow n\}]\!]_w = [\![\sigma(G\{A\})\{x\leftarrow n\}]\!]_u$. By Theorem 4.18(2), $\tau(P) = P \in \tau([\![\sigma(G\{A\})\{x\leftarrow n\}]\!]_u) = [\![\sigma(G\{A\})\{x\leftarrow m\}]\!]_u = [\![\sigma(G\{A\})\{x\leftarrow m\}]\!]_w$.
  By the induction hypothesis we conclude $P \in [\![\sigma(G\{B\})\{x\leftarrow m\}]\!]_w$.
  Since $m \notin fn^w(\sigma(G\{B\}))$, we obtain $P \in [\![\sigma(F\{B\})]\!]_w$.
  (Right to left inclusion) Symmetrical.
- Case $F\{-\} = X$. Then $F\{A\} = X = F\{B\}$ and we have $[\![\sigma(F\{A\})]\!]_w = [\![\sigma(X)]\!]_w = w(X) = [\![\sigma(X)]\!]_w = [\![\sigma(F\{B\})]\!]_w$.
- Case $F\{-\} = \boldsymbol{\nu}Z.G\{-\}$.
  (Left to right inclusion) Take $P \in [\![\sigma(\boldsymbol{\nu}Z.G\{A\})]\!]_w$. By definition, $P \in \Psi$ where $\Psi \in \mathbb{P}_{M_A}$, and $\Psi \subseteq [\![\sigma(G\{A\})]\!]_{w[Z\leftarrow\Psi]}$, and $M_A = fn^w(\sigma(\boldsymbol{\nu}Z.G\{A\}))$.
  Let $M_B = fn^w(\sigma(\boldsymbol{\nu}Z.G\{B\}))$. We have $\Psi \in \mathbb{P}_{M_A\cup M_B}$, by Lemma 4.2(1).
  By induction hypothesis, $\Psi \subseteq [\![\sigma(G\{B\})]\!]_{w[Z\leftarrow\Psi]}$. Then we conclude that $\Psi \subseteq [\![\sigma(G\{B\})]\!]_{w[Z\leftarrow\Psi]}$, so $\Psi \subseteq \mathsf{gfix}_{M_A\cup M_B}(\boldsymbol{\lambda}\Phi.[\![\sigma(G\{B\})]\!]_{w[Z\leftarrow\Phi]})$.
  Since $fn^w(\sigma(\boldsymbol{\nu}Z.G\{B\})) \subseteq M_A \cup M_B$, by Lemma 4.25, we conclude $\Psi \subseteq [\![\sigma(\boldsymbol{\nu}Z.G\{B\})]\!]_w$. Therefore $P \in [\![\sigma(\boldsymbol{\nu}Z.G\{B\})]\!]_w$.
  (Right to left inclusion) Handled symmetrically.