A Spatial Logic for Concurrency (Part II)

Luís Caires [a,*], Luca Cardelli [b]

[a] *Departamento de Informática, FCT/UNL, Portugal*

[b] *Microsoft Research, Cambridge, UK*

Abstract

We present a modal logic for describing the spatial organization and the behavior of distributed systems. In addition to standard logical and temporal operators, our logic includes spatial operations corresponding to process composition and name hiding, and a fresh name quantifier. In Part I of this work we study the fundamental semantic properties of our logic; the focus of the present Part II is on proof theory. The main contributions are a sequent-based proof system for our logic, and a proof of cut-elimination for its first-order fragment.

1 Introduction

We develop a logic to describe properties of distributed concurrent systems, for specification and model-checking purposes; we believe that the peculiar characteristics of such systems justify the introduction of new logical constructs.

Our first emphasis is on *distributed* systems, meaning that we should be able to talk about properties of distinct subsystems, such as subsystems that reside at different locations, and subsystems that privately share hidden resources. For this purpose, we introduce *spatial* (as opposed to temporal) logical operators; for example, we may talk about a property holding somewhere (as opposed to sometimes). Our second emphasis is on *concurrent* systems: we want a logic that unambiguously talks about concurrency and (nowadays) privacy. For this purpose, the intended model of our logic is built explicitly from a standard process calculus (an asynchronous π-calculus). Our formulas denote collections of processes subject to certain closure conditions, with some logical operators mapping directly to process composition and name hiding.

[*] Corresponding author.
Email addresses: Luis.Caires@di.fct.unl.pt (Luís Caires), luca@microsoft.com (Luca Cardelli).

In Part I of this paper [2,4] we study this intended model, which is used here to establish the soundness of the logical rules. The central focus of this Part II, however, is proof theory. We regularize and generalize the logics introduced in [1,10,11], and we prove a cut-elimination result for the first-order fragment, including cut-elimination for a fresh name quantifier (*cf.* Nominal Logic [18]).

A formula in our logic describes a property of a particular part of a concurrent system (a *world*) at a particular time; therefore it is modal in space as well as in time. In our sequents, formulas are indexed by the worlds they predicate over [21], so a sequent can talk about many distinct worlds at once. Each sequent incorporates also a finite set of constraints over the worlds, including process reduction and congruence constraints. In general, the constraint structure can be fashioned as an algebra [24]; which in our case is a relatively complex process algebra.

The fragment of our logic that deals with process composition is relatively straightforward: composition shows up in the logic as a tensor, which is strongly related to linear connectives. The sequent-style presentation of this fragment should look relatively familiar, except for the constraints part. The relevant constraints are essentially constraints over a (concurrency) monoid, with some specific interactions with reduction. Along these lines, we could also easily add an explicit structure of locations to the process calculus, and related logical operators, as done in [10].

Far less obvious is what to do about hiding of private resources, which is represented in π-calculus by the name hiding operator. The hiding of a name in a process should correspond, logically, to a "hiding quantifier" that binds a private name in a formula; such a formula could then describe the use of that private name in the process. The study of such a quantifier, from a logical point of view, was started in [5,1], and later independently in [11]. Our current understanding is that it is best to decompose such a hiding quantifier into two operators: a modal version of the fresh quantifier of Gabbay and Pitts [14], and a logical operator, called *revelation* [11], that relates to name hiding in strong analogy to the way tensor relates to process composition. A simple combination of fresh quantification and revelation then yields hiding, in the intuitive sense that if something is hidden, we can choose to name it (reveal it) by any name that is fresh.

Many natural examples of use of our logic involve recursive formulas. Two typical examples of recursion that attract us in our context are: (1) a process having an arbitrary number of hidden resources, and (2) a process generating an infinite supply of fresh names. Particularly, the interaction of recursion and freshness is semantically quite challenging, and was investigated in Part I.

Structurally, our logic consists of a collection of left-right rules for logical operators, including essentially the standard rules of classical sequent calculus, plus the ones for temporal and spatial operators. In addition, there are special

$\langle S\rangle\, \Gamma \vdash \Delta$	Sequents, of the form $\langle S\rangle\, u_1 : A_1, \ldots , u_n : A_n \vdash v_1 : B_1, \ldots , v_m : B_m$
A_i, B_i	Formulas
u_i, v_j	Indexes, members of a process algebra (the worlds)
S	finite set of constraints (*e.g.*, equations, reductions)

Fig. 1. Sequents.

rules about the worlds: they add meaning to the logical operators, allowing us to capture deep properties of process calculi without interfering very much with the core left-right rules.

We highlight here the left and right rules for composition, $A|B$, which include many of the interesting features of our sequents.

Sequents (Figure 1) have the form $\langle S\rangle\, \Gamma \vdash \Delta$, where $\langle S\rangle$ is a finite set of constraints, and Γ , Δ are multisets of indexed formulas. Constraints include equality constraints, $u \doteq v$, stating that u and v represent structurally congruent processes.

$$[x \text{ and } y \text{ not free in the conclusion}]$$

$$\frac{\langle S, u \doteq x\,|y\rangle\, \Gamma, x : A, y : B \vdash \Delta}{\langle S\rangle\, \Gamma, u : A|B \vdash \Delta}\ (|\mathrm{L})$$

$$\frac{\langle S\rangle\, \Gamma \vdash v : A, \Delta \quad \langle S\rangle\, \Gamma \vdash t : B, \Delta \quad u \doteq_S v|t}{\langle S\rangle\, \Gamma \vdash u : A|B, \Delta}\ (|\mathrm{R})$$

The ($|$R) rule says: if we can show that index v satisfies formula A (i.e, that A holds at world v, written $v : A$), and that t satisfies B, and if we can show from the constraints in S that u is structurally congruent to $v|t$, then we can conclude that u satisfies $A|B$. Hence, the reading of this logical rules incorporates much of the intended satisfaction semantics [21]. The ($|$L) rule features the assumption "x and y not free in the conclusion (of the rule)". This assumption means, in particular, that x and y are completely generic and unconstrained variables. A reading is: to show that $u : A|B$ entails Δ, we must show that for an arbitrary decomposition of u as $x\,|y$, we have that $x : A$ and $y : B$ entail Δ.

Composition also has a number of "rules about the world", as mentioned above. Here is a simple one:

$$\frac{\langle S, u \doteq \mathbf{0}\rangle\, \Gamma \vdash \Delta \quad u|v \doteq_S \mathbf{0}}{\langle S\rangle\, \Gamma \vdash \Delta}\ (\mathrm{S}|\mathbf{0})$$

Note that these world rules do not involve the logical connectives (we have $\Gamma \vdash \Delta$ above and below), and instead affect the $\langle S\rangle$ part. In most process calculi we have that if $u|v$ is structurally congruent to $\mathbf{0}$ then both u and v are structurally congruent to $\mathbf{0}$. This property does not derive from (|L) and (|R), but is embedded in (S|**0**). The rule reads as follows: if we can already infer from the S part of the constraints that $u|v \doteq \mathbf{0}$, and we have an additional constraint that $u \doteq \mathbf{0}$, that constraint is redundant and we can remove it. In this style, we can incorporate many peculiar properties of process calculi as world rules; many such rules analyze the consequences of an equation between two spatial operators (above, | vs. **0**), and are listed in Figure 12. All such rules have a similar reading in terms of eliminating "redundant" constraints.

Because of the regular left-right structure of our core rules, cut elimination falls largely along predictable lines; the indexes do not hinder, and rules such as (S|**0**) can be dealt with separately. The main difficulty is in the cut elimination case for the freshness quantifier. As in Nominal Logic, the result depends on an "equivariance" property of the logic [18], which is used to perform an α-conversion of fresh names over a whole derivation. Equivariance is embedded, in our case, in the (TL/TR) rules in Figure 7. Expressing these rules in the general case of open formulas, requires introducing explicit transpositions over formulas, which entail some technical complications.

Related Work A logic for a process calculus including a tensor operator and a hiding quantifier was developed by Luís Caires in [5,1], but a satisfactory semantic treatment for the latter connective was not achieved before the contributions of [11,2]. Andy Gordon was a coauthor with Luca Cardelli of initial versions of spatial logics for the Ambient Calculus [10,11], which also investigated connections with linear logic. The present paper contains the first presentation of such a logic as a proper sequent calculus. Moreover, we now target the logic towards a more standard π-calculus.

The first main difference between our logic and standard logics of concurrency (e.g. [15]) is the presence in our case of a tensor operator that corresponds to process composition. Usually, those other logics require formulas to denote processes up to bisimulation, which is difficult to reconcile with a tensor operator that can make distinctions between bisimilar processes (however, such an operator was anticipated by Dam [12]). In our case, we only require formulas to denote processes up to structural equivalence, so that a tensor operator makes easy sense. Sangiorgi, Hirshkoff and Lozes have shown, for a closely related logic, that the equivalence induced by the logic is then essentially structural equivalence [20,16]. Compositional proof systems for behavioral equivalences on the π-calculus have also been recently proposed by Dam [13].

The work of Gabbay and Pitts on the freshness quantifier [14] has become central to our logic. The work of O'Hearn and Pym on Bunched Logics [17] and of Reynolds on Separation Logic [19] is closely related to ours, at least in intent. Spatial logics for trees and graphs have also been investigated in [9,7].

The style in which our logic is formalized is an extension of work by Alex Simpson [21,22], and is also related, at least superficially, to labeled deductive systems [24]. The use of formal transpositions, adopted here as a technique for manipulating freshness constraints, turned out to be useful also in the setting of programming languages for semi-structured data [8]. A decidable and complete propositional fragment of a related logic has been recently investigated [6].

Structure of the paper In Section 2 we recall the syntax and semantics of our logic of Part I. In Section 3 we present the various ingredients that constitute the proof system. In Section 3.1 we introduce the π-algebra that is used in the constraints and indexes of our sequents. A π-algebra is an abstraction of π-calculi, incorporating most of the characteristic properties of composition and hiding. In Section 3.5 we introduce our sequent calculus, which can be shown sound by an interpretation in the model of Part I [2]. In Section 4 we show how recursive properties can be fully handled inside our logic. In Section 5 we investigate proof theory, and in particular cut elimination for the first-order fragment of our logic. In Section 6 we go through a set of basic examples, to illustrate the expressive power of the logic. In the Appendix, we collect proofs of results.

2 The Logic and its Semantics

In this section, we review the syntax and semantics of our spatial logic for concurrency. Our intended model [4] is a fixed nominal process calculus (we use asynchronous π-calculus) over a set of pure names Λ ; let $\mathcal{P}$ be the collection of such processes. On $\mathcal{P}$ is defined the relation $\equiv$ of *structural congruence*, that equates processes that possess the same spatial structure, and the binary relation $\rightarrow$ of *reduction*, that captures the dynamic behavior of processes. A *property* is a set of processes; a subset of $\mathcal{P}$. Then, a formula of our logic denotes a property, namely, it denotes the collection of processes satisfying that formula.

Given the sets $\mathcal{V}$ and $\mathcal{Z}$ of name variables and propositional variables, respectively, formulas are defined in Fig. 2. They include classical propositional connectives, $\mathbf{F}$, $\wedge$, $\Rightarrow$, and the basic spatial operators: $A|B$ (the tensor, representing the parallel composition of processes), $\mathbf{0}$ (the unit of the tensor, representing the collection of void processes), and $A \triangleright B$ (the linear implication associated with the tensor). This last operator corresponds to context-system specification of processes, which are the concurrency-theory equivalent of pre/post conditions.

First-order quantification allows us to quantify over the set of pure names

m,n,p ::=		Name Terms	$(m,n,p \in \mathcal{N})$
	x	Name variable	$(x \in \mathcal{V})$
	$(m \leftrightarrow n)p$	Transposition term	
A,B ::=		Formulas	$(A,B \in \Phi)$
	$\mathbf{F}$	False	
	$(m \leftrightarrow n)A$	Transposition	
	$A \wedge B$	Conjunction	
	$A \Rightarrow B$	Implication	
	$\mathbf{0}$	Void	
	$A \vert B$	Composition	
	$A \triangleright B$	Guarantee	
	$n \circledR A$	Revelation	
	$n \oslash A$	Hiding	
	$m\langle n \rangle$	Message	
	$\Diamond A$	Next	
	$\forall x.A$	First-order universal quantification	
	$\text{И}x.A$	Freshness quantification	
	X	Propositional variable	$(X \in \mathcal{X})$
	$\forall X.A$	Second-order universal quantification	

Fig. 2. Formulas

Λ of the π-calculus. Pure names $(n,m,p \in \Lambda)$ are represented in our logic by *name terms*: a name variable x denotes some name, while a transposition term $(m \leftrightarrow n)p$ denotes the name obtained by applying the transposition of the names denoted by the name terms m and n to the name denoted by the name term p. The use of name terms in formulas and the presence of a explicit transposition formula $(m \leftrightarrow n)A$ are some convenient additions we introduce here to the basic logic of [4,3] (*cf.*, transposition types in [8]). We do not allow pure names to appear in the syntax of formulas: only name variables and their transpositions are used there. As discussed below, these additions can be integrated in a fairly straightforward way into the semantic framework already developed in [4].

Name hiding induces a pair of adjunct logical operators. The formula $n \circledR A$

means that a hidden name, denoted by the name term n, exists in a restricted scope that satisfies property A. It is matched by a π-calculus term $(\boldsymbol{\nu} n)u$ provided that u satisfies A and n denotes the name n (see the semantic clause for $\mathsf{n} \circledR A$ in Fig. 3, inference rule for ($\circledR$R) in Fig. 8, and the example in Section 6.5; see [11,4] for further discussion.) The formula $A \oslash \mathsf{n}$ is the logical adjunct of $\mathsf{n} \circledR A$, indicating that A can be satisfied by a process after hiding the name denoted by n.

The notion of *fresh name* is introduced by a quantifier $\mathsf{И}x.A$; a process P satisfies $\mathsf{И}x.A$ if P satisfies A for some name fresh in the process P and in the formula $\mathsf{И}x.A$. $\mathsf{И}x.A$ is defined along the lines of the freshness quantifier of Gabbay-Pitts [14,18], and its semantics is designed to be compatible with recursive formulas.

A logical operator $\mathsf{n}\langle\mathsf{m}\rangle$ allows us to assert that a process consists precisely of a message m over a channel n, giving us some minimal power to observe its behavior. A next-step temporal operator, $\Diamond A$, allows us to talk about a process after a single (unspecified) reduction step. Finally, we have a second-order quantifier and related propositional variables.

In $\forall x.A$, $\mathsf{И}x.A$ (and $\forall X.A$), the variables x (and X) are bound with scope the formula A. We assume defined on formulas the standard relation $\equiv_\alpha$ of α-conversion (safe renaming of bound variables), but we never implicitly take formulas "up to α-conversion": our manipulation of variables via α-conversion steps is always quite explicit. The set $fv(A)$ of *free name variables* in A, and the set $fpv(A)$ of *free propositional variables* in A, are defined in the usual way. Then, we define the set of *logically free variables* of a formula A by $lfv(A) \triangleq fv(A) \cup fpv(A)$. If m is a name term and A is a formula then $A\{x \leftarrow \mathsf{m}\}$ denotes the formula obtained by replacing of all free occurrences of x in A by the name term m, renaming bound name variables as needed to avoid capture of name variables occurring in the name term m. We also define the set $ft(A)$ of *free terms* in A, to be the set of all maximal name terms in A that do not contain occurrences of variables bound in A; and the set of *logically free terms* of a formula A by $lft(A) \triangleq ft(A) \cup fpv(A)$.

We now review the semantics of our logic; if needed, further details can be found in [4]. The denotation of formulas is defined in terms of sets of processes that satisfy certain natural closure conditions. These conditions are motivated by the following facts. First, we expect satisfaction to be closed under structural congruence (processes with the same spatial structure must satisfy the same formulas). Second, a property should depend only on a finite set of relevant names (related to the denotation of the free name variables of a formula); such a set of names is called the *support* of the property. The collection of all properties has the structure of a Boolean algebra under set inclusion, so we naturally get propositional connectives in the logic. The collection of all properties has also the structure of a commutative quantale, due to the parallel composition operator over processes; this induces the basic spatial connectives of the logic. Other process operators induce further spatial connectives.

The support of a set of processes is defined using name transpositions. A transposition $\{m\leftrightarrow n\}$ acts on a process P by swapping all occurrences (free and bound) of the names n and m in the process P. From [4], we recall

Definition 2.1 (PSet) *A* property set *is a set of processes* Ψ *such that*

(1) For all Q*, if* $P \in \Psi$ *and* $P \equiv Q$ *then* $Q \in \Psi$*.*
(2) There is a finite set of names N *such that, for all* $n, m \notin N$*, if* $P \in \Phi$ *then* $P\{n\leftrightarrow m\} \in \Phi$*.*

We denote by $\mathbb{P}$ the set of all Psets. Every Pset $\Phi \in \mathbb{P}$ has a least support [18,4], that we denote by $supp(\Phi)$. Hence, in our semantics, the denotation of any formula A is given by a Pset $[\![A]\!] \in \mathbb{P}$. Since a formula A may contain free occurrences of propositional and name variables, its denotation depends on the denotation of such variables, which is given by a valuation. A valuation v is a finite mapping assigning to each name variable in its domain a name in Λ (the set of π-calculus pure names), and each propositional variable in its domain a Pset in $\mathbb{P}$. The application of transpositions to Psets and valuations is defined pointwise [4]. The following semantic characterization for the "free" names of a formula A under a valuation [4] is also useful.

Definition 2.2 (Free Names under 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*

$$fn^v(A) \triangleq \bigcup\{v(x) \mid x \in fv(A)\} \cup \bigcup\{supp(v(X)) \mid X \in fpv(A)\}$$

Intuitively, $fn^v(A)$ is basically $fn(v(A))$ except that we set $fn(X) \triangleq supp(v(X))$ for any $X \in fpv(A)$, hence $fn^v(A) = fn(A)$ for closed A. The set $fn^v(A)$ is useful in the definition of the semantics of the fresh name quantifier, where the quantification witness must be fresh with respect to the property set denoted by a formula that in general may contain free occurrences of propositional (and name) variables.

The semantics of formulas is defined in Fig. 3. The denotation mapping $[\![-]\!]_v$ satisfies certain fundamental properties, listed in the next theorem.

Theorem 2.3 *For all formulas* A *and valuations* v

(1) $[\![A]\!]_v \in \mathbb{P}$ *with* $supp([\![A]\!]_v) \subseteq fn^v(A)$*.*
(2) For all transpositions τ*,* $\tau([\![A]\!]_v) = [\![A]\!]_{\tau(v)}$*.*
(3) Let $M = fn^v(\text{И}x.A) \cup fn(P)$*. If* $P \in [\![A]\!]_{v[x\leftarrow p]}$ *for some* $p \notin M$*, then* $P \in [\![A]\!]_{v[x\leftarrow p]}$ *for all* $p \notin M$*.*

Proof. (1-2) By induction on the structure of the formula A; a straightforward adaptation of the proof of Theorem 4.21 in [4]. (3) A consequence of (2). ■

$$
\begin{aligned}
\llbracket x \rrbracket_v &\triangleq v(x)\\
\llbracket (m \leftrightarrow n)p \rrbracket_v &\triangleq \{\llbracket m \rrbracket_v \leftrightarrow \llbracket n \rrbracket_v\}\llbracket p \rrbracket_v
\end{aligned}
$$

$$
\begin{aligned}
\llbracket \mathbf{F} \rrbracket_v &\triangleq \emptyset\\
\llbracket (m \leftrightarrow n)A \rrbracket_v &\triangleq \{\llbracket m \rrbracket_v \leftrightarrow \llbracket n \rrbracket_v\}\llbracket A \rrbracket_v\\
\llbracket A \wedge B \rrbracket_v &\triangleq \llbracket A \rrbracket_v \cap \llbracket B \rrbracket_v\\
\llbracket A \Rightarrow B \rrbracket_v &\triangleq \{P \mid \mathit{if}\ P \in \llbracket A \rrbracket_v\ \mathit{then}\ P \in \llbracket B \rrbracket_v\}\\
\llbracket \mathbf{0} \rrbracket_v &\triangleq \{P \mid P \equiv \mathbf{0}\}\\
\llbracket A|B \rrbracket_v &\triangleq \{P \mid \mathit{Exists}\ Q, R.\ P \equiv Q|R\ \mathit{and}\ Q \in \llbracket A \rrbracket_v\ \mathit{and}\ R \in \llbracket B \rrbracket_v\}\\
\llbracket A \triangleright B \rrbracket_v &\triangleq \{P \mid \mathit{Forall}\ Q.\ \mathit{if}\ Q \in \llbracket A \rrbracket_v\ \mathit{then}\ P|Q \in \llbracket B \rrbracket_v\}\\
\llbracket n \circledR A \rrbracket_v &\triangleq \{P \mid \mathit{Exists}\ Q.\ P \equiv (\boldsymbol{\nu}\llbracket n \rrbracket_v)Q\ \mathit{and}\ Q \in \llbracket A \rrbracket_v\}\\
\llbracket A \oslash n \rrbracket_v &\triangleq \{P \mid (\boldsymbol{\nu}\llbracket n \rrbracket_v)P \in \llbracket A \rrbracket_v\}\\
\llbracket m\langle n \rangle \rrbracket_v &\triangleq \{P \mid P \equiv \llbracket m \rrbracket_v\langle\llbracket n \rrbracket_v\rangle\}\\
\llbracket \forall x.A \rrbracket_v &\triangleq \textstyle\bigcap_{n \in \Lambda} \llbracket A \rrbracket_{v[x \leftarrow n]}\\
\llbracket \text{И}x.A \rrbracket_v &\triangleq \textstyle\bigcup_{n \notin fn^v(\text{И}x.A)} (\llbracket A \rrbracket_{v[x \leftarrow n]} \setminus \{P \mid n \in fn(P)\})\\
\llbracket \Diamond A \rrbracket_v &\triangleq \{P \mid \mathit{Exists}\ Q.\ P \rightarrow Q\ \mathit{and}\ Q \in \llbracket A \rrbracket_v\}\\
\llbracket X \rrbracket_v &\triangleq v(X)\\
\llbracket \forall X.A \rrbracket_v &\triangleq \textstyle\bigcap_{\Psi \in \mathbb{P}} \llbracket A \rrbracket_{v[X \leftarrow \Psi]}
\end{aligned}
$$

Fig. 3. Denotation of terms and formulas.

3 The Proof System

In this section, we present a sequent calculus based proof system for our logic. The inference rules of our system follow the pattern one expects from a Gentzen-style sequent calculus, that is, a system where there is a symmetric pair of left and right introduction rules for each logical connective. As discussed in the introduction, sequents have the form $\langle S \rangle\, \Gamma \vdash \Delta$, where $\langle S \rangle$ is a finite set of constraints, and Γ, Δ are multisets of index-tagged formulas. Indexes denote the worlds (the processes) of our modal logic. Such indexes are elements of the term π-algebra.

We now introduce π-algebras, and constraint theories over the term π-algebra. A π-algebra is a sorted algebra, with a sort for names, a sort for processes, and a sort for collections of processes (properties), and equipped with the basic process operations of composition, name hiding and name transposition. Hence, many process calculi are π-algebras, in particular the asynchronous π-calculus Aπ which is the intended model of our logic.

Definition 3.1 (π-algebra) *A π-algebra is a structure*

$$\Pi = \langle \mathcal{L}, \mathcal{P}, \mathcal{C}, \mathbf{0}, |, \boldsymbol{\nu}, (\leftrightarrow)_{\mathcal{L}}, (\leftrightarrow)_{\mathcal{P}}, (\leftrightarrow)_{\mathcal{C}} \rangle$$

such that $\mathcal{L}$ is a countable set of labels (ℓ)*, $\mathcal{P}$ is the set of* processes (P, Q, R)*, $\mathcal{C}$ is a collection of properties (F, G), and*

- $\mathbf{0}$ *(void) is a distinguished process in $\mathcal{P}$*
- $-|-$ *(*composition*) is an operation $\mathcal{P} \times \mathcal{P} \to \mathcal{P}$*
- $(\boldsymbol{\nu}-)-$ *(*name hiding*, a.k.a. restriction) is an operation $\mathcal{L} \times \mathcal{P} \to \mathcal{P}$*
- $(-\leftrightarrow-)_{\mathcal{L}}-$ *(*transposition on labels*) is an operation $\mathcal{L} \times \mathcal{L} \times \mathcal{L} \to \mathcal{L}$*
- $(-\leftrightarrow-)_{\mathcal{P}}-$ *(*transposition on processes*) is an operation $\mathcal{L} \times \mathcal{L} \times \mathcal{P} \to \mathcal{P}$*
- $(-\leftrightarrow-)_{\mathcal{C}}-$ *(*transposition on properties*) is an operation $\mathcal{L} \times \mathcal{L} \times \mathcal{C} \to \mathcal{C}$*

We refer to the $\mathcal{L}$ part of a π-algebra Π by $\Pi_{\mathcal{L}}$, and likewise for the remaining components (*e.g.*, $\Pi_{\mathcal{P}}$). For example, the asynchronous π-calculus Aπ is the π-algebra where A$\pi_{\mathcal{L}}$ is the set of π-calculus names, A$\pi_{\mathcal{P}}$ is the set of π-calculus processes, and $(m \leftrightarrow n)P$ denotes the process $\{m\leftrightarrow n\}\cdot P$ obtained by swapping the names m, n in the process P.

Of particular interest to us is the term π-algebra, which supports the syntactical manipulation of (schematic) processes and names in a general way.

Definition 3.2 (Term π-algebra) *Consider given a set $\mathcal{V}$ of* names variables*, a set $\mathcal{Z}$ of* process variables*, and a set $\mathcal{X}$ of* propositional variables*. The term π-algebra is the free π-algebra*

$$\mathbf{P} = \langle \mathcal{N}, \mathcal{I}, \mathcal{F}, \mathbf{0}, |, \boldsymbol{\nu}, (\leftrightarrow)_{\mathcal{N}}, (\leftrightarrow)_{\mathcal{I}}, (\leftrightarrow)_{\mathcal{F}} \rangle$$

where $\mathcal{N}$ is the set of all terms freely built from the variables in $\mathcal{V}$ and name transposition, $\mathcal{F}$ is the set of all terms freely built from the variables in $\mathcal{X}$ and name transposition, and $\mathcal{I}$ is the set of all terms freely built from the variables in $\mathcal{Z}$, name terms in $\mathcal{N}$, and the process operations $\mathbf{0}, |, \boldsymbol{\nu}$ and $(\leftrightarrow)_{\mathcal{I}}$. In the term π-algebra, the labels $\mathcal{N}$ are called name terms*, the processes $\mathcal{I}$ are called* indexes*, and the properties $\mathcal{F}$ are called* propositional terms*. We use* *the meta-variables*

$x, y, z \in \mathcal{V}$	(Name Variables)	$\mathsf{m}, \mathsf{n}, \mathsf{p} \in \mathcal{N}$	(Name Terms)
$\mathscr{x}, \mathscr{y}, \mathscr{z} \in \mathcal{Z}$	(Process Variables)	$u, v, t \in \mathcal{I}$	(Indexes)
$X, Y, Z \in \mathcal{X}$	(Propositional Variables)	$F, G, H \in \mathcal{F}$	(Propositional Terms)

$$\gamma, \delta \in \mathcal{G} \triangleq \mathcal{F} \cup \mathcal{N} \quad \xi \in \mathcal{T} \triangleq \mathcal{I} \cup \mathcal{G}$$

The elements of the term π-algebra that we have called *indexes* denote elements of the intended process algebra (processes, the worlds of our modal logic), while the *name terms* denote the pure names used in processes. For example, x, $(x \leftrightarrow y)z$ and $(x \leftrightarrow ((y \leftrightarrow z)x))z$ are name terms, while $\mathscr{x}$, $(x \leftrightarrow y)\mathscr{x}$ and $\mathscr{x}\,|(\boldsymbol{\nu}(x \leftrightarrow y)z)\mathscr{y}$ are indexes. N.B., in the term π-algebra, $(\mathsf{m} \leftrightarrow \mathsf{n})P$ (respectively, $(\mathsf{m} \leftrightarrow \mathsf{n})\mathsf{p}$) is a particular index (respectively, name term) in which transposition is interpreted as a formal operation.

A *propositional term* F denotes a property (a collection of processes). The intention is that the process denoted by the index u belongs to the property denoted by $(\mathsf{n} \leftrightarrow \mathsf{m})F$ whenever the process denoted by $(\mathsf{n} \leftrightarrow \mathsf{m})u$ belongs to the property denoted by F.

Definition 3.3 (Interpretation) *Given any π-algebra Π, an* interpretation $\mathcal{J}$ of the term π-algebra into Π *is a triple of mappings $\mathcal{J}_{\mathcal{L}} : \mathcal{V} \to \Pi_{\mathcal{L}}$ and $\mathcal{J}_{\mathcal{P}} : \mathcal{Z} \to \Pi_{\mathcal{P}}$, $\mathcal{J}_{\mathcal{C}} : \mathcal{X} \to \Pi_{\mathcal{C}}$.*

Every interpretation $\mathcal{J}$ extends to the unique homomorphism $\hat{\mathcal{J}} : \mathbf{P} \to \Pi$ of π-algebras in the standard way. Note that the term π algebra can be straightforwardly interpreted into any nominal calculi (e.g., the π-calculus, the ambient calculus), by mapping the (formal) operators of the term π-algebra into the corresponding process model operators.

Definition 3.4 (Algebraic free variables) *Given an index, name term, or propositional term ξ, we denote by $afv(\xi)$ its set of algebraic free (name, process and property) variables, defined simply as the collection of all the variables in $\mathcal{V}$, $\mathcal{Z}$ and $\mathcal{X}$ occurring in such terms.*

Remark 3.5 A variable x is algebraic free, in, *e.g.*, the index $(\boldsymbol{\nu}x)\mathbf{0}$, while the name n is not free in the usual sense in the π-calculus process $(\boldsymbol{\nu}n)\mathbf{0}$. In particular, a π-substitution acts on all algebraic free variables of indexes and name terms. *E.g.*, if $u \triangleq (\boldsymbol{\nu}x)(\mathscr{x}\,|\mathscr{y})$, then $u\{x \leftarrow y\}\{\mathscr{x} \leftarrow (\boldsymbol{\nu}x)z\} = (\boldsymbol{\nu}y)((\boldsymbol{\nu}x)z|\mathscr{y})$.

Definition 3.6 (π-substitution) *A π-substitution is an interpretation from* $\mathbf{P}$ *into* $\mathbf{P}$.

Every π-substitution σ extends to the homomorphism $\hat{\sigma} : \mathbf{P} \to \mathbf{P}$ of term π-algebras that acts as a syntactic substitution on indexes. We denote by $\{x \leftarrow \mathsf{n}\}$ the π-substitution that maps x into n and acts like the identity elsewhere, and likewise for $\{x\leftarrow u\}$ and $\{X\leftarrow F\}$. If $I_{\mathcal{L}}$ is a mapping $\mathcal{V}\rightarrow\mathcal{N}$ then we note by $I_{\mathcal{L}}\{x\leftarrow\mathsf{n}\}$ the mapping $I'_{\mathcal{L}}$ such that $I'_{\mathcal{L}}(z)\triangleq I(z)$ for $z\neq x$ and $I'_{\mathcal{L}}(x)\triangleq n$. Likewise, if $\mathcal{J}$ is an interpretation, we write $\mathcal{J}\{x\leftarrow n\}\{x\leftarrow u\}$ for the interpretation that behaves like $\mathcal{J}$ except that it maps x to n and x to u.

Usually, we write just σ for the homomorphic extension $\hat{\sigma}$ of a π-substitution σ.

3.2 Constraint theories

The worlds of our logic relate to each other both by spatial and temporal constraints: spatial constraints express that the processes denoted by the equated indexes have the same spatial structure (*cf.* π-calculus structural congruence), while temporal constraints express that a process has a reduction to another process (*cf.* π-calculus reduction). Intuitively, a constraint theory defines a class of models for the spatial logic, namely those models that satisfy all of its spatial and temporal constraints.

Definition 3.7 (Constraint and constraint theory) *A* constraint c *is either an index, name or property equation, a reduction, a name or property apartness, defined by*

$$
\begin{array}{llll}
c ::= & & \text{Constraints} & \\
 & u \doteq v & \textit{Index equation} & (u, v \in \mathcal{I}) \\
 & \mathsf{n} \doteq \mathsf{m} & \textit{Name equation} & (\mathsf{n}, \mathsf{m} \in \mathcal{N}) \\
 & \mathsf{m} \# \mathsf{n} & \text{Name apartness} & (\mathsf{m}, \mathsf{n} \in \mathcal{N}) \\
 & F \doteq G & \textit{Property equation} & (F, G \in \mathcal{F}) \\
 & \mathsf{m} \# F & \textit{Property apartness} & (\mathsf{m} \in \mathcal{N}, F \in \mathcal{F}) \\
 & u \rightarrow v & \textit{Reduction} & (u, v \in \mathcal{I})
\end{array}
$$

A constraint theory *is a finite set of constraints.*

An equation $u \doteq v$ states that the indexes u and v denote structurally congruent processes, while a reduction $u \rightarrow v$ asserts that the process denoted by the index u reduces to the process denoted by the index v.

In order to handle freshness constraints explicitly, we also introduce apartness constraints: $\mathsf{m}\#\mathsf{n}$ meaning that the name terms m and n denote distinct names, and $\mathsf{m}\# F$ meaning that the name term m denotes a name distinct from any name in the (finite) support of the property (set of processes) denoted by the propositional term F (so the name n is fresh in such a property).

A constraint $F \doteq G$ asserts that the propositional terms F and G denote the same property.

(Basic)

$\xi \doteq \xi' \in S \Rightarrow \xi \doteq_S \xi'$ (Basic Equ)

$\gamma \# \gamma' \in S \Rightarrow \gamma \#_S \gamma'$ (Basic Apart)

$u \to v \in S \Rightarrow u \to_S v$ (Basic Red)

(Spatial)

$u|\mathbf{0} \doteq_S u$ (Sp Void)

$u|v \doteq_S v|u$ (Sp Par Comm)

$(u|v)|t \doteq_S u|(v|t)$ (Sp Par Assoc)

$(\boldsymbol{\nu} n)\mathbf{0} \doteq_S \mathbf{0}$ (Sp Res Void)

$(\boldsymbol{\nu} n)(\boldsymbol{\nu} n)u \doteq_S (\boldsymbol{\nu} n)u$ (Sp Res Res)

$(\boldsymbol{\nu} m)(\boldsymbol{\nu} n)u \doteq_S (\boldsymbol{\nu} n)(\boldsymbol{\nu} m)u$ (Sp Res Comm)

$(\boldsymbol{\nu} n)(u|(\boldsymbol{\nu} n)v) \doteq_S ((\boldsymbol{\nu} n)u)|(\boldsymbol{\nu} n)v$ (Sp Res Par)

(Congruence)

$\xi \doteq_S \xi$ (Cong Refl)

$\xi \doteq_S \xi' \Rightarrow \xi' \doteq_S \xi$ (Cong Sym)

$\xi \doteq_S \xi', \xi' \doteq_S \xi'' \Rightarrow \xi \doteq_S \xi''$ (Cong Trans)

$u \doteq_S v \Rightarrow u|t \doteq_S v|t$ (Cong Par)

$u \doteq_S v, m \doteq_S n \Rightarrow (\boldsymbol{\nu} m)u \doteq_S (\boldsymbol{\nu} n)v$ (Cong Res)

$m \doteq_S n, r \doteq_S q, \gamma \doteq_S \gamma' \Rightarrow (m \leftrightarrow r)\gamma \doteq_S (n \leftrightarrow q)\gamma'$ (Cong Swap)

$\gamma \#_S \gamma', r \doteq_S r', q \doteq_S q' \Rightarrow (r \leftrightarrow q)\gamma \#_S (r' \leftrightarrow q')\gamma'$ (Cong Apart)

Fig. 4. Closure of constraint theories (Basic, Spatial and Congruence).

Definition 3.8 (Closure of a constraint theory) *Given a constraint theory S, the relations*

$\doteq_S \ \subseteq \mathcal{I} \times \mathcal{I}$	Index Equality	$\doteq_S \ \subseteq \mathcal{F} \times \mathcal{F}$	Property Equality
$\doteq_S \ \subseteq \mathcal{N} \times \mathcal{N}$	Name Equality	$\#_S \ \subseteq \mathcal{N} \times \mathcal{F}$	Property Apartness
$\#_S \ \subseteq \mathcal{N} \times \mathcal{N}$	Name Apartness	$\to_S \ \subseteq \mathcal{I} \times \mathcal{I}$	Index Reduction

are inductively defined by the set of closure rules in Figs. 4-5.

Closure rules axiomatize some basic structural properties of our intended models. For instance, rules in (Spatial) characterize the basic properties of structural congruence; in particular (Sp Res Par) expresses the usual name extrusion property of π-calculus.

(Apartness)

$\mathsf{m}\#_S\,\gamma, \mathsf{n}\#_S\,\gamma \Rightarrow (\mathsf{m}\leftrightarrow\mathsf{n})\gamma \doteq_S \gamma$ (Swap Fresh)

$\mathsf{m}\#_S\,\mathsf{n} \Rightarrow \mathsf{n}\#_S\,\mathsf{m}$ (Apart Sym)

$\gamma\,\#_S\,\delta, \gamma \doteq_S \gamma', \delta \doteq_S \delta' \Rightarrow \gamma'\,\#_S\,\delta'$ (Cong Apr)

(Transposition)

$(\mathsf{n}\leftrightarrow\mathsf{m})\mathbf{0} \doteq_S \mathbf{0}$ (Swap Void)

$(\mathsf{n}\leftrightarrow\mathsf{m})(u|v) \doteq_S (\mathsf{n}\leftrightarrow\mathsf{m})u|(\mathsf{n}\leftrightarrow\mathsf{m})v$ (Swap Par)

$(\mathsf{n}\leftrightarrow\mathsf{m})(\boldsymbol{\nu}\mathsf{p})u \doteq_S (\boldsymbol{\nu}(\mathsf{n}\leftrightarrow\mathsf{m})\mathsf{p})(\mathsf{n}\leftrightarrow\mathsf{m})u$ (Swap Res)

$(\mathsf{n}\leftrightarrow\mathsf{m})(\mathsf{p}\leftrightarrow\mathsf{q})\gamma \doteq_S ((\mathsf{n}\leftrightarrow\mathsf{m})\mathsf{p}\leftrightarrow(\mathsf{n}\leftrightarrow\mathsf{m})\mathsf{q})(\mathsf{n}\leftrightarrow\mathsf{m})\gamma$ (Swap Swap)

$(\mathsf{n}\leftrightarrow\mathsf{m})(\mathsf{n}\leftrightarrow\mathsf{m})\xi \doteq_S \xi$ (Swap Inv)

$(\mathsf{n}\leftrightarrow\mathsf{n})\xi \doteq_S \xi$ (Swap Id)

$(\mathsf{m}\leftrightarrow\mathsf{n})\mathsf{m} \doteq_S \mathsf{n}$ (Swap App)

$u \doteq_S (\boldsymbol{\nu}\mathsf{n})t, u \doteq_S (\boldsymbol{\nu}\mathsf{m})v \Rightarrow (\mathsf{n}\leftrightarrow\mathsf{m})u \doteq_S u$ (Swap Erase)

(Reduction)

$u \to_S t, v \doteq_S u, t \doteq_S w \Rightarrow v \to_S w$ (Red Cong)

$u \to_S t \Rightarrow u|v \to_S t|v$ (Red Par)

$u \to_S t \Rightarrow (\boldsymbol{\nu}\mathsf{n})u \to_S (\boldsymbol{\nu}\mathsf{n})t$ (Red Res)

$u \to_S t \Rightarrow (\mathsf{n}\leftrightarrow\mathsf{m})u \to_S (\mathsf{n}\leftrightarrow\mathsf{m})t$ (Red Transp)

Fig. 5. Closure of constraint theories (Apartness, Transposition and Reduction).

Remark 3.9 Let u be the index $(\boldsymbol{\nu}x)x\,|(\boldsymbol{\nu}x)z$ and v the index $(\boldsymbol{\nu}x)(x\,|(\boldsymbol{\nu}x)z)$. Let I be any interpretation into Aπ, we then have $I(u) = (\boldsymbol{\nu}n)P|(\boldsymbol{\nu}n)Q$, for some processes P and Q and name n. Since name n is not free in the process $(\boldsymbol{\nu}n)Q$ (in the usual π-calculus sense), by the scope extrusion axiom of structural congruence we have $(\boldsymbol{\nu}n)P|(\boldsymbol{\nu}n)Q \equiv (\boldsymbol{\nu}n)(P|(\boldsymbol{\nu}n)Q) = I(v)$. This shows the soundness of the (Sp Res Par) axiom with respect to our intended interpretation.

Rules in (Transposition) and (Apartness) express the action of transpositions on indexes and name terms. The notation $\tau\gamma$ is used to represent the application of the transposition τ to some (index or name term) γ, and $\rho\gamma$ to represent the application of an arbitrary sequence of transpositions (that is, a permutation) to the element γ. For example, (Swap Erase) expresses that transposition of names which are not free in a process act as the identity: in fact, if $u \doteq_S (\boldsymbol{\nu}\mathsf{n})t$ holds then n denotes a name which is not free in the process denoted by the name term u. We write $S \vdash \mathsf{n}\#\,\mathsf{m}$ to denote that $\mathsf{n}\#_S\,\mathsf{m}$, and likewise for the other kinds of constraints. We have the following basic properties

Lemma 3.10 *For all constraint theories S and S', for all constraints c and c', for all π-substitutions σ, we have*

(1) $S \vdash c$ implies $S \cup S' \vdash c$.
(2) If $S \vdash c$ and $S, c \vdash c'$ then $S \vdash c'$.
(3) If $S \vdash c$ then $\sigma(S) \vdash \sigma(c)$.

In the remainder of this section, we present some basic concepts related to the semantics of constraint theories. An interpretation for a constraint theory assigns an appropriate denotation to all propositional, process and name variables occurring on it. As in Part I, we are interested on a version of the spatial logic for the asynchronous π-calculus (we use the standard notations $\equiv$ and $\rightarrow$ for asynchronous π-calculus structural congruence and reduction). Therefore, interpretations that concern us here map process variables into $\mathrm{A}\pi$ processes, name variables into $\mathrm{A}\pi$ names, and propositional variables into property sets.

For convenience, we present $\mathrm{A}\pi$ as a π-algebra

$$\mathrm{A}\pi = \langle \Lambda, \mathcal{P}, \mathbb{P}, \mathbf{0}, |, \boldsymbol{\nu}, (\leftrightarrow)_{\Lambda}, (\leftrightarrow)_{\mathcal{P}}, (\leftrightarrow)_{\mathbb{P}} \rangle$$

where Λ is the set of pure names, $\mathcal{P}$ is the set of processes, and $\mathbb{P}$ is the collection of all Psets (Definition 2.1). We now define

Definition 3.11 ($\mathrm{A}\pi$-interpretation) *A $\mathrm{A}\pi$-interpretation $\mathcal{J}$ is an interpretation of the term π-algebra into $\mathrm{A}\pi$.*

As noticed above, we can then see that an $\mathrm{A}\pi$-interpretation $\mathcal{J}$ contains a valuation, so that it also makes sense to write $\llbracket A \rrbracket_{\mathcal{J}}$ for the denotation of the formula A under the valuation determined by $\mathcal{J}$. Also, for a name term n, we can verify that $\mathcal{J}(\mathsf{n}) = \llbracket \mathsf{n} \rrbracket_{\mathcal{J}}$.

Definition 3.12 (Satisfaction and Validity) *The relation of* satisfaction *between an $\mathrm{A}\pi$-interpretation $\mathcal{J}$ and constraints is defined thus:*

1. $\mathcal{J}$ *sat* $\mathsf{m} \doteq \mathsf{n} \Leftrightarrow \llbracket \mathsf{m} \rrbracket_{\mathcal{J}} = \llbracket \mathsf{n} \rrbracket_{\mathcal{J}}$
2. $\mathcal{J}$ *sat* $u \doteq v \Leftrightarrow \mathcal{J}(u) \equiv \mathcal{J}(v)$
3. $\mathcal{J}$ *sat* $u \rightarrow v \Leftrightarrow \mathcal{J}(u) \rightarrow \mathcal{J}(v)$
4. $\mathcal{J}$ *sat* $\mathsf{m} \# \mathsf{n} \Leftrightarrow \llbracket \mathsf{m} \rrbracket_{\mathcal{J}} \neq \llbracket \mathsf{n} \rrbracket_{\mathcal{J}}$
5. $\mathcal{J}$ *sat* $F \doteq G \Leftrightarrow \llbracket F \rrbracket_{\mathcal{J}} = \llbracket G \rrbracket_{\mathcal{J}}$
6. $\mathcal{J}$ *sat* $\mathsf{n} \# F \Leftrightarrow \llbracket \mathsf{n} \rrbracket_{\mathcal{J}} \notin \mathit{supp}(\llbracket F \rrbracket_{\mathcal{J}})$

$\mathcal{J}$ satisfies *the constraint theory S if $\mathcal{J}$ satisfies all constraints in S. A constraint $S \vdash c$* is valid *if every interpretation that satisfies S also satisfies c.*

The following lemma establishes the soundness of the closure of constraint theories.

Lemma 3.13 (Soundness) *Let S be a constraint theory and $\mathcal{J}$ a $\mathrm{A}\pi$-interpretation that satisfies S. For all name terms m and n, for all indexes u and* *t, and for all propositional terms F and G, we have:*

1. If $\mathsf{m} \doteq_S \mathsf{n}$ then $[\![\mathsf{m}]\!]_{\mathcal{J}} = [\![\mathsf{n}]\!]_{\mathcal{J}}$

2. If $u \doteq_S t$ then $\mathcal{J}(u) \equiv \mathcal{J}(t)$

3. If $u \to_S t$ then $\mathcal{J}(u) \to \mathcal{J}(t)$

4. If $\mathsf{m} \#_S \mathsf{n}$ then $[\![\mathsf{m}]\!]_{\mathcal{J}} \neq [\![\mathsf{n}]\!]_{\mathcal{J}}$

5. If $F \doteq_S G$ then $\mathcal{J}(F) = \mathcal{J}(G)$

6. If $\mathsf{m} \#_S F$ then $[\![\mathsf{m}]\!]_{\mathcal{J}} \notin supp([\![F]\!]_{\mathcal{J}})$

Proof. By induction on the derivations of $\gamma \doteq_S \gamma'$, $\mathsf{n} \#_S \mathsf{m}$, $\mathsf{m} \#_S F$, and $u \to_S v$ using well-known properties of structural congruence, name transposition and reduction of the asynchronous π-calculus. ■

3.3 Sequents

Having introduced indexes and constraint theories, we can now define the sequents of our logic. First, a *context* is a finite multiset of indexed formulas of the form $u : A$ where u is an index (Definition 3.2) and A is a formula. We use Δ, Γ to denote contexts. Then

Definition 3.14 (Sequent) *A* sequent *is a judgment of the form $\langle S \rangle\, \Gamma \vdash \Delta$ where S is a constraint theory, and Δ and Γ are contexts.*

As usual, the *right context* Δ is interpreted as the disjunction of its formulas, the *left context* Γ is interpreted as the conjunction of the formulas in it. Defining contexts as multisets allows for the implicit use of exchange (but not contraction!) in proofs. We write $\Delta \equiv_\alpha \Delta'$ if Δ' is obtained from Δ by α-converting some formulas in it.

Definition 3.15 (Variables in sequents) *The set of free (name, process, and propositional) variables of a context Δ is given by*

$$lfv(\Delta) \triangleq \bigcup \{ afv(u) \cup lfv(A) \mid u : A \in \Delta \}$$

The set of free (name, process, and propositional) variables in a sequent $\langle S \rangle\, \Gamma \vdash \Delta$ is given by

$$fv(\langle S \rangle\, \Gamma \vdash \Delta) \triangleq afv(S) \cup fv(\Gamma) \cup fv(\Delta)$$

N.B.: name variables x occur both in constraints and in formulas A; process variables x occur only in indexes; propositional variables X also may occur in formulas and constraints. Given a Aπ-interpretation $\mathcal{J}$ and a context Γ, we say that $\mathcal{J}$ *satisfies all of* Γ if $\mathcal{J}(u) \in [\![A]\!]_{\mathcal{J}}$ for all $u : A \in \Gamma$. Likewise, we say that $\mathcal{J}$ *satisfies some of* Γ if $\mathcal{J}(u) \in [\![A]\!]_{\mathcal{J}}$ for some $u : A \in \Gamma$. Hence we have

Definition 3.16 (Valid Sequent) *A sequent $\langle S \rangle\, \Gamma \vdash \Delta$ is* valid *if for all interpretations $\mathcal{J}$ such that $\mathcal{J}$ satisfies S, and $\mathcal{J}$ satisfies all of Γ, then $\mathcal{J}$ satisfies some of Δ.*

$$
\begin{array}{lll}
A & \equiv_S A' & \text{if } A \equiv_\alpha A' \\
(n \leftrightarrow m)\mathbf{0} & \equiv_S \mathbf{0} & \\
(n \leftrightarrow m)\mathbf{F} & \equiv_S \mathbf{F} & \\
(n \leftrightarrow m)(A \wedge B) & \equiv_S (n \leftrightarrow m)A \wedge (n \leftrightarrow m)B & \\
(n \leftrightarrow m)(A \Rightarrow B) & \equiv_S (n \leftrightarrow m)A \Rightarrow (n \leftrightarrow m)B & \\
(n \leftrightarrow m)(A|B) & \equiv_S (n \leftrightarrow m)A|(n \leftrightarrow m)B & \\
(n \leftrightarrow m)(A \triangleright B) & \equiv_S (n \leftrightarrow m)A \triangleright (n \leftrightarrow m)B & \\
(n \leftrightarrow m)\Diamond A & \equiv_S \Diamond(n \leftrightarrow m)A & \\
(n \leftrightarrow m)(\reflectbox{N}x.A) & \equiv_S \reflectbox{N}x.(n \leftrightarrow m)(A\{x \leftarrow (n \leftrightarrow m)x\}) & \text{if } x \notin fv(m) \cup fv(n) \\
(n \leftrightarrow m)(\forall x.A) & \equiv_S \forall x.(n \leftrightarrow m)(A\{x \leftarrow (n \leftrightarrow m)x\}) & \text{if } x \notin fv(m) \cup fv(n) \\
(n \leftrightarrow m)(\forall X.A) & \equiv_S \forall X.(n \leftrightarrow m)A\{X \leftarrow (n \leftrightarrow m)X\} & \\
(n \leftrightarrow m)(p \circledR A) & \equiv_S ((n \leftrightarrow m)p) \circledR (n \leftrightarrow m)A & \\
(n \leftrightarrow m)(A \oslash p) & \equiv_S (n \leftrightarrow m)A \oslash ((n \leftrightarrow m)p) & \\
(n \leftrightarrow m)(p\langle q\rangle) & \equiv_S ((n \leftrightarrow m)p)\langle (n \leftrightarrow m)q\rangle & \\
n \circledR A & \equiv_S m \circledR A & \text{if } n \doteq_S m \\
A \oslash n & \equiv_S A \oslash m & \text{if } n \doteq_S m \\
n\langle m\rangle & \equiv_S p\langle q\rangle & \text{if } m \doteq_S p \text{ and } n \doteq_S q \\
F & \equiv_S G & \text{if } F \doteq_S G
\end{array}
$$

Fig. 6. Formula Equivalence.

For example, if A and B are closed formulas, the sequent $\langle\rangle\, x : A \vdash x : B$ is valid if and only if every process that satisfies the formula A also satisfies the formula B.

3.4 Assertions

An assertion $A \equiv_S B$ states that, under any interpretation that satisfies all constraints in S, the formulas A and B denote the same property.

Definition 3.17 (Equational equivalence of formulas) *Equational equivalence of formulas, written $\equiv_S$, is the least congruence relation on formulas inductively defined in Figure 6.*

We call a formula *normalized* if all occurrences of transpositions occur at the term level (so it contains no subformula of the form $(n \leftrightarrow m)A$). In general, given a constraint theory S, any formula A can be converted into a semantically equivalent but normalized formula A', using the equations in Figure 6 as left-to-right rewrite rules. We then define

Definition 3.18 (Normalized) *We assert $A \Downarrow_S B$ whenever $A \equiv_S B$ and B is normalized.*

Note that if $A \Downarrow_S B$ and $A \Downarrow_S B'$, we must have $B \Downarrow_S B'$. We also use the notation $\Gamma \Downarrow_S \Gamma'$ to denote that the sequent context Γ' results from normalizing the sequent context Γ under the constraints S. Thus, we also call a sequent or sequent context *normalized* whenever all formulas in it are normalized. Moreover

Lemma 3.19 *For all formulas A, B and constraint theory S we have*

(1) For every π-substitution σ, if $A \equiv_S B$ then $\sigma(A) \equiv_{\sigma(S)} \sigma(B)$.
(2) If $A \equiv_S B$ then there is A' such that $A \Downarrow_S A'$ and $B \Downarrow_S A'$.
(3) If $A \equiv_S B$ and $A \Downarrow_S A'$ for formula some A', then also $B \Downarrow_S A'$.

Proof. (1–3) Induction on the derivation of $A \equiv_S B$. ■

An assertion $\mathsf{n} \#_S A$ states that, under any interpretation that satisfies all constraints in S, the name denoted by the name term n is fresh in the property denoted by the formula A. More precisely, given a formula A with $\mathit{lft}(A) = \{\mathsf{m}_1, \ldots, \mathsf{m}_k\}$, and a constraint theory S, we write $\mathsf{n} \#_S A$ as an abbreviation for the set (understood as the conjunction) containing the constraints $\mathsf{n} \#_S \mathsf{m}_1, \ldots, \mathsf{n} \#_S \mathsf{m}_k$. N.B.: each m_i is either a name term or a propositional variable. The following facts are important:

Lemma 3.20 *For all normalized formulas A and name terms p, q,*

(1) Let $\mathsf{p} \#_S A$ and $\mathsf{q} \#_S A$. Then $(\mathsf{p} \leftrightarrow \mathsf{q}) A \Downarrow_S A$.
(2) Let $\mathsf{p} \#_S \text{И} x.A$ and $\mathsf{q} \#_S \text{И} x.A$. Then $(\mathsf{p} \leftrightarrow \mathsf{q}) A\{x \leftarrow \mathsf{p}\} \Downarrow_S A\{x \leftarrow \mathsf{q}\}$.

Proof. Follows from Lemma 8.2 in appendix. ■.

We can verify that the relations $\equiv_S$ (between formulas), and $\#_S$ (between name terms and formulas) defined above are sound with respect to their intended interpretations.

Lemma 3.21 (Soundness) *Let $\mathcal{J}$ be a $\mathrm{A}\pi$-interpretation. For all formulas A, B and name terms n,*

(1) If $\mathcal{J}$ satisfies S and $A \equiv_S B$ then $[\![A]\!]_{\mathcal{J}} = [\![B]\!]_{\mathcal{J}}$.
(2) If $\mathcal{J}$ satisfies S and $\mathsf{n} \#_S A$ then $[\![\mathsf{n}]\!]_{\mathcal{J}} \notin \mathit{fn}^{\mathcal{J}}(A)$.
(3) If $\mathcal{J}$ satisfies S and $\mathsf{n} \#_S A$ then $[\![\mathsf{n}]\!]_{\mathcal{J}} \notin \mathit{supp}([\![A]\!]_{\mathcal{J}})$.

Proof. (1) Induction on the derivation of $A \equiv_S B$. (2) By Lemma 3.13. (3) By (2) and Theorem 2.3(1). ■

$$\frac{[A \textit{ is an atomic formula }]}{\langle S\rangle\, \Gamma, u : A \vdash u : A, \Delta}\,(\mathrm{Id}) \qquad \frac{\langle S\rangle\, \Gamma \vdash u : A, \Delta \quad \langle S\rangle\, \Gamma, u : A \vdash \Delta}{\langle S\rangle\, \Gamma \vdash \Delta}\,(\mathrm{Cut})$$

$$\frac{\langle S\rangle\, \Gamma, u : A, u : A \vdash \Delta}{\langle S\rangle\, \Gamma, u : A \vdash \Delta}\,(\mathrm{CL}) \qquad \frac{\langle S\rangle\, \Gamma \vdash u : A, u : A, \Delta}{\langle S\rangle\, \Gamma \vdash u : A, \Delta}\,(\mathrm{CR})$$

$$\frac{\begin{array}{r}(n\leftrightarrow m)A \equiv_S A' \\ \langle S\rangle\, \Gamma, u' : A' \vdash \Delta \;\; (m\leftrightarrow n)u \doteq_S u'\end{array}}{\langle S\rangle\, \Gamma, u : A \vdash \Delta}\,(\mathrm{TL}) \quad \frac{\begin{array}{r}(n\leftrightarrow m)A \equiv_S A' \\ \langle S\rangle\, \Gamma \vdash u' : A', \Delta \;\; (m\leftrightarrow n)u \doteq_S u'\end{array}}{\langle S\rangle\, \Gamma \vdash u : A, \Delta}\,(\mathrm{TR})$$

$$\frac{}{\langle S\rangle\, \Gamma, u : \mathbf{F} \vdash \Delta}\,(\mathbf{F}\mathrm{L}) \qquad \frac{\langle S\rangle\, \Gamma \vdash \Delta}{\langle S\rangle\, \Gamma \vdash u : \mathbf{F}, \Delta}\,(\mathbf{F}\mathrm{R})$$

$$\frac{\langle S\rangle\, \Gamma, u : A, u : B \vdash \Delta}{\langle S\rangle\, \Gamma, u : A \wedge B \vdash \Delta}\,(\wedge\mathrm{L}) \qquad \frac{\begin{array}{c}\langle S\rangle\, \Gamma \vdash u : A, \Delta \\ \langle S\rangle\, \Gamma \vdash u : B, \Delta\end{array}}{\langle S\rangle\, \Gamma \vdash u : A \wedge B, \Delta}\,(\wedge\mathrm{R})$$

$$\frac{\langle S\rangle\, \Gamma \vdash u : A, \Delta \quad \langle S\rangle\, \Gamma, u : B \vdash \Delta}{\langle S\rangle\, \Gamma, u : A \Rightarrow B \vdash \Delta}\,(\Rightarrow\mathrm{L}) \quad \frac{\langle S\rangle\, \Gamma, u : A \vdash u : B, \Delta}{\langle S\rangle\, \Gamma \vdash u : A \Rightarrow B, \Delta}\,(\Rightarrow\mathrm{R})$$

Fig. 7. Structural and Propositional Rules.

3.5 Inference Rules

We now present the set of inference rules of our base proof system **S**. Inference rules may have for premises not just sequents but also assertions over the closure of the constraint theory S that appears in the conclusion. Such assertions are of the form $u \doteq_S v$ (mostly in the rules for spatial connectives), $A \equiv_S B$ (in (TL) and (TR) rules), $u \rightarrow_S v$ (in the temporal rules) or $n \#_S A$ (in the freshness rules).

The rules in the identity, structural and propositional group (see Figures 7) follow the standard format. We use the simplest possible form for the (Id) axiom, where the formula A is required to be atomic. Recall that in general a formula is called *atomic* if it is not built from a logical connective at the top level, in our case, if it is either a propositional variable X or a message $n\langle m\rangle$. This is without loss of generality, since the general form of (Id) where the identified formula can be an arbitrary one is admissible (Lemma 5.5). We include explicit contraction rules (CL) and (CR); weakening is admissible, and exchange may be dealt with implicitly, since sequent contexts are multisets.

The transposition rules (TL) and (TR) capture the property of invariance of the semantics under transposition of names (Theorem 2.3). They also incorporate the theory of equality of indexes and names terms defined by the

$$\frac{\langle S, t \doteq \mathbf{0}\rangle\, \Gamma \vdash \Delta}{\langle S\rangle\, \Gamma, t : \mathbf{0} \vdash \Delta}\ (\mathbf{0}\mathrm{L}) \qquad \frac{u \doteq_S \mathbf{0}}{\langle S\rangle\, \Gamma \vdash u : \mathbf{0}, \Delta}\ (\mathbf{0}\mathrm{R})$$

$$\begin{array}{c}[x \textit{ and } y \textit{ not free in the conclusion}]\\ \dfrac{\langle S, u \doteq x|y\rangle\, \Gamma, x : A, y : B \vdash \Delta}{\langle S\rangle\, \Gamma, u : A|B \vdash \Delta}\ (|\mathrm{L})\end{array} \qquad \dfrac{\begin{array}{c}\langle S\rangle\, \Gamma \vdash v : A, \Delta\\ \langle S\rangle\, \Gamma \vdash t : B, \Delta \quad u \doteq_S v|t\end{array}}{\langle S\rangle\, \Gamma \vdash u : A|B, \Delta}\ (|\mathrm{R})$$

$$\dfrac{\langle S\rangle\, \Gamma \vdash t : A, \Delta \quad \langle S\rangle\, \Gamma, t|u : B \vdash \Delta}{\langle S\rangle\, \Gamma, u : A \triangleright B \vdash \Delta}\ (\triangleright\mathrm{L}) \quad \begin{array}{c}[x \textit{ not free in the conclusion}]\\ \dfrac{\langle S\rangle\, \Gamma, x : A \vdash v : B, \Delta \quad v \doteq_S x|u}{\langle S\rangle\, \Gamma \vdash u : A \triangleright B, \Delta}\ (\triangleright\mathrm{R})\end{array}$$

$$\begin{array}{c}[x \textit{ not free in the conclusion}]\\ \dfrac{\langle S, u \doteq (\boldsymbol{\nu}\mathsf{n})x\rangle\, \Gamma, x : A \vdash \Delta}{\langle S\rangle\, \Gamma, u : \mathsf{n}\circledR A \vdash \Delta}\ (\circledR\mathrm{L})\end{array} \qquad \dfrac{\langle S\rangle\, \Gamma \vdash u : A, \Delta \quad t \doteq_S (\boldsymbol{\nu}\mathsf{n})u}{\langle S\rangle\, \Gamma \vdash t : \mathsf{n}\circledR A, \Delta}\ (\circledR\mathrm{R})$$

$$\dfrac{\langle S\rangle\, \Gamma, t : A \vdash \Delta \quad t \doteq_S (\boldsymbol{\nu}\mathsf{n})u}{\langle S\rangle\, \Gamma, u : A \oslash \mathsf{n} \vdash \Delta}\ (\oslash\mathrm{L}) \qquad \dfrac{\langle S\rangle\, \Gamma \vdash u : A, \Delta \quad u \doteq_S (\boldsymbol{\nu}\mathsf{n})t}{\langle S\rangle\, \Gamma \vdash t : A \oslash \mathsf{n}, \Delta}\ (\oslash\mathrm{R})$$

Fig. 8. Spatial Rules.

$$\begin{array}{c}[x \textit{ not free in the conclusion}]\\ \dfrac{\langle S, u \to x\rangle\, \Gamma, x : A \vdash \Delta}{\langle S\rangle\, \Gamma, u : \Diamond A \vdash \Delta}\ (\Diamond\mathrm{L})\end{array} \quad \dfrac{\langle S\rangle\, \Gamma \vdash v : A, \Delta \quad u \to_S v}{\langle S\rangle\, \Gamma \vdash u : \Diamond A, \Delta}\ (\Diamond\mathrm{R})$$

Fig. 9. Temporal Rules.

$$\dfrac{\langle S, x \# N, u \doteq (\boldsymbol{\nu}x)x\rangle\, \Gamma \vdash \Delta}{\langle S\rangle\, \Gamma \vdash \Delta}\ (\text{И})$$

$$\dfrac{\begin{array}{c}u \doteq_S (\boldsymbol{\nu}\mathsf{n})v\\ \mathsf{n} \#_S \text{И}x.A\\ \langle S\rangle\, \Gamma, u : A\{x \leftarrow \mathsf{n}\} \vdash \Delta\end{array}}{\langle S\rangle\, \Gamma, u : \text{И}x.A \vdash \Delta}\ (\text{И}\mathrm{L}) \quad \dfrac{\begin{array}{c}u \doteq_S (\boldsymbol{\nu}\mathsf{n})v\\ \mathsf{n} \#_S \text{И}x.A\\ \langle S\rangle\, \Gamma \vdash u : A\{x \leftarrow \mathsf{n}\}, \Delta\end{array}}{\langle S\rangle\, \Gamma \vdash u : \text{И}x.A, \Delta}\ (\text{И}\mathrm{R})$$

Fig. 10. Freshness Rules

constraint closure (Definition 3.8) into the proof system, in particular axiomatizing the principle of substitution of equals for equals of name terms in formulas. Note that, in these rules, indexes are identified up to $\doteq_S$, while formulas are identified up to $\equiv_S$. As we shall discuss in Section 5.2, explicit transpositions and the transpositions rule also play a crucial role in obtaining cut-elimination for the freshness quantifier.

$$\frac{\langle S\rangle\, \Gamma, u : A\{x{\leftarrow}\boldsymbol{n}\} \vdash \Delta}{\langle S\rangle\, \Gamma, u : \forall x.A \vdash \Delta}\ (\forall\mathrm{L}) \qquad \overset{[y\ \textit{not free in the conclusion}]}{\frac{\langle S\rangle\, \Gamma \vdash u : A\{x{\leftarrow}y\}, \Delta}{\langle S\rangle\, \Gamma \vdash u : \forall x.A, \Delta}}\ (\forall\mathrm{R})$$

$$\frac{\langle S\rangle\, \Gamma, u : A\{X{\leftarrow}B\} \vdash \Delta}{\langle S\rangle\, \Gamma, u : \forall X.A \vdash \Delta}\ (\forall^2\mathrm{L}) \qquad \overset{[Y\ \textit{not free in the conclusion}]}{\frac{\langle S\rangle\, \Gamma \vdash u : A\{X{\leftarrow}Y\}, \Delta}{\langle S\rangle\, \Gamma \vdash u : \forall X.A, \Delta}}(\forall^2\mathrm{R})$$

Fig. 11. Quantifier Rules.

$$\frac{\langle S, u \doteq \mathbf{0}\rangle\, \Gamma \vdash \Delta \quad u|v \doteq_S \mathbf{0}}{\langle S\rangle\, \Gamma \vdash \Delta}\ (\mathrm{S}|\mathbf{0}) \quad \frac{\langle S, u \doteq \mathbf{0}\rangle\, \Gamma \vdash \Delta \quad (\boldsymbol{\nu}\boldsymbol{n})u \doteq_S \mathbf{0}}{\langle S\rangle\, \Gamma \vdash \Delta}\ (\mathrm{S}\boldsymbol{\nu}\mathbf{0})$$

$$\overset{[\mathcal{x}\ \textit{and}\ \mathcal{y}\ \textit{not free in the conclusion}]}{\frac{\langle S, u \doteq \mathcal{x}|\mathcal{y}, (\boldsymbol{\nu}\boldsymbol{n})\mathcal{x} \doteq t, (\boldsymbol{\nu}\boldsymbol{n})\mathcal{y} \doteq v\rangle\, \Gamma \vdash \Delta \quad (\boldsymbol{\nu}\boldsymbol{n})u \doteq_S t|v}{\langle S\rangle\, \Gamma \vdash \Delta}}\ (\mathrm{S}\boldsymbol{\nu}|)$$

$$\overset{[\mathcal{x},\mathcal{x}',\mathcal{y}\ \textit{and}\ \mathcal{y}'\ \textit{not free in the conclusion}]}{\frac{\langle S, u \doteq \mathcal{x}|\mathcal{x}', w \doteq \mathcal{y}|\mathcal{y}', t \doteq \mathcal{x}|\mathcal{y}, v \doteq \mathcal{x}'|\mathcal{y}'\rangle\, \Gamma \vdash \Delta \quad u|w \doteq_S t|v}{\langle S\rangle\, \Gamma \vdash \Delta}}\ (\mathrm{S}||)$$

$$\overset{[\mathcal{x}\ \textit{not free in the conclusion}]}{\frac{\begin{array}{l}\langle S, u \doteq (\boldsymbol{n}{\leftrightarrow}\boldsymbol{m})v\rangle\, \Gamma \vdash \Delta \\ \langle S, u \doteq (\boldsymbol{\nu}\boldsymbol{m})\mathcal{x}, v \doteq (\boldsymbol{\nu}\boldsymbol{n})\mathcal{x}\rangle\, \Gamma \vdash \Delta \quad (\boldsymbol{\nu}\boldsymbol{n})u \doteq_S (\boldsymbol{\nu}\boldsymbol{m})v\end{array}}{\langle S\rangle\, \Gamma \vdash \Delta}}\ (\mathrm{S}\boldsymbol{\nu}\boldsymbol{\nu})$$

$$\frac{\mathbf{0} \to_S u}{\langle S\rangle\, \Gamma \vdash \Delta}\ (\mathrm{S}\mathbf{0} \to) \quad \overset{[\mathcal{x}\ \textit{not free in the conclusion}]}{\frac{\langle S, u \to \mathcal{x}, v \doteq (\boldsymbol{\nu}n)\mathcal{x}\rangle\, \Gamma \vdash \Delta \quad (\boldsymbol{\nu}n)u \to_S v}{\langle S\rangle\, \Gamma \vdash \Delta}}\ (\mathrm{S}\boldsymbol{\nu} \to)$$

Fig. 12. World Rules.

In the rules for propositional connectives, indexes keep track of the processes for which the formulas are asserted to hold, but do not interfere in any way with the constraint part of sequents. This is not the case in rules for the spatial connectives (Figure 8), that and make essential use of the constraint theories in sequents. Note that the left rules, when read bottom-up, introduce spatial constraints into the constraint theories, and the respective right rules, when read top-down, check corresponding constraints. While spatial rules rely on spatial constraints, temporal rules (Figure 9) rely on reduction constraints.

The rules for first and second order quantifiers have the expected form (Figure 11). We then introduce the rules for freshness (Figure 10). Rule ($\reflectbox{N}$) asserts, when read bottom-up, that there is always a name (denoted by) x that is fresh with respect to the free names of (the process denoted by) the index u, and that is also fresh with respect to a set of names (denoted by the name and propositional variables in) N. Hence, rule ($\reflectbox{N}$) corresponds to the (Fresh) axiom of Pitts' Nominal Logic [18].

The rules ($\reflectbox{N}$L/R) for the fresh quantifier do not show the symmetry one might expect of a left / right rule pair. This fact relates to the existential / universal ambivalence of freshness quantification (the Gabbay-Pitts property): note that ($\reflectbox{N}$L) follows the pattern of ($\forall$L), while ($\reflectbox{N}$R) follows the pattern of ($\exists$R). Then, ($\reflectbox{N}$) embodies the introduction of fresh witnesses usually present in *both* ($\forall$R) and ($\exists$L). Both ($\reflectbox{N}$L) and ($\reflectbox{N}$R) include a premise of the form $\boldsymbol{n} \#_S \reflectbox{N}x.A$, asserting that the name term $\boldsymbol{n}$ must denote a name distinct from all free names in the support of the property denoted by formula A. Moreover, in the rules for $\reflectbox{N}x.A$, in addition to the freshness condition $\boldsymbol{n} \#_S \reflectbox{N}x.A$, the assumption $u \doteq_S (\boldsymbol{\nu n})v$ ensures that $\boldsymbol{n}$ denotes a name that does not occur free in the process denoted by u, *cf.* the semantics of $\reflectbox{N}x.A$.

Finally, world rules (Figure 12) axiomatize certain deep (extra-logical) properties of the worlds. Moreover, the properties captured by the proposed set of world rules (inversion principles for structural congruence and for process reduction) are expected to hold in any natural variation of the π-calculus. It is important to note that none of the studied proof-theoretic properties of our logic (*e.g.*, cut-elimination) depend on the chosen set of world rules. This means that the proof system is completely open to the addition of further world rules, provided their soundness is granted, that they do not change logical contexts of sequents (Γ and Δ), and that they just check or eliminate constraints from the constraint part of sequents.

We assert $\vdash \langle S \rangle \, \Gamma \Vdash \Delta$ to state that the sequent $\langle S \rangle \, \Gamma \Vdash \Delta$ has a derivation. We now state soundness of our system with respect to the intended model.

Theorem 3.22 (Soundness) *All sequents derivable in* **S** *are valid in* Aπ.

Proof. See appendix. ■

$\neg A$	$\triangleq A \Rightarrow \mathbf{F}$	(Negation)
$\mathbf{T}$	$\triangleq \neg\mathbf{F}$	(True)
$A \vee B$	$\triangleq \neg A \Rightarrow B$	(Disjunction)
$A \Vert B$	$\triangleq \neg(\neg A \mid \neg B)$	(Decomposition)
$\copyright n$	$\triangleq \neg n \circledR \mathbf{T}$	(Free name)
$\Box A$	$\triangleq \neg\Diamond\neg A$	(All next)
$\sim A$	$\triangleq A \triangleright \mathbf{F}$	(Inconsistency)
$!A$	$\triangleq \sim \neg A$	(Validity)
$A \Mapsto B$	$\triangleq !(A \Rightarrow B)$	(Entailment)
$\exists x.A$	$\triangleq \neg\forall x.\neg A$	(First-order existential quantification)
$\exists X.A$	$\triangleq \neg\forall X.\neg A$	(Second-order existential quantification)
$\mathsf{H}x.A$	$\triangleq \text{И}x.x \circledR A$	(Hidden name quantification)

Fig. 13. Derived Connectives.

3.6 Derived Connectives and Inference Rules

Before closing the section, we introduce some useful derived connectives (see Figure 13). These include the usual operations of the classical predicate calculus, namely $\neg A$ (Negation), $\exists x.A$ (Existential quantification), $A \vee B$ (Disjunction) and $\mathbf{T}$ (True), with the expected meaning. *Decomposition* $A \Vert B$ is the DeMorgan dual of composition $A \mid B$. For instance, a process satisfies $\mathbf{0} \Vert \mathbf{0}$ if it is single-threaded (or void). We also have the standard temporal modality $\Box$, the dual of $\Diamond$. The free name predicate $\copyright n$ holds of any process with some free occurrence of the name (denoted by the name term) n. *Inconsistency* $\sim A$ expresses internally to the logic that A is false of every process and *validity* $!A$ that A holds of every process [10]. Thus, *entailment* $A \Mapsto B$ internalizes the consequence relation induced by the logic. The hidden name quantifier is defined as in [2]. For these connectives the inference rules presented in Figures 14 and 15 can easily be shown to be admissible.

4 Inductive and Coinductive Definitions

In this section, we present our treatment of recursive formulas. First, as shown in Section 3.6 we can combine the spatial operator $\triangleright$ with classical negation to obtain an operator $!A \triangleq (A \Rightarrow \mathbf{F}) \triangleright \mathbf{F}$ that has the meaning that A is valid (is satisfied by any process). $!A$ is an example of a classical formula [10]: the truth value of classical formulas does not depend on the particular world (process)

$$\frac{\langle S\rangle\,\Gamma \vdash \Delta}{\langle S\rangle\,\Gamma, u : \mathbf{T} \vdash \Delta}\ (\mathbf{T}\mathrm{L}) \qquad\qquad \frac{}{\langle S\rangle\,\Gamma \vdash u : \mathbf{T}, \Delta}\ (\mathbf{T}\mathrm{R})$$

$$\frac{\begin{array}{c}\langle S\rangle\,\Gamma, u : A \vdash \Delta\\ \langle S\rangle\,\Gamma, u : B \vdash \Delta\end{array}}{\langle S\rangle\,\Gamma, u : A \vee B \vdash \Delta}\ (\vee\mathrm{L}) \qquad\qquad \frac{\langle S\rangle\,\Gamma \vdash u : A, u : B, \Delta}{\langle S\rangle\,\Gamma \vdash u : A \vee B, \Delta}\ (\vee\mathrm{R})$$

$$\frac{\langle S\rangle\,\Gamma \vdash u : A, \Delta}{\langle S\rangle\,\Gamma, u : \neg A \vdash \Delta}\ (\neg\mathrm{L}) \qquad\qquad \frac{\langle S\rangle\,\Gamma, u : A \vdash \Delta}{\langle S\rangle\,\Gamma \vdash u : \neg A, \Delta}\ (\neg\mathrm{R})$$

$$\frac{\begin{array}{l}\langle S\rangle\,\Gamma, v : A \vdash \Delta\\ \langle S\rangle\,\Gamma, t : B \vdash \Delta \quad u \doteq_S v|t\end{array}}{\langle S\rangle\,\Gamma, u : A\|B \vdash \Delta}\ (\|\mathrm{L}) \qquad \begin{array}{c}[\mathscr{x}\ \textit{and}\ \mathscr{y}\ \textit{not free in the conclusion}]\\ \dfrac{\langle S, u \doteq \mathscr{x}|\mathscr{y}\rangle\,\Gamma \vdash \mathscr{x} : A, \mathscr{y} : B, \Delta}{\langle S\rangle\,\Gamma \vdash u : A\|B, \Delta}\end{array}(\|\mathrm{R})$$

$$\frac{u \doteq_S (\boldsymbol{\nu} \boldsymbol{n})v}{\langle S\rangle\,\Gamma, u : \copyright\boldsymbol{n} \vdash \Delta}\ (\copyright\mathrm{L}) \qquad \begin{array}{c}[\mathscr{x}\ \textit{not free in the conclusion}]\\ \dfrac{\langle S, u = (\boldsymbol{\nu} \boldsymbol{n})\mathscr{x}\rangle\,\Gamma \vdash \Delta}{\langle S\rangle\,\Gamma \vdash u : \copyright\boldsymbol{n}, \Delta}\end{array}\ (\copyright\mathrm{R})$$

$$\frac{\langle S\rangle\,\Gamma, v : A \vdash \Delta \quad u \rightarrow_S v}{\langle S\rangle\,\Gamma, u : \Box A \vdash \Delta}\ (\Box\mathrm{L}) \qquad \begin{array}{c}[X\ \textit{not free in the conclusion}]\\ \dfrac{\langle S, u \rightarrow \mathscr{x}\rangle\,\Gamma \vdash \mathscr{x} : A, \Delta}{\langle S\rangle\,\Gamma \vdash u : \Box A, \Delta}\end{array}\ (\Box\mathrm{R})$$

Fig. 14. Inference Rules for derived connectives.

at which they are evaluated. Then, the formula

$$A \Mapsto B \triangleq\ !(A \Rightarrow B)$$

means that the denotation of formula A is contained in the denotation of formula B. Now, given a formula A with a free propositional variable X, we say that A *is monotonic in* X if the mapping that assigns $[\![A]\!]_{v[X \leftarrow \Psi]}$ to every property Ψ is monotonic. Writing A as $A\{X\}$ and $A\{X \leftarrow B\}$ as $A\{B\}$, through second-order quantification we can express inside the logic that A is monotonic in X as follows:

$$A\{X^+\} \triangleq\ \mathbf{0} :!\forall X.\forall Y.(X \Mapsto Y) \Rightarrow (A\{X\} \Mapsto A\{Y\})$$

We may check that $A\{X^+\}$ is valid if and only if A is monotonic in X (note that $A\{X^+\}$ is an indexed formula, where the index is $\mathbf{0}$).We then define least and greatest fixpoint operators in a style similar to F-algebraic encodings.

$$\frac{\langle S\rangle\,\Gamma, v : A \vdash \Delta}{\langle S\rangle\,\Gamma, u : !A \vdash \Delta}\;(!\mathrm{L}) \qquad \frac{\begin{array}{c}[x\ \textit{not free in the conclusion}]\\ \langle S\rangle\,\Gamma \vdash x : A, \Delta\end{array}}{\langle S\rangle\,\Gamma \vdash u : !A, \Delta}\;(!\mathrm{R})$$

$$\frac{\langle S\rangle\,\Gamma \vdash v : A, \Delta \quad \langle S\rangle\,\Gamma, v : B \vdash \Delta}{\langle S\rangle\,\Gamma, u : A \Mapsto B \vdash \Delta}\;(\Mapsto\mathrm{L}) \qquad \frac{\begin{array}{c}[x\ \textit{not free in the conclusion}]\\ \langle S\rangle\,\Gamma, x : A \vdash x : B, \Delta\end{array}}{\langle S\rangle\,\Gamma \vdash u : A \Mapsto B, \Delta}\;(\Mapsto\mathrm{R})$$

$$\frac{\begin{array}{c}[x\ \textit{not free in the conclusion}]\\ \langle S\rangle\,\Gamma, u : A \vdash \Delta\end{array}}{\langle S\rangle\,\Gamma, u : \exists x.A \vdash \Delta}\;(\exists\mathrm{L}) \qquad \frac{\langle S\rangle\,\Gamma \vdash u : A\{x\leftarrow \boldsymbol{n}\}, \Delta}{\langle S\rangle\,\Gamma \vdash u : \exists x.A, \Delta}\;(\exists\mathrm{R})$$

$$\frac{\begin{array}{c}[X\ \textit{not free in the conclusion}]\\ \langle S\rangle\,\Gamma, u : A \vdash \Delta\end{array}}{\langle S\rangle\,\Gamma, u : \exists X.A \vdash \Delta}\;(\exists^2\mathrm{L}) \qquad \frac{\langle S\rangle\,\Gamma \vdash u : A\{X\leftarrow B\}, \Delta}{\langle S\rangle\,\Gamma \vdash u : \exists X.A, \Delta}\;(\exists^2\mathrm{R})$$

$$\frac{\begin{array}{c}[x\ \textit{not free in the conclusion}]\\ \boldsymbol{n}\#_S\,\mathsf{H}x.A\\ \langle S, u \doteq (\boldsymbol{\nu n})x\rangle\,\Gamma, x : A\{x\leftarrow \boldsymbol{n}\} \vdash \Delta\end{array}}{\langle S\rangle\,\Gamma, u : \mathsf{H}x.A \vdash \Delta}\;(\mathsf{HL}) \qquad \frac{\begin{array}{c}\boldsymbol{n}\#_S\,\mathsf{H}x.A \quad u \doteq_S (\boldsymbol{\nu n})v\\ \langle S\rangle\,\Gamma \vdash v : A\{x\leftarrow \boldsymbol{n}\}, \Delta\end{array}}{\langle S\rangle\,\Gamma \vdash u : \mathsf{H}x.A, \Delta}\;(\mathsf{HR})$$

Fig. 15. Inference Rules for derived connectives.

$$\boldsymbol{\mu}Y.A\{Y\} \triangleq \forall Y.(A\{Y\} \Mapsto Y) \Rightarrow Y \qquad \boldsymbol{\nu}Y.A\{Y\} \triangleq \neg\boldsymbol{\mu}X.\neg A\{X\}$$

These definitions turn out to enjoy the expected properties of recursive formulas, in the form of the derivable left and right rules in Figure 16. For example, the derivable rule (νR) corresponds to a coinduction principle. The folding and unfolding principles for $\boldsymbol{\mu}X.A$ and $\boldsymbol{\nu}X.A$ can also be derived, by making an essential use of monotonicity assumptions. We show in detail the case for folding the least fixpoint operator, using the abbreviation $F\triangleq\boldsymbol{\mu}X.A\{X\}$ to make the proof more readable.

$\langle S\rangle\,\Gamma, A\{X^+\} \vdash u : A\{\boldsymbol{\mu}X.A\{X\}\} \Mapsto \boldsymbol{\mu}X.A\{X\}, \Delta$ (Fold)

5. $\langle S\rangle\,\Gamma, A\{X^+\}, x : A\{F\}, x : A\{X\} \Mapsto X, x : A\{F\} \Mapsto A\{X\} \vdash x : X, \Delta$ (by Id)

4. $\langle S\rangle\,\Gamma, A\{X^+\}, x : A\{F\}, x : A\{X\} \Mapsto X, x : F \Mapsto X \vdash x : X, \Delta$ (by 5, (MonL))

3. $\langle S\rangle\,\Gamma, A\{X^+\}, x : A\{F\}, x : A\{X\} \Mapsto X \vdash x : X, \Delta$ (by 4, (μFixL))

2. $\langle S\rangle\,\Gamma, A\{X^+\}, x : A\{F\} \vdash x : F, \Delta$ (by 3, ($\forall$2R), (!R), ($\Rightarrow$R)

1. $\langle S\rangle\,\Gamma, A\{X^+\} \vdash u : A\{F\} \Mapsto F, \Delta$ (by 2, (!R), ($\Rightarrow$R))

$$\frac{\langle S\rangle\,\Gamma, A\{X^+\}, u : A\{B\} \Mapsto A\{C\} \vdash \Delta}{\langle S\rangle\,\Gamma, A\{X^+\}, u : B \Mapsto C \vdash \Delta}\;(\mathrm{MonL})$$

$$\frac{\langle S\rangle\,\Gamma, A\{X^+\} \vdash u : B \Mapsto C, \Delta}{\langle S\rangle\,\Gamma, A\{X^+\} \vdash u : A\{B\} \Mapsto A\{C\}, \Delta}\;(\mathrm{MonR})$$

[X *is not free in the conclusion*]

$$\frac{\langle S\rangle\,\Gamma, u : X, u : X \Mapsto A\{X\} \vdash \Delta}{\langle S\rangle\,\Gamma, u : \boldsymbol{\nu}X.A \vdash \Delta}\;(\nu\mathrm{L})$$

[x *is not free in the conclusion*]

$$\frac{\langle S\rangle\,\Gamma, x : B \vdash x : A\{X{\leftarrow}B\}, \Delta \quad \langle S\rangle\,\Gamma \vdash u : B, \Delta}{\langle S\rangle\,\Gamma \vdash u : \boldsymbol{\nu}X.A, \Delta}\;(\nu\mathrm{R})$$

[X *is not free in the conclusion*]

$$\frac{\langle S\rangle\,\Gamma, u : A\{X\} \Mapsto X \vdash u : X, \Delta}{\langle S\rangle\,\Gamma \vdash u : \boldsymbol{\mu}X.A, \Delta}\;(\mu\mathrm{R})$$

[x *is not free in the conclusion*]

$$\frac{\langle S\rangle\,\Gamma, x : A\{B\} \vdash x : B, \Delta \quad \langle S\rangle\,\Gamma, u : B \vdash \Delta}{\langle S\rangle\,\Gamma, u : \boldsymbol{\mu}X.A \vdash \Delta}\;(\mu\mathrm{L})$$

$$\frac{\langle S\rangle\,\Gamma, u : \boldsymbol{\mu}X.A\{X\} \Mapsto B \vdash \Delta}{\langle S\rangle\,\Gamma, u : A\{B\} \Mapsto B \vdash \Delta}\;(\mu\mathrm{FixL}) \qquad \frac{\langle S\rangle\,\Gamma \vdash u : A\{B\} \Mapsto B, \Delta}{\langle S\rangle\,\Gamma \vdash u : \boldsymbol{\mu}X.A\{X\} \Mapsto B, \Delta}\;(\mu\mathrm{FixR})$$

Fig. 16. Derived rules for the fixpoint operators.

In section 6.6 we give further examples illustrating the use of recursion.

5 Basic Proof Theory

In this section we develop some proof-theory for our logic, stating several admissible proof principles and a cut elimination result for the first-order fragment.

5.1 Admissible Rules

Most of the presented proof principles are size-preserving, and instrumental to the proof of cut elimination. We introduce a measure for the *size* of a derivation, in which certain occurrences of the (TL/TR) rules are not weighted. We will show below that any derivation can be transformed into a derivation for the same sequent where all occurrences of the (TL/TR) rules are simple.

Definition 5.1 (Simple occurrence) *In a derivation, an occurrence of a* (TL/TR) *inference rule is* simple *if it applies either to an instance of* (Id)*, or to another simple occurrence of a* (TL/TR) *inference rule.*

Definition 5.2 (Size of a derivation) *The* size of a derivation *is the number of rule occurrences it contains, other than simple occurrences of* (TL/TR) *inference rules.*

We then assert $\vdash_n \langle S\rangle\, \Gamma \boldsymbol{\vdash} \Delta$ to state that the given sequent has a derivation of size not exceeding n. We have the following useful admissible rules

Lemma 5.3 (Basic Admissible Rules) *The following size-preserving proof principles are admissible:*

$$\frac{[\ \varphi, \varphi' \in \mathcal{V} \cup \mathcal{X} \cup \mathcal{Z},\ \varphi' \textit{ not free in premise}\ ] \quad \vdash_n \langle S\rangle\, \Gamma \boldsymbol{\vdash} \Delta}{\vdash_n \langle S\{\varphi{\leftarrow}\varphi'\}\rangle\, \Gamma\{\varphi{\leftarrow}\varphi'\} \boldsymbol{\vdash} \Delta\{\varphi{\leftarrow}\varphi'\}}\ \text{(Ren)} \qquad \frac{[\ \Gamma \equiv_\alpha \Gamma' \textit{ and } \Delta \equiv_\alpha \Delta'\ ] \quad \vdash_n \langle S\rangle\, \Gamma \boldsymbol{\vdash} \Delta}{\vdash_n \langle S\rangle\, \Gamma' \boldsymbol{\vdash} \Delta'}\ (\alpha)$$

$$\frac{\vdash_n \langle S\rangle\, \Gamma \boldsymbol{\vdash} \Delta}{\vdash_n \langle S\{x{\leftarrow}\mathsf{m}\}\rangle\, \Gamma\{x{\leftarrow}\mathsf{m}\} \boldsymbol{\vdash} \Delta\{x{\leftarrow}\mathsf{m}\}}\ (\text{In}\mathcal{N}) \qquad \frac{\vdash_n \langle S, c\rangle\, \Gamma \boldsymbol{\vdash} \Delta \quad S \vdash c}{\vdash_n \langle S\rangle\, \Gamma \boldsymbol{\vdash} \Delta}\ \text{(CS)}$$

$$\frac{\vdash_n \langle S\rangle\, \Gamma \boldsymbol{\vdash} \Delta}{\vdash_n \langle S\{x{\leftarrow}u\}\rangle\, \Gamma\{x{\leftarrow}u\} \boldsymbol{\vdash} \Delta\{x{\leftarrow}u\}}\ (\text{In}\mathcal{I}) \qquad \frac{\vdash_n \langle S\rangle\, \Gamma \boldsymbol{\vdash} \Delta}{\vdash_n \langle S, S'\rangle\, \Gamma, \Gamma' \boldsymbol{\vdash} \Delta, \Delta'}\ \text{(W)}$$

Proof. See appendix. ∎

Lemma 5.4 (Replacement and Instantiation) *The inference rules presented below are admissible*

$$\frac{[\ x \textit{ not free in } S\ ] \quad \langle S\rangle\, x : A \boldsymbol{\vdash} x : B \quad \langle S\rangle\, x : B \boldsymbol{\vdash} x : A}{\langle S\rangle\, y : C[A] \boldsymbol{\vdash} y : C[B]}\ \text{(Rep)} \qquad \frac{[\ X \textit{ not free in the conclusion}\ ] \quad \langle S\rangle\, \Gamma \boldsymbol{\vdash} \Delta}{\langle S\rangle\, \Gamma\{X{\leftarrow}A\} \boldsymbol{\vdash} \Delta\{X{\leftarrow}A\}}\ \text{(In2)}$$

Proof. (Rep) By induction on the structure of the context $C[-]$. (In2) By induction on the derivation. ∎

Our primitive (Id) axiom is restricted to atomic formulas, however we have the following standard property for unrestricted formulas.

Lemma 5.5 *Every sequent of the form* $\langle S\rangle\, \Gamma, u : A \boldsymbol{\vdash} u : A, \Delta$*, where* A *in not atomic, has a cut- and contraction-free derivation.*

Proof. See appendix. ∎

We now introduce the following useful variants of the (TL) and (TR) rules.

$$\frac{\begin{array}{c}(\mathsf{n}\leftrightarrow\mathsf{m})A \Downarrow_S A' \\ \langle S\rangle\,\Gamma, u' : A' \vdash \Delta \;\; (\mathsf{m}\leftrightarrow\mathsf{n})u \doteq_S u'\end{array}}{\langle S\rangle\,\Gamma, u : A \vdash \Delta}\ (\mathrm{SL}) \qquad \frac{\begin{array}{c}(\mathsf{n}\leftrightarrow\mathsf{m})A \Downarrow_S A' \\ \langle S\rangle\,\Gamma \vdash u' : A', \Delta \;\; (\mathsf{m}\leftrightarrow\mathsf{n})u \doteq_S u'\end{array}}{\langle S\rangle\,\Gamma \vdash u : A, \Delta}\ (\mathrm{SR})$$

Definition 5.6 **S1** *is the proof system obtained from the base proof system* **S** *by replacing rules* (TL) *and* (TR) *with the rules* (SL) *and* (SR).

It is easy to see that if a sequent is derivable in **S1** then it is also derivable in the base system since $A \equiv_S B$ whenever $A \Downarrow_S B$. In fact, every **S1** derivation can be seen as a derivation in the base system just by interpreting (SR) and (SL) as (TR) and (TL) respectively. Conversely, if a sequent is derivable in the base system, it is also derivable in **S1** since any instance of (TL) or (TR) can be emulated using Cut, (SL) and (SR). Like with **S** derivations we call *simple* to any **S1** derivation in which all instances of (SL) and (SR) inference rules are simple (*cf.*, Definition 5.1). Hence, according to Definition 5.2, in a simple **S1** derivation no occurrence of the (SL) and (SR) rule is weighted.

Remark 5.7 The main difference between the system **S** and the system **S1**, is that all formulas occurring in a cut-free **S1** proof of a normalized sequent are normalized (Definition 3.18). Moreover, as the following Lemma shows, every **S** or **S1** proof of a normalized sequent can be transformed into a **S1** proof of the same sequent in which all formulas are normalized.

Lemma 5.8 (Simplification) *Assume* $(\mathsf{n}\leftrightarrow\mathsf{m})u \doteq_S u'$ *and* $(\mathsf{n}\leftrightarrow\mathsf{m})A \Downarrow_S A'$, $\Gamma \Downarrow_S \Gamma'$ *and* $\Delta \Downarrow_S \Delta'$. *Then the following size-preserving proof principles are admissible:*

(1) If $\vdash_n \langle S\rangle\,\Gamma, u : A \vdash \Delta$ *in* **S** *then* $\vdash_n \langle S\rangle\,\Gamma', u' : A' \vdash \Delta'$ *in* **S1**.
(2) If $\vdash_n \langle S\rangle\,\Gamma \vdash u : A, \Delta$ *in* **S** *then* $\vdash_n \langle S\rangle\,\Gamma' \vdash u' : A', \Delta'$ *in* **S1**.

The resulting derivations in **S1** *are simple and normalized. Moreover, if the original derivations in* **S** *are cut-free the resulting ones in* **S1** *are also cut-free.*

Proof. See appendix. ∎

A useful special case of Lemma 5.8 is the following fact.

Lemma 5.9 *Assume* $\Gamma \Downarrow_S \Gamma'$ *and* $\Delta \Downarrow_S \Delta'$. *If* $\vdash_n \langle S\rangle\,\Gamma \vdash \Delta$ *in* **S** *then* $\vdash_n \langle S\rangle\,\Gamma' \vdash \Delta'$ *in* **S1**.

Proof. By Lemma 5.8(2): let $u : A = \mathbf{0} : \mathbf{F}$ and note that if $\vdash_n \langle S\rangle\,\Gamma' \vdash \mathbf{0} : \mathbf{F}, \Delta'$ then $\vdash_n \langle S\rangle\,\Gamma' \vdash \Delta'$. ∎

$$\frac{\begin{array}{c}\langle S\rangle\,\Gamma \vdash v : A, u : A|B, \Delta \\ \langle S\rangle\,\Gamma \vdash t : B, u : A|B, \Delta \quad u \doteq_S v|t\end{array}}{\langle S\rangle\,\Gamma \vdash u : A|B, \Delta}\;(|\mathrm{RK}) \qquad \frac{\begin{array}{c}\langle S\rangle\,\Gamma, u : A \triangleright B \vdash t : A, \Delta \\ \langle S\rangle\,\Gamma, u : A \triangleright B, t|u : B \vdash \Delta\end{array}}{\langle S\rangle\,\Gamma, u : A \triangleright B \vdash \Delta}\;(\triangleright\mathrm{LK})$$

$$\frac{\begin{array}{c}u \doteq_S (\boldsymbol{\nu} \boldsymbol{n})t \\ \langle S\rangle\,\Gamma \vdash t : A, u : \boldsymbol{n}\circledR A, \Delta\end{array}}{\langle S\rangle\,\Gamma \vdash u : \boldsymbol{n}\circledR A, \Delta}\;(\circledR\mathrm{RK})$$

$$\frac{\langle S\rangle\,\Gamma \vdash t : A, u : \Diamond A, \Delta \quad u \rightarrow_S t}{\langle S\rangle\,\Gamma \vdash u : \Diamond A, \Delta}\;(\Diamond\mathrm{RK}) \qquad \frac{\langle S\rangle\,\Gamma, u : \forall x.A, u : A\{x\leftarrow \boldsymbol{m}\} \vdash \Delta}{\langle S\rangle\,\Gamma, u : \forall x.A \vdash \Delta}\;(\forall\mathrm{LK})$$

$$\frac{\begin{array}{c}u \doteq_S (\boldsymbol{\nu} \boldsymbol{n})v \quad \boldsymbol{n} \#_S \text{И}x.A \\ \langle S\rangle\,\Gamma \vdash u : \text{И}x.A, u : A\{x\leftarrow \boldsymbol{n}\}, \Delta\end{array}}{\langle S\rangle\,\Gamma \vdash u : \text{И}x.A, \Delta}\;(\text{И}\mathrm{R}) \qquad \frac{\begin{array}{c}u \doteq_S (\boldsymbol{\nu} \boldsymbol{n})v \quad \boldsymbol{n} \#_S \text{И}x.A \\ \langle S\rangle\,\Gamma, u : \text{И}x.A, u : A\{x\leftarrow \boldsymbol{n}\} \vdash \Delta\end{array}}{\langle S\rangle\,\Gamma, u : \text{И}x.A \vdash \Delta}\;(\text{И}\mathrm{L})$$

Fig. 17. Rules of the contraction-free system **CF**.

5.2 Cut Elimination

Our aim is now to prove the cut-elimination property for the first-order fragment of our logic. First, we introduce an alternative proof system **CF**. The system **CF** has no primitive contraction rules, but admits an admissible size-preserving contraction principle that plays an important role in the base case of the Cut Lemma 5.17 below. Then, we show that there are transformations between derivations in **CF**, **S1**, and **S**, such that the cut-elimination property for **CF** implies the cut-elimination property for **S**. From now on, we restrict to the first-order fragment of our logic.

Definition 5.10 **CF** *is the proof system obtained from the system* **S1** *by removing the contraction rules* (CL) *and* (CR)*, and replacing the rules* (∀L), (|R), (▷L), (ИL), (ИR), (®R)*, and* (◇R) *by the rules shown in Figure 17.*

The **CF** rules are identical to the corresponding ones in system **S**, except in that they embed a contraction step (*cf.* the system **G3c** in [23]), that is, the principal formula is copied in the premise. The replaced rules are precisely the non-invertible ones. Note that in sequent calculus presentations of classical logic (*e.g.*, Gentzen's LK) (∀L) is not invertible, and in classical linear logic (⊗R) is not invertible (*cf.*, (|R)) and (⊸L) is not invertible (*cf.* (▷)).

Note that any derivation in **CF** can be immediately transformed into a derivation in the basic system, since each **CF** rule that does not belong to the system **S1** can be easily simulated by the corresponding rule followed by contraction.

Lemma 5.11 *If a sequent has a derivation in the system* **CF***, it has a derivation in the system* **S1***. Moreover, if the original derivation is cut-free, so is the resulting one.*

Moreover, since the proof transformations given in Lemmas 5.3 and 5.8 are completely structure-preserving, we can also verify that

Lemma 5.12 (Admissible Rules for the CF system) *The proof principles in Lemma 5.3 and Lemma 5.8 hold exactly as stated for the* **CF** *system.*

Lemma 5.13 (Inversion) *The following size-preserving proof principles are admissible in the system* **CF***, provided the sequents shown are normalized.*

(1) If $\vdash_n \langle S\rangle\, \Gamma, u : A \wedge B \Vdash \Delta$ *then* $\vdash_n \langle S\rangle\, \Gamma, u : A, u : B \Vdash \Delta$.
(2) If $\vdash_n \langle S\rangle\, \Gamma \Vdash u : A \wedge B, \Delta$ *then*
$\vdash_n \langle S\rangle\, \Gamma \Vdash u : A, \Delta$ *and* $\vdash_n \langle S\rangle\, \Gamma \Vdash u : B, \Delta$.
(3) If $\vdash_n \langle S\rangle\, \Gamma \Vdash u : A \Rightarrow B, \Delta$ *then* $\langle S\rangle\, \Gamma, u : A \Vdash u : B, \Delta$.
(4) If $\vdash_n \langle S\rangle\, \Gamma, u : A \Rightarrow B \Vdash \Delta$ *then*
$\vdash_n \langle S\rangle\, \Gamma, u : B \Vdash \Delta$ *and* $\vdash_n \langle S\rangle\, \Gamma \Vdash u : A, \Delta$.
(5) If $\vdash_n \langle S\rangle\, \Gamma \Vdash u : \forall x.A, \Delta$ *then*
$\vdash_n \langle S\rangle\, \Gamma \Vdash u : A\{x\leftarrow y\}, \Delta$*, for any fresh* y.
(6) If $\vdash_n \langle S\rangle\, \Gamma \Vdash u : A \rhd B, \Delta$ *then*
$\vdash_n \langle S\rangle\, \Gamma, \mathscr{x} : A \Vdash \mathscr{x}|u : B, \Delta$*, for any fresh* $\mathscr{x}$.
(7) If $\vdash_n \langle S\rangle\, \Gamma, u : A|B \Vdash \Delta$ *then*
$\vdash_n \langle S, u \doteq \mathscr{x}|\mathscr{y}\rangle\, \Gamma, \mathscr{x} : A, \mathscr{y} : B \Vdash \Delta$*, for any fresh* $\mathscr{x}, \mathscr{y}$.
(8) If $\vdash_n \langle S\rangle\, \Gamma, u : n \circledR A \Vdash \Delta$ *then*
$\vdash_n \langle S, u \doteq (\boldsymbol{\nu} n)\mathscr{x}\rangle\, \Gamma, \mathscr{x} : A, \Vdash \Delta$*, for any fresh* $\mathscr{x}$.
(9) If $\vdash_n \langle S\rangle\, \Gamma, u : \mathbf{0} \Vdash \Delta$ *then* $\vdash_n \langle S, u \doteq \mathbf{0}\rangle\, \Gamma \Vdash \Delta$.

The resulting derivations are normalized. Moreover, if the original derivations are cut-free, so are the resulting derivations; if the original derivations are simple, so are the resulting derivations.

Proof. See appendix. ■

Lemma 5.14 (Contraction Elimination) *The size-preserving proof principles given below are admissible in the system* **CF***, provided the sequents shown are normalized:*

$$\frac{\vdash_n \langle S\rangle\, \Gamma \Vdash u : A, u : A, \Delta}{\vdash_n \langle S\rangle\, \Gamma, u : A \Vdash \Delta}\ (\mathrm{CR}) \quad \frac{\vdash_n \langle S\rangle\, \Gamma, u : A, u : A \Vdash \Delta}{\vdash_n \langle S\rangle\, \Gamma, u : A \Vdash \Delta}\ (\mathrm{CL})$$

The resulting derivations are normalized. Moreover, if the original derivations are cut-free, so are the resulting derivations; if the original derivations are simple, so are the resulting derivations.

Proof. See appendix. ■

We can now state:

Proposition 5.15 *If a normalized sequent is derivable in* **S1** *then it is derivable in* **CF**. *The resulting derivation is normalized. Moreover, if the original derivation is simple, so is the resulting one.*

Proof. By induction on the structure of the original derivation, we construct a **CF** derivation by replacing every occurrence of ($\forall$L), ($|$R), ($\triangleright$L), ($\reflectbox{N}$L), ($\reflectbox{N}$R), (®R), and ($\Diamond$R) by the corresponding **CF** rule, after adding the extra required formula in the premise using (W), and removing every occurrence of (CL) and (CR) using Lemma 5.14. ∎

We are now in a position to show that the first-order fragment of the spatial logic enjoys the cut elimination property. This result is reasonable evidence that our addition of structural and freshness constraints to sequents and inference rules is rather canonical. For instance, cuts on spatial formulas are eliminated quite uniformly, by matching fresh process variables (on one side) against the given witnesses (on the other), and then eliminating the remaining redundant structural constraints. The cut elimination case for freshness quantifications deserves a more detailed discussion. Consider the following cut

$$\dfrac{\dfrac{\langle S\rangle\, \Gamma \vdash u : A\{x{\leftarrow}\mathsf{n}\}, \Delta}{\langle S\rangle\, \Gamma \vdash u : \reflectbox{N}x.A, \Delta} \quad \dfrac{\langle S\rangle\, \Gamma, u : A\{x{\leftarrow}\mathsf{m}\} \vdash u : \Delta}{\langle S\rangle\, \Gamma, u : \reflectbox{N}x.A \vdash \Delta}}{\langle S\rangle\, \Gamma \vdash \Delta}$$

To eliminate this we need to cut $u : A\{x{\leftarrow}\mathsf{n}\}$ against $u : A\{x{\leftarrow}\mathsf{m}\}$, while preserving the sequent contexts Γ, Δ untouched. In general m and n are different name terms denoting distinct names (we could even have $\mathsf{m}\#_S\,\mathsf{n}$ provably). In fact, soundness of this cut follows from the fact that a fresh name is (in the sense of equivariance in Nominal Logic) indistinguishable from any other fresh name. In proof-theoretic terms, the equivariance property has as consequence that, in the apartness conditions made explicit by the premises of such a cut, we can actually transform (using Lemma 5.8) the derivation of $\langle S\rangle\, \Gamma \vdash u : A\{x{\leftarrow}\mathsf{m}\}, \Delta$ into a derivation of $\langle S\rangle\, \Gamma \vdash u : A\{x{\leftarrow}\mathsf{n}\}, \Delta$. For this transformation to go through the use of formal transpositions seems to be essential both in the π-algebra and in the syntax of formulas and terms.

Definition 5.16 (Single-cut derivation) *A* single-cut derivation *is a derivation with a single instance of the* (Cut) *rule, occurring at its root.*

Lemma 5.17 (Cut Lemma) *If a normalized sequent has a single-cut simple and normalized* **CF** *derivation then it has a simple and normalized* **CF** *cut-free derivation.*

Proof. The root of the derivation of the given sequent has the form

$$\dfrac{\dfrac{\pi_1(n)}{\langle S\rangle\, \Gamma \vdash u : A, \Delta} \quad \dfrac{\pi_2(m)}{\langle S\rangle\, \Gamma, u : A \vdash \Delta}}{\langle S\rangle\, \Gamma \vdash \Delta}\ (\text{Cut})$$

where $\pi_1(n)$ and $\pi_2(m)$ are cut-free simple derivations for the sequents $\langle S\rangle\,\Gamma \Vdash u : A, \Delta$ and $\langle S\rangle\,\Gamma, u : A \Vdash \Delta$, of sizes n and m respectively. We use the notation $\pi(n)$ to assert that π is a derivation of size n of the sequent in the conclusion of the rule. The proof proceeds by induction on the measure $(|A|, n+m)$, where $|A|$ is the structural complexity of the cut-formula A, $n+m$ is the sum of the sizes n and m of the derivations that occurs as premises of the cut, and the pairs $(|A|, m+n)$ are ordered lexicographically. We split the various possible forms of such premises as follows: (1) one of the premises is an instance of (Id) or (SL), (2) one of the premises is an instance of a world rule, (3) one of the premises is an instance of (И), (4) one of the premises is an instance of a logical rule that does not introduce the cut-formula, or (5) both premises are instances of logical rules, both introducing the cut formula.

(1) (Case (Cut) - (Id/SL)) Suppose (Id/SL) occurs in the right premise of the cut. Since the derivation is simple by assumption, it must have the form $(1.A)$ below

$$\dfrac{\dfrac{\pi_1(n)}{\langle S\rangle\,\Gamma \Vdash u : A, \Delta} \qquad \begin{array}{c}\langle S\rangle\,\Gamma^* \Vdash \Delta^* \ \ (\mathrm{Id}) \\ \vdots \\ \langle S\rangle\,\Gamma, u : A \Vdash \Delta\end{array}}{\langle S\rangle\,\Gamma \Vdash \Delta}\ (1.A) \qquad \dfrac{{\pi_1}'(n)}{\langle S\rangle\,\Gamma \Vdash v : C, v : C, \Delta''}\ (1.B)$$

where the dots stand for a sequence of $k \geq 0$ applications of the (SL) or (SR) rules. Hence, Γ^* has the form $\Gamma', t : B$, and Δ^* has the form $t : B, \Delta'$. We must consider two cases: either the occurrence $t : B$ in Γ^* results from $u : A$ below in $\langle S\rangle\,\Gamma, u : A \Vdash \Delta$, or it does not. In the first case, there is a sequence of transpositions ρ such that $\rho u \doteq_S t$ and $\rho A \Downarrow_S B$, and a sequence of transpositions σ such that $\sigma v \doteq_S t$ and $\sigma C \Downarrow_S B$, for some $v : C$ in Δ (so Δ has the form $v : C, \Delta''$). Therefore we have that $\sigma^{-1}\rho A \Downarrow_S C$ and $\sigma^{-1}\rho u \doteq_S v$. Hence, by Lemma 5.8(1), there is the derivation $(1.B)$ above, and we conclude by (CR) Lemma 5.14. In the second case, we can then build a cut-free simple proof of $\langle S\rangle\,\Gamma \Vdash \Delta$ of size equal to one by removing the premise $u : A$ and its ancestors from every sequent above $\langle S\rangle\,\Gamma, u : A \vdash \Delta$. The case in which the instance of (Id/SL) occurs as the left premise of the cut is handled symmetrically, also by Lemma 5.8.

(2) (Case Cut - (S$-$)) We have

$$\dfrac{\dfrac{\dfrac{\pi_1(n)}{\langle S, S'\rangle\,\Gamma \Vdash u : A, \Delta \quad u_1 \doteq_S v_1}}{\langle S\rangle\,\Gamma \Vdash u : A, \Delta}\ (\mathrm{S}-) \qquad \dfrac{\pi_2(m)}{\langle S\rangle\,\Gamma, u : A \Vdash \Delta}}{\langle S\rangle\,\Gamma \Vdash \Delta}$$

We can now build the derivation

$$\dfrac{\dfrac{\dfrac{\pi_1(n)}{\langle S,S'\rangle\,\Gamma,u:A\vdash\Delta}\qquad\dfrac{{\pi_2}'(m)}{\langle S,S'\rangle\,\Gamma,u:A\vdash\Delta}}{\langle S,S'\rangle\,\Gamma\vdash\Delta}\quad u_1\doteq_S v_1}{\langle S\rangle\,\Gamma\vdash\Delta}\;(\mathrm{S}-)$$

where $\pi_2'(m)$ is obtained from $\pi_2(m)$ by (W). By induction hypothesis, there is a cut-free derivation of $\langle S,S'\rangle\,\Gamma\vdash\Delta$, so we conclude by (S−).

(3) (Case Cut - (И) We handle the case in which the conclusion of (И) is the left premise of the cut, being the right case handled symmetrically. Hence, we have

$$\dfrac{\dfrac{\dfrac{\pi_1(n)}{\langle S,t\doteq(\boldsymbol{\nu}x)x\,,x\,\#\,N\rangle\,\Gamma\vdash u:A,\Delta}}{\langle S\rangle\,\Gamma\vdash u:A,\Delta}\;(\text{И})\qquad\dfrac{\pi_2(m)}{\langle S\rangle\,\Gamma,u:A\vdash\Delta}}{\langle S\rangle\,\Gamma\vdash\Delta}$$

We can now build the derivation

$$\dfrac{\dfrac{\dfrac{\pi_1(n)}{\langle S,t\doteq(\boldsymbol{\nu}x)x\,,x\,\#\,N\rangle\,\Gamma\vdash u:A,\Delta}\qquad\dfrac{{\pi_2}'(m)}{\langle S,t\doteq(\boldsymbol{\nu}x)x\,,x\,\#\,N\rangle\,\Gamma,u:A\vdash\Delta}}{\langle S,t\doteq(\boldsymbol{\nu}x)x\,,x\,\#\,N\rangle\,\Gamma\vdash\Delta}}{\langle S\rangle\,\Gamma\vdash\Delta}\;(\text{И})$$

where $\pi_2'(m)$ is obtained from $\pi_2(m)$ by (W). By induction hypothesis, there is a cut-free derivation of $\langle S,t\doteq(\boldsymbol{\nu}x)x\,,x\,\#\,N\rangle\,\Gamma\vdash\Delta$, so we conclude by (И).

(4) (Case 4.LR) We consider here the case in which the left premise of the cut rule is the conclusion of a right logical rule that *does not introduce* the cut formula. We consider the general case of a two-premise rule, but the argument can be replicated for single premise rule like (∀R), or (⇒R), which adds an hypothesis (*e.g.*, Γ_1) to the left context. Hence we have,

$$\dfrac{\dfrac{\dfrac{\pi_1(n)}{\langle S\rangle\,\Gamma,\Gamma_1\vdash u:A,\Delta_1}\qquad\dfrac{\pi_2(m)}{\langle S\rangle\,\Gamma,\Gamma_2\vdash u:A,\Delta_2}\quad C_S}{\langle S\rangle\,\Gamma\vdash u:A,\Delta}\;(-\mathrm{R})\qquad\dfrac{\pi_3(k)}{\langle S\rangle\,\Gamma,u:A\vdash\Delta}}{\langle S\rangle\,\Gamma\vdash\Delta}$$

where in general the instance of (−R) may also have some assertions C_S as premises. Now, by (W) we can build derivations

$$\dfrac{\pi_3'(k)}{\langle S\rangle\,\Gamma,u:A,\Gamma_1\vdash\Delta,\Delta_1}\qquad\qquad\dfrac{\pi_3''(k)}{\langle S\rangle\,\Gamma,u:A,\Gamma_2\vdash\Delta,\Delta_2}$$

hence we can construct the derivations

$$\dfrac{\dfrac{\pi_1(n)}{\langle S\rangle\,\Gamma,\Gamma_1 \vdash u:A,\Delta,\Delta_1} \qquad \dfrac{\pi_3'(k)}{\langle S\rangle\,\Gamma,u:A,\Gamma_1 \vdash \Delta,\Delta_1}}{\langle S\rangle\,\Gamma,\Gamma_1 \vdash \Delta,\Delta_1}$$

and

$$\dfrac{\dfrac{\pi_2(m)}{\langle S\rangle\,\Gamma,\Gamma_2 \vdash u:A,\Delta,\Delta_2} \qquad \dfrac{\pi_3''(k)}{\langle S\rangle\,\Gamma,u:A,\Gamma_2 \vdash \Delta,\Delta_2}}{\langle S\rangle\,\Gamma,\Gamma_2 \vdash \Delta,\Delta_2}$$

By induction hypothesis, there are cut-free derivations for the $\langle S\rangle\,\Gamma,\Gamma_1 \vdash \Delta,\Delta_1$ and $\langle S\rangle\,\Gamma,\Gamma_2 \vdash \Delta,\Delta_2$. Hence, by (−R) we can build a cut-free derivation of $\langle S\rangle\,\Gamma \vdash \Delta$, since all possibly required assertions C_S still apply.

(Case 4.LL) The left premise of the cut is the conclusion of a left logical rule that *does not introduce* the cut formula. Note that in our proof system all left rules have at most one premise, although some require testing certain assertions C (namely (ИL)). Hence we have in general

$$\dfrac{\dfrac{\dfrac{\pi_1(n)}{\langle S,S'\rangle\,\Gamma' \vdash u:A,\Delta \quad C_S}}{\langle S\rangle\,\Gamma \vdash u:A,\Delta}\,(-\mathrm{L}) \qquad \dfrac{\pi_2(m)}{\langle S\rangle\,\Gamma,u:A \vdash \Delta}}{\langle S\rangle\,\Gamma \vdash \Delta}$$

Using (W) on $\pi_1(n)$ and $\pi_2(m)$ we can build the derivation

$$\dfrac{\dfrac{\pi_1'(n)}{\langle S,S'\rangle\,\Gamma,\Gamma' \vdash u:A,\Delta} \qquad \dfrac{\pi_2'(m)}{\langle S,S'\rangle\,\Gamma,u:A,\Gamma' \vdash \Delta}}{\langle S,S'\rangle\,\Gamma,\Gamma' \vdash \Delta}$$

By induction hypothesis, we obtain a cut-free derivation of $\langle S,S'\rangle\,\Gamma,\Gamma' \vdash \Delta$. Since $C_{S,S'}$ holds, by (−L) we obtain a cut-free derivation of $\langle S\rangle\,\Gamma \vdash \Delta$.

(Case 4.RR) The right premise of the cut is the conclusion of a right logical rule that does not introduce the cut formula. Like (Case 4.LL) above.

(Case 4.RL) The right premise of the cut is the conclusion of a left logical rule that does not introduce the cut formula. Like (Case 4.LR) above.

(5) We now consider all cases where the premises of the cut are conclusions of (left and right) logical rules, both introducing the cut formula. Then the rule that occurs in the left (resp. left) premise is a right-rule (resp. left-rule). We consider the various possible rule pairs, there is one such pair for each logical connective. We show in detail the most interesting cases.

(Case of $|$) We have

$$\dfrac{\dfrac{\dfrac{\pi_1(n)}{\langle S\rangle\,\Gamma \vdash u' : A, u : A|B, \Delta} \quad \dfrac{\pi_2(m)}{\langle S\rangle\,\Gamma \vdash u'' : B, u : A|B, \Delta}}{\langle S\rangle\,\Gamma \vdash u : A|B, \Delta} \quad \dfrac{\dfrac{\pi_3(k)}{\langle S'\rangle\,\Gamma, x : A, y : B \vdash \Delta}}{\langle S\rangle\,\Gamma, u : A|B \vdash \Delta}}{\langle S\rangle\,\Gamma \vdash \Delta}$$

where $u \doteq_S u'|u''$, and $S' = S, u \doteq x\,|\,y$. By (In$\mathcal{I}$) with $\{x \leftarrow u'\}$ and $\{y \leftarrow u''\}$ on π_3 we get ${\pi'}_3(k)$

$$\dfrac{{\pi'}_3(k)}{\langle S, u \doteq u'|u''\rangle\,\Gamma, u' : A, u'' : B \vdash \Delta}$$

(note that by the side condition on ($|$L) x and y do not occur in S, Γ, Δ). Since $u \doteq_S u'|u''$, by (CS) we get ${\pi''}_3(k)$

$$\dfrac{{\pi''}_3(k)}{\langle S\rangle\,\Gamma, u' : A, u'' : B \vdash \Delta}$$

We now build

$$\dfrac{\langle S\rangle\,\Gamma \vdash u' : A, u : A|B, \Delta \quad \langle S\rangle\,\Gamma, u : A|B \vdash \Delta}{\langle S\rangle\,\Gamma \vdash u' : A, \Delta}$$

and

$$\dfrac{\langle S\rangle\,\Gamma \vdash u'' : B, u : A|B, \Delta \quad \langle S\rangle\,\Gamma, u : A|B \vdash \Delta}{\langle S\rangle\,\Gamma \vdash u'' : B, \Delta}$$

By induction hypothesis, these cuts can be eliminated. By (W) from the derivation of $\langle S\rangle\,\Gamma \vdash u' : A, \Delta$ above, we obtain a derivation of $\langle S\rangle\,\Gamma, u'' : B \vdash u' : A, \Delta$. Now we construct

$$\dfrac{\begin{array}{c}\vdots\\ \langle S\rangle\,\Gamma \vdash u'' : B, \Delta\end{array} \quad \dfrac{\begin{array}{c}\vdots\\ \langle S\rangle\,\Gamma, u'' : B \vdash u' : A, \Delta\end{array} \quad \begin{array}{c}\vdots\\ \langle S\rangle\,\Gamma, u' : A, u'' : B \vdash \Delta\end{array}}{\langle S\rangle\,\Gamma, u'' : B \vdash \Delta}}{\langle S\rangle\,\Gamma \vdash \Delta}$$

By induction hypothesis, these two cuts can be successively eliminated.

(Case of $\rhd$) Let

$$\dfrac{\dfrac{\dfrac{\pi_1(n)}{\langle S\rangle\,\Gamma, x : A \vdash v' : B, \Delta}}{\langle S\rangle\,\Gamma \vdash u : A \rhd B, \Delta} \quad \dfrac{\dfrac{\pi_2(m)}{\langle S\rangle\,\Gamma, u : A \rhd B \vdash t : A, \Delta} \quad \dfrac{\pi_3(k)}{\langle S\rangle\,\Gamma, u : A \rhd B, t|u : B \vdash \Delta}}{\langle S\rangle\,\Gamma, u : A \rhd B \vdash \Delta}}{\langle S\rangle\,\Gamma \vdash \Delta}$$

where $v' \doteq_S x\,|u$. By (In$\mathcal{I}$) with $\{x\leftarrow t\}$ on $\pi_1(n)$ and Lemma 5.8(1) we get

$$\dfrac{\pi'_1(n)}{\langle S\rangle\, \Gamma, t : A \vdash t|u : B, \Delta}$$

since by Lemma 3.10(1) $v'\{x\leftarrow t\} \doteq_S t|u$ (note that by the side condition on ($\triangleright$R) x does not occur in S, Γ, Δ). We can now build

$$\dfrac{\dfrac{\pi'(n+1)}{\langle S\rangle\, \Gamma \vdash u : A \triangleright B, t : A, \Delta} \qquad \dfrac{\pi_2(m)}{\langle S\rangle\, \Gamma, u : A \triangleright B \vdash t : A, \Delta}}{\langle S\rangle\, \Gamma \vdash t : A, \Delta}$$

where $\pi'(n+1)$ is obtained from the left premise of the original cut by (W). In a similar way we construct

$$\dfrac{\dfrac{\pi''(n+1)}{\langle S\rangle\, \Gamma, t|u : B \vdash u : A \triangleright B, \Delta} \qquad \dfrac{\pi_3(k)}{\langle S\rangle\, \Gamma, u : A \triangleright B, t|u : B \vdash \Delta}}{\langle S\rangle\, \Gamma, t|u : B \vdash \Delta}$$

By induction hypothesis, these two cuts can be eliminated. By (W) on the first subderivation above we get

$$\begin{array}{c}\vdots\\ \langle S\rangle\, \Gamma \vdash t : A, t|u : B, \Delta\end{array}$$

We now build the following derivation

$$\dfrac{\begin{array}{c}\vdots\\ \langle S\rangle\, \Gamma, t|u : B \vdash \Delta\end{array} \qquad \dfrac{\begin{array}{c}\vdots\\ \langle S\rangle\, \Gamma \vdash t : A, t|u : B, \Delta\end{array} \quad \dfrac{\pi'_1(n)}{\langle S\rangle\, \Gamma, t : A \vdash t|u : B, \Delta}}{\langle S\rangle\, \Gamma \vdash t|u : B, \Delta}}{\langle S\rangle\, \Gamma \vdash \Delta}$$

To conclude, we use the induction hypothesis to eliminate the cut on B, and then the cut on A, like in the cases for $\wedge$ and $\Rightarrow$ above.

(Case of $\forall$) We have

$$\dfrac{\dfrac{\dfrac{\pi_1(n)}{\langle S\rangle\, \Gamma \vdash u : A\{x\leftarrow y\}, \Delta}}{\langle S\rangle\, \Gamma \vdash u : \forall x.A, \Delta} \qquad \dfrac{\dfrac{\pi_2(m)}{\langle S\rangle\, \Gamma, u : A\{x\leftarrow \boldsymbol{p}\}, u : \forall x.A \vdash \Delta}}{\langle S\rangle\, \Gamma, u : \forall x.A \vdash \Delta}}{\langle S\rangle\, \Gamma \vdash \Delta}$$

where y does not occur free in u, S, Γ, Δ. By (In$\mathcal{N}$) with $\{x\leftarrow \boldsymbol{p}\}$ on $\pi_1(n)$

$$\dfrac{\pi'_1(n)}{\langle S\rangle\, \Gamma \vdash u : A\{x\leftarrow \boldsymbol{p}\}, \Delta}$$

Using (Cut) we can build

$$\dfrac{\dfrac{\pi_2(n+2)}{\langle S\rangle\,\Gamma, u : A\{x\leftarrow\boldsymbol{p}\} \vdash u : \forall x.A, \Delta} \qquad \dfrac{\pi_2(m)}{\langle S\rangle\,\Gamma, u : A\{x\leftarrow\boldsymbol{p}\}, u : \forall x.A \vdash \Delta}}{\langle S\rangle\,\Gamma, u : A\{x\leftarrow\boldsymbol{p}\} \vdash \Delta}$$

where the left premise comes from the left premise of the original cut by (W). By induction hypothesis, this cut can be eliminated. We now build the following single-cut derivation

$$\dfrac{\dfrac{}{\langle S\rangle\,\Gamma \vdash u : A\{x\leftarrow\boldsymbol{p}\}, \Delta} \qquad \dfrac{}{\langle S\rangle\,\Gamma, u : A\{x\leftarrow\boldsymbol{p}\} \vdash \Delta}}{\langle S\rangle\,\Gamma \vdash \Delta}$$

and conclude by the induction hypothesis.

(Case of И) We have

$$\dfrac{\dfrac{\dfrac{\pi_1(n)}{\langle S\rangle\,\Gamma \vdash u : A\{z\leftarrow\boldsymbol{p}\}, u : \text{И}z.A, \Delta}}{\langle S\rangle\,\Gamma \vdash u : \text{И}z.A, \Delta} \qquad \dfrac{\dfrac{\pi_2(m)}{\langle S\rangle\,\Gamma, u : A\{z\leftarrow\boldsymbol{q}\}, u : \text{И}z.A \vdash \Delta}}{\langle S\rangle\,\Gamma, u : \text{И}z.A \vdash \Delta}}{\langle S\rangle\,\Gamma \vdash \Delta}$$

where $\boldsymbol{p}\#_S\,\text{И}z.A$ and $\boldsymbol{q}\#_S\,\text{И}z.A$, and $u \doteq_S (\boldsymbol{\nu p})u'$ and $u \doteq_S (\boldsymbol{\nu q})u''$. We can now build the derivation

$$\dfrac{\dfrac{\pi_1(n)}{\langle S\rangle\,\Gamma \vdash u : A\{z\leftarrow\boldsymbol{p}\}, u : \text{И}z.A, \Delta} \qquad \dfrac{\pi'(m+1)}{\langle S\rangle\,\Gamma, u : \text{И}z.A \vdash u : A\{z\leftarrow\boldsymbol{p}\}, \Delta}}{\langle S\rangle\,\Gamma \vdash u : A\{z\leftarrow\boldsymbol{p}\}, \Delta}$$

where the right premise is obtained from the right premise of the initial cut by (W). Symmetrically, we can build the derivation

$$\dfrac{\dfrac{\pi''(n+1)}{\langle S\rangle\,\Gamma, u : A\{z\leftarrow\boldsymbol{q}\} \vdash u : \text{И}z.A, \Delta} \qquad \dfrac{\pi_2(m)}{\langle S\rangle\,\Gamma, u : A\{z\leftarrow\boldsymbol{q}\}, u : \text{И}z.A \vdash \Delta}}{\langle S\rangle\,\Gamma, u : A\{z\leftarrow\boldsymbol{q}\} \vdash \Delta}$$

where the left premise is obtained from the left premise of the initial cut by (W). These two cuts can be eliminated by the induction hypothesis. Since $(\boldsymbol{p}\leftrightarrow\boldsymbol{q})A\{z\leftarrow\boldsymbol{p}\} \Downarrow_S A\{z\leftarrow\boldsymbol{q}\}$ by Lemma 3.20 and $(\boldsymbol{p}\leftrightarrow\boldsymbol{q})u \doteq_S u$ by (Swap Erase), by Lemma 5.8 there is a derivation of

$$\Gamma \vdash u : A\{z\leftarrow\boldsymbol{q}\}, \Delta$$

Then, we build the single-cut derivation

$$\dfrac{\dfrac{}{\langle S\rangle\,\Gamma \vdash u : A\{z\leftarrow\boldsymbol{q}\}, \Delta} \qquad \dfrac{}{\langle S\rangle\,\Gamma, u : A\{z\leftarrow\boldsymbol{q}\} \vdash \Delta}}{\langle S\rangle\,\Gamma \vdash \Delta}$$

and conclude by the induction hypothesis. ■

Theorem 5.18 (Cut Elimination) *If a sequent has a first-order derivation in* **S** *then it has a derivation in* **S** *without any instance of the* (Cut) *rule.*

Proof. Assume that a sequent $\langle S \rangle\, \Gamma \vdash \Delta$ has a first-order derivation π in the base system **S**. Without loss of generality, we assume that the sequent is normalized. If the sequent has a derivation in **S**, by Lemma 5.9 it has a simple and normalized **S1** derivation π'. By Proposition 5.15, we conclude that $\langle S \rangle\, \Gamma \vdash \Delta$ has a simple and normalized derivation π'' in **CF**. Now, by induction on the number of instances of (Cut) in π'', we can build a cut-free simple and normalized derivation of the same sequent by iterating Lemma 5.17 for each minimal single-cut subderivation of the derivation π'', thus ending up with a cut-free **CF** derivation of the original sequent. By Lemma 5.11, we conclude that the sequent has a cut-free derivation in **S1** and thus also in **S**. ■

6 Examples

In this Section we go though a sequence of short examples to show how our logic is applicable to reasoning about distributed concurrent systems. We are necessarily brief here, and show only very elementary examples, but most interesting logical operators are covered.

6.1 Some Simple Spatial Properties

We show a simple derivation of the fact that $(A|B) \wedge \mathbf{0}$ entails A, meaning that if a process satisfies $(A|B) \wedge \mathbf{0}$ then it satisfies A. The intuition is that if a process P satisfies both $(A|B)$ and $\mathbf{0}$, then P is (structurally equivalent to) the $\mathbf{0}$ process, which is the same as $\mathbf{0}|\mathbf{0}$; hence $\mathbf{0}$ satisfies A (and B). We conclude that P satisfies A. This derivation illustrates: a property combining spatial and propositional operators; the use of constraint manipulation; and the use of one of the world rules, namely, $(S|\mathbf{0})$ corresponding to the "zero law" of π-calculus processes: if $P|Q \equiv \mathbf{0}$ then $P \equiv \mathbf{0}$.

5. $\langle S, u \doteq x|y, u \doteq \mathbf{0}, x \doteq \mathbf{0} \rangle\, \Gamma, x : A, y : B \vdash u : A, \Delta$ (by (Id) since $u \doteq_S x$)

4. $\langle S, u \doteq x|y, u \doteq \mathbf{0} \rangle\, \Gamma, x : A, y : B \vdash u : A, \Delta$ (by 5, (S|**0**) since $x|y \doteq_S \mathbf{0}$)

3. $\langle S, u \doteq x|y \rangle\, \Gamma, x : A, y : B, u : \mathbf{0} \vdash u : A, \Delta$ (by 4, (**0**L))

2. $\langle S \rangle\, \Gamma, u : (A|B), u : \mathbf{0} \vdash u : A, \Delta$ (by 3, (|L))

1. $\langle S \rangle\, \Gamma, u : (A|B) \wedge \mathbf{0} \vdash u : A, \Delta$ (by 2, ($\wedge$L))

Note that the proof is fairly simple, particularly if conducted bottom up. Most constraints are generated from the goal by using all the applicable left rules, and the final constraint $x \doteq \mathbf{0}$ is generated by closing up the constraint set under deduction, via (S|$\mathbf{0}$). Finally, (Id) involves a simple equivalence check in S. It is common for our derivations, when read bottom-up, to have this mechanical flavor.

As a further interesting example, we prove a sequent for which does not exists a contraction free-proof in our system.

11. $\langle\rangle\, x : A \vdash x : \mathbf{0}, x : A, x : \mathbf{0}$ (by (Id))

10. $\langle\rangle \vdash x : \neg A, x : \mathbf{0}, x : A, x : \mathbf{0}$ (by 11 ($\neg$R))

9. $\langle\rangle \vdash x : \neg A, x : \mathbf{0}, \mathbf{0} : \neg A, \mathbf{0} : \mathbf{0}$ (by ($\mathbf{0}$R))

8. $\langle\rangle \vdash x : \neg A, x : \mathbf{0}, x : (A \vee \mathbf{0})$ (by 10 ($\vee$R))

7. $\langle\rangle \vdash x : \neg A, x : \mathbf{0}, \mathbf{0} : (\neg A \vee \mathbf{0})$ (by 9 ($\vee$R))

6. $\langle\rangle \vdash x : \neg A, x : \mathbf{0}, x : (A \vee \mathbf{0})|(\neg A \vee \mathbf{0})$ (by 7,8, (|R), since $u \doteq u|\mathbf{0}$))

5. $\langle\rangle \vdash \mathbf{0} : A, \mathbf{0} : \mathbf{0}, x : (A \vee \mathbf{0})|(\neg A \vee \mathbf{0})$ (by ($\mathbf{0}$R))

4. $\langle\rangle \vdash \mathbf{0} : A \vee \mathbf{0}, x : (A \vee \mathbf{0})|(\neg A \vee \mathbf{0})$ (by 5, ($\vee$R))

3. $\langle\rangle \vdash x : \neg A \vee \mathbf{0}, x : (A \vee \mathbf{0})|(\neg A \vee \mathbf{0})$ (by 6, ($\vee$R))

2. $\langle\rangle \vdash x : (A \vee \mathbf{0})|(\neg A \vee \mathbf{0}), x : (A \vee \mathbf{0})|(\neg A \vee \mathbf{0})$ (by 3,4, (|R), since $u \doteq \mathbf{0}|u$)

1. $\langle\rangle \vdash x : (A \vee \mathbf{0})|(\neg A \vee \mathbf{0})$ (by 2, (CR))

Indeed, any cut-free proof of $\langle\rangle \vdash x : (A \vee \mathbf{0})|(\neg A \vee \mathbf{0})$ must end either by an application of contraction or by an application of (|R). So, in absence of contraction, the only possible premises are either $\langle\rangle \vdash x : (A \vee \mathbf{0})$ and $\langle\rangle \vdash \mathbf{0} : (\neg A \vee 0)$, or $\langle\rangle \vdash \mathbf{0} : (A \vee \mathbf{0})$ and $\langle\rangle \vdash x : (\neg A \vee 0)$. In either case, by soundness we can verify that neither $\langle\rangle \vdash x : (\neg A \vee 0)$ nor $\langle\rangle \vdash x : (A \vee \mathbf{0})$ can be derivable in general.

6.2 Freshness

We show a derivation of the fact that $\neg\text{И}x.A$ entails $\text{И}x.\neg A$. This (and its converse) is a well-known property of $\text{И}x.A$ [14]; the purpose here is to show the use of the rules for freshness in a simple case.

6. $\langle S, y \# \text{И}x.A, u \doteq (\boldsymbol{\nu} y)x\rangle\ \Gamma, u : A\{x{\leftarrow}y\} \vdash u : A\{x{\leftarrow}y\}, \Delta$

(by (Id) choose y, x fresh)

5. $\langle S, y \# \text{И}x.A, u \doteq (\boldsymbol{\nu} y)x\rangle\ \Gamma \vdash u : A\{x{\leftarrow}y\}, u : \neg A\{x{\leftarrow}y\}, \Delta$ (by 6, ($\neg$ R))

4. $\langle S, y \# \text{И}x.A, u \doteq (\boldsymbol{\nu} y)x\rangle\ \Gamma \vdash u : \text{И}x.A, u : \neg A\{x{\leftarrow}y\}, \Delta$ (by 5, (ИR))

3. $\langle S, y \# \text{И}x.A, u \doteq (\boldsymbol{\nu} y)x\rangle\ \Gamma \vdash u : \text{И}x.A, u : \text{И}x.\neg A, \Delta$ (by 4, (ИR))

2. $\langle S, y \# \text{И}x.A, u \doteq (\boldsymbol{\nu} y)x\rangle\ \Gamma, u : \neg\text{И}x.A \vdash u : \text{И}x.\neg A, \Delta$ (by 3, ($\neg$ L))

1. $\langle S\rangle\ \Gamma, u : \neg\text{И}x.A \vdash u : \text{И}x.\neg A, \Delta$ (by 2, (И) y, x not in conclusion)

We start with $A\{x{\leftarrow}y\}$ for a fresh y, instead of simply with A, so that we can apply (И) in the last step even when x occurs free in Γ, Δ. It is usually the case that an application of rules (И L) or (И R) is followed by an application of rule (И), to clean up the constraints. Note, however, that having (И) decoupled from (ИL) and (ИR) allow us to apply, in this case, (ИR) twice before applying (И).

Along similar lines, we can derive interesting properties combining $\text{И}x.A$ with spatial operators, for example the following one, which is important for deriving properties of the hiding quantifier (it takes about eight steps in each direction, but with a rather more involved set of constraints):

$$\langle S\rangle\ \Gamma, u : (\text{И}x.A)|(\text{И}x.B) \dashv\vdash u : \text{И}x.(A|B), \Delta$$

This derivation uses the world rule (Sν|), which embeds a rather deep lemma about π-calculus structural congruence; namely, that if $(\boldsymbol{\nu} n)P \equiv Q|R$ then there exist P', Q' such that $P \equiv P'|P''$ and $(\boldsymbol{\nu} n)P' \equiv Q$ and $(\boldsymbol{\nu} n)P'' \equiv R$.

6.3 Equivariance

In general terms, we have that an Aπ process P satisfies the formula $\mathsf{n}\circledR A$ if $P \equiv (\boldsymbol{\nu} n)Q$, where Q is a process that satisfies A, and n is the name denoted by n. Then n denotes a name which is hidden, and hence not free, in P. Therefore, the revelation operator has a useful meaning also in the special case $\mathsf{n}\circledR\mathbf{T}$: the process P satisfies $\mathsf{n}\circledR\mathbf{T}$ if and only if the name denoted by n is fresh in P (In Section 3.6 we introduced $\copyright\mathsf{n}$ as an abbreviation for $\neg\mathsf{n}\circledR\mathbf{T}$). We can show than $A \wedge \mathsf{m}\circledR\mathbf{T} \wedge \mathsf{n}\circledR\mathbf{T}$ entails $(\mathsf{n}{\leftrightarrow}\mathsf{m})A$:

3. $\langle z \doteq (\boldsymbol{\nu}\mathsf{n})x, z \doteq (\boldsymbol{\nu}\mathsf{m})y\rangle\ (\mathsf{n}{\leftrightarrow}\mathsf{m})z : (\mathsf{n}{\leftrightarrow}\mathsf{m})A, x : \mathbf{T}, y : \mathbf{T} \vdash z : (\mathsf{n}{\leftrightarrow}\mathsf{m})A$

2. $\langle z \doteq (\boldsymbol{\nu}\mathsf{n})x, z \doteq (\boldsymbol{\nu}\mathsf{m})y\rangle\ z : A, x : \mathbf{T}, y : \mathbf{T} \vdash z : (\mathsf{n}{\leftrightarrow}\mathsf{m})A$ (by 3, (TL))

1. $\langle\rangle\ z : A \wedge \mathsf{m}\circledR\mathbf{T} \wedge \mathsf{n}\circledR\mathbf{T} \vdash z : (\mathsf{n}{\leftrightarrow}\mathsf{m})A$ (by 2, ($\wedge$L and $\circledR$L))

(Note the use of (Swap Erase) in step 3, proved by (Id), to show $(n\leftrightarrow m)z \doteq z$ w.r.t. the constraint part of the sequent) This property can be interpreted as saying that, for any process P, if it satisfies A, it also satisfies $(n \leftrightarrow m)A$ for any fresh names m and n. This fact is a consequence of the equivariance property of the semantics: intuitively, if the name denoted by (say) m occurs in the formula A but not in the process P, then we would expect the name m to be irrelevant to the fact that P satisfies A. This means that if we swap in formula A the name m by any other fresh name n, we would expect that P would still satisfy it (since a fresh name is as good as any other fresh name). For example, the following provable sequent

$$\langle n\# p, m\# p\rangle\, x : n\circledR(p\langle n\rangle|\mathbf{T}) \wedge m\circledR\,\mathbf{T} \wedge n\circledR\,\mathbf{T} \vdash x : m\circledR(p\langle m\rangle|\mathbf{T})$$

says that if a process is about to send a fresh name on a public channel p, it can send any other fresh name as well.

6.4 Input

In our logic we have a primitive formula to observe messages, $n\langle m\rangle$, corresponding to the output operator of the asynchronous π-calculus. We do not have a corresponding input formula, but it can be expressed from output along the lines of [20]. The guarantee operator is crucial to this; recall that a process P satisfies $A \triangleright B$ if for any Q that satisfies A, we have that $P|Q$ satisfies B (this can be read out from ($\triangleright$R)). We say that P satisfies B "in presence" of any Q that satisfies A. We can take the following definition of input:

$$x(y).A \triangleq \forall y.x\langle y\rangle \triangleright \Diamond A$$

The intention is that a process satisfies the input specification $x(y).A$ if it performs an input over a given channel x of any name y (with y bound in A), and then satisfies the property A. The above definition says literally, that an input process is one that, in presence of any output message y over the given channel x, at the next step (after input) it behaves according to A.

It is then easy to verify that because of the adjunction between $|$ and $\triangleright$, input and output interact as expected in π-calculus communication, that is, $x\langle z\rangle|x(y).A$ entails $\Diamond A\{y\leftarrow z\}$:

4.2. $\langle S, u = \mathscr{x}|\mathscr{y}\rangle\, \Gamma, \mathscr{x} : x\langle z\rangle \vdash \mathscr{x} : x\langle z\rangle, u : \Diamond A\{y\leftarrow z\}, \Delta$ (by (Id))

4.1. $\langle S, u = \mathscr{x}|\mathscr{y}\rangle\, \Gamma, \mathscr{x} : x\langle z\rangle, \mathscr{x}|\mathscr{y} : \Diamond A\{y\leftarrow z\} \vdash u : \Diamond A\{y\leftarrow z\}, \Delta$ (by (Id))

3. $\langle S, u = \mathscr{x}|\mathscr{y}\rangle\, \Gamma, \mathscr{x} : x\langle z\rangle, \mathscr{y} : x\langle z\rangle \triangleright \Diamond A\{y\leftarrow z\} \vdash u : \Diamond A\{y\leftarrow z\}, \Delta$ (by 4.1-2, ($\triangleright$L))

2. $\langle S, u = \mathscr{x}|\mathscr{y}\rangle\, \Gamma, \mathscr{x} : x\langle z\rangle, \mathscr{y} : \forall y.x\langle y\rangle \triangleright \Diamond A \vdash u : \Diamond A\{y\leftarrow z\}, \Delta$ (by 3, ($\forall$L))

1. $\langle S\rangle\, \Gamma, u : x\langle z\rangle|(\forall y.x\langle y\rangle \triangleright \Diamond A) \vdash u : \Diamond A\{y\leftarrow z\}, \Delta$ (by 2, (|L))

So we have that the following sequent is derivable:

$$\langle S \rangle\, \Gamma, u : x\langle z\rangle | x(y).A \vdash u : \Diamond A\{y \leftarrow z\}, \Delta \quad \text{(I/O)}$$

6.5 *Hiding*

In Part I and Section 3.6 we defined a hiding quantifier: $\mathsf{H}x.A \triangleq \text{И}x.x \circledR A$ which is related to π-calculus name restriction in an appropriate way; namely, that if process P satisfies formula $A\{x \leftarrow \boldsymbol{n}\}$, then $(\boldsymbol{\nu} n)P$ satisfies $\mathsf{H}x.A$, where n is a (fresh) name denoted by $\boldsymbol{n}$. An interesting use of $\mathsf{H}x.A$ is in specifying "nonce generators", that is processes that output freshly generated names on a given channel. In π-calculus, a nonce generator can be written simply as $(\boldsymbol{\nu} n)nc\langle n\rangle$, for a given channel nc. A nonce generator over nc can then be specified by the following formula:

$$\mathcal{N}_c \triangleq \mathsf{H}x.nc\langle x\rangle$$

We can show that, when a nonce generator interacts with an input, the result is the acquisition of a private name:

$$\langle S \rangle\, \Gamma, u : \mathcal{N}_c | nc(y).A \vdash u : \Diamond \mathsf{H}z.A\{y \leftarrow z\}, \Delta \quad \text{(BI/O)}$$

Before input we have a nonce generator $\mathcal{N}_c$ separate from the input process. After one step, we have that the A part has acquired a name z; but noticeably this z is "hidden" within $A\{y \leftarrow z\}$ by the scope of the hiding quantifier. Hence the A part of the system has acquired, from the nonce generator, a private name not shared with other parts of the system (at least, not yet).

6.6 *Recursive Properties*

We show a couple of derivations involving recursive formulas and freshness. As a first example, consider the following formulas

$$\mathit{Writer} \triangleq \boldsymbol{\nu} X.(x\langle y\rangle | X) \quad \mathit{Reader} \triangleq \boldsymbol{\nu} Y.(x(y).Y) \quad \mathit{LiveLock} \triangleq \boldsymbol{\nu} Z.\Diamond Z$$

Thus, a process that satisfies *Writer* is able to send an unbounded number of messages $x\langle y\rangle$. Likewise, a process that satisfies *Reader* has continuously enabled the capability of consuming the message $x\langle y\rangle$. We can prove that the composition of *Writer* and *Reader* has a non-terminating computation path: this fact can be expressed by the sequent

$$\langle\rangle\; x : \mathit{Reader} | \mathit{Writer} \vdash x : \mathit{Livelock}$$

We abbreviate $B \triangleq x\langle y\rangle | \mathit{Writer} | x(y).\mathit{Reader}$, so that the formula B is the one step unfolding of the formula $\mathit{Reader}|\mathit{Writer}$. Let also M be the sequent context expressing the monotonicity assumptions (see Section 4) for the recursive formulas *Reader* and *Writer* in the example (the proof of M is also rather mechanical): $M \triangleq (x\langle y\rangle|X)\{X^+\}, (x(y).Y)\{Y^+\}$. We can then use a standard coinductive argument to show the statement:

$$\begin{array}{lr}
4.\ \langle\rangle\ M, \mathcal{x} : \mathit{Reader}|\ \mathit{Writer}, \mathcal{y} : B \vdash \mathcal{y} : \Diamond \mathit{Reader}|\ \mathit{Writer} & \text{(by I/O)} \\
3.\ \langle\rangle\ M, \mathcal{x} : \mathit{Reader}|\ \mathit{Writer}, \mathcal{y} : B \vdash \mathcal{y} : \Diamond B & \text{(by 4, (Unfold))} \\
2.\ \langle\rangle\ M, \mathcal{x} : \mathit{Reader}|\ \mathit{Writer} \vdash \mathcal{x} : B & \text{(by (Unfold), (Id))} \\
1.\ \langle\rangle\ M, \mathcal{x} : \mathit{Reader}|\ \mathit{Writer} \vdash \mathcal{x} : \mathit{Livelock} & \text{(by 2, 3 (νR))} \\
1.\ \langle\rangle\ \mathcal{x} : \mathit{Reader}|\ \mathit{Writer} \vdash \mathcal{x} : \mathit{Livelock} & \text{(by (Cut), with $\langle\rangle \vdash M$)}
\end{array}$$

As a second example of the use of recursion, extending the one in Section 6.5, we specify a recursive nonce generator (a process producing an unbounded number of fresh names) by follows: $\mathcal{U}\mathcal{N}_c \triangleq \boldsymbol{\nu} X.\mathcal{N}_c|X$. As in our last example, we can then show

$$\mathcal{x} : \mathcal{U}\mathcal{N}_c|\mathcal{U}\mathcal{N}_c \vdash \mathcal{x} : \mathcal{U}\mathcal{N}_c$$

This is simple but significant: it means that (without any knowledge of the π-calculus implementation) two recursive nonce generators running in parallel behave like a single recursive nonce generator; in particular, the two generators do not risk generating independently the same name twice.

7 Conclusion

We have presented a sequent calculus that has a direct interpretation in terms of distributed concurrent behaviors, including notions of resource hiding. We believe we have obtained a unique combination of, on one hand, good proof-theoretical structures and properties, and, on the other hand, direct applicability to concurrency. These twin aims have driven us towards a "many worlds" formulation of modal sequents that has been able to accommodate a wide range of unusual but strongly motivated logical constructions.

Acknowledgements We thank Alex Simpson for useful comments on an early version of this paper. The anonymous referees are also acknowledged. Caires acknowledges Profundis FET IST-2001-33100 and Microsoft Research for a visit to the Cambridge Laboratory during the time we worked on this paper.

References

[1] L. Caires. *A Model for Declarative Programming and Specification with Concurrency and Mobility.* PhD thesis, Dept. de Informática, FCT, Universidade Nova de Lisboa, 1999.

[2] L. Caires and L. Cardelli. A Spatial Logic for Concurrency (Part I). In N. Kobayashi and B.C. Pierce, editors, *10th Symposium on Theoretical Aspects of Computer Science*, volume 2215 of *Lecture Notes in Computer Science*, pages 1–30. Springer-Verlag, 2001.

[3] L. Caires and L. Cardelli. A Spatial Logic for Concurrency (Part II). In *CONCUR 2002 (13th International Conference)*, Lecture Notes in Computer Science. Springer-Verlag, 2002.

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

[5] L. Caires and L. Monteiro. Verifiable and Executable Specifications of Concurrent Objects in $\mathcal{L}_\pi$. In C. Hankin, editor, *7th European Symp. on Programming (ESOP 1998)*, number 1381 in Lecture Notes in Computer Science, pages 42–56. Springer-Verlag, 1998.

[6] C. Calcagno, L. Cardelli, and A. Gordon. Deciding Validity in a Spatial Logic of Trees. In *ACM Workshop on Types in Language Design and Implementation*, pages 62–73, New Orleans, USA, 2003. ACM Press.

[7] L. Cardelli, P. Gardner, and G. Ghelli. A spatial logic for querying graphs. In *29td Colloquium on Automata, Languages and Programming (ICALP 2002)*, Lecture Notes in Computer Science, pages 597–610. Springer-Verlag, 2002.

[8] L. Cardelli, P. Gardner, and G. Ghelli. Manipulating Trees with Hidden Labels. In A. D. Gordon, editor, *Proceedings of the First International Conference on Foundations of Software Science and Computation Structures (FoSSaCS '03)*, Lecture Notes in Computer Science, pages 216–232. Springer-Verlag, 2003.

[9] L. Cardelli and G. Ghelli. A Query Language Based on the Ambient Logic. In D. Sands, editor, *10th European Symp. on Programming (ESOP 2001)*, volume 2028 of *Lecture Notes in Computer Science*, pages 1–22. Springer-Verlag, 2001.

[10] L. Cardelli and A. D. Gordon. Anytime, Anywhere. Modal Logics for Mobile Ambients. In *27th ACM Symp. on Principles of Programming Languages*, pages 365–377. ACM, 2000.

[11] 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, pages 46–60. Springer-Verlag, 2001.

[12] 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.

[13] M. Dam. Proof systems for π-calculus logics. In de Queiroz, editor, *Logic for Concurrency and Synchronisation*, Studies in Logic and Computation. Oxford University Press, 2003.

[14] M. Gabbay and A. 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.

[15] M. Hennessy and R. Milner. Algebraic laws for Nondeterminism and Concurrency. *JACM*, 32(1):137–161, 1985.

[16] D. Hirschkoff, E. Lozes, and D. Sangiorgi. Separability, Expressiveness and Decidability in the Ambient Logic. In *Third Annual Symp. on Logic in Computer Science*, pages 423–432, Copenhagen, Denmark, 2002. IEEE Press.

[17] P. O'Hearn and D. Pym. The Logic of Bunched Implications. *The Bulletin of Symbolic Logic*, 5(2):215–243, 1999.

[18] A. Pitts. Nominal Logic: A First Order Theory of Names and Binding. In B.C. Pierce N. Kobayashi, editor, *10th Symposium on Theoretical Aspects of Computer Science (TACS 2001)*, volume 2215 of *Lecture Notes in Computer Science*, pages 219–235. Springer-Verlag, 2001.

[19] J. C. Reynolds. Separation Logic: A Logic for Shared Mutable Data Structures. In *Third Annual Symp. on Logic in Computer Science*, pages 55–74, Copenhagen, Denmark, 2002. IEEE Press.

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

[21] A. Simpson. *The Proof Theory and Semantics of Intuitionistic Modal Logic.* Ph.D. thesis, Dept. of Computer Science, Edinburgh University, 1994.

[22] A. Simpson. Compositionality via cut-elimination: Hennessy-Milner logic for an arbitrary GSOS. In *Proceedings of the 10th IEEE Symposium on Logic in Computer Science*, pages 420–430. IEEE Press, 1995.

[23] A. Troelstra and H. Schwichtenberg. *Basic Proof Theory.* Cambridge University Press, 2000.

[24] Luca Viganò. *Labelled Non-Classical Logics.* Kluwer Academic Publishers, Dordrecht, 2000.

8 Appendix

We introduce some auxiliary notation, useful for the proofs of the next two Lemmas. If $\vec{x}$ is a list of distinct name variables and $\vec{n}$ is a list of terms, we use the notation $\{\vec{x}\leftarrow\vec{n}\}$ to denote the substitution that assigns n_i to x_i, and the notation $\{\vec{x}\leftarrow(p\leftrightarrow q)\vec{n}\}$ to denote the substitution that assigns $(p\leftrightarrow q)n_i$ to x_i. We also assume then that no name variable in $\vec{x}$ occurs in p, q or n.

Given a name term m and a list of distinct name variables $\vec{x}$, we write $S(m, \vec{x})$ for the set of all maximal subterms of the name term m that do not contain occurrences of name variables x in the list $\vec{x}$. In a similar way, given a formula A and a list of distinct (name or propositional variables) $\vec{x}$, we write $S(A, \vec{x})$ for the set of all logically free terms in formula A that do not contain occurrences of some name or (propositional) variable x in the list $\vec{x}$. More precisely: $S(A, \vec{x}) \triangleq \{n \mid n \in \mathit{lft}(A) \text{ and } \mathit{fv}(n) \cap \vec{x} = \emptyset\}$ N.B. $S(A, \emptyset) = \mathit{lft}(A)$.

Lemma 8.1 *Let p, q, m be name terms such that $p, q \#_S S(A, \vec{x})$, where $\vec{x}$ is a list of distinct name variables, and $\vec{n}$ is a matching list of name terms (for $\vec{x}$). Then $(p \leftrightarrow q)(m\{\vec{x} \leftarrow \vec{n}\}) \doteq_S m\{\vec{x} \leftarrow (p \leftrightarrow q)\vec{n}\}$.*

Proof. Induction on the structure of the name term m. ■

Lemma 8.2 *Let p, q be name terms and A a normalized formula such that $p, q \#_S S(A, \vec{x})$, where $\vec{x}$ is a list of distinct variables, and $\vec{n}$ is a matching list of name and propositional terms. Then $(p \leftrightarrow q)(A\{\vec{x} \leftarrow \vec{n}\}) \Downarrow_S A\{\vec{x} \leftarrow (p \overset{\rightarrow}{\leftrightarrow} q)n\}$.*

Proof. Induction on the structure of the formula A. The result is in all cases a direct consequence of the induction hypothesis; in the case of formulas mentioning name terms, the result follows from Lemma 8.1. We detail two cases.

(Case of $A = m \circledR B$**)** We must have $(p \leftrightarrow q)(A\{\vec{x} \leftarrow \vec{n}\}) \Downarrow_S m' \circledR B'$ where $(p \leftrightarrow q)(B\{\vec{x} \leftarrow \vec{n}\}) \Downarrow_S B'$ and $m' \doteq_S (p \leftrightarrow q)(m\{\vec{x} \leftarrow \vec{n}\})$. Note that we must have $p, q \#_S S(m, \vec{x})$. Therefore, by Lemma 8.1, we conclude $m' \doteq_S m\{\vec{x} \leftarrow (p \leftrightarrow q)\vec{n}\}$. By induction hypothesis, we conclude $(p \leftrightarrow q)(B\{\vec{x} \leftarrow \vec{n}\}) \Downarrow_S B\{\vec{x} \leftarrow (p \leftrightarrow q)\vec{n}\}$. Hence $(p \leftrightarrow q)(A\{\vec{x} \leftarrow \vec{n}\}) \Downarrow_S m\{\vec{x} \leftarrow (p \leftrightarrow q)\vec{n}\} \circledR (B\{\vec{x} \leftarrow (p \leftrightarrow q)\vec{n}\}) = A\{\vec{x} \leftarrow (p \leftrightarrow q)\vec{n}\}$.

(Case of $A = \text{И} z.B$**)** We have $(p \leftrightarrow q)A\{\vec{x} \leftarrow \vec{n}\} \Downarrow_S \text{И} z.B'$ where $(p \leftrightarrow q)B\{\vec{x}, z \leftarrow \vec{n}, (p \leftrightarrow q)z\} \Downarrow_S B'$. We can the apply the induction hypothesis (note that $S(\text{И} z.B, \vec{x}) = S(B, \vec{x} \cup \{z\})$, and conclude $(p \leftrightarrow q)B\{\vec{x}, z \leftarrow \vec{n}, (p \leftrightarrow q)z\} \Downarrow_S B\{\vec{x}, z \leftarrow (p \leftrightarrow q)\vec{n}, z\}$. Hence $(p \leftrightarrow q)A\{\vec{x} \leftarrow \vec{n}\} \Downarrow_S \text{И} z.(B\{\vec{x} \leftarrow (p \leftrightarrow q)\vec{n}\}) = A\{\vec{x} \leftarrow (p \leftrightarrow q)\vec{n}\}$. ■

Theorem 3.22 [Soundness] *All sequents derivable in* **S** *are valid in* $\mathrm{A}\pi$.

Proof. We show that all inference rules are sound. An inference rule is *sound* if the sequent in the conclusion is valid provided all the sequents and assertions occurring as premises are valid (see Definitions 3.12 and 3.16). Cases of (Id), (Cut), (**F**L), (**F**R), ($\wedge$L), ($\wedge$R), ($\Rightarrow$L) and ($\Rightarrow$R) are standard.

- **(Case of (TL))** Let $\mathcal{J}$ be an interpretation for the sequent $\langle S \rangle\, \Gamma, u : A \vdash \Delta$ such that $\mathcal{J}$ satisfies S and all of $\Gamma, u : A$. Then $\mathcal{J}$ satisfies all of Γ and $\mathcal{J}(u) \in [\![A]\!]_{\mathcal{J}}$. Therefore, $\{[\![n]\!]_{\mathcal{J}} \leftrightarrow [\![m]\!]_{\mathcal{J}}\}\mathcal{J}(u) \in \{[\![n]\!]_{\mathcal{J}} \leftrightarrow [\![m]\!]_{\mathcal{J}}\}[\![A]\!]_{\mathcal{J}} = [\![(n \leftrightarrow m)A]\!]_{\mathcal{J}}$.

  Since $(m \leftrightarrow n)A \equiv_S A'$ and $(m \leftrightarrow n)u \doteq_S u'$, by Lemma 3.21(1) and Lemma 3.13(2) we have $[\![(m \leftrightarrow n)A]\!]_{\mathcal{J}} = [\![A']\!]_{\mathcal{J}}$ and $\{[\![n]\!]_{\mathcal{J}} \leftrightarrow [\![m]\!]_{\mathcal{J}}\}\mathcal{J}(u) \equiv \mathcal{J}(u')$. We conclude $\mathcal{J}(u') \in [\![A']\!]_{\mathcal{J}}$.
- **(Case of (TR))** Similar to (TL).

- (**Case of (S$\boldsymbol{\nu}$|)**) Let $\mathcal{J}$ be an interpretation for the sequent $\langle S\rangle\, \Gamma \vdash \Delta$ in the conclusion such that $\mathcal{J}$ satisfies S and $\mathcal{J}$ satisfies all of Γ. In particular, we have $(\boldsymbol{\nu}\mathcal{J}(x))\mathcal{J}(u) \equiv \mathcal{J}(t)|\mathcal{J}(v)$. By Proposition (Part I)2.13(2) [4], there are processes P and Q such that $t \equiv (\boldsymbol{\nu}\mathcal{J}(x))P$, $v \equiv (\boldsymbol{\nu}\mathcal{J}(x))Q$, and $\mathcal{J}(u) \equiv P|Q$. Let $\mathcal{J}' \triangleq \mathcal{J}\{x \leftarrow P\}\{y \leftarrow Q\}$.

  $\mathcal{J}'$ satisfies $\langle S, u \doteq x\,|y, (\boldsymbol{\nu}x)x \doteq t, (\boldsymbol{\nu}x)y \doteq v\rangle$. Since x and y do not occur in Γ and Δ, we have that $\mathcal{J}'$ satisfies all of Γ, hence by validity of the premises it also satisfies some of Δ. So $\mathcal{J}$ satisfies some of Δ.
- **(Case of other (S$-$) rules)**. Like with (S$\boldsymbol{\nu}$|) above, soundness is a consequence of the inversion properties of Proposition (Part I)2.13 [4].
- **(Case of (0R))** Let $\mathcal{J}$ be an interpretation for the sequent $\langle S\rangle\, \Gamma \vdash \Delta$ in the conclusion, and assume that $\mathcal{J}$ satisfies S. Hence $\mathcal{J}(u) \equiv \mathbf{0}$, thus $\mathcal{J}(u) \in [\![\mathbf{0}]\!]_v$.
- **(Case of (0L))** Let $\mathcal{J}$ be an interpretation for $\langle S\rangle\, \Gamma \vdash \Delta$, and assume that $\mathcal{J}$ satisfies all of $\Gamma, u : \mathbf{0}$. Hence $\mathcal{J}(u) \equiv \mathbf{0}$, and $\mathcal{J}$ satisfies $\langle S, u \doteq \mathbf{0}\rangle$. By validity of the premise, $\mathcal{J}$ satisfies some of Δ.
- **(Case of (|R))** Let $\mathcal{J}$ be an interpretation for the sequent $\langle S\rangle\, \Gamma \vdash \Delta$ in the conclusion, and assume that $\mathcal{J}$ satisfies S and $\mathcal{J}$ satisfies all of Γ. By assumption, $\mathcal{J}(u) \equiv \mathcal{J}(v)|\mathcal{J}(t)$. If $\mathcal{J}$ satisfies some of Δ, we have the conclusion. Otherwise, by validity of the premises, we must have $\mathcal{J}(v) \in [\![A]\!]_{\mathcal{J}}$ and $\mathcal{J}(t) \in [\![B]\!]_{\mathcal{J}}$. From that, we conclude $\mathcal{J}(u) \in [\![A|B]\!]_{\mathcal{J}}$.
- **(Case of (|L))** Let $\mathcal{J}$ be an interpretation for the sequent in the conclusion, and assume that $\mathcal{J}$ satisfies S and $\mathcal{J}$ satisfies all of $\Gamma, u : A|B$. Thus, there are P and Q such that $\mathcal{J}(u) \equiv P|Q$, $P \in [\![A]\!]_{\mathcal{J}}$ and $Q \in [\![B]\!]_{\mathcal{J}}$. Let $\mathcal{J}' \triangleq \mathcal{J}\{x \leftarrow P\}\{y \leftarrow Q\}$: then $\mathcal{J}'$ satisfies $\langle S, u \doteq x\,|y\rangle$ and $\mathcal{J}'$ satisfies all of $\Gamma, x : A, y : B$. To conclude, note that by assumption $\mathcal{J}'$ satisfies some of Δ, and that $\mathcal{J}'$ agrees with $\mathcal{J}$ on Δ.
- **(Case of ($\triangleright$R))** Let $\mathcal{J}$ be an interpretation for the sequent in the conclusion, and assume that $\mathcal{J}$ satisfies S and $\mathcal{J}$ satisfies all of Γ. Pick any process $P \in [\![A]\!]_{\mathcal{J}}$. Since x does not occur in the conclusion, the interpretation $\mathcal{J}^P \triangleq \mathcal{J}\{x \leftarrow P\})$ also satisfies S and all of $\Gamma, x : A$. By assumption, $\mathcal{J}^P(v) \in [\![B]\!]_{\mathcal{J}^P} = [\![B]\!]_{\mathcal{J}}$. But $\mathcal{J}^P(v) \equiv P|\mathcal{J}^P(u) = P|\mathcal{J}(u)$. Hence $P|\mathcal{J}(u) \in [\![B]\!]_{\mathcal{J}}$, for all processes $P \in [\![A]\!]_{\mathcal{J}}$. We conclude $\mathcal{J}(u) \in [\![A \triangleright B]\!]_{\mathcal{J}}$.
- **(Case of ($\triangleright$L))** Let $\mathcal{J}$ be an interpretation for the sequent in the conclusion, and assume that $\mathcal{J}$ satisfies S and $\mathcal{J}$ satisfies all of $\Gamma, u : A \triangleright B$. Thus, for all processes P such that $P \in [\![A]\!]_{\mathcal{J}}$ we have that $P|\mathcal{J}(u) \in [\![B]\!]_{\mathcal{J}}$. Since $\mathcal{J}$ satisfies all of Γ, by validity of the left premise either $\mathcal{J}$ satisfies $t : A$ or $\mathcal{J}$ satisfies some of Δ. In the latter case, we can conclude. Otherwise, $\mathcal{J}(t) \in [\![A]\!]_{\mathcal{J}}$. Then $\mathcal{J}(t|u) \in [\![B]\!]_{\mathcal{J}}$, hence $\mathcal{J}$ satisfies all of $\Gamma, t|u : B$. By validity of the right premise, we also conclude that $\mathcal{J}$ satisfies some of Δ.
- **(Case of ($\Diamond$R))** and **(Case of ($\Diamond$L))** By Lemma 3.13(2).
- **(Cases of ($\circledR$L), ($\circledR$R), ($\oslash$L), and ($\oslash$R))** Like (|R), (|L), ($\triangleright$L) and ($\triangleright$R).
- **(Case of ($\forall$R))** Let $\mathcal{J}$ be an interpretation for the sequent $\langle S\rangle\, \Gamma \vdash u : \forall x.A, \Delta$ in the conclusion such that $\mathcal{J}$ satisfies S and $\mathcal{J}$ satisfies all of Γ. Pick any name $n \in \Lambda$ and define $\mathcal{J}^n \triangleq \mathcal{J}\{x \leftarrow n\}$; $\mathcal{J}^n$ is then an interpretation

for the sequent $\langle S\rangle\, \Gamma \vdash u : A, \Delta$ in the premise. Note that for all names n, $\mathcal{J}^n$ satisfies S and $\mathcal{J}^n$ satisfies all of Γ, since x does not occur free in the conclusion of the rule. Hence, by validity of the premise, $\mathcal{J}^n$ satisfies some of $u : A, \Delta$, for all n. Now, suppose there is an interpretation $\mathcal{J}^n$ that satisfies some of Δ. Then also $\mathcal{J}$ satisfies some of Δ since x is not free in Δ, and we have the conclusion. Otherwise, we must have $\mathcal{J}^n(u) \in [\![A]\!]_{\mathcal{J}^n}$ for all names n. But then, $\mathcal{J}(u) \in [\![\forall x.A]\!]_{\mathcal{J}}$.

- **(Case of ($\forall$L))** Let $\mathcal{J}$ be an interpretation for the sequent $\langle S\rangle\, \Gamma, u : \forall x.A \vdash \Delta$ in the conclusion such that $\mathcal{J}$ satisfies S, $\mathcal{J}$ satisfies all of $\Gamma, u : \forall x.A$. Hence, we have $\mathcal{J}(u) \in [\![A]\!]_{\mathcal{J}[x\leftarrow p]}$ for all names p, in particular for $n = [\![\mathsf{m}]\!]_{\mathcal{J}}$. Hence, we conclude $\mathcal{J}(u) \in [\![A\{x\leftarrow \mathsf{m}\}]\!]_{\mathcal{J}}$. By validity of the sequent in the premise, we conclude that $\mathcal{J}$ satisfies some of Δ.
- **(Cases of ($\forall^2$R) and ($\forall^2$L))** The proof is similar to ($\forall$L) and ($\forall$R) above.
- **(Case of (И))** Let $\mathcal{J}$ be an interpretation for the sequent $\langle S\rangle\, \Gamma \vdash \Delta$ such that $\mathcal{J}$ satisfies S and $\mathcal{J}$ satisfies all of Γ. Now, let $P = \mathcal{J}(u)$ and pick any name $n \notin \mathit{fn}(P)$ such that $n \neq \mathcal{J}(y)$ for all $y \in \mathit{fv}(N)$ and $n \notin \mathit{supp}(\mathcal{J}(X))$, for all $X \in \mathit{fpv}(N)$.

  When then have $\mathcal{J}(u) \equiv (\boldsymbol{\nu} n)\mathcal{J}(u)$. Define $\mathcal{J}' \triangleq \mathcal{J}\{x\leftarrow n\}\{x\leftarrow P\}$. Hence $\mathcal{J}'$ is an interpretation for the sequent in the premise, where $\mathcal{J}'$ satisfies $\langle S, u \doteq (\boldsymbol{\nu} x)x, x \,\#\, N\rangle$ and $\mathcal{J}'$ satisfies all of Γ (since x and x are fresh). By validity of such sequent, we conclude that $\mathcal{J}'$ satisfies some of Δ. Since $\mathcal{J}'$ agrees with $\mathcal{J}$ on Δ, we conclude that $\mathcal{J}$ satisfies some of Δ.
- **(Case of (ИR))** Let $\mathcal{J}$ be an interpretation for the sequent $\langle S\rangle\, \Gamma \vdash u : \text{И}x.A, \Delta$ such that $\mathcal{J}$ satisfies S and $\mathcal{J}$ satisfies all of Γ. By validity of the premise, $\mathcal{J}$ satisfies some of $u : A\{x\leftarrow \mathsf{n}\}, \Delta$. If $\mathcal{J}$ satisfies some of Δ the proof is concluded. Otherwise, $\mathcal{J}(u) \in [\![A\{x\leftarrow \mathsf{n}\}]\!]_{\mathcal{J}} = [\![A]\!]_{\mathcal{J}[x\leftarrow [\![\mathsf{n}]\!]_{\mathcal{J}}]}$. By assumption, the assertion $u \doteq_S (\boldsymbol{\nu} \mathsf{n})v$ is valid, hence $\mathcal{J}(u) \equiv (\boldsymbol{\nu} \mathcal{J}(\mathsf{n}))\mathcal{J}(v)$. So, $[\![\mathsf{n}]\!]_{\mathcal{J}} \notin \mathit{fn}(\mathcal{J}(u))$. Moreover, since $\mathsf{n} \#_S \text{И}x.A$, by Lemma 3.21(2) we have $[\![\mathsf{n}]\!]_{\mathcal{J}} \notin \mathit{fn}^{\mathcal{J}}(\text{И}x.A)$. Since $\mathcal{J}(u) \in [\![A]\!]_{\mathcal{J}[x\leftarrow [\![\mathsf{n}]\!]_{\mathcal{J}}]}$, $\mathcal{J}(u) \in [\![\text{И}x.A]\!]_v$.
- **(Case of (ИL))** Let $\mathcal{J}$ be an interpretation for the sequent $\langle S\rangle\, \Gamma, u : \text{И}x.A \vdash \Delta$ such that $\mathcal{J}$ satisfies S and $\mathcal{J}$ satisfies all of $\Gamma, u : \text{И}x.A$. In particular, we have $\mathcal{J}(u) \in [\![A]\!]_{\mathcal{J}[x\leftarrow n]}$ for some $n \notin \mathit{fn}(\mathcal{J}(u)) \cup \mathit{fn}^v(\text{И}x.A)$. Thus, by Theorem 2.3(3), for all names $p \in \Lambda$ such that $p \notin \mathit{fn}(\mathcal{J}(u)) \cup \mathit{fn}^v(\text{И}x.A)$ we have $\mathcal{J}(u) \in [\![A]\!]_{\mathcal{J}[x\leftarrow p]}$. Like in the case above for (ИR), we can verify that $[\![\mathsf{n}]\!]_{\mathcal{J}} \notin \mathit{fn}(\mathcal{J}(u)) \cup \mathit{fn}^v(\text{И}x.A)$, so that n denotes a possible freshness witness. Hence, we have $\mathcal{J}(u) \in [\![A]\!]_{\mathcal{J}[x\leftarrow [\![\mathsf{n}]\!]_{\mathcal{J}}]} = [\![A\{x\leftarrow \mathsf{n}\}]\!]_{\mathcal{J}}$. Since the premise of the rule is valid by assumption, $\mathcal{J}$ satisfies some of Δ. ■

Lemma 5.3 (Basic) The size-preserving proof principles (CS), (Ren), (W), (In$\mathcal{I}$) and (In$\mathcal{N}$) are admissible.

Proof. **(CS)** By induction on the structure of derivations, using Lemma 3.10(2).

(Ren) and (α) By simultaneous induction on the structure of of derivations.

(W) By induction on the structure of derivations, using Lemma 3.10(1) to show that provability of constraint premises is preserved.

(In$\mathcal{I}$) For clarity, we abbreviate the substitution $\{x{\leftarrow}u\}$ by σ. Proof by induction on the structure of derivations, using Lemma 3.10(3) to show that $u \doteq_S v$ implies $\sigma(u) \doteq_{\sigma(S)} \sigma(v)$ in all rule instances with assertions $u \doteq_S v$ as premises, and likewise for premises of the form $u \rightarrow_S v$. The most interesting cases are the ones which introduce process eigenvariables, *e.g.*,

- **(Case of (|L))** $\langle S\rangle\, \Gamma, u : A|B \vdash \Delta$ is concluded from $\langle S, u \doteq x'|\mathcal{Y}'\rangle\, \Gamma, x' : A, \mathcal{Y}' : B \vdash \Delta$. By (Ren), there is a derivation of $\langle S, u \doteq x''|\mathcal{Y}''\rangle\, \Gamma, x'' : A, \mathcal{Y}'' : B \vdash \Delta$, where x'' and $\mathcal{Y}''$ are distinct from x and do not belong to $\mathit{afv}(u)$. By induction hypothesis, we have $\langle \sigma(S), u \doteq x''|\mathcal{Y}''\rangle\, \Gamma, x'' : \sigma(A), \mathcal{Y}'' : \sigma(B) \vdash \sigma(\Delta)$. We then conclude by (|L).

(In$\mathcal{N}$) For clarity, we abbreviate the substitution $\{x{\leftarrow}\mathsf{m}\}$ by σ. The proof proceeds by induction on the structure of derivations and case analysis on the last rule used, using Lemma 3.10(3) to show that all assertions that occur as premises of rule instances in the derivation are preserved. We present a detailed proof for one of the spatial rules, the (И) and (TL) rules, and all the quantifier rules. In each case, note that the structure of the derivation is preserved by the transformation.

- **(Case of (|R))** $\langle S\rangle\, \Gamma \vdash u : A|B, \Delta$ is concluded from $\langle S\rangle\, \Gamma \vdash t : A, \Delta$ and $\langle S\rangle\, \Gamma \vdash v : B, \Delta$ and $u \doteq_S t|v$. By induction hypothesis, we have $\langle \sigma(S)\rangle\, \Gamma \vdash \sigma(t) : \sigma(A), \sigma(\Delta)$ and $\langle \sigma(S)\rangle\, \Gamma \vdash \sigma(v) : \sigma(B), \sigma(\Delta)$. By Lemma 3.10, we have $\sigma(u) \doteq_{\sigma(S)} \sigma(t)|\sigma(v)$. We conclude by (|R).
- **(Case of (TL))** Suppose the instance of (TL) is not simple. Then $\langle S\rangle\, \Gamma, u : A \vdash \Delta$ is obtained by (TL) from $\langle S\rangle\, \Gamma, u' : A' \vdash \Delta$, where $(\mathsf{n}{\leftrightarrow}\mathsf{p})A \equiv_S A'$ and $(\mathsf{n} \leftrightarrow \mathsf{p})u \equiv_S u'$. By induction hypothesis, there is a derivation of $\langle \sigma(S)\rangle\, \Gamma, \sigma(u') : \sigma(A') \vdash \sigma(\Delta)$. By Lemma 3.19(1), we have $\sigma(A') \equiv_{\sigma(S)} \sigma((\mathsf{n}{\leftrightarrow}\mathsf{p})A) = (\sigma(\mathsf{n}){\leftrightarrow}\sigma(\mathsf{p}))\sigma(A)$. By Lemma 3.10, we have $\sigma(u') \equiv_{\sigma(S)} \sigma((\mathsf{n}{\leftrightarrow}\mathsf{p})u) = (\sigma(\mathsf{n}){\leftrightarrow}\sigma(\mathsf{p}))\sigma(u)$. We then obtain the conclusion by (TL). In the case where the instance of (TL) is simple, we can along similar lines obtain a derivation of size equal to one for $\langle \sigma(S)\rangle\, \sigma(\Gamma), \sigma(u) : \sigma(A) \vdash \sigma(\Delta)$ by instantiating every sequent in the given derivation with σ.
- **(Case of (И))** $\langle S\rangle\, \Gamma \vdash \Delta$ is obtained by (И) from $\langle S, u \doteq (\boldsymbol{\nu} z)x, z \,\#\, N\rangle\, \Gamma \vdash \Delta$, where z and x do not occur free in the conclusion and u, and N is a finite set of names not containing z.

  By (α) we may assume that $z \neq x$ and $z \notin \mathit{afv}(\mathsf{m})$. By induction hypothesis, we have $\langle \sigma(S), \sigma(u) \doteq (\boldsymbol{\nu} z)x, z \,\#\, \sigma(N)\rangle\, \sigma(\Gamma) \vdash \sigma(\Delta)$. Let $M = \mathit{afv}(\sigma(N))$. By (W), $\langle \sigma(S), \sigma(u) \doteq (\boldsymbol{\nu} z)x, z \,\#\, \sigma(N), z \,\#\, M\rangle\, \sigma(\Gamma) \vdash \sigma(\Delta)$. Write $S' \triangleq \langle \sigma(S), \sigma(u) \doteq (\boldsymbol{\nu} z)x, z \,\#\, M\rangle$. Since $M = \mathit{afv}(\sigma(N))$ and $z \,\#_{S'}\, M$, we can verify that $z \,\#_{S'}\, \sigma(N)$.

  By (CS) we have $\langle \sigma(S), \sigma(u) \doteq (\boldsymbol{\nu} z)x, z \,\#\, M\rangle\, \sigma(\Gamma) \vdash \sigma(\Delta)$. Now, note that z does not occur free in $\sigma(S)$, in $\sigma(\Gamma)$, u or M, or in $\sigma(\Delta)$, because it does not occur free in m, nor in the conclusion of the original sequent. Hence, by (И), we obtain the conclusion $\langle \sigma(S)\rangle\, \sigma(\Gamma) \vdash \sigma(\Delta)$.
- **(Case of ($\forall$R))** The sequent $\langle S\rangle\, \Gamma \vdash u : \forall z.A, \Delta$ is concluded from the sequent $\langle S\rangle\, \Gamma \vdash u : A\{z{\leftarrow}y\}, \Delta$, where y does not occur free in the conclu-

sion. By (Ren) we can assume that y does not occur (free or bound) neither in the initially given sequent nor in m. By induction hypothesis, we have $\langle\sigma(S)\rangle\,\sigma(\Gamma) \vdash \sigma(v) : \sigma(A\{z{\leftarrow}y\}), \sigma(\Delta)$. We have $\sigma\{z{\leftarrow}z\}(A)\{z{\leftarrow}y\} \equiv_\alpha \sigma(A\{z{\leftarrow}y\})$. By (α), $\langle\sigma(S)\rangle\,\sigma(\Gamma) \vdash \sigma(v) : \sigma\{z{\leftarrow}z\}(A)\{z{\leftarrow}y\}, \sigma(\Delta)$. By $(\forall\text{R})$ we conclude $\langle\sigma(S)\rangle\,\sigma(\Gamma) \vdash \sigma(v) : \sigma(\forall z.A), \sigma(\Delta)$, since $\sigma(\forall z.A) = \forall z.\sigma(A\{z{\leftarrow}z\})$.

- **(Case of (ИR))** We have the sequent $\langle S\rangle\,\Gamma \vdash u : \text{И}z.A, \Delta$, concluded by (И) from a derivation of $\langle S\rangle\,\Gamma \vdash u : A\{z{\leftarrow}\mathsf{n}\}, \Delta$, where $u \doteq_S (\boldsymbol{\nu}\mathsf{n})v$ and $\mathsf{n}\#_S \text{И}z.A$. By induction hypothesis, we have $\langle\sigma(S)\rangle\,\sigma(\Gamma) \vdash \sigma(u) : \sigma(A\{z{\leftarrow}\mathsf{n}\}), \sigma(\Delta)$. By Lemma 3.10(3), we have $\sigma(u) \doteq_{\sigma(S)} (\boldsymbol{\nu}\sigma(\mathsf{n}))\sigma(v)$.

  Note that $\sigma(\text{И}z.A) \equiv_\alpha \text{И}y.\sigma(A\{z{\leftarrow}y\})$ for some $y \notin \mathit{afv}(\mathsf{m}, x) \cup \mathit{fv}(A)$. Since $\mathit{lfv}(\text{И}z.A) = \mathit{lfv}(\text{И}y.A\{z{\leftarrow}y\})$, we also have $\mathsf{n}\#_S \text{И}y.A\{z{\leftarrow}y\}$. Note that $\mathit{lft}(\sigma(\text{И}z.A)) = \{\sigma(\mathsf{n}) \mid \mathsf{n} \in \mathit{lft}(\text{И}z.A)\}$.

  So, by Lemma 3.10(3), we conclude $\sigma(\mathsf{n})\,\#_{\sigma(S)}\,\sigma(\text{И}y.A\{z{\leftarrow}y\})$. By (ИR) and (α), we can build a derivation $\langle\sigma(S)\rangle\,\sigma(\Gamma) \vdash \sigma(u) : \sigma(\text{И}z.A), \sigma(\Delta)$, since $\sigma(A\{z{\leftarrow}y\})\{y{\leftarrow}\sigma(\mathsf{n})\} \equiv_\alpha \sigma(A\{z{\leftarrow}\mathsf{n}\})$.
- **(Case of (∀L))** We have a derivation of $\langle S\rangle\,\Gamma, u : \forall z.A \vdash \Delta$ concluded from a derivation of $\langle S\rangle\,\Gamma, u : A\{z{\leftarrow}\mathsf{n}\} \vdash \Delta$. By induction hypothesis, we have $\langle\sigma(S)\rangle\,\sigma(\Gamma), \sigma(u) : \sigma(A\{z{\leftarrow}\mathsf{n}\}) \vdash \sigma(\Delta)$. By (α) and $(\forall\text{L})$, we conclude $\langle\sigma(S)\rangle\,\sigma(\Gamma), \sigma(u) : \sigma(\forall z.A) \vdash \sigma(\Delta)$, since we have that $\sigma(\forall z.A) \equiv_\alpha \forall y.\sigma(A\{z{\leftarrow}y\})$ for some $y \in \mathit{afv}(\mathsf{m}, x) \cup \mathit{fv}(A)$ and we can verify that $\sigma(A\{z{\leftarrow}y\})\{y{\leftarrow}\sigma(\mathsf{n})\} \equiv_\alpha \sigma(A\{z{\leftarrow}\mathsf{n}\})$.
- **(Case of (ИL))** Similar to (ИR). ■

Lemma 5.5 *Every sequent of the form $\langle S\rangle\,\Gamma, u : A \vdash u : A, \Delta$, where A is not atomic, has a cut- and contraction-free derivation.*

Proof. By induction on the structure of the formula A we show that this sequent has a derivation in the stated conditions: in the base case the sequent is itself an instance of (Id). We show a few cases:

- **(Case of $A = A_1|A_2$)** By induction hypothesis, there are derivations of $\langle S, u \doteq x|y\rangle\,\Gamma, x : A_1, y : A_2 \vdash x : A_1, \Delta$ and $\langle S, u \doteq x|y\rangle\,\Gamma, x : A_1, y : A_2 \vdash y : A_2, \Delta$. By (|R) we get $\langle S, u \doteq x|y\rangle\,\Gamma, x : B_1, y : B_2 \vdash u : A_1|A_2, \Delta$. We then conclude by (|L).
- **(Case of $A = A_1 \triangleright A_2$)** By induction hypothesis, there are derivations of $\langle S\rangle\,\Gamma, x : A_1 \vdash x : A_1, \Delta$ and $\langle S\rangle\,\Gamma, x : A_1, x|u : A_2 \vdash x|u : A_2, \Delta$. By $(\triangleright\text{L})$ we get $\langle S\rangle\,\Gamma, u : A, x : A_1 \vdash x|u : A_2, \Delta$. We conclude by $(\triangleright\text{R})$.
- **(Case of $A = \text{И}x.B$)** Let N be the set of all name and propositional variables occurring free in the given sequent. Let $\langle S'\rangle \triangleq \langle S, u \doteq (\boldsymbol{\nu}y)x, y\,\#\,N\rangle$, where x and y are also chosen not free in the sequents under consideration. By induction hypothesis, $\langle S'\rangle\,\Gamma, u : B\{x{\leftarrow}y\} \vdash v : B\{x{\leftarrow}y\}\Delta$. Note that $y\,\#_{S'}\,\text{И}x.B$ and $u \doteq_{S'} (\boldsymbol{\nu}y)x$. We conclude by (ИL), (ИR), and (И). ■

Lemma 8.3 (Basic Simplification) *Assume $(\mathsf{n} \leftrightarrow \mathsf{m})u \doteq_S u'$ and $(\mathsf{n} \leftrightarrow \mathsf{m})A \Downarrow_S A'$, $\Gamma \Downarrow_S \Gamma'$ and $\Delta \Downarrow_S \Delta'$. Then we have:*

(1) If $\vdash_1 \langle S\rangle\, \Gamma \vdash u : A, \Delta$ *in* $\mathbf{S}$ *then* $\vdash_1 \langle S\rangle\, \Gamma' \vdash u' : A', \Delta'$ *in* $\mathbf{S1}$.
(2) If $\vdash_1 \langle S\rangle\, \Gamma, u : A \vdash \Delta$ *in* $\mathbf{S}$ *then* $\vdash_1 \langle S\rangle\, \Gamma', u' : A' \vdash \Delta'$ *in* $\mathbf{S1}$.

Proof. We prove (1), the proof for (2) is similar. The proof rests on the following observation: if there is a derivation of $\langle S\rangle\, \Gamma \vdash u : A, \Delta$ built just from (Id), (TL) and (TR), then there are formulas B and B' such that $\Gamma = \Gamma_l, t : B$ and $u : A, \Delta = t' : B', \Delta_r$ and $\rho B \equiv_S \sigma B'$ and $\rho t \doteq_S \sigma t'$, hence $\sigma^{-1}\rho B \equiv_S B'$. ■

Lemma 5.8 (Simplification) *Assume* $(\mathsf{n}\leftrightarrow\mathsf{m})u \doteq_S u'$ *and* $(\mathsf{n}\leftrightarrow\mathsf{m})A \Downarrow_S A'$, $\Gamma \Downarrow_S \Gamma'$ *and* $\Delta \Downarrow_S \Delta'$*. Then the following size-preserving proof principles are admissible:*

(1) If $\vdash_n \langle S\rangle\, \Gamma \vdash u : A, \Delta$ *in* $\mathbf{S}$ *then* $\vdash_n \langle S\rangle\, \Gamma' \vdash u' : A', \Delta'$ *in* $\mathbf{S1}$.
(2) If $\vdash_n \langle S\rangle\, \Gamma, u : A \vdash \Delta$ *in* $\mathbf{S}$ *then* $\vdash_n \langle S\rangle\, \Gamma', u' : A' \vdash \Delta'$ *in* $\mathbf{S1}$.

The resulting derivations are simple and normalized. Moreover, if the original derivations are cut-free then the resulting ones are also cut-free.

Proof. The proof proceeds by mutual induction on the size of the derivations (1) $\vdash_n \langle S\rangle\, \Gamma \vdash u : A, \Delta$ and (2) $\vdash_n \langle S\rangle\, \Gamma, u : A \vdash \Delta$. We show the proof for (1), the case of (2) is handled in a similar way.

(Case of (1)). Assume $\vdash_n \langle S\rangle\, \Gamma \vdash \Delta, u : A$. Possible ways of deriving this sequent are: (1) the last rule is a logical rule acting on a formula in Δ or Γ, (2) the last rule is a world rule acting on S or (И), or (3) the last rule is (Id), (Cut) or a logical right rule acting on the principal formula $u : A$.

(Subcase 1) The result follows from the inductive hypothesis, possibly using (Ren) in the ($\forall$R) case.

(Subcase 2) If the last rule is some world (S$-$) rule, the result is an immediate consequence of the induction hypothesis. If the last rule is (И), the sequent $\langle S\rangle\, \Gamma \vdash u : A, \Delta$ is concluded from $\vdash_{n-1} \langle S, v \doteq (\boldsymbol{\nu} x)x, x \# N\rangle\, \Gamma \vdash u : A, \Delta$. By (Ren), there is a derivation $\vdash_{n-1} \langle S, v \doteq (\boldsymbol{\nu} y)x, y \# N\rangle\, \Gamma \vdash t : A, \Delta$, where y is chosen not free neither in the original sequent, nor in $\Gamma', \Delta', u' : A'$. By induction hypothesis, we conclude $\vdash_{n-1} \langle S, v \doteq (\boldsymbol{\nu} y)x, y \# N\rangle\, \Gamma' \vdash u' : A', \Delta'$. By (И), $\langle S\rangle\, \Gamma' \vdash u' : A', \Delta'$ is obtained. We now address (Subcase 3).

- **(Case of (Id))** By Lemma 8.3(1).
- **(Case of (Cut))** We have $\langle S\rangle\, \Gamma \vdash \Delta, u : A$ concluded from $\vdash_{n-1} \langle S\rangle\, \Gamma \vdash \Delta, u : A, v : B$ and $\vdash_{n-1} \langle S\rangle\, \Gamma, v : B \vdash \Delta, u : A$. Let $B \Downarrow_S B'$. By induction hypothesis, we have $\vdash_{n-1} \langle S\rangle\, \Gamma' \vdash \Delta', u' : A', u : B'$ and $\vdash_{n-1} \langle S\rangle\, \Gamma', u : B' \vdash \Delta', u' : A'$. We then conclude by (Cut).
- **(Case of (CR))** We have $\langle S\rangle\, \Gamma \vdash \Delta, u : A$ concluded from $\vdash_{n-1} \langle S\rangle\, \Gamma \vdash \Delta, u : A, u : A$. By the induction hypothesis, we have $\vdash_{n-1} \langle S\rangle\, \Gamma' \vdash \Delta', u' : A', u : A''$, where $A \Downarrow_S A''$. Again by induction hypothesis, we conclude $\vdash_{n-1} \langle S\rangle\, \Gamma' \vdash \Delta', u' : A', u' : A'$, since $A'' \Downarrow_S A'$. We then conclude by (CR).
- **(Case of (TR))** We consider first the case where the application of (TR) is not simple. We have $\langle S\rangle\, \Gamma \vdash u : A, \Delta$ concluded from $\vdash_{n-1} \langle S\rangle\, \Gamma \vdash v : B, \Delta$, where $B \equiv_S (\mathsf{p}\leftrightarrow\mathsf{q})A$ and $v \doteq_S (\mathsf{p}\leftrightarrow\mathsf{q})u$. Hence $(\mathsf{p}\leftrightarrow\mathsf{q})B \equiv_S A$ and $(\mathsf{p}\leftrightarrow\mathsf{q})v \doteq_S u$. By Lemma 3.19(2) there is B' such that $(\mathsf{p}\leftrightarrow\mathsf{q})B \Downarrow_S B'$ and

$A \Downarrow_S B'$. By induction hypothesis, we conclude $\vdash_{n-1} \langle S\rangle\, \Gamma' \vdash u : B', \Delta'$. Since $(\mathsf{n} \leftrightarrow \mathsf{m})A \Downarrow_S A'$ we also have $(\mathsf{n} \leftrightarrow \mathsf{m})B' \Downarrow_S A'$. Again by the induction hypothesis, we conclude $\vdash_{n-1} \langle S\rangle\, \Gamma' \vdash u' : A', \Delta'$. Otherwise, suppose the application of (TR) is simple. Then, we have $\vdash_1 \langle S\rangle\, \Gamma \vdash u : A, \Delta$. By Lemma 8.3(1), we conclude $\vdash_1 \langle S\rangle\, \Gamma' \vdash u' : A', \Delta'$.

- **(Case of ($\wedge$R))** We have $A = B \wedge C$ and $\langle S\rangle\, \Gamma \vdash \Delta, u : A$ concluded from $\vdash_{n-1} \langle S\rangle\, \Gamma \vdash \Delta, u : B$ and $\vdash_{n-1} \langle S\rangle\, \Gamma \vdash \Delta, u : C$. We have $A' = B' \wedge C'$ with $(\mathsf{n}\leftrightarrow\mathsf{m})B \Downarrow_S B'$ and $(\mathsf{n}\leftrightarrow\mathsf{m}) \Downarrow_S C'$. By induction hypothesis, we have $\vdash_{n-1} \langle S\rangle\, \Gamma' \vdash \Delta', u' : B'$ and $\vdash_{n-1} \langle S\rangle\, \Gamma' \vdash \Delta', u' : C'$. By ($\wedge$R), we conclude $\vdash_n \langle S\rangle\, \Gamma' \vdash \Delta', u' : A'$.
- **(Case of ($\Rightarrow$R))** We have $A = B \Rightarrow C$ and $\langle S\rangle\, \Gamma \vdash \Delta, u : A$ concluded from $\vdash_{n-1} \langle S\rangle\, \Gamma, u : B \vdash \Delta, u : C$. We have $A' = B' \Rightarrow C'$ with $(\mathsf{n}\leftrightarrow\mathsf{m})B \Downarrow_S B'$ and $(\mathsf{n}\leftrightarrow\mathsf{m}) \Downarrow_S C'$. By induction hypothesis, we have $\vdash_{n-1} \langle S\rangle\, \Gamma', u' : B' \vdash \Delta', u : C$. By induction hypothesis again, $\vdash_{n-1} \langle S\rangle\, \Gamma', u' : B' \vdash \Delta', u' : C'$. By ($\Rightarrow$R), we conclude $\vdash_n \langle S\rangle\, \Gamma' \vdash \Delta', u' : A'$.
- **(Case of ($|$R))** We have $A = B|C$ and $\langle S\rangle\, \Gamma \vdash \Delta, u : A$ concluded from $\langle S\rangle\, \Gamma \vdash \Delta, t : B$ and $\langle S\rangle\, \Gamma \vdash \Delta, v : C$, where $t|v \doteq_S u$. We have $A' = B'|C'$ with $(\mathsf{n}\leftrightarrow\mathsf{m})B \Downarrow_S B'$ and $(\mathsf{n}\leftrightarrow\mathsf{m}) \Downarrow_S C'$ and $u' \doteq_S (\mathsf{n}\leftrightarrow\mathsf{m})u \doteq_S (\mathsf{n}\leftrightarrow\mathsf{m})t|(\mathsf{n}\leftrightarrow\mathsf{m})v$. By induction hypothesis, we have $\langle S\rangle\, \Gamma' \vdash \Delta', (\mathsf{n}\leftrightarrow\mathsf{m})t : B'$ and $\langle S\rangle\, \Gamma' \vdash \Delta', (\mathsf{n}\leftrightarrow\mathsf{m})v : C'$. By ($|$R), we conclude $\vdash_n \langle S\rangle\, \Gamma' \vdash \Delta', u' : A'$.
- **(Case of ($\triangleright$R))** We have $A = B \triangleright C$ and $\langle S\rangle\, \Gamma \vdash \Delta, u : A$ concluded from $\vdash_{n-1} \langle S\rangle\, \Gamma, x : B \vdash \Delta, v : C$ and $v \doteq_S x|u$, where x does not occur in the conclusion. We have $A' = B' \triangleright C'$ with $(\mathsf{n}\leftrightarrow\mathsf{m})B \Downarrow_S B'$ and $(\mathsf{n}\leftrightarrow\mathsf{m}) \Downarrow_S C'$. By induction hypothesis (twice) we have $\vdash_{n-1} \langle S\rangle\, \Gamma', (\mathsf{n}\leftrightarrow\mathsf{m})x : B' \vdash \Delta', (\mathsf{n}\leftrightarrow\mathsf{m})v : C'$.

  By (In$\mathcal{I}$) with $\{x\leftarrow(\mathsf{n}\leftrightarrow\mathsf{m})x\}$ and induction hypothesis, we have $\vdash_{n-1} \langle S\rangle\, \Gamma', x : B' \vdash \Delta', x|v : C'$, since $((\mathsf{n}\leftrightarrow\mathsf{m})v)\{x\leftarrow(\mathsf{n}\leftrightarrow\mathsf{m})x\} \doteq_S ((\mathsf{n}\leftrightarrow\mathsf{m})x|u)\{x\leftarrow(\mathsf{n}\leftrightarrow\mathsf{m})x\} \doteq_S x|u'$, because $u' \doteq_S (\mathsf{n}\leftrightarrow\mathsf{m})u$ by assumption. By ($\triangleright$R), we conclude $\vdash_n \langle S\rangle\, \Gamma' \vdash \Delta', u' : A'$.
- **(Case of ($\Diamond$R))** We have $A = \Diamond B$ and $\langle S\rangle\, \Gamma \vdash \Delta, u : A$ concluded from $\vdash_{n-1} \langle S\rangle\, \Gamma \vdash \Delta, v : B$ and $t \rightarrow_S v$. We have $A' = \Diamond B'$ where $(\mathsf{n}\leftrightarrow\mathsf{m})B \Downarrow_S B'$. By induction hypothesis, $\vdash_{n-1} \langle S\rangle\, \Gamma' \vdash \Delta', u' : B'$. By ($\Diamond$R), we conclude $\vdash_n \langle S\rangle\, \Gamma' \vdash \Delta', u' : A'$, since $u = (\mathsf{n}\leftrightarrow\mathsf{m})t \rightarrow_S (\mathsf{n}\leftrightarrow\mathsf{m})v$ by (Swap Red).
- **(Case of ($\circledR$R))** We have $A = \mathsf{q} \circledR B$ and $\langle S\rangle\, \Gamma \vdash \Delta, u : A$ concluded from $\vdash_{n-1} \langle S\rangle\, \Gamma \vdash \Delta, v : B$ and $t \doteq_S (\boldsymbol{\nu}\mathsf{q})v$. We have $A' = \mathsf{q}' \circledR B'$ with $\mathsf{q}' \doteq_S (\mathsf{n}\leftrightarrow\mathsf{m})\mathsf{q}$ and $(\mathsf{n}\leftrightarrow\mathsf{m})B \Downarrow_S B'$. By induction hypothesis, $\vdash_{n-1} \langle S\rangle\, \Gamma' \vdash \Delta', (\mathsf{n}\leftrightarrow\mathsf{m})v : B'$. By ($\circledR R$), we conclude $\vdash_n \langle S\rangle\, \Gamma' \vdash \Delta', u' : A'$, since $(\boldsymbol{\nu}\mathsf{q}')(\mathsf{n}\leftrightarrow\mathsf{m})v \doteq_S u' \doteq_S (\mathsf{n}\leftrightarrow\mathsf{m})u$.
- **(Case of ($\oslash$R))** Similar to ($\circledR$R).
- **(Case of (ИR))** We have $A = \text{И}x.B$ and $\langle S\rangle\, \Gamma \vdash u : A, \Delta$ concluded from $\vdash_{n-1} \langle S\rangle\, \Gamma \vdash u : B\{x\leftarrow\mathsf{p}\}, \Delta$ where $\mathsf{p} \#_S \text{И}x.B$, and $u \doteq_S (\boldsymbol{\nu}\mathsf{p})v$. By ($\alpha$) we can assume that x is not free in n, m or S, so that $A' = \text{И}x.B'$ with $(\mathsf{n}\leftrightarrow\mathsf{m})B\{x\leftarrow(\mathsf{n}\leftrightarrow\mathsf{m})x\} \Downarrow_S B'$. Let $(\mathsf{n}\leftrightarrow\mathsf{m})B\{x\leftarrow\mathsf{p}\} \Downarrow_S B''$, by induction hypothesis we conclude $\vdash_{n-1} \langle S\rangle\, \Gamma' \vdash u' : B'', \Delta'$. We have $(\mathsf{n}\leftrightarrow\mathsf{m})B\{x\leftarrow(\mathsf{n}\leftrightarrow\mathsf{m})x\}\{x\leftarrow(\mathsf{n}\leftrightarrow\mathsf{m})\mathsf{p}\} \equiv_S (\mathsf{n}\leftrightarrow\mathsf{m})B\{x\leftarrow\mathsf{p}\}$.

By Lemma 3.19(1), we have $(\mathsf{n} \leftrightarrow \mathsf{m})B\{x{\leftarrow}\mathsf{p}\} \Downarrow_S B'\{x{\leftarrow}(\mathsf{n} \leftrightarrow \mathsf{m})\mathsf{p}\}$. Hence $B'\{x{\leftarrow}(\mathsf{n}{\leftrightarrow}\mathsf{m})\mathsf{p}\} \equiv_S B''$, and actually $B'' \Downarrow_S B'\{x{\leftarrow}(\mathsf{n}{\leftrightarrow}\mathsf{m})\mathsf{p}\}$, since both formulas are normalized. By induction hypothesis again, we have $\vdash_{n-1} \langle S\rangle\, \Gamma' \Vdash u' : B'\{x{\leftarrow}(\mathsf{n} \leftrightarrow \mathsf{m})\mathsf{p}\}, \Delta'$. By (ИR), we conclude $\vdash_n \langle S\rangle\, \Gamma' \Vdash \Delta', u' : A'$, since by Lemma 3.10(3) $(\mathsf{n} \leftrightarrow \mathsf{m})\mathsf{p} \#_S A'$, and $u' \doteq_S (\mathsf{n}{\leftrightarrow}\mathsf{m})u \doteq_S (\boldsymbol{\nu}(\mathsf{n}{\leftrightarrow}\mathsf{m})\mathsf{p})(\mathsf{n}{\leftrightarrow}\mathsf{m})v$.

- **(Case of ($\forall$R))** We have $A = \forall x.B$ and $\langle S\rangle\, \Gamma \Vdash u : A, \Delta$ concluded from $\vdash_{n-1} \langle S\rangle\, \Gamma \Vdash u : B\{x{\leftarrow}y\}, \Delta$, where y is not free in the conclusion, and by (Ren) we can also assume that y is not free in Γ', Δ', A', u'. By (α) we can assume that x does not occur in $\mathsf{n}, \mathsf{m}, S$, so that $A' = \forall x.B'$ where $(\mathsf{n} \leftrightarrow \mathsf{m})B\{x{\leftarrow}(\mathsf{n} \leftrightarrow \mathsf{m})x\} \Downarrow_S B'$. By Lemma 3.19(1), we obtain $(\mathsf{n}{\leftrightarrow}\mathsf{m})B\{x{\leftarrow}(\mathsf{n}{\leftrightarrow}\mathsf{m})y\} \Downarrow_S B'\{x{\leftarrow}y\}$. By (In$\mathcal{N}$) with $\{y{\leftarrow}(\mathsf{n}{\leftrightarrow}\mathsf{m})y\}$ we have $\vdash_{n-1} \langle S\rangle\, \Gamma \Vdash u : B\{x{\leftarrow}(\mathsf{n}{\leftrightarrow}\mathsf{m})y\}, \Delta$. Since $(\mathsf{n}{\leftrightarrow}\mathsf{m})B\{x{\leftarrow}(\mathsf{n}{\leftrightarrow}\mathsf{m})y\} \Downarrow_S B'\{x{\leftarrow}y\}$, by induction hypothesis, we have $\vdash_{n-1} \langle S\rangle\, \Gamma' \Vdash u' : B'\{x{\leftarrow}y\}, \Delta'$. By ($\forall$R) we conclude $\vdash_n \langle S\rangle\, \Gamma' \Vdash u' : A', \Delta'$. ■

Lemma 5.13 (Inversion)

Proof. By induction on the size of the derivation of the given sequents and case analysis in the last rule used. We present a detailed argument for (7), the other cases are handled in a similar way.

If $\vdash_n \langle S\rangle\, \Gamma, u : A|B \Vdash \Delta$ *then* $\vdash_n \langle S, u \doteq \mathscr{x}\,|\mathscr{y}\rangle\, \Gamma, \mathscr{x} : A, \mathscr{y} : B \Vdash \Delta$*, for any* $\mathscr{x}, \mathscr{y}$ *not free in the first sequent.*

We consider three subcases: (a) If the last step is an application of (|L) to the distinguished formula $u : A|B$, the proof is concluded.

(b) The last step in an application of (SL) to the distinguished formula $u : A|B$. Hence we have $\vdash_{n-1} \langle S\rangle\, \Gamma, u' : A'|B' \Vdash \Delta$, where $(\mathsf{n}{\leftrightarrow}\mathsf{m})A \Downarrow_S A'|B'$ and $u' \doteq_S (\mathsf{n}{\leftrightarrow}\mathsf{m})u$. By induction hypothesis, we have $\vdash_{n-1} \langle S, u' \doteq \mathscr{x}\,|\mathscr{y}\rangle\, \Gamma, \mathscr{x} : A', \mathscr{y} : B' \Vdash \Delta$. By (In$\mathcal{I}$) we have $\vdash_{n-1} \langle S, u' \doteq (\mathsf{n}{\leftrightarrow}\mathsf{m})\mathscr{x}\,|(\mathsf{n}{\leftrightarrow}\mathsf{m})\mathscr{y}\rangle\, \Gamma, (\mathsf{n}{\leftrightarrow}\mathsf{m})\mathscr{x} : A', (\mathsf{n} \leftrightarrow \mathsf{m})\mathscr{y} : B' \Vdash \Delta$. By Lemma 5.8(2) (twice), we have $\vdash_{n-1} \langle S, u' \doteq (\mathsf{n}{\leftrightarrow}\mathsf{m})\mathscr{x}\,|(\mathsf{n}{\leftrightarrow}\mathsf{m})\mathscr{y}\rangle\, \Gamma, \mathscr{x} : A, \mathscr{y} : B \Vdash \Delta$, since $(\mathsf{n}{\leftrightarrow}\mathsf{m})B' \Downarrow_S B$, $(\mathsf{n}{\leftrightarrow}\mathsf{m})A' \Downarrow_S A$ and all formulas in Γ and Δ are normalized. By (W) and (CS), we conclude $\vdash_{n-1} \langle S, u \doteq \mathscr{x}\,|\mathscr{y}\rangle\, \Gamma, \mathscr{x} : A, \mathscr{y} : B \Vdash \Delta$.

(c) Otherwise, the sequent $\langle S\rangle\, \Gamma, u : A|B \Vdash \Delta$ is concluded from $k = 1$ or $k = 2$ premises of the form $\langle S_i\rangle\, \Gamma_i, u : A|B \Vdash \Delta_i$, for $i = 1, \ldots, k$ and possibly some premises of the form $S \vdash c$, by an application of some inference rule $(\mathscr{R})$ acting either on a principal formula in Γ or Δ, or in $\langle S\rangle$. By induction hypothesis, we conclude $\vdash_{n-1} \langle S_i, u \doteq \mathscr{x}\,|\mathscr{y}\rangle\, \Gamma_i, \mathscr{x} : A, \mathscr{y} : B \Vdash \Delta_i$ for $i = 1, \ldots, k$, where $\mathscr{x}$ and $\mathscr{y}$ can be chosen fresh with respect to S, S_i, Γ, Δ, Γ_i and Δ_i (so that any eigenvariable condition required for applying $(\mathscr{R})$ still holds). By $(\mathscr{R})$, we conclude $\vdash_n \langle S, u \doteq \mathscr{x}\,|\mathscr{y}\rangle\, \Gamma, u : A, u : B \Vdash \Delta$. ■

Lemma 5.14 (Contraction Elimination) *In the system* **CF** *the following size-preserving proof principles are admissible, provided the sequents shown are normalized*

$$\frac{\vdash_n \langle S\rangle\, \Gamma \Vdash u : A, u : A, \Delta}{\vdash_n \langle S\rangle\, \Gamma, u : A \Vdash \Delta}\ (\mathrm{CR}) \quad \frac{\vdash_n \langle S\rangle\, \Gamma, u : A, u : A \Vdash \Delta}{\vdash_n \langle S\rangle\, \Gamma, u : A \Vdash \Delta}\ (\mathrm{CL})$$

The resulting derivations are normalized. Moreover, if the original derivations are cut-free, so are the resulting derivations; if the original derivations are simple, so are the resulting derivations.

Proof. The principles (CL) and (CR) are proved by mutual induction on the size of the respective derivations. If the last rule of the derivation is a world (S−) rule, (Cut), or a logical rule other than (Id), applying to some formula in Γ or Δ, the result follows directly by the induction hypothesis. If last rule is (Id), identifying atomic formulas in Γ or Δ, then $\langle S\rangle\, \Gamma \Vdash \Delta$ is an instance of (Id). The conclusion can then be obtained by adding the required formulas to the left and right context of this sequent. Otherwise, we consider the case of each possible rule acting on one of the distinguished occurrences of $u : A$. We consider a few cases for (CR), (CL) is handled in a similar way.

- **(Case of (Id))** Immediate, for just one of the $u : A$ can be relevant to (Id).
- **(Case of (SR)** If this occurrence of (SR) is simple, then just one of the occurrences of $u : A$ is used in the (Id) axiom below it, so we immediately conclude $\vdash_1 \langle S\rangle\, \Gamma \Vdash u : A, \Delta$. Otherwise, we have that $\langle S\rangle\, \Gamma \Vdash u : A, u : A, \Delta$ results from $\vdash_{n-1} \langle S\rangle\, \Gamma \Vdash u' : A', u : A, \Delta$ where $(\mathsf{n}{\leftrightarrow}\mathsf{m})u \doteq_S u'$ and $(\mathsf{n}{\leftrightarrow}\mathsf{m})A \Downarrow_S A'$. By Lemma 5.8(1), we have $\vdash_{n-1} \langle S\rangle\, \Gamma \Vdash u' : A', u' : A', \Delta$. By induction hypothesis, we conclude $\vdash_{n-1} \langle S\rangle\, \Gamma \Vdash u' : A', \Delta$. By (SR) we conclude $\vdash_n \langle S\rangle\, \Gamma \Vdash u : A, \Delta$.
- **(Case of (|RK))** We have $A = B|C$ and $\langle S\rangle\, \Gamma \Vdash u : A, u : A, \Delta$ concluded from $\vdash_{n-1} \langle S\rangle\, \Gamma \Vdash v : B, u : A, u : A, \Delta$ and $\vdash_{n-1} \langle S\rangle\, \Gamma \Vdash t : C, u : A, u : A, \Delta$ and $v|t \doteq_S u$. By induction hypothesis, we have $\vdash_{n-1} \langle S\rangle\, \Gamma \Vdash v : B, u : A, \Delta$ and $\vdash_{n-1} \langle S\rangle\, \Gamma \Vdash t : C, u : A, \Delta$. We conclude by (|RK).
- **(Case of (▷R))** We have $A = B \triangleright C$ and $\vdash_n \langle S\rangle\, \Gamma \Vdash u : A, u : A, \Delta$ concluded from $\vdash_{n-1} \langle S\rangle\, \Gamma, \mathscr{x} : B \Vdash v : C, u : A, \Delta$ and $v \doteq_S \mathscr{x}|u$. By Lemma 5.13(6), we have $\vdash_{n-1} \langle S\rangle\, \Gamma, \mathscr{x} : B, \mathscr{y} : B \Vdash v : C, \mathscr{y}|u : C, \Delta$ for some fresh $\mathscr{y}$. By $(\mathrm{In}\mathcal{I})$ we get $\vdash_{n-1} \langle S\rangle\, \Gamma, \mathscr{x} : B, \mathscr{x} : B \Vdash v : C, \mathscr{x}|u : C, \Delta$. By Lemma 5.8(1) with the identity permutation, we conclude $\vdash_{n-1} \langle S\rangle\, \Gamma, \mathscr{x} : B, \mathscr{x} : B \Vdash v : C, v : C, \Delta$, since $v \doteq_S \mathscr{x}|u$. By induction hypothesis, we get $\vdash_{n-1} \langle S\rangle\, \Gamma, \mathscr{x} : B \Vdash v : C, \Delta$. The conclusion follows by (▷R).
- **(Case of (∀R))** We have $A = \forall x.B$ and $\langle S\rangle\, \Gamma \Vdash u : A, u : A, \Delta$ concluded from $\vdash_{n-1} \langle S\rangle\, \Gamma \Vdash u : B\{x{\leftarrow}y\}, u : A, \Delta$, where y is not free in the conclusion. By Lemma 5.13(5), we have $\vdash_{n-1} \langle S\rangle\, \Gamma \Vdash u : B\{x{\leftarrow}y\}, u : B\{x{\leftarrow}z\}, \Delta$, where z is not free in the conclusion. By $(\mathrm{In}\mathcal{N})$ with $\{z{\leftarrow}y\}$, we have $\vdash_{n-1} \langle S\rangle\, \Gamma \Vdash u : B\{x{\leftarrow}y\}, u : B\{x{\leftarrow}y\}, \Delta$. By the induction hypothesis, we get $\vdash_{n-1} \langle S\rangle\, \Gamma \Vdash u : B\{x{\leftarrow}y\}, \Delta$ and we conclude by (∀R).
- **(Case of (ИRK))** We have $A = \text{И}x.B$ and $\langle S\rangle\, \Gamma \Vdash u : A, u : A, \Delta$ concluded from $\vdash_{n-1} \langle S\rangle\, \Gamma \Vdash u : B\{x{\leftarrow}\boldsymbol{p}\}, u : A, u : A, \Delta$ where $\boldsymbol{p} \#_S \text{И}x.B$, and $t \doteq_S (\boldsymbol{\nu p})v$. By induction hypothesis, $\vdash_{n-1} \langle S\rangle\, \Gamma \Vdash u : B\{x{\leftarrow}\boldsymbol{p}\}, u : A, \Delta$, we then conclude by (ИRK). ■