Sequential Process Logics: Soundness Proofs

Kohei Honda

Queen Mary, London

Abstract. We prove soundness results of the proof systems for basic specification logics for sequentially typed π-calculi, offering a basis for their applications. The proofs for different logics enjoy modularity in correspondence with that of their proof sustems.

1 Introduction

In the companion papers [6, 7], we introduced specification logics for the linear/affine π-calculus and its extensions, and illustrated how they can be used for reasoning about processes and programs. This paper proves the soundness results for these logics, establishing that the proof rules as presented in [6, 7] can only derive those statements which are sound with respect to their interpretation in the corresponding models. We cover the following logics, presented in the order given below.

- The logic for processes with the linear type discipline (Section 2).
- The total logic for processes with the affine type discipline (Section 3).
- The total logic for processes with the affine type discipline and open state (Section 4).
- The total logic for processes with the affine type discipline and local state (Section 5).
- The partial counterpart of the affine total logic (Section 6).

In the linear type discipline, the total and partial logics coincide, due to strong normalisability. In the affine calculi, we need distinction between total logics and partial ones (in the sense of partial/total correctness in the standard Hoare logics). Different nature between these two kinds of logics, both in syntax and semantics, demands their separate treatment. Thus, for a smoother presentation, we treat the total logics for various affine calculi, discussing their partial counterparts at the end. Reflecting the modularity of the logics and their proof systems, the proof of soundness for each logic is incremental over that of the preceding ones. All proofs are operational and elementary. Some use is made of basic unfolding techniques when treating circular process composition.

The present paper only contains, in condensed form, those definitions needed for making the technical development self-contained. For discussions on the underlying ideas, applications and related studies, see [6, 7].

$$\frac{}{\vdash \mathbf{0} \triangleright \_}\ (\mathsf{Zero}) \qquad \frac{\vdash P_1 \triangleright \Gamma_1\,;\,\Delta_1, \Theta \quad \vdash P_2 \triangleright \Gamma_2, \overline{\Theta}\,;\,\Delta_2}{\vdash P_1|P_2 \triangleright \Gamma_1 \cup \Gamma_2\,;\,\Delta_1, \Delta_2, \Theta'}\ (\mathsf{Par})\ (\Theta \odot \overline{\Theta} = \Theta')$$

$$\frac{\vdash P \triangleright \Gamma, x{:}\tau \quad \mathsf{md}(\tau) \in \{\updownarrow, !\}}{\vdash (\boldsymbol{\nu}\, x)P \triangleright \Gamma}\ (\mathsf{Res}) \qquad \frac{\vdash P \triangleright \Gamma^{-x} \quad \mathsf{md}(\tau) \in \{\updownarrow, ?\}}{\vdash P \triangleright \Gamma\,,\, x{:}\tau}\ (\mathsf{Weak})$$

$$\frac{\forall i.\ \vdash P_i \triangleright ?\Gamma^{-x},\, \vec{y}_i{:}\vec{\tau}_i,\, v{:}\rho^{\uparrow}}{\vdash x[\&_i(\vec{y}_i).P_i] \triangleright \Gamma,\, x{:}[\&_i\vec{\tau}_i]^{\downarrow},\, v{:}\rho}\ (\mathsf{Bra}^{\downarrow}) \qquad \frac{\vdash P \triangleright ?\Gamma,\, \vec{y}{:}\vec{\tau}_i}{\vdash \overline{x}\mathtt{in}_i(\vec{y})P \triangleright \Gamma,\, x{:}[\oplus_{i\in I}\vec{\tau}_i]^{\uparrow}}\ (\mathsf{Sel}^{\uparrow})$$

$$\frac{\vdash P \triangleright ?\Gamma^{-x},\, \vec{y}{:}\vec{\tau},\, z{:}\rho}{\vdash !\,x(\vec{y}z).P \triangleright \Gamma,\, x{:}(\vec{\tau}\rho)^{!}}\ (\mathsf{In}^{!}) \qquad \frac{\vdash R \triangleright ?\Gamma,\, x{:}\sigma,\, \vec{y}{:}\vec{\tau} \quad \vdash P \triangleright ?\Gamma,\, x{:}\sigma,\, z{:}\rho',\, v{:}\rho^{\uparrow}}{\vdash \overline{x}(\vec{y}z)(R|P) \triangleright \Gamma, x{:}\sigma, v{:}\rho}\ (\mathsf{Out}^{?})\ (\sigma = (\vec{\tau}\rho')^{?})$$

Fig. 1. Typing Rules for Sequential Linear Processes

2 Logic for Linear Sequential Processes

2.1 Linear Sequential Typing

We use the following set of channel types.

$$\tau \ ::= \ (\vec{\tau}^{?}\rho^{\uparrow})^{!} \ \mid\ [\&_{i\in I}\vec{\tau}_i^{?}]^{\downarrow} \ \mid\ (\vec{\tau}^{!}\rho^{\downarrow})^{?} \ \mid\ [\oplus_{i\in I}\vec{\tau}_i^{!}]^{\uparrow} \ \mid\ \updownarrow$$

Types of mode $!, \uparrow$ are *positive*, while their duals are *negative*. $\updownarrow$ is *neutral*. An *action type* $(\Gamma, \Delta, \Theta, \ldots)$ is a finite map from names $(x, y, \ldots)$ to channel types. An action type is *well-formed* if it contains at most one $\uparrow$-type. $\Gamma;\Delta$ stands for an action type whose negative (resp. positive/neutral) component is Γ (resp. Δ).

We use a version of sequential linear typing [15] without causality edges, with the sequent $\vdash P \triangleright \Gamma$. We list the typing rules in Figure 1 ($\odot$ is given in Appendix). In each rule, we assume all (action) types are well-formed, with "," in "Γ, Δ" indicating disjointness of the domains. $?\Gamma$ etc. indicates the mode of the types in Γ, while Γ^{-x} says x does not occur in Γ. Processes typable in this system are *linearly typable*. A typed process is often written P^{Γ}. Below $\Downarrow$ denotes the convergence with respect to the standard reduction relation, written $\longrightarrow$.

Proposition 1. (strong normalisability [15]) *If P is linearly typable, $P \Downarrow$.*

Further P is sequential in the sense that $P \longrightarrow P_{1,2}$ implies $P_1 \equiv P_2$. We write $\cong$ for the standard contextual congruence on linearly typed processes.

2.2 Logic for Linear Processes

2.2.1 Terms and Formulae. The *terms* of the specification logic is given by the following grammar.

$$\begin{array}{rcl} e & ::= & * \mid \mathtt{true} \mid \mathtt{false} \mid n \mid i^{\mathsf{nat}} \mid i^{\mathsf{bool}} \mid e + e' \mid e \wedge e' \mid \ldots \\ a & ::= & x^{\tau} \mid a^{(\vec{\rho}\tau)^!}\bullet \vec{b}^{\vec{\rho}} \mid \mathtt{in}_e(\vec{a}^{\vec{\rho}^!})^{\uparrow} \end{array}$$

where $i, i', \ldots$ range over a couple set of *variables* (which are disjoint from channel names). Variables and channel names are are often called *names*. $e, e', \ldots$ are called *data expressions*, while $a, a', \ldots$ *behavioural expressions*. Terms are naturally typed, with $a^{(\vec{\rho}\tau)^!}\bullet \vec{b}^{\vec{\rho}}$ typed as τ (note this in effect means the last two terms only take names as operands). We only treat well-typed terms.

The set of formulae are given by the following grammar. Below $\star$ ranges over $\{\wedge, \vee, \supset\}$ and $\mathcal{Q}$ over $\{\forall, \exists\}$.

$$A \ ::= \ e_1 = e_2 \mid a_1 = a_2 \mid \neg A \mid A_1 \star A_2 \mid \mathcal{Q}i.A \mid \mathcal{Q}x.A$$

We also use the truth T (definable as, for example, $1 = 1$) and the falsity F (which is $\neg\mathsf{T}$). The quantifications induce binding, for which we assume the standard bound name convention. $\mathsf{fn}(A)$ denotes the set of free names in A. A is *well-typed* if whenever a variable/name occurs twice they own the same type and each pair of equated terms have the same type. *We only consider well-typed formulae from now on.* Fixing **primary names** $\mathsf{prim}(A)$ and **auxiliary names** $\mathsf{aux}(A)$ of A which together disjointly cover all names in A with all variables in the latter, we write $\Gamma;;\Theta \vdash A$ if primary (resp. auxiliary) names in A are well-typed under Γ (resp. under Θ). We also write A^{Γ} for A such that $\Gamma;;\Theta \vdash A$ for some Θ.

2.2.2 Model. For defining the interpretation we use the set of *behavioural constants*, generated from the following grammar.

$$\alpha^{!,?} \ ::= \ \wedge_{i\in J}(\vec{\alpha}_i^{?}\beta_i^{\uparrow})^{!} \mid \vee_{i\in J}\,(\vec{\alpha}_i^{!}\beta_i^{\downarrow})^{?} \qquad \alpha^{\downarrow,\uparrow} \ ::= \ \mathtt{in}_i(\vec{\alpha}^{?})^{\downarrow} \mid \mathtt{in}_i(\vec{\alpha}^{!})^{\uparrow}$$

A behavioural constant is *well-formed* if, in $\wedge_{i\in I}(\vec{\alpha}_i\beta_i)^{!}$ and $\vee_{i\in I}(\vec{\alpha}_i\beta_i)^{?}$, $\vec{\alpha}_i \mapsto \beta_i$ defines a total function. A *model* ξ is a finite map from names to well-formed positive behavioural constants (*positive* means their types are positive). We write ξ^{Γ} for ξ with a typing Γ. Given ξ^{Γ}, we say P^{Γ} *defines* ξ, written $P^{\Gamma} \models_{\mathsf{b}} \xi$, when the following inductive conditions hold. $!\xi$ indicates ξ^{Γ} s.t. $\mathsf{md}(\Gamma) = !$. "$\cdot$" is used for "," for the sake of clarity. We treat ξ as if it is a sequence, which does not affect the resulting relation.

1. $P \models_{\mathsf{b}} \xi \cdot x{:}\mathtt{in}_i(\vec{\alpha})^{\uparrow}$ iff $P \stackrel{\overline{x}\mathtt{in}_i(\vec{y})}{\Longrightarrow} P'$ such that $P' \models_{\mathsf{b}} \xi \cdot \vec{y}{:}\vec{\alpha}$.

2. $P \models_{\mathsf{b}} !\xi \cdot x : \wedge_{i\in I}(\vec{\alpha}_i\beta_i)^!$ iff $P \xrightarrow{x(\vec{y}z)} P'$ such that, for each $i \in I$, $R \models_{\mathsf{b}} \vec{y} : \vec{\alpha}_i$ implies $(\boldsymbol{\nu}\, x\vec{y})(P'|R) \models_{\mathsf{b}} \xi/x \cdot z : \beta_i$.

We say ξ is *definable* when $P \models_{\mathsf{b}} \xi$ for some P. α is *definable* when $x : \alpha$ is definable (clearly definability does not depend on the choice of x). *From now on, we only consider definable constants and models.*

We write $\xi_1 \simeq_{\mathsf{b}} \xi_2$ when $\xi_{1,2}$ define the identical set of processes. $\alpha_1 \simeq_{\mathsf{b}} \alpha_2$ iff $x{:}\alpha_1 \simeq_{\mathsf{b}} x{:}\alpha_2$.

2.2.3 Interpretation. Let $\Gamma;;\Theta \vdash A$ and fix an *interpretation* $\mathcal{I}$ of type Θ, which is a well-typed map from $\mathsf{dom}(\Theta)$ to constants. $[\![e]\!]_{\mathcal{I}\cdot\xi}$ is given in the standard manner. With ξ disjoint from I, $[\![a]\!]_{\mathcal{I}\cdot\xi}$ is given by:

- $[\![x]\!]_{\mathcal{I}\cdot\xi} = (I \cdot \xi)(x)$,
- $[\![a \bullet \vec{b}]\!]_{\mathcal{I}\cdot\xi} = \gamma_i$, if $[\![a]\!]_{\mathcal{I}\cdot\xi} : [\![\vec{b_i}]\!] \mapsto \gamma_i$, and
- $[\![\mathtt{in}(a_1..a_n)^\uparrow]\!]_{\mathcal{I}\cdot\xi} = \mathtt{in}_i(\alpha_1..\alpha_n)^\uparrow$ (if $[\![a_i]\!]_{\mathcal{I}\cdot\xi} = \alpha_i$, $1 \le i \le n$).

Then $\xi \models^{\mathcal{I}} A$ is defined by induction on the structure of A, including:

$$\begin{array}{rclcrcl} \xi \models^{\mathcal{I}} e_1 = e_2 & \equiv & [\![e_1]\!]_{\mathcal{I}\cdot\xi} = [\![e_2]\!]_{\mathcal{I}\cdot\xi}, & \qquad & \xi \models^{\mathcal{I}} \forall x^\tau.A & \equiv & \forall \alpha^\tau.\ \xi \models^{I\cdot x:\alpha} A, \\ \xi \models^{\mathcal{I}} a_1 = a_2 & \equiv & [\![a_1]\!]_{\mathcal{I}\cdot\xi} \simeq_{\mathsf{b}} [\![a_2]\!]_{\mathcal{I}\cdot\xi}, & & \xi \models^{\mathcal{I}} \exists x^\tau.A & \equiv & \exists \alpha^\tau.\ \xi \models^{I\cdot x:\alpha} A. \end{array}$$

The rest is the standard interpretation of logical connectives. Note names are treated just like variables in the interpretation of quantifiers (due to stateless nature of linear processes). T is satisfied by all models, while F is never satisfied by any model. The following defines the semantics of assertions. Below $\overline{\Gamma}$ denotes the point-wise dual of Γ, defined from $\overline{\tau}$ given in the obvious way.

Definition 1. (semantics of assertions) Let $\vdash P \triangleright \overline{\Gamma};\Delta$, $\Gamma;;\Theta \vdash A$ and $\Delta;;\Theta \vdash B$. Then we write $P^{\overline{\Gamma};\Delta} \models \mathbf{rely}\ A\ \mathbf{guar}\ B$ when, for each appropriate $\mathcal{I}$ of type Θ,

$$R^\Gamma \models^{\mathcal{I}} A \quad \supset \quad (\boldsymbol{\nu}\, \mathsf{fn}(\Gamma))(P|R) \models^{\mathcal{I}} B,$$

where $P \models^{\mathcal{I}} A$ stands for: $\exists \xi.\, (P \models_{\mathsf{b}} \xi \wedge \xi \models^{\mathcal{I}} A)$.

Given a sequent $P^{\overline{\Gamma};\Delta} \models \mathbf{rely}\ A\ \mathbf{guar}\ B$, the names in $\mathsf{dom}(\Gamma{\cdot}\Delta)$ are its **primary names**, while those in $\mathsf{dom}(\Theta) \cap \mathsf{fn}(A, B)$ are its **auxiliary names**. For brevity we often write $P \models \mathbf{rely}\ A\ \mathbf{guar}\ B$, assuming well-typedness.

2.3 Basic Properties of $\models_{\mathsf{b}}$.

In this and next subsection, we list a few preliminary results which we shall use in the soundness proofs. *Below and henceforth we assume all mentioned processes, sequents, etc. are well-typed.*

Lemma 1. (1) $\equiv, \longrightarrow, \approx$ *are subrelations of* $\cong$.
(2) $!x(\vec{y}).R|P_1|P_2 \cong !x(\vec{y}).R|P_1|(\boldsymbol{\nu}\, x)(!x(\vec{y}).R|P_2)$. *Also* $(\boldsymbol{\nu}\, x)!x(\vec{y}).R \cong \mathbf{0}$.

(3) *If* $\vdash P \triangleright \,!\Gamma\cdot\uparrow!\Delta$, *then* $P \cong Q|R$ *such that* $\vdash Q \triangleright \Gamma$ *and* $\vdash Q \triangleright \Delta$.

(4) *Let* $\vdash P_{1,2} \triangleright\, !\Gamma \cdot x\!:\!\rho^{\uparrow}$ *and* $P_i \stackrel{\overline{x}\mathtt{in}(\vec{y})}{\Longrightarrow} P_i'$. *Then* $P_1 \cong P_2$ *iff* $P_1' \cong P_2'$.

(5) *Let* $\vdash P_{1,2} \triangleright\, !\Gamma \cdot x\!:\!(\vec{\tau}\rho)^{!}$ *and* $P_i \stackrel{x(\vec{y}z)}{\Longrightarrow} P_i'$. *Then* $P_1 \cong P_2$ *iff* $P_1' \cong P_2'$ *iff, for each* $\vdash R \triangleright \vec{y}\!:\!\vec{\tau}$, $(\boldsymbol{\nu}\,\vec{y})(R|P_1) \cong (\boldsymbol{\nu}\,\vec{y})(R|P_2)$.

Proof. (1) and (2) are standard. (3) is by hiding all **?**-typed channels by (2) so that P is transformed into, modulo $\cong$, a process of the form ΠP_i with each P_i having a single (positive) free channel. For (4), we use (3) to reduce the statement to the case when $\Gamma = \emptyset$, so that (via Proposition 1) we have only to show $\overline{x}\mathtt{in}_i(\vec{y})P_1 \cong \overline{x}\mathtt{in}_i(\vec{y})P_2$ iff $P_1 \cong P_2$, which is direct from the congruency of $\cong$. (5) is by congruency and the standard contextual lemma [4]. □

Lemma 2. (1) $P_{1,2}^{\Gamma} \models_{\mathsf{b}} \xi$ *implies* $P_1 \cong P_2$. (2) $P_1^{\Gamma} \models_{\mathsf{b}} \xi$ *and* $P_1 \cong P_2$ *imply* $P_2^{\Gamma} \models_{\mathsf{b}} \xi$. (3) *Let* $\xi_{1,2}$ *be definable. Then* $\xi_1 \simeq_{\mathsf{b}} \xi_2$ *iff* $P_1 \cong P_2$ *for some* $P_{1,2}$ *defining* $\xi_{1,2}$ *respectively iff* $P_1 \cong P_2$ *for any* $P_{1,2}$ *defining* $\xi_{1,2}$ *respectively.*

Proof. (1) and (2) are by induction on the height of Γ. For (1):

Case $\Gamma = \emptyset$. By $P_{1,2} \cong \mathbf{0}$.

Case $\Gamma = \Gamma' \cdot x\!:\!\rho^{\uparrow}$. Let $\xi = \xi' \cdot x\!:\!\mathtt{in}(\vec{\alpha})$. By assumption $P_i \stackrel{\overline{x}\mathtt{in}_i(\vec{y})}{\Longrightarrow} P_i'$ such that $P_i' \models_{\mathsf{b}} \xi' \cdot \vec{y}\!:\!\vec{\alpha}$ for $i = 1, 2$. By induction $P_1' \cong P_2'$ hence done by Lemma 1 (4).

Case $\Gamma = \Gamma' \cdot x\!:\!\rho^{!}$. Let $\xi = \xi' \cdot x\!:\!\wedge_i(\vec{\alpha}_i\beta_i)^{!}$. By assumption $P_i \stackrel{x(\vec{y}z)}{\Longrightarrow} P_i'$ such that, for each j, whenever $R \models_{\mathsf{b}} \vec{y}\!:\!\vec{\alpha}_j$, we have $P_i'' \stackrel{\mathrm{def}}{=} (\boldsymbol{\nu}\,y)(P_i'|R) \models_{\mathsf{b}} z\!:\!\beta_i$. Noting $\vec{\alpha}_j$ range over all behavioural constants of the given types, and because by induction $P_1'' \cong P_2''$, by Lemma 1 (5) we are done.

For (2):

Case $\Gamma = \emptyset$. Because by typing we always have $P_2 \models_{\mathsf{b}} \xi$.

Case $\Gamma = \Gamma' \cdot x : \rho^{\uparrow}$. Let $\xi = \xi' \cdot x : \mathtt{in}(\vec{\alpha})$. By the assumption $P_1 \models_{\mathsf{b}} \xi$, we have $P_1 \stackrel{\overline{x}\mathtt{in}_i(\vec{y})}{\Longrightarrow} P_1'$ such that $P_1' \models_{\mathsf{b}} \xi' \cdot \vec{y}\!:\!\vec{\alpha}$. By the assumption $P_1 \cong P_2$, we have $P_2 \stackrel{\overline{x}\mathtt{in}_i(\vec{y})}{\Longrightarrow} P_2'$ such that $P_1' \cong P_2'$ [because: if not, by Lemma 1 (4) we have $P_1 \ncong P_2$]. By induction $P_2' \models_{\mathsf{b}} \xi', \vec{y}\!:\!\vec{\alpha}$, which by definition means $P_2 \models_{\mathsf{b}} \xi$.

Case $\Gamma = \Gamma' \cdot x\!:\!\rho^{!}$. Let $\xi = \xi' \cdot x\!:\!\wedge_i(\vec{\alpha}_i\beta_i)^{!}$. By assumption $P_1 \stackrel{x(\vec{y}z)}{\Longrightarrow} P_1'$ such that, for each j, whenever $R \models_{\mathsf{b}} \vec{y}\!:\!\vec{\alpha}_j$, we have $(\boldsymbol{\nu}\,y)(P_1'|R) \models_{\mathsf{b}} z\!:\!\beta_i$. By assumption we have $P_2 \stackrel{x(\vec{y}z)}{\Longrightarrow} P_2' \cong P_1'$ for which, by induction, the same condition as P_1' holds, hence by definition $P_1 \models_{\mathsf{b}} \xi$. we are done.

By (1) and (2) above each ξ is inhabited by one and only one congruence class of $\cong$, which immediately implies all logical equivalences in (3). □

Corollary 1. (1) *If* $P_1 \models^{\mathcal{I}} A$ *and* $P_1 \cong P_2$ *then* $P_2 \models^{\mathcal{I}} A$. (2) *If* $P \models^{\mathcal{I}} A$ *and* $P \models_{\mathsf{b}} \xi$ *then* $\xi \models^{\mathcal{I}} A$.

Proof. Immediate from Lemma 2 (2) and (3), respectively. □

Lemma 3. (1) *If* $\mathsf{fn}(\Gamma) \cap \mathsf{fn}(\Delta) = \emptyset$ *then* $P^{\Gamma} | Q^{\Delta} \models_{\mathsf{b}} \xi^{\Gamma} \cdot \xi'^{\Delta}$ *iff* $P \models_{\mathsf{b}} \xi$ *and* $Q \models_{\mathsf{b}} \xi'$. (2) *If* $P \models_{\mathsf{b}} \xi \cdot x{:}\alpha^{!}$ *then* $(\boldsymbol{\nu}\, x)P \models_{\mathsf{b}} \xi$.

Proof. (1) is immediate from Lemma 1 (3) and the definition of $\models_{\mathsf{b}}$. For (2), again by Lemma 1 (3) and (1) above we have $P \cong P' | P_0$ such that $P' \models_{\mathsf{b}} \xi$ and $P_0 \models_{\mathsf{b}} x{:}\alpha$. Since $(\boldsymbol{\nu}\, x)P \cong P'$ we are done. □

Lemma 4. *If* $\alpha_1 \simeq_{\mathsf{b}} \alpha_2$ *and, for* $1 \leq i \leq n$, *we have* $\beta_{1i} \simeq_{\mathsf{b}} \beta_{2i}$, *then* $\alpha_1 \bullet \beta_{11}..\beta_{1n} \simeq_{\mathsf{b}} \alpha_2 \bullet \beta_{2i}..\beta_{2n}$, *where* $\bullet$ *denotes function application.*

Proof. Since, by Lemma 2, both sides are defined by composition of congruent processes. □

Lemma 5. *If* $P \models_{\mathsf{b}} \xi_{1,2}$ *then* $[\![a]\!]_{\mathcal{I}\cdot\xi_1} \simeq_{\mathsf{b}} [\![a]\!]_{\mathcal{I}\cdot\xi_2}$.

Proof. By induction on the structure of a.

1. $a = x$. Immediate from Lemma 2 (3).
2. $a = x \bullet \vec{y}$. From Lemma 4.
3. $\mathtt{in}_i(\vec{y})$. Again immediate from Lemma 2 (3). □

2.4 Basic Properties of $\models^{\mathcal{I}}$.

Below we again assume well-typedness of sequents, formulae, etc.

Lemma 6. *Let* $P \models_{\mathsf{b}} \xi_{1,2}$. *Then* $\xi_1 \models^{\mathcal{I}} A$ *iff* $\xi_2 \models^{\mathcal{I}} A$.

Proof. By induction on the structure of A (the case when A is $e = e'$ is vacuous hence omitted).

Case $\xi_1 \models^{\mathcal{I}} a = a'$. Because $[\![a]\!]_{\mathcal{I}\cdot\xi_2} \simeq_{\mathsf{b}} [\![a]\!]_{\mathcal{I}\cdot\xi_1} \simeq_{\mathsf{b}} [\![a']\!]_{\mathcal{I}\cdot\xi_1} \simeq_{\mathsf{b}} [\![a']\!]_{\mathcal{I}\cdot\xi_2}$, where the first and third equalities are from Lemma 5.

Case $\xi_1 \models^{\mathcal{I}} A \wedge B$, $\xi_1 \models^{\mathcal{I}} A \vee B$, $\xi_1 \models^{\mathcal{I}} \forall i.A$, $\xi_1 \models^{\mathcal{I}} \forall x.A$, $\xi_1 \models^{\mathcal{I}} \exists i.A$, $\xi_1 \models^{\mathcal{I}} \exists x.A$. All immediate.

Case $\xi_1 \models^{\mathcal{I}} \neg A$. Then it is not the case $\xi_1 \models^{\mathcal{I}} A$. But if $\xi_2 \models^{\mathcal{I}} A$ then $\xi_1 \models^{\mathcal{I}} A$ by induction hypothesis, a contradiction. So it is not the case $\xi_2 \models^{\mathcal{I}} A$ either (note we need two directions in this case). □

Lemma 7. *If* $\xi \models^{\mathcal{I}} A$ *and* $A \supset B$ *then* $\xi \models^{\mathcal{I}} B$.

Proof. Since $A \supset B$ means $\xi \models A$ implies $\xi \models B$ for each ξ. □

Corollary 2. *If* $P \models^{\mathcal{I}} A$ *and* $A \supset B$ *then* $P \models^{\mathcal{I}} B$.

Proof. If $P \models^{\mathcal{I}} A$ then $P \models_{\mathsf{b}} \xi$ and $\xi \models^{\mathcal{I}} A$ for some ξ. By Lemma 7 we have $\xi \models^{\mathcal{I}} B$, hence $P \models^{\mathcal{I}} B$, as required. □

Lemma 8. *Let* $\mathsf{dom}(\xi') \cap \mathsf{fn}(A) = \emptyset$. *Then* $\xi \cdot \xi' \models^{\mathcal{I}} A$ *iff* $\xi \models^{\mathcal{I}} A$.

Proof. By induction on the structure of A. Assume the condition. For equations on names in A, ξ' does not matter, so it cannot affect their validity. All other cases are immediate by induction. □

Lemma 9. (monotinicity of $\boldsymbol{\nu}$) *Let* $x \notin \mathsf{fn}(A)$. *Then* $P \models^{\mathcal{I}} A$ *iff* $(\boldsymbol{\nu}\, x)P \models^{\mathcal{I}} A$.

Proof. By Lemma 3 (3) and Lemma 8. □

Lemma 10. (cut in $\models^{\mathcal{I}}$) *Let* $!\Gamma' \subset \Gamma$, $\Gamma_0 \subset \Gamma$ *and* $\Gamma';;\Theta \vdash A$ *such that* $\Theta \vdash I$. *Then* $P^{\overline{\Gamma_0};\Delta} \models \mathbf{rely}\, A_0\, \mathbf{guar}\, B$ *and* $R^{\Gamma} \models^{\mathcal{I}} A \wedge A_0$ *imply* $P|R \models^{\mathcal{I}} A \wedge B$.

Proof. In the second line to the last, note $P|R \cong P'^{\Delta}|R^{\Gamma} \models_{\mathsf{b}} \xi_1^{\Delta} \cdot \xi_2^{\Gamma}$.

$$
\begin{array}{lcll}
R^{\Gamma_0\cdot\Gamma_1} \models^{\mathcal{I}} A \wedge A_0 & \supset & R^{\Gamma_0\cdot\Gamma_1} \models^{\mathcal{I}} A \ \wedge\ R^{\Gamma_0\cdot\Gamma_1} \models^{\mathcal{I}} A_0 & (\wedge) \\
& \supset & R^{\Gamma_0\cdot\Gamma_1} \models^{\mathcal{I}} A \ \wedge\ (\boldsymbol{\nu}\,\mathsf{fn}(\Gamma_1))R \models^{\mathcal{I}} A_0 & (\text{Lem. 9}) \\
& \supset & R^{\Gamma_0\cdot\Gamma_1} \models^{\mathcal{I}} A \ \wedge\ (\boldsymbol{\nu}\,\mathsf{fn}(\Gamma))(P|R) \models^{\mathcal{I}} B & (\text{IH}) \\
& \supset & R^{\Gamma_0\cdot\Gamma_1} \models^{\mathcal{I}} A \ \wedge\ P|R \models^{\mathcal{I}} B & (\text{Lem. 9}) \\
& \supset & P|R \models^{\mathcal{I}} A \ \wedge\ P|R \models^{\mathcal{I}} B & (\text{Lem. 8}) \\
& \supset & P|R \models^{\mathcal{I}} A \wedge B & (\wedge).
\end{array}
$$
□

Lemma 11. *If* $x \notin \mathsf{fn}(A, a)$, *then* $\xi \models^{\mathcal{I}} A[a/x]$ *iff* $\xi \models^{\mathcal{I}} \exists x.(A \wedge x = a)$.

Proof. We show the two inverse implications one by one.

$$
\begin{array}{lcll}
\xi \models^{\mathcal{I}} A[a/x] & \supset & \xi \cdot x : [\![a]\!]_{\mathcal{I}\cdot\xi} \models A & (\text{Lemma 8}) \\
& \supset & \xi \cdot x : [\![a]\!]_{\mathcal{I}\cdot\xi} \models (A \wedge x = a) & (\text{definition of } \wedge) \\
& \supset & \xi \models \exists x.\,(A \wedge x = a) & (\text{definition of } \exists).
\end{array}
$$

For the other direction, we reason:

$$
\begin{array}{lcll}
\xi \models \exists x.\,(A \wedge x = a) & \supset & \xi \cdot x : \beta \models (A \wedge x = a) & (\exists) \\
& \supset & \xi \cdot x \mapsto [\![a]\!]_{\mathcal{I}\cdot\xi} \models (A \wedge x = a) & (\wedge) \\
& \supset & \xi \cdot x \mapsto [\![a]\!]_{\mathcal{I}\cdot\xi} \models A & (\wedge).
\end{array}
$$
□

Lemma 12. $\xi \cdot x : \mathtt{in}_i(\vec{\alpha})^{\uparrow} \models^{\mathcal{I}} A$ *iff* $\xi \cdot \vec{y} : \vec{\alpha} \models^{\mathcal{I}} A[\mathtt{in}_i(\vec{y})^{\uparrow}/x]$.

Remark 1. In the statement above, $\vec{y}$ are introduced into the right-hand sequent without being mentioned in the preceding sequent. In such cases, we always assume newly introduced names are freshly chosen, e.g. $(\mathsf{fn}(A) \cup \{x\}) \cap \{\vec{y}\} = \emptyset$ in the statement above.

Proof. Because:

$$
\begin{array}{lcl}
\Gamma \cdot x : \mathtt{in}_i(\vec{\alpha})^{\uparrow} \models^{\mathcal{I}} A & \equiv & \Gamma \cdot x : \mathtt{in}_i(\vec{\alpha})^{\uparrow}, \vec{y} : \vec{\alpha} \models^{\mathcal{I}} (A \wedge x = \mathtt{in}_i(\vec{\alpha})^{\uparrow}) \\
& \equiv & \Gamma \cdot \vec{y} : \vec{\alpha} \models^{\mathcal{I}} \exists x.(A \wedge x = \mathtt{in}_i(\vec{y})^{\uparrow}) \\
& \equiv & \Gamma \cdot \vec{y} : \vec{\alpha} \models^{\mathcal{I}} A[\mathtt{in}_i(\vec{y})^{\uparrow}/x].
\end{array}
$$
□

Lemma 13. *Let* $\mathsf{fn}(B) \cap \{x\vec{y}\} = \emptyset$. *Then* $\xi \cdot x : \wedge_{i\in J}(\vec{\alpha}_i\beta_i)^{!} \cdot \vec{y} : \vec{\alpha}_i \models^{\mathcal{I}} B[x \bullet \vec{y}/z]$ *iff* $\xi \cdot z : \beta_i \models^{\mathcal{I}} B$.

Proof. Letting $\theta = \wedge_{i\in J}(\vec{\alpha}_i\beta_i)^!$, we have:

$$\begin{array}{lll}
\xi\cdot x{:}\theta\cdot\vec{y}{:}\vec{\alpha}_i \models^{\mathcal{I}} B[x\bullet\vec{y}/z] & & \\
\quad\equiv\quad \xi\cdot x{:}\theta\cdot\vec{y}{:}\vec{\alpha}_i \models^{\mathcal{I}} \exists z.\,(B\wedge z = x\bullet\vec{y}) & \text{(Lemma 11)} \\
\quad\equiv\quad x{:}\theta\cdot\vec{y}{:}\vec{\alpha}_i\cdot z{:}\beta' \models^{\mathcal{I}} B\wedge z = x\bullet\vec{y} & (\exists) \\
\quad\equiv\quad \xi\cdot x{:}\theta\cdot\vec{y}{:}\vec{\alpha}_i\cdot z{:}\beta_i \models^{\mathcal{I}} B & (\wedge) \\
\quad\equiv\quad \xi\cdot z{:}\beta_i \models^{\mathcal{I}} B, & \text{(Lemma 8)}
\end{array}$$

as required. □

2.5 Soundness

We list the proof rules for the specification logic in Figure 2. The sequent has the form $P^{\Gamma;\Delta} \vdash \mathbf{rely}\, A\, \mathbf{guar}\, B$, which is *well-typed* when (in addition to the well-typedness of $P^{\Gamma;\Delta}$), we have $\overline{\Gamma}\cdot\Theta \vdash A$ and $\Delta\cdot\Theta \vdash B$ for some Θ. For legibility we often omit the action type, writing $P \vdash \mathbf{rely}\, A\, \mathbf{guar}\, B$. While most rules in Figure 2 omit the type annotations in this way, the derived sequents are always well-typed, assuming, in each rule, all sequents are well-typed, as well as the standard bound name convention (over both processes and formulae). In the rules, $B^{\text{-}\vec{x}}$ means $\{\vec{x}\}\cap \mathsf{fn}(B) = \emptyset$. $A[a/x]$ is the result of syntactically substituting a for x in A. $\vec{l}$ indicate auxiliary names. For detailed illustration of each rule, see [7]. The aim of this section is to prove:

Theorem 1. *The proof system of the linear process logic is sound, that is:* $P^{\Gamma;\Delta}\vdash \mathbf{rely}\, A\, \mathbf{guar}\, B$ *implies* $P^{\Gamma;\Delta}\models \mathbf{rely}\, A\, \mathbf{guar}\, B$ *for each linearly typed* P.

Remark. The corresponding soundness result in the main exposition is stated with respect to a model which is directly constructed from typed processes modulo $\cong$ (for which the properties of satisfaction as presented in the previous subsections are direct from its definition). Since the validity of formulae with respect to these two models easily coincide, no difference comes about (in fact exactly the same proof is applicable for the alternative model, line by line).

Proof. We show, for each rule, that if its antecedent is valid semantically, then its conclusion is also valid semantically. As before, we always assume well-typedness of processes, formulae and sequents. We often ignore neutral types ($\updownarrow$) since they are insignificant both logically and semantically. The first rule is for inaction:

$$(\mathsf{Zero})\ \frac{-}{\mathbf{0}^{\emptyset} \vdash \mathbf{rely}\, A\, \mathbf{guar}\, A}$$

Let $\Theta \vdash A$ and $\mathcal{I}$ has type Θ (by typing A and Θ only contain variables). Then

$$\begin{array}{lll}
R^{\emptyset} \models^{\mathcal{I}} A & \supset & \mathbf{0}|R \cong R \models^{\mathcal{I}} A \\
& \supset & \mathbf{0}|R \models^{\mathcal{I}} A, \qquad \text{(Corollary 1)},
\end{array}$$

as required. The second rule is:

$$(\mathsf{Res})\ \frac{P \vdash \mathbf{rely}\, A\, \mathbf{guar}\, B^{\text{-}x}}{(\boldsymbol{\nu}\, x)P \vdash \mathbf{rely}\, A\, \mathbf{guar}\, B}$$

$$\frac{}{\mathbf{0}^{\emptyset} \vdash \mathbf{rely}\ A\ \mathbf{guar}\ A}\ (\mathsf{Zero})$$

$$\frac{P_1 \vdash \mathbf{rely}\ A_1\ \mathbf{guar}\ B_1 \wedge E \quad P_2 \vdash \mathbf{rely}\ A_2 \wedge E\ \mathbf{guar}\ B_2}{P_1|P_2 \vdash \mathbf{rely}\ A_1 \wedge A_2\ \mathbf{guar}\ B_1 \wedge B_2}\ (\mathsf{Par})$$

$$\frac{P^{\Gamma,x:\tau} \vdash \mathbf{rely}\ A\ \mathbf{guar}\ B^{\text{-}x}}{(\boldsymbol{\nu}\, x)P^{\Gamma} \vdash \mathbf{rely}\ A\ \mathbf{guar}\ B}\ (\mathsf{Res})$$

$$\frac{P^{\Gamma} \vdash \mathbf{rely}\ A\ \mathbf{guar}\ B}{P^{\Gamma,x:\tau} \vdash \mathbf{rely}\ A\ \mathbf{guar}\ B}\ (\mathsf{Weak})$$

$$\frac{\forall i.\ P_i \vdash \mathbf{rely}\ A[\mathtt{in}_i(\vec{y}_i)^{\uparrow}/x]\ \mathbf{guar}\ B}{x[\&_i(\vec{y}_i).P_i] \vdash \mathbf{rely}\ A\ \mathbf{guar}\ B}\ (\mathsf{Bra}^{\downarrow})$$

$$\frac{P \vdash \mathbf{rely}\ A\ \mathbf{guar}\ B[\mathtt{in}_i(\vec{y})^{\uparrow}/x]}{\overline{x}\mathtt{in}_i(\vec{y})P \vdash \mathbf{rely}\ A\ \mathbf{guar}\ B}\ (\mathsf{Sel}^{\uparrow})$$

$$\frac{P \vdash \mathbf{rely}\ A^{\text{-}\vec{y}} \wedge B_1^{\vec{y}}\ \mathbf{guar}\ B_2}{!x(\vec{y}z).P \vdash \mathbf{rely}\ A\ \mathbf{guar}\ B}\ (\mathsf{In}^{!})\quad (\forall \vec{y}.\,(B_1 \supset B_2[x\bullet\vec{y}/z]) \supset B)$$

$$\frac{P \vdash \mathbf{rely}\ A^{\text{-}z} \wedge A_2^{z}\ \mathbf{guar}\ A_1^{\vec{y}} \wedge B^{\text{-}\vec{y}}}{\overline{x}(\vec{y}z)P \vdash \mathbf{rely}\ A\ \mathbf{guar}\ B}\ (\mathsf{Out}^{?})\quad (A \supset A_1 \supset A_2[x\bullet\vec{y}/z])$$

Fig. 2. Proof Rules for Linear Processes

Assume $P^{\overline{\Gamma};\Delta\cdot x:\tau}$, A^{Γ} and B^{Δ} with $\mathsf{md}(\tau) = !$. Fixing an appropriate $\mathcal{I}$:

$$\begin{array}{lll} R^{\Gamma} \models^{\mathcal{I}} A & \supset & (\boldsymbol{\nu}\,\mathsf{fn}(\Gamma))(P|R) \models_{\mathsf{b}} \xi \cdot x{:}\alpha \models^{\mathcal{I}} B \quad \text{(IH)} \\ & \supset & (\boldsymbol{\nu}\, x)(\boldsymbol{\nu}\,\mathsf{fn}(\Gamma))(P|R) \models_{\mathsf{b}} \xi \models^{\mathcal{I}} B \quad \text{(Lemma 9)}, \end{array}$$

as required. The case when the mode is $\updownarrow$ is direct from (IH).

The (Par) rule is closely related with the cut rule in the sequent calculus.

$$(\mathsf{Par})\ \frac{P_1 \vdash \mathbf{rely}\ A_1\ \mathbf{guar}\ B_1 \wedge E \quad P_2 \vdash \mathbf{rely}\ A_2 \wedge E\ \mathbf{guar}\ B_2}{P_1|P_2 \vdash \mathbf{rely}\ A_1 \wedge A_2\ \mathbf{guar}\ B_1 \wedge B_2}$$

Assume $P_1^{\overline{\Gamma_1};\Delta_1}$ and $P_2^{\overline{\Gamma_2};\Delta_2}$. We have:

$$\begin{array}{lll} R^{\Gamma} \models^{\mathcal{I}} A_1 \wedge A_2 & \supset & P_1|R \models^{\mathcal{I}} A_2 \wedge (B_1 \wedge E) \quad \text{(IH, Lem. 10)} \\ & \supset & P_1|P_2|R \models^{\mathcal{I}} B_1 \wedge B_2 \quad \text{(IH, Lem. 10)} \\ & \supset & (\boldsymbol{\nu}\,\mathsf{fn}(\Gamma_1 \cup \Gamma_2))(P_1|P_2|R) \models^{\mathcal{I}} B_1 \wedge B_2 \quad \text{(Lem. 9)}, \end{array}$$

as required (in the first two lines, we used associativity of $\wedge$, which is obvious from the definition of $\models^{\mathcal{I}}$).

Recall a name occurring in a formula in the statement is *auxiliary* if it does not occur in the action type of the concerned process. The weakening rule needs some care in the treatment of auxiliary names.

$$(\mathsf{Weak})\ \frac{P^{\Gamma} \vdash \mathbf{rely}\ A\ \mathbf{guar}\ B}{P^{\Gamma\cdot x:\tau} \vdash \mathbf{rely}\ A\ \mathbf{guar}\ B}$$

Let $\mathsf{md}(\tau) = \mathbf{?}$, in which case x is auxiliary in the antecedent but is primary in the conclusion. We first set, without loss of generality by Corollary 1:

$$R^{\Gamma\cdot x:\overline{\tau}} \stackrel{\text{def}}{=} {R'}^{\Gamma}|R_0^{x:\tau} \qquad \text{such that } R' \models_{\mathsf{b}} \xi \text{ and } R_0 \models_{\mathsf{b}} x:\alpha.$$

Note $(\boldsymbol{\nu}\,\mathsf{fn}(\Gamma))(P|R') \cong (\boldsymbol{\nu}\,\mathsf{fn}(\Gamma), x)(P|R)$ and $x \notin \mathsf{fn}(B)$ (the latter by typing).

$$\begin{array}{lcll} R \models^{\mathcal{I}} A & \supset & R \models_{\mathsf{b}} \xi\cdot x{:}\alpha \models^{\mathcal{I}} A & \text{(Corollary 1)} \\ & \supset & R' \models_{\mathsf{b}} \xi \models^{I\cdot x:\alpha} A & \text{(Definition of } \models^{\mathcal{I}}\text{)} \\ & \supset & (\boldsymbol{\nu}\,\mathsf{fn}(\Gamma))(P|R') \models^{I\cdot x:\alpha} B & \text{(IH)} \\ & \supset & (\boldsymbol{\nu}\,\mathsf{fn}(\Gamma))(P|R')|(\boldsymbol{\nu}\,x)R_0 \models^{I\cdot x:\alpha} B & \text{(Corollary 1)} \\ & \supset & (\boldsymbol{\nu}\,\mathsf{fn}(\Gamma), x)(P|R) \models^{\mathcal{I}} B, & \text{(Lemma 8),} \end{array}$$

as required. The case when $\tau = \updownarrow$ is direct from (IH).

We move to the prefix rules. The first is the linear input.

$$(\mathsf{Bra}^{\downarrow})\ \frac{\forall i.\ P_i \vdash \mathbf{rely}\ A[\mathtt{in}_i(\vec{y}_i)^{\uparrow}/x]\ \mathbf{guar}\ B}{x[\&_i(\vec{y}_i).P_i] \vdash \mathbf{rely}\ A\,\mathbf{guar}\ B}$$

Note $\{\vec{y}\}\cap\mathsf{fn}(A) = \emptyset$ by the bound name convention. Below let $R^{\Gamma} \equiv R'|\overline{x}\mathtt{in}_i(\vec{y})R_0$ with $\mathsf{dom}(\Gamma) = \{\vec{w}x\}$.

$$\begin{array}{lcll} R \models_{\mathsf{b}} \vec{w}{:}\vec{\gamma}\cdot x{:}\mathtt{in}_i(\vec{\alpha})^{\uparrow} \models^{\mathcal{I}} A & & & \\ & \supset & R'|R_0 \models_{\mathsf{b}} \vec{w}{:}\vec{\gamma}\cdot\vec{y}{:}\vec{\alpha} \models^{\mathcal{I}} A[\mathtt{in}_i(\vec{y})^{\uparrow}/x] & \text{(Def. of } \models_{\mathsf{b}}\text{, Lemma 12)} \\ & \supset & (\boldsymbol{\nu}\,\vec{y}\vec{w})(P_i|R'|R_0) \models^{\mathcal{I}} B & \text{(IH)} \\ & \supset & (\boldsymbol{\nu}\,\vec{w}x)(x[\&(\vec{y}_i).P_i]|R'|\overline{x}\mathtt{in}_i(\vec{y})R_0) \models_{\mathsf{b}} B & \text{(Corollary 1),} \end{array}$$

as required. In the last step we used $x[\&(\vec{y}_i).P_i]|\overline{x}\mathtt{in}_i(\vec{y})R_0 \longrightarrow (\boldsymbol{\nu}\,\vec{y})(P_i|R_0)$.

For the linear output:

$$(\mathsf{Sel}^{\uparrow})\ \frac{P \vdash \mathbf{rely}\ A\ \mathbf{guar}\ B[\mathtt{in}_i(\vec{y})^{\uparrow}/x]}{\overline{x}\mathtt{in}_i(\vec{y})P \vdash \mathbf{rely}\ A\ \mathbf{guar}\ B}$$

The reasoning is precisely dual to that for (Bra):

$$\begin{array}{lcll} R \models_{\mathsf{b}} \vec{w}{:}\vec{\gamma} \models^{\mathcal{I}} A & & & \\ & \supset & (\boldsymbol{\nu}\,\vec{w})(P|R) \models_{\mathsf{b}} \vec{y}{:}\vec{\alpha} \models^{\mathcal{I}} B[\mathtt{in}_i(\vec{y})^{\uparrow}/x] & \text{(IH)} \\ & \supset & \overline{x}\mathtt{in}_i(\vec{y})(\boldsymbol{\nu}\,\vec{w})(P|R) \models_{\mathsf{b}} x{:}\mathtt{in}_i(\vec{\alpha}) \models^{\mathcal{I}} B & \text{(Def. of } \models_{\mathsf{b}}\text{, Lemma 12)} \\ & \supset & (\boldsymbol{\nu}\,\vec{w}x)(\overline{x}\mathtt{in}_i(\vec{y})R_0|R) \models^{\mathcal{I}} B & \text{(Corollary 1),} \end{array}$$

as required. In the final step we used $\overline{x}\mathtt{in}_i(\vec{y}(\boldsymbol{\nu}\,\vec{w})(P|R) \cong (\boldsymbol{\nu}\,\vec{w}x)(\overline{x}\mathtt{in}_i(\vec{y})R_0|R)$.

We next prove the rule for replication and its dual.

$$(\mathsf{In}^{!})\ \frac{P \vdash \mathbf{rely}\ A^{\vec{y}\vec{l}} \wedge B_1^{\vec{y}\vec{l}}\ \mathbf{guar}\ B_2\ ,\ \ \forall\vec{y}\vec{l}.(B_1 \supset B_2[x\bullet\vec{y}/z]) \supset B}{!x(\vec{y}z).P \vdash \mathbf{rely}\ A\ \mathbf{guar}\ B}$$

Let $R^{\Gamma} \models^{\mathcal{I}} A$ with $\mathsf{dom}(\Gamma) = \{\vec{w}\}$. We first construct the behavioural constant $\theta \stackrel{\text{def}}{=} \wedge_i(\vec{\alpha}_i\beta_i)^{!}$ as follows (assume the type of x be $(\vec{\tau}\rho)^{!}$):

($\star$) For each $\vec{\alpha}_i^{\vec{i}}$, we choose $S_i \models_{\mathsf{b}} \vec{y}{:}\vec{\alpha}$ and let $(\boldsymbol{\nu}\, \vec{w}\vec{y})(P|R|S_i) \models_{\mathsf{b}} z{:}\beta_i$.

For this θ we show $(\vec{w})(!x(\vec{y}z).P|R) \models_{\mathsf{b}} x:\theta \models^{\mathcal{I}} B$. The reasoning is decomposed into the part for $\models_{\mathsf{b}}$ and the part for $\models^{\mathcal{I}}$.

(1) $(\boldsymbol{\nu}\, \vec{w})(!x(\vec{y}z).P|R) \models_{\mathsf{b}} x{:}\theta$**.** Since

$$(\boldsymbol{\nu}\, \vec{w})(!x(\vec{y}z).P|R) \xrightarrow{x(\vec{y}z)} (\boldsymbol{\nu}\, \vec{w})(!x(\vec{y}z).P|R)|(\boldsymbol{\nu}\, \vec{w})(P|R).$$

it suffices to show, by the definition of $\models_{\mathsf{b}}$, that, for each $S \models_{\mathsf{b}} \vec{y}{:}\vec{\alpha}_i$:

$$(\boldsymbol{\nu}\, \vec{y})((\boldsymbol{\nu}\, \vec{w})(P|R) \mid S) \equiv (\boldsymbol{\nu}\, \vec{y}\vec{w})(P|R|S) \models_{\mathsf{b}} z{:}\beta_i$$

Below S_i is the process used for constructing θ in ($\star$) above.

$$\begin{array}{lll} S \models_{\mathsf{b}} \vec{y}{:}\vec{\alpha}_i & & \\ \quad \supset & S_i \cong S & (\star, \text{Lemma 2 (1)}) \\ \quad \supset & (\boldsymbol{\nu}\, \vec{y}\vec{w})(P|R|S) \cong (\boldsymbol{\nu}\, \vec{y}\vec{w})(P|R|S_i) \models_{\mathsf{b}} z{:}\beta_i & (\text{congruency}, \star) \\ \quad \supset & (\boldsymbol{\nu}\, \vec{y}\vec{w})(P|R|S) \models_{\mathsf{b}} z{:}\beta_i & (\text{Corollary 1}). \quad \Box \end{array}$$

(2) $x:\theta \models^{\mathcal{I}} B$. Since $\forall \vec{y}\vec{l}.(B_1 \supset B_2[x \bullet \vec{y}/z]) \supset B$ by assumption, it suffices to show $x:\theta \models^{\mathcal{I}} \forall \vec{y}\vec{l}.(B_1 \supset B_2[x \bullet \vec{y}/z])$ (by Lemma 7). Below we let J be an arbitrary well-typed assignment to $\vec{l}$. As before, S_i is the process used in $\star$.

$$\begin{array}{lll} x{:}\theta \cdot \vec{y}{:}\vec{\alpha}_i \cdot J \models^{\mathcal{I}} B_1 & & \\ \quad \equiv & \vec{y}{:}\vec{\alpha}_i \models^{I\cdot J} B_1 & (\text{Lemma 8}) \\ \quad \supset & (\boldsymbol{\nu}\, \vec{w}\vec{y})(P|S_i|R) \models^{I\cdot J} B_2,\ (\boldsymbol{\nu}\, \vec{w}\vec{y})(P|S_i|R) \models_{\mathsf{b}} z{:}\beta_i & (\text{IH}, \star) \\ \quad \supset & z{:}\beta_i \models^{I\cdot J} B_2 & (\text{Corollary 1 (2)}) \\ \quad \supset & x{:}\theta \cdot \vec{y}{:}\vec{\alpha}_i \cdot J \models^{\mathcal{I}} B_2[x \bullet \vec{y}/z] & (\text{Lemma 13}), \end{array}$$

as required.

The final prefix rule is a replicated output.

$$(\mathsf{Out}^{?})\ \frac{P \vdash \mathbf{rely}\ A^{\text{-}z} \wedge A_2^z\ \mathbf{guar}\ A_1^{\vec{y}} \wedge B^{\text{-}\vec{y}} \qquad A \supset A_1 \supset A_2[x \bullet \vec{y}/z]}{\overline{x}(\vec{y}z)P \vdash \mathbf{rely}\ A\ \mathbf{guar}\ B}$$

The rule above is equivalent to the following rule up to $\cong$ [because: using Lemma 1 (2), we can decompose P above into two terms of the required shape, but the formulae in the antecedent are completely determined by types, thus, via Lemma 1 (3), we know each term satisfies corresponding formulae].

$$(\mathsf{Out}^{?}_{\mathsf{dec}})\ \frac{S \vdash \mathbf{rely}\ A\, \mathbf{guar}\, A_1^{\vec{y}}, \quad P \vdash \mathbf{rely}\ A^{\text{-}z} \wedge A_2^z\ \mathbf{guar}\ B, \quad A \supset A_1 \supset A_2[x \bullet \vec{y}/z]}{\overline{x}(\vec{y}z)(S|P) \vdash \mathbf{rely}\ A\ \mathbf{guar}\ B}$$

Since the validity does not change up to $\cong$, we reason using this decomposed version of the rule, which gives a clearer articulation. We first fix names as follows: $\vdash S \triangleright \overline{\Gamma}, \Delta$ and $\vdash P \triangleright \overline{\Gamma}, z{:}\rho^{\downarrow}, u{:}\sigma^{\uparrow}$ where $\mathsf{dom}(\Gamma) = \{\vec{w}\}$ and $\mathsf{dom}(\Delta) = \{\vec{y}\}$. Also note, by the binder convention, we have $\mathsf{fn}(\{\vec{y}z\}) \wedge \mathsf{fn}(A, B) = \emptyset$.

Assume $\vdash R \triangleright \Gamma$ such that, by Lemma 1 (3), $R \equiv (!x(\vec{y}z).R_0) \,|\, R'$ where $!x(\vec{y}z).R_0 \models_{\mathsf{b}} x:\theta$ with $\theta = \wedge_i(\vec{\alpha}_i\beta_i)^!$ whose $\vec{\alpha}_i$ ranges over all constants of the appropriate type. We can now reason as follows. Below we let $\xi \stackrel{\text{def}}{=} x:\theta \cdot \vec{w}:\vec{\gamma}$ and write $\ddagger$ for the condition $A \supset A_1 \supset A_2[x \bullet \vec{y}/z]$.

$$
\begin{array}{lll}
R \models_{\mathsf{b}} x:\theta \cdot \vec{w}:\vec{\gamma} \models^{\mathcal{I}} A & & \\
\quad \supset & S|R \models_{\mathsf{b}} \xi \cdot \vec{y}:\vec{\alpha} \models^{\mathcal{I}} A \wedge A_1 & \text{(IH, Corollary 1)} \\
\quad \supset & S|R \models_{\mathsf{b}} \xi \cdot \vec{y}:\vec{\alpha}_i \models^{\mathcal{I}} A \wedge A_2[x \bullet \vec{y}/z] & (\ddagger\text{, Corollary 2}) \\
\quad \supset & S|R|R_0 \models_{\mathsf{b}} \xi \cdot \vec{y}:\vec{\alpha}_i \cdot z:\beta_i, & \text{(Lemma 3 (1))} \\
 & \xi \cdot \vec{y}:\vec{\alpha}_i \cdot z:\beta_i \models_{\mathsf{b}} A \wedge A_2 \wedge x \bullet \vec{y} = z & \text{(Lemma 13)} \\
\quad \supset & (\boldsymbol{\nu}\,\vec{y})(S|R|R_0) \models_{\mathsf{b}} \xi \cdot z:\beta_i \models^{\mathcal{I}} A \wedge A_2 & \text{(Cor. 2, Lem. 3 (2))} \\
\quad \supset & (\boldsymbol{\nu}\,\vec{w}xz)(P|(\boldsymbol{\nu}\,\vec{y})(S|R|R_0)) \models^{\mathcal{I}} B & \text{(IH)} \\
\quad \supset & (\boldsymbol{\nu}\,\vec{w}x)(\overline{x}(\vec{y}z)(S|P) \,|\, R) \models^{\mathcal{I}} B & \text{(Corollary 1).}
\end{array}
$$

The last step used: $(\boldsymbol{\nu}\,\vec{w}xz)(\overline{x}(\vec{y}z)(S|P) \,|\, R) \longrightarrow (\boldsymbol{\nu}\,\vec{w}\vec{y}xz)(S|P|R|R_0)$.

Finally, the soundness of

$$
(\mathsf{Consequence})\ \frac{P \vdash \mathbf{rely}\ A'\ \mathbf{guar}\ B' \quad A \supset A' \quad B' \supset B}{P \vdash \mathbf{rely}\ A\ \mathbf{guar}\ B}
$$

is proved as follows. Let $\vdash P \triangleright \overline{\Gamma}, \Delta$ with $\mathsf{dom}(\Gamma) = \vec{w}$.

$$
\begin{array}{lll}
R \models^{\mathcal{I}} A \quad \supset & R \models^{\mathcal{I}} A' & \text{(Corollary 2)} \\
\quad \supset & (\boldsymbol{\nu}\,\vec{w})(P|R) \models^{\mathcal{I}} B' & \text{(IH)} \\
\quad \supset & (\boldsymbol{\nu}\,\vec{w})(P|R) \models^{\mathcal{I}} B & \text{(Corollary 2),}
\end{array}
$$

as required. We have now exhausted all rules. □

$$\frac{\vdash P \triangleright \Gamma, y{:}\overline{\tau}^{?}\ ;\ x{:}\tau^{!}}{\vdash \mu y{=}x.P \triangleright \Gamma, x{:}\tau}\ (\mathsf{Rec})$$

Fig. 3. Typing Rule for Recursion

3 Total Logic for Affine Sequential Processes

3.1 Affine Sequential Typing

In this section we consider sequential processes which add *recursion* to processes, hence nontermination. While this effect is easily obtained by allowing circular composition in (Par) in Figure 1, here we take an incremental approach, adding the rule for a single circular composition in Figure 3. $\mu y{=}x.P$ is best understood as $(\boldsymbol{\nu}\, y)(P|![y \to x]^{\tau})$ using the standard copy-cat (see Appendix) with τ being the type of x, any request to y would be forwarded to x. All sequential affine processes generated possibly using circular composition are representable when we regard recursion $\mu y = x.P$ as substitution $P[x/y]$ (to be precise we also need a recursion for a circular composition at linear channels: since, however, the resulting processes are realisable without such composition up to strong bisimilarity, we ignore such circularity).

To substantiate this point, we need the operational semantics of recursion, which we present below first as transition and second as reduction.

$$\frac{P \xrightarrow{\overline{y}(\vec{w})x(\vec{w}')} \xrightarrow{} P'}{\mu y{=}x.P \xrightarrow{\tau} \mu y{=}x.(\boldsymbol{\nu}\,\vec{w})(P'[\vec{w}/\vec{w}'])}\ (\mathsf{Rec\text{-}Unfold}) \qquad \frac{P \xrightarrow{l} P' \quad y \notin \mathsf{fn}(l)}{\mu y{=}x.P \xrightarrow{l} \mu y{=}x.P'}\ (\mathsf{Rec\text{-}Comp})$$

For reduction, we first add the structural rule.

$$\mu y{=}x.(P|Q) \equiv (\mu y{=}x.P)|Q \qquad (x \notin \mathsf{fn}(P), y \notin \mathsf{fn}(Q)),$$

Then we add the following reduction rule.

$$\mu y{=}x.(!x(\vec{w}z).P|\overline{y}(\vec{w}z)Q|R) \longrightarrow \mu y{=}x.(!x(\vec{w}z).P|(\boldsymbol{\nu}\, y\vec{w}z)(P|Q)|R)$$

In either presentation, we can check that the semantics is precisely the same as having $P[x/y]$ instead: thus, semantically speaking, it neither adds nor takes off anything to/from the original affine system. The significance of having the new construct is then solely for having the proof system which is more modular and with better articulation. We also note, since $(\mu y{=}x.!x(c).\overline{y}(e)e.\overline{c}) \mid !\overline{x}(c)c.\overline{w}$ (which immediately diverges) is typable, Proposition 1 no longer holds.

3.2 Affine Process Logic

We use the same terms and formulae as before for our logic. To incorporate possibly divergent processes in the logic, we need a couple of refinement.

1. To behavioural constants, we add ω (to be precise, ω^τ for each $\tau^\uparrow$), which indicates the divergence. The well-formedness of (co-)replicated behaviours is refined so that "total functions" become "total continuous functions" (regarding ω as the bottom element).
2. $\models_{\mathsf{b}}$ in §2 is extended by adding the clause: $P \models_{\mathsf{b}} !\xi, x:\omega$ iff $P \Uparrow$. We say ξ is *total* if $\omega \notin \mathsf{ran}(\xi)$ ($\mathsf{ran}(\xi)$ denotes the range of ξ).
3. $\xi \models^{\mathcal{I}} A$ is given by the same condition as before assuming ξ and $\mathcal{I}$ are total.

Other than these changes, all definitions in Section 2.2 remain the same, leading to the satisfaction relation which we again write $P^{\Gamma;\Delta} \models \mathbf{rely}\, A\, \mathbf{guar}\, B$. By the condition 3 above, assertions in this logic are about **total correctness** in the standard sense, that is $P \models^{\mathcal{I}} \mathbf{rely}\, A\, \mathbf{guar}\, B$ holds only when (assuming A is satisfiable) $P|R$ converges for all $R \models^{\mathcal{I}} A$. We call this logic the *basic affine process logic*, which enjoys a smooth transition from the linear process logic in the previous section (alternatively we may take off the restriction of totality of ξ in $\xi \models^{\mathcal{I}} A$, resulting in a more expressive logic, which is briefly discussed at the end of this section). We observe:

Lemma 14. *All lemmas and corollaries of Sections 2.3 and 2.4 hold in the affine logic, except for: (i) changing Lemma 1 (3) into "If* $\vdash P \triangleright !\Gamma, \uparrow!\Delta$ *and* $P \Downarrow$ *then* $P \cong Q|R$ *such that* $\vdash Q \triangleright \Gamma$ *and* $\vdash R \triangleright \Delta$*", (ii) adding the condition "*$\omega \notin \mathsf{ran}(\xi \cdot \xi')$*" in Lemma 3 (1), and (iii) adding the assumption "*$[\![a]\!]_{\mathcal{I}\cdot\xi} \neq \omega$*" in (the both sides of the implications of) Lemma 11.*

Proof. The first part of the refined clause of Lemma 1 is standard, while the second part is by the replication theorem. The rest of Lemma 1 is proved exactly as before, line by line. For Lemma 3 (1), the condition is needed since if $P \Uparrow$ $P \models_{\mathsf{b}} x{:}\omega \cdot \xi$ for any (well-typed) ξ by definition.a For Lemma 2, the proof of (1) is the same as before, except when $\xi = \xi' \cdot x{:}\omega$ in which case we use the refined clause of Lemma 1 (3) noted above. Similarly for (2). (3) is from (1) and (2). The proof of Lemma 11 is precisely as before by using $[\![a]\!]_{\mathcal{I}\cdot\xi} \neq \omega$. The proofs of Lemma 4, Lemma 5, Lemma 6, Lemma 7, Corollary 2, Lemma 8 (noting $\xi \cdot \xi'$ is total hence ξ' does not use ω: if it does use ω, the statement does not hold), Lemma 10, Corollary 2 and Lemma 13 are precisely as before. □

The following property holds for all logics we shall consider in this paper.

Lemma 15. *Let* $\boldsymbol{\sigma}$ *be an arbitrary permutation (injective map) on names. Then* $P \models_{\mathsf{b}} \xi \models^{\mathcal{I}} A$ *iff* $P\boldsymbol{\sigma} \models_{\mathsf{b}} \xi\boldsymbol{\sigma} \models^{\mathcal{I}\sigma} A\boldsymbol{\sigma}$.

Proof. We omit mechanical and uninteresting proofs. Generally this holds since, in each definition, names occur via their metavariables. So a particular choice of names never matters. □

We also need a couple of properties of recursion.

Lemma 16. *Let* $\vdash \mu y = x.P \triangleright x : \tau, ?\Gamma, \Uparrow !\updownarrow \Delta$ *with* $\mathsf{dom}(\Gamma) = \{\vec{w}\}$. *Then* $P \cong (!x(\vec{w}v).P_0)|R$ *such that* $\mathsf{fn}(!x(\vec{w}v).P_0) \subset \{xy\vec{w}\}$. *Further in this case we have* $\mu y{=}x.P \cong \mu y{=}x.P_0|R$.

Proof. The first statement is by folding other names by the replication theorem. The second statement is by $\mu y{=}x.(!x(\vec{w}v).P_0)|R \sim \mu y{=}x.!x(\vec{w}v).P_0 \mid R$ where $\sim$ is the strong bisimilarity. □

Lemma 17. *Let* $\vdash P \triangleright ?\Gamma, y{:}\overline{\tau}^?, x{:}\tau^!$. *Then* $\mu y{=}x.P \cong (\boldsymbol{\nu}\, y)(P|(\mu y{=}x.P)[y/x])$.

Proof. This is the standard unfolding. We first enlarge typable terms to the standard sequential affine calculus, where $\mu y{=}x.P \cong (\boldsymbol{\nu}\, y)(P|![y \to x])$ by the correspondence in transition. Using this and standard properties of copycat:

$$\begin{array}{ll} \mu y{=}x.P & \cong (\boldsymbol{\nu}\, y)(P|![y \to x]) \\ & \cong (\boldsymbol{\nu}\, y)(P|(\boldsymbol{\nu}\, x)(![y \to x]|P)) \\ & \cong (\boldsymbol{\nu}\, y)(P|(\boldsymbol{\nu}\, xw)(![y \to x]|P[w/y]|[w \to y])) \\ & \cong (\boldsymbol{\nu}\, y)(P|(\boldsymbol{\nu}\, w)(P[yw/xy]|[w \to y])) \cong (\boldsymbol{\nu}\, y)(P|(\mu y{=}x.P)[x/y]). \end{array}$$ □

Lemma 18. *Let* $\vdash P \triangleright \Gamma, y : \overline{\tau}^? \,;\, \Delta, x : \tau^!$. *Then* $P \models \mathbf{rely}\ A^{\text{-}y}\ \mathbf{guar}\ B$ *implies* $\mu y{=}x.P \models \mathbf{rely}\ A\ \mathbf{guar}\ B$.

Proof. The statement is intuitively reasonable since any behavioural guarantee at y is unused to ensure B. Formally we reason as follows. Let $\mathsf{dom}(\Gamma) = \vec{w}$ and set P_0 so that $P \cong P_0|P'$ with $\vdash P_0\Gamma, y{:}\overline{\tau}\,;\, x{:}\tau$ (by Lemma 16).

$$\begin{array}{lll} R^{\Gamma} \models^{\mathcal{I}} A & & \\ \quad \supset & \forall S^{y:\tau}.\ (\boldsymbol{\nu}\, \vec{w}y)(P|R|S) \models^{\mathcal{I}} B & \text{(IH, Lemma 8)} \\ \quad \supset & (\boldsymbol{\nu}\, \vec{w}y)(P|R|(\boldsymbol{\nu}\, \vec{w})((\mu y{=}x.P_0)[y/x]|R) \models^{\mathcal{I}} B & ((\boldsymbol{\nu}\, \vec{w})((\mu y{=}x.P_0)|R)^{y:\tau}) \\ \quad \supset & (\boldsymbol{\nu}\, \vec{w}y)(P|R|(\mu y{=}x.P_0)[y/x]) \models^{\mathcal{I}} B & \text{(Corollary 1)} \\ \quad \supset & (\boldsymbol{\nu}\, \vec{w}y)(P|R|(\mu y{=}x.P)[y/x]) \models^{\mathcal{I}} B & \text{(Lemma 17),} \end{array}$$

as required. □

3.3 Soundness of Proof Rules

The proof system for the affine logic uses the same sequent as before:

$$P^{\Gamma;\Delta} \vdash \mathbf{rely}\ A\ \mathbf{guar}\ B.$$

The provability relation is derived by all rules in Figure 2 except (Open) which is refined to incorporate the termination guarantee, plus one additional rule for recursion. These two rules are listed in Figure 4. In the rule, we use the notation $A(e)$ which denotes the result of substituting e for a fixed variable in A. For example, if i is the fixed variable for $A \stackrel{\text{def}}{=} u \bullet i = i!$, then $A(1)$ denotes $u \bullet 1 = 1!$ while $A(j+1)$ denotes $u \bullet j + 1 = (j+1)!$.

$$(\mathsf{Out}^{?})\quad \frac{(A \supset A_1 \supset A_2[x\bullet\vec{y}/z] \wedge x \bullet \vec{y}\Downarrow)\quad R \vdash \mathbf{rely}\ A\ \mathbf{guar}\ A_1\quad P \vdash \mathbf{rely}\ A \wedge A_2\ \mathbf{guar}\ B}{\overline{x}(\vec{y}z)(R|P) \vdash \mathbf{rely}\ A\ \mathbf{guar}\ B}$$

$$(\mathsf{Rec})\quad \frac{P^{\Gamma\cdot\vec{y}:\overline{\vec{r}};\vec{x}:\vec{r}^{\,!}} \vdash \mathbf{rely}\ A^{\vec{y}}\ \mathbf{guar}\ B(0)\quad P \vdash \mathbf{rely}\ A \wedge B(i)[\vec{y}/\vec{x}]\ \mathbf{guar}\ B(i+1)}{\mu\vec{y}{=}\vec{x}.P \vdash \mathbf{rely}\ A\ \mathbf{guar}\ B}$$

Fig. 4. Altered/Added Proof Rules for Total Affine Logic

The (Out) rule now ensures that the result of invocation should terminate. In the rule, "$a\Downarrow$" stands for "$\exists i, \vec{y}.\, a = \mathtt{in}_i(\vec{y})^{\uparrow}$", assuming a is typed with a linear output. Except for this addition, the rule is read just as in the original rule in the linear process logic. The need to give a further constraint on convergence is essentially because this total logic is *not* about ensuring strong normalisability: a soundly inferred term may as well be partial when it is invoked with a specific argument, thus it is necessary to explicitly stipulate the convergence. In practice, whenever we infer a non-trivial property such as $x : y = \mathtt{in}_3(u)^{\uparrow}$, it automatically ensures convergence.

The (Rec) rule, which resembles the standard proof rule for the while loop (of Hoare Logic for total correctness), combines mathematical induction and recursive behaviour, so that we can ensure total correctness of typed processes (in particular their convergence) by the well-founded induction. The choice of natural numbers as a domain for well-founded induction is not significant but is convenient and general enough, as has been practised in the foregoing studies of program logics. For non-trivial examples of reasoning using this rule, see [6, 7]. Note we can derive $\mu y = x.P \vdash \mathbf{rely}\ \mathsf{T}\ \mathbf{guar}\ \mathsf{T}$ from $P \vdash \mathbf{rely}\ \mathsf{T}\ \mathbf{guar}\ \mathsf{T}$. In fact, $P^{\Gamma;!\Delta} \vdash \mathbf{rely}\ A\ \mathbf{guar}\ \mathsf{T}$ for any A (as far as the sequent is well-typed).

In the following we establish the soundness of the proof system for the basic affine logic. Firsly, the rules in Figure 2 except (Out) are immediately sound (via Lemma 14) with exactly the same proofs as in Section 2.5 (starting from Page 8). Next, for the refined (Out), we verify:

Proposition 2. *(Out) in Figure 2 is sound in the affine logic.*

Proof. As before, it suffices to show the soundness of the following rule, which is equivalent to the rule in Figure 4.

$$(\mathsf{Out}^{?}_{\mathsf{dec}})\quad \frac{R \vdash \mathbf{rely}\, A \mathbf{guar} A_1,\ P \vdash \mathbf{rely}\, A \wedge A_2 \mathbf{guar}\, B,\ A \supset A_1 \supset A_2[x\bullet\vec{y}/z] \wedge x \bullet \vec{y}\Downarrow}{\overline{x}(\vec{y}z)(R|P) \vdash \mathbf{rely}\ A\ \mathbf{guar}\ B}$$

The following proof is essentially idential with the one given in Page 11, §2.5, except for a single line for checking a divergence. For precision we list the proof below. Assume $\vdash R \triangleright \Gamma$ such that $R \Downarrow$ and, by Lemma 14 (i) which says the affine calculus preserves Lemma 1 (3) under convergence assumption, we can set $R \equiv (!x(\vec{y}z).R_0) \mid R'$ where $!x(\vec{y}z).R_0 \models_{\mathsf{b}} x : \theta$ with $\theta = \wedge_i(\vec{\alpha}_i\beta_i)^{!}$ whose $\vec{\alpha}_i$

ranges over all constants of the appropriate types. Let $\xi \stackrel{\text{def}}{=} x{:}\theta \cdot \vec{w}{:}\vec{\gamma}$ and write $\ddagger$ for the condition $A \supset A_1 \supset A_2[x\bullet\vec{y}/z] \wedge x\bullet\vec{y}\Downarrow$. Below Corollary 1), Corollary 2), Lemma 3 (1)) and Lemma 13 are the corresponding lemmas for the affine logic, which hold by Lemma 14. Note below, in the third entailment, we infer the totality of z from $x\bullet\vec{y}\Downarrow$ (without which $\models^{\mathcal{I}}$ is not even defined).

$$
\begin{array}{lll}
R \models_{\mathsf{b}} x{:}\theta \cdot \vec{w}{:}\vec{\gamma} \models^{\mathcal{I}} A & & \\
\quad \supset & S|R \models_{\mathsf{b}} \xi \cdot \vec{y}{:}\vec{\alpha} \models^{\mathcal{I}} A \wedge A_1 & \text{(IH, Corollary 1)} \\
\quad \supset & S|R \models_{\mathsf{b}} \xi \cdot \vec{y}{:}\vec{\alpha}_i \models^{\mathcal{I}} A \wedge A_2[x \bullet \vec{y}/z] \wedge x\bullet\vec{y}\Downarrow & (\ddagger\text{, Corollary 2}) \\
\quad \supset & S|R|R_0 \models_{\mathsf{b}} \xi \cdot \vec{y}{:}\vec{\alpha}_i \cdot z{:}\beta_i & \text{(Lemma 3 (1))} \\
 & \xi \cdot \vec{y}{:}\vec{\alpha}_i \cdot z{:}\beta_i \models_{\mathsf{b}} A \wedge A_2 \wedge x \bullet \vec{y} = z & \text{(Lemma 13)} \\
\quad \supset & (\boldsymbol{\nu}\,\vec{y})(S|R|R_0) \models_{\mathsf{b}} \xi \cdot z{:}\beta_i \models^{\mathcal{I}} A \wedge A_2 & \text{(Cor. 2, Lem. 3 (2))} \\
\quad \supset & (\boldsymbol{\nu}\,\vec{w}xz)(P|(\boldsymbol{\nu}\,\vec{y})(S|R|R_0)) \models^{\mathcal{I}} B & \text{(IH)} \\
\quad \supset & (\boldsymbol{\nu}\,\vec{w}x)(\overline{x}(\vec{y}z)(S|P) \mid R) \models^{\mathcal{I}} B & \text{(Corollary 1).}
\end{array}
$$

The last step used: $(\boldsymbol{\nu}\,\vec{w}xz)(\overline{x}(\vec{y}z)(S|P) \mid R) \longrightarrow (\boldsymbol{\nu}\,\vec{w}\vec{y}xz)(S|P|R|R_0)$. □

rule is verified precisely following the reasoning for the original (Out) rule except for the termination guarantee, the soundness of the provability can be verified solely via establishing the soundness of (Rec).

Proposition 3. *(Rec) in Figure 2 is sound in the affine logic.*

Proof. As before, we assume the antecedent of (Rec) holds (reading $\vdash$ as $\models$), and show that its conclusion holds (again reading $\vdash$ as $\models$). Let $\mathcal{I}$ be an arbitrary (and appropriately typed) interpretation of auxiliary names. Let us assume, by induction hypothesis, that:

(IH-1) $P \models \mathbf{rely}\, A\, \mathbf{guar}\, B(0)$.
(IH-2) $P \models \mathbf{rely}\, A \wedge B(i)\, \mathbf{guar}\, B(i+1)$.

Under these conditions we show $\mu y{=}x.P \models \mathbf{rely}\, A\, \mathbf{guar}\, B$.

First we fix the action type of P be $\overline{\Gamma}, y : \overline{\tau}^{?}\,;\, \Delta, x : \tau^{!}$ with $\mathsf{fn}(\Gamma) = \vec{w}$. By Lemma 16, we safely assume $P \equiv P_0|P'$ such that $\vdash P_0 \triangleright \Gamma, x : \tau$. Our argument is by mathematical induction, showing, under the assumption $R^{\Gamma} \models^{\mathcal{I}} A$ and writing $\Psi(i)$ for $(\boldsymbol{\nu}\,\vec{w}y)(P|R|(\mu y{=}x.P)[y/x]) \models^{\mathcal{I}} B(i)$:

(Base step) $\Psi(0)$ is valid;
(Inductive step) $\Psi(n)$ implies $\Psi(n+1)$ for each n,

from which we can infer $\Psi(n)$ for all n.

We first show the base case. Below we observe B can only contain (at most) x as its prime name by typing (since if not it cannot occur in the negative position in the second sequent in the antecedent).

$$
\begin{array}{lll}
R^{\Gamma} \models^{\mathcal{I}} A \quad \supset & (\boldsymbol{\nu}\,\vec{w}y)(P|R|(\mu y{=}x.P)[y/x]) \models^{\mathcal{I}} B(0) & \text{(IH-1, Lemma 18)} \\
\quad \supset & (\boldsymbol{\nu}\,\vec{w})(\mu y{=}x.P|R) \models^{\mathcal{I}} B(0) & \text{(Lemma 17).}
\end{array}
$$

For the inductive step, for an arbitrary natural number n:

$$
\begin{array}{lll}
R^\Gamma \models^{\mathcal{I}} A,\ (\boldsymbol{\nu}\,\vec{w})(\mu y{=}x.P|R) \models^{\mathcal{I}} B(n) & & \\
\quad\equiv\quad R^\Gamma \models^{\mathcal{I}} A,\ (\boldsymbol{\nu}\,\vec{w})(\mu y{=}x.P|R)[y/x] \models^{\mathcal{I}} B(n)[y/x] & & \text{(Lemma 15)} \\
\quad\supset\quad (\mu y{=}x.P)[y/x]\,|\,R \models^{\mathcal{I}} A \wedge B(n)[y/x] & & \text{(Lemma 10)} \\
\quad\supset\quad (\boldsymbol{\nu}\,y\vec{w})(P|(\mu y{=}x.P)[y/x]|R) \models^{\mathcal{I}} B(n+1) & & \text{(IH-2)} \\
\quad\supset\quad (\boldsymbol{\nu}\,\vec{w})(\mu y{=}x.P|R) \models^{\mathcal{I}} B(n+1) & & \text{(Lemma 17).}
\end{array}
$$

Hence we conclude $R^\Gamma \models^{\mathcal{I}} A$ implies $(\boldsymbol{\nu}\,\vec{w})(\mu y{=}x.P|R) \models^{\mathcal{I}} B(n)$ for an arbitrary natural number n, that is (assuming the fixed variable in $B(\,\cdot\,)$ is j) we have $(\boldsymbol{\nu}\,\vec{w})(\mu y = x.P|R) \models^{I\cdot j\mapsto n} B$ for an arbitrary n. By definition this means $\mu y{=}x.P \models \mathbf{rely}\ A\ \mathbf{guar}\ B$, as required. □

Theorem 2. *The proof system of the basic affine logic is sound.*

We conclude this section with a note on an alternative formulation of affine logic whose universe includes divergent computation.

Remark 2. We briefly discuss an alternative affine process logic which can directly assert on divergent computation. To start with, the following shows that the language of our basic affine logic can in fact express the divergence of a process, even though its total model prohibits such expressions.

$$P^{x:()^\uparrow} \models^{\mathcal{I}} \mathsf{T}\,\mathbf{guar}\,x \neq ()^\uparrow$$

This assertion is not valid for any P in the basic affine logic: however it is often useful to be able to talk about divergence in assertions, motivating an extension of the affine logic where assertions are meaningful. For this purpose we extend the relation $\models^{\mathcal{I}}$ so that its domain is the whole set of (possibly non-total) models, while retaining the identical inductive definition. $\models_{\mathsf{b}}$, which already treats divergence, stays unchanged.

The resulting logic allows us to discuss both divergence and convergence. Let us write "$x \Downarrow$" for "$\exists i, \vec{y}.\, x = \mathtt{in}_i(\vec{y})^\uparrow$", assuming x is typed as $[\oplus_{i\in I}\vec{\alpha}_i]^\uparrow$, and "$x \Uparrow$" for "$\neg x \Downarrow$". Then the assertion:

$$P^{\overline{\Gamma};\Delta,x:\tau^\uparrow} \models^{\mathcal{I}} \mathbf{rely}\,A\,\mathbf{guar}\,x \Uparrow,$$

in the generalised affine logic means that, for each $R^\Gamma \models^{\mathcal{I}} A$, the process $(\boldsymbol{\nu}\,\mathsf{fn}(\Gamma))(P|R)$ diverges. On the other hand, the assertion:

$$P^{\overline{\Gamma};\Delta,x:\tau^\uparrow} \models^{\mathcal{I}} \mathbf{rely}\,A\,\mathbf{guar}\,x \Downarrow,$$

says that $(\boldsymbol{\nu}\,\mathsf{fn}(\Gamma))(P|R)$ necessarily converges for each $R^\Gamma \models^{\mathcal{I}} A$. As another example, the following statement is always valid in the general affine logic.

$$P^{y:\rho^\downarrow,x:\tau^\uparrow} \models \mathbf{rely}\,y \Uparrow\ \mathbf{guar}\,x \Uparrow .$$

In fact, if P under the given typing is to converge, it should rely on the convergence of the environment at y, since, if not, P may not be activated at y, which, by typing, should always precede a linear output at x.

The general affine logic satisfies all essential properties of the basic logic, in the sense that all lemmas and corollaries of Sections 2.3 and 2.4 hold in the affine logic, except for changing Lemma 1 (3) as noted in Lemma 14, and for changing the statement of Lemma 8 into:

Let $\mathsf{dom}(\xi') \cap \mathsf{fn}(A) = \emptyset$ and $\omega \notin \mathsf{ran}(\xi')$. Then $\xi \cdot \xi' \models^{\mathcal{I}} A$ iff $\xi \models^{\mathcal{I}} A$

This is immediate for the properties of $\models_{\mathsf{b}}$ since it is not changed. for Lemma 8, if ω is used in ξ', then any well-typed ξ satisfies $\xi \cdot \xi' \models^{\mathcal{I}} A$, hence we need the additional condition. For other statements, the proofs are identical line by line.

The proof system for the general affine logic uses the same rules as the basic one, except that we have an additional rule for recursion (whose details we omit: in brief, the rule infers $\mu y = x.P \vdash \mathbf{rely}\,A\,\mathbf{guar}\,B$ from $P \vdash \mathbf{rely}\,A \wedge$

$B[y/x]\,\mathbf{guar}\,B$ when B is satisfiable by a lazy divergent behaviour at x under any interpretation) as well as the following refinement of the branching rule:

$$(\mathsf{Bra}^{\downarrow}\Downarrow)\ \frac{\forall i.\ P_i^{\Gamma_i, z:\tau^{\uparrow}} \vdash \mathbf{rely}\ A[\mathtt{in}_i(\vec{y}_i)^{\uparrow}/x]\,\mathbf{guar}\ B \quad B\Downarrow_z \supset A\Downarrow_x}{x[\&_i(\vec{y}_i).P_i] \vdash \mathbf{rely}\ A\,\mathbf{guar}\ B}$$

In the rule above, "$A\Downarrow_x$" ("*A ensures convergence at x*") stands for "$A \supset x\Downarrow$." The added side condition says that if we need to guarantee the convergence of the process, then we should rely on the convergence of the environment (a criticism can be lodged against the use of such a "semantic" predicate in the syntactic rule: various syntactic aspects of the general affine process logic would need further scrutiny). Without this condition, we can easily derive an unsound assertion, as the following example shows.

$$(\mathsf{Bra})\ \frac{(\mathsf{Sel})\ \dfrac{}{\overline{z} \vdash \mathbf{rely}\ \mathsf{T}\,\mathbf{guar}\,z\Downarrow}}{x.\overline{z} \vdash \mathbf{rely}\ \mathsf{T}\,\mathbf{guar}\,z\Downarrow}$$

This conclusion is unsound since, given $\vdash R \triangleright x{:}()^{\uparrow}$ which is diverging, we have $R \models^{\mathcal{I}} \mathsf{T}$ (note $x:\omega \models^{\mathcal{I}} \mathsf{T}$ in the general affine logic, unlike in the basic affine logic), using which we know $(\boldsymbol{\nu}\, x)(x.\overline{z}|R)\Uparrow$, contradicting the conclusion above. However, under the side condition, this is not possible since it says that, because $z\Downarrow \supset z\Downarrow$, we should have $\mathsf{T} \supset x\Downarrow$, which obviously does not hold, precluding the unsound inference above. Generalising this reasoning, we can easily show the rule is semantically sound. Assume $R^{\Gamma} \models^{\mathcal{I}} A$. If $R\Downarrow$ then we have precisely the same reasoning as for Theorem 1. For the other case, noting $R\Uparrow$ implies $(\boldsymbol{\nu}\,\mathsf{fn}(\Gamma)x)(x[\&_i(\vec{y}_i).P_i]|R)\Uparrow$:

$$\begin{array}{lll} R \models_{\mathsf{b}} \xi\cdot x{:}\omega \models^{\mathcal{I}} A & \supset & \models^{\mathcal{I}} \neg(A \supset z\Downarrow) \\ & \supset & \models^{\mathcal{I}} \neg(B \supset x\Downarrow) \qquad \text{(IH)} \\ & \supset & (\boldsymbol{\nu}\,\mathsf{fn}(\Gamma)x)(x[\&_i(\vec{y}_i).P_i]|R) \models_{\mathsf{b}} z:\omega \models^{\mathcal{I}} B, \end{array}$$

as required. The soundness of the remaining rules can be checked by the same reasoning as before except (Par) needs to treat the divergent case separately, as in the reasoning for $(\mathsf{Bra}^{\downarrow}\Downarrow)$ above.

$$\frac{\vdash P \triangleright \Gamma, z{:}\rho^{\uparrow}}{\vdash P \triangleright \Gamma, z{:}\rho^{\uparrow}, x{:}\tau^{?_{rw}}}\ (\mathsf{Weak}\text{-}?_{rw}) \qquad \frac{\vdash P \triangleright ?\Gamma^{\text{-}x},\ ?_{rw}\Delta^{\text{-}x},\ \vec{y}z{:}\vec{\tau}\rho}{\vdash !\, x(\vec{y}z).P \triangleright \Gamma,\ x{:}(\vec{\tau}\rho)^{!}_{\Delta}}\ (\mathsf{In}^{!}) \qquad \frac{\vdash R \triangleright ?\Gamma,\ \vec{y}{:}\vec{\tau} \quad \vdash P \triangleright ?\Gamma,\ ?_{rw}\Delta,\ z{:}\rho^{\downarrow},\ v{:}\sigma^{\uparrow}}{\vdash \overline{x}(\vec{y}z)(R|P) \triangleright \Gamma,\ ?_{rw}\Delta,\ v{:}\sigma}\ (\mathsf{Out}^{?})\ (\Gamma(x)=(\vec{\tau}\rho)^{?}_{\Delta'},\ \Delta'\subset\Delta)$$

$$\frac{-}{\vdash \mathsf{Ref}\langle xy\rangle \triangleright x{:}ref(\tau), y{:}\overline{\tau}}\ (\mathsf{Ref})\quad (x \notin \mathsf{fn}(\tau)) \qquad \frac{\vdash P \triangleright \uparrow ?\,?_{rw}\Gamma,\ c{:}(\overline{\tau})^{\downarrow}}{\vdash \overline{x}\mathsf{read}(c)P \triangleright \Gamma}\ (\mathsf{Read})\quad (\Gamma(x)=rw(\tau)) \qquad \frac{\vdash P \triangleright \uparrow ?\,?_{rw}\Gamma,\ v{:}\tau,\ c{:}()^{\downarrow}}{\vdash \overline{x}\mathsf{write}(vc)P \triangleright \Gamma}\ (\mathsf{Write})\quad (\Gamma(x)=rw(\tau))$$

Fig. 5. Typing Rules for Open State

4 Logic for Sequential Processes with Open State

4.1 Affine Behaviour with State

Stateful types add the following constructs to the grammar of types:

$$\tau \quad ::= \quad (\vec{\tau}^{?}\rho^{\uparrow})^{!}_{?_{rw}\Gamma} \ \mid\ (\vec{\tau}^{!}\rho^{\downarrow})^{?}_{?_{rw}\Gamma} \ \mid\ ref(\tau^{!}) \ \mid\ rw(\tau^{?})$$

where $\Gamma, \Delta, \ldots$ range over, as before, finite maps from names to channel types ($?_{rw}\,\Gamma$ indicate the mode of types in Γ is $?_{rw}$, with $?_{rw}$ being the mode for the type $rw(\tau)$, whose dual is written as $!_{rw}$). Note types never carry reference types: this does not restrict representability of behaviour. When $\Gamma = \emptyset$, we identify $(\vec{\tau}^{?}\rho^{\uparrow})^{!}_{\Gamma}$ with $(\vec{\tau}^{?}\rho^{\uparrow})^{!}$ (in the types without state). Note that, by the lack of recursively defined types, whenever $\Gamma(x) = \tau$, we have $x \notin \mathsf{fn}(\tau)$.

Γ in $(\vec{\tau}^{?}\rho^{\uparrow})^{!}_{\Gamma}$ are *effects* in the sense of the type and effect discipline [14]. We call Γ (resp. $\mathsf{dom}(\Gamma)$) in $(\vec{\tau}^{?}\rho^{\uparrow})^{!}_{\Gamma}$, its *effect typing* (resp. its *effect names*). Effects arise naturally as part of the basic type structure the present logic is built upon.

For processes, we add:

$$P \quad ::= \quad \ldots \mid \mathsf{Ref}\langle xy\rangle \mid \overline{x}\mathsf{read}(c)P \mid \overline{x}\mathsf{write}(yc)(R|P)$$

The reduction rules are standard:

$$\mathsf{Ref}\langle xy\rangle|\overline{x}\mathsf{read}(c)P \longrightarrow \mathsf{Ref}\langle xy\rangle|(\boldsymbol{\nu}\, c)(\overline{c}\langle y\rangle|P)$$
$$\mathsf{Ref}\langle xy\rangle|\overline{x}\mathsf{write}(y'c)P \longrightarrow (\boldsymbol{\nu}\, y')(\mathsf{Ref}\langle xy'\rangle|\overline{c}|P)$$

We add a reference as a constant, which is easily encodable into the standard π-calculus process. Introducing this constant as such however makes the logic reasoning tractable. The rules may not need any illustration ($\overline{c}$ in the write rule is the acknowledgement P would receive).

The typing rules are listed in 5. We add the rules for references and two actions on them, as well as the rule which introduces a now effect-annotated replicated type and the dual rule, each subsuming the corresponding rule in Figure 1. We also add the weakening of co-reference (read/write) channels, which is restricted to the case when P contains a linear output. This restriction can also be applied to the original weakening rule without losing typability, since we only input-abstract these names in such a process.

By retaining the original (Res) rule without change, we do not allow the hiding of the subject of a reference. Because of this, we call the typed processes in this system *processes with open state*: adding hiding of references leads to the existence of local state, which substantially changes the nature of logics.

4.2 Logic for Affine Processes with Open State

The following affine logic with open state extends the basic affine logic in the previous section to open stateful behaviour. We first extend terms and formulae (now mutually defined) by the following grammars.

$$a \ ::= \ ... \mid !x^{ref(\tau)} \qquad A \ ::= \ ... \mid \{A^{\Delta}\}\, a^{(\vec{\rho}\tau)^{!}_{\Delta}} \bullet \vec{b}^{\vec{\rho}}\, \{A'^{\Delta}\} \mid \langle \{A^{\Delta}\} a^{(\vec{\rho}\tau)^{!}_{\Delta}} \bullet \vec{b}^{\vec{\rho}} / z^{\rho} \rangle B.$$

For legibility we often omit type annotations. $\mathsf{fn}(a)$ and $\mathsf{fn}(A)$ now include names of types, starting from $\mathsf{fn}(x^{\tau}) = \{x\} \cup \mathsf{fn}(\tau)$ where $\mathsf{fn}(\tau)$ may include effect names when τ is replicated. $!x^{ref(\tau)}$ is a *dereference*, resulting in type τ. $\{A\}\, a \bullet \vec{b}\, \{A'\}$ says invoking a with $\vec{b}$ changes the state from A to A', and $\langle \{A\} a \bullet \vec{b} / z \rangle B$ says invoking a with $\vec{b}$ at A returns z for which B holds. For brevity we set:

$$\langle \{C\} a \bullet \vec{b}\, \{C'\} / z \rangle A \qquad \equiv \qquad \{C\}\, a \bullet \vec{b}\, \{C'\} \ \wedge \ \langle \{C\} a \bullet \vec{b} / z \rangle A$$

Terms and formulae should obey a natural notion of typing. For example, for a formula $\langle \{C_1\} a^{(\vec{\tau}\rho)^{!}_{\Delta}} \bullet \vec{b}^{\vec{\rho}}\, \{C_2\} / z^{\rho} \rangle A$ to be well-typed under $\Gamma ; ; \Theta$, we should have $\Delta ; ; \Theta \vdash C_i$ $(i = 1, 2)$ as well as $(\Gamma \cdot z : \rho) ; ; \Theta \vdash A$, similarly for others.

In accordance with channel types, behavioural constants are extended as:

$$\alpha \ ::= \ ... \mid \wedge_{i \in I} \{\xi_i\} (\vec{\alpha}_i^{?} \beta_i^{\uparrow})^{!} \{\xi_i'\} \mid \vee_{i \in I} \{\xi_i\} (\vec{\alpha}_i^{!} \beta_i^{\downarrow})^{?} \{\xi_i'\} \mid \mathsf{Ref}(\alpha^{!}) \mid \mathsf{RW}(\alpha^{?}) \ .$$

We only consider well-typed constants. For example, $\wedge_{i \in I} \{\xi_i\} (\vec{\alpha}_i^{?} \beta_i^{\uparrow})^{!} \{\xi_i'\}$ is well-typed with type $(\vec{\tau}\rho)^{!}_{\Delta}$ iff, for each i, $\vec{\alpha}_i \beta_i$ are typed by $\vec{\tau}_i \rho_i$ and each of ξ_i and ξ_i' has an effect type Δ, similarly for its dual. Further each (co-)replicated behaviour should define a continuous function from states and arguments to states and answers. A *model* ξ is defined as before using only well-formed constants.

$P^{\Gamma} \models_{\mathsf{b}} \xi$ is given by the following inductions (only the third clause is a substantial addition).

1. $P \models_{\mathsf{b}} !!_{rw} \xi \cdot x : \mathtt{in}_i(\vec{\alpha})^{\uparrow}$ when $P \stackrel{\overline{x}\mathtt{in}_i(\vec{y})}{\Longrightarrow} P'$ such that $P' \models_{\mathsf{b}} \xi \cdot \vec{y} : \vec{\alpha}$. Further $P \models_{\mathsf{b}} \xi \cdot x : \omega$ when $P \Uparrow$.

2. $P \models_{\mathsf{b}} !\xi \cdot x : \wedge_{i\in I}\{\xi_i\}(\vec{\alpha}_i\beta_i)^!\{\xi_i'\}$ when $P \xrightarrow{x(\vec{y}z)} P'$ such that, for each $i \in I$, $R \models_{\mathsf{b}} \vec{y} : \vec{\alpha}_i \cdot \overline{\xi_i}$ implies $(\boldsymbol{\nu}\, x\vec{y})(P'|R) \models_{\mathsf{b}} \xi \cdot \xi_i' \cdot z : \beta_i$,
3. $P \models_{\mathsf{b}} !!_{rw}\xi \cdot x : \mathsf{Ref}(\alpha)$ when $P \xrightarrow{x\mathsf{read}(c)} C[\mathsf{Ref}\langle xz\rangle]$ s.t. $C[\mathbf{0}] \models_{\mathsf{b}} \xi \cdot c : (\alpha)^{\uparrow}$.

In the third clause we take off the reference in conclusion, for testing replications with different values (needed because references are global so that their values are transient). $\xi_1 \cong \xi_2$ and $\alpha_1 \cong \alpha_2$ are defined as before.

$[\![a]\!]_{\mathcal{I}\cdot\xi}$ and $\xi \models^{\mathcal{I}} A$, the latter with ξ total (cf. Section 3), are simultaneously extended as follows, keeping all other rules. Below in the second and third clauses, we let $[\![a]\!]_{\mathcal{I}\cdot\xi} = \wedge_{i\in J}\{\xi_i\}(\vec{\alpha}_i\beta_i)^!\{\xi_i'\}$ and $\eta = \xi/\mathsf{dom}(\xi_i)$.

- $[\![x^{ref(\tau)}]\!]_{\mathcal{I}\cdot\xi} \stackrel{\text{def}}{=} x$.
- $[\![!x]\!]_{\mathcal{I}\cdot\xi} \stackrel{\text{def}}{=} \alpha \quad (\text{if } (\xi \cdot I)(x) = \mathsf{Ref}(\alpha))$.
- $\xi \models^{\mathcal{I}} \{A\}a \bullet \vec{b}\{A'\}$ iff $\forall i \in J.\, (\eta \cdot \xi_i \models^{\mathcal{I}} A \supset [\![\vec{b}]\!]_{\mathcal{I}\cdot\xi} = \vec{\alpha}_i \supset \eta \cdot \xi_i' \models^{\mathcal{I}} A')$.
- $\xi \models^{\mathcal{I}} \langle\{A\}a \bullet \vec{b}/z\rangle B$ iff $\forall i \in J.\, (\eta \cdot \xi_i \models^{\mathcal{I}} A \supset [\![\vec{b}]\!]_{\mathcal{I}\cdot\xi} = \vec{\alpha}_i \supset \xi \cdot z : \beta_i \models^{\mathcal{I}} B)$.
- $\xi \models^{\mathcal{I}} \forall x^{ref(\tau)}.A$ iff $\forall \alpha^{\tau}.\xi \models^{I\cdot x:\mathsf{Ref}(\alpha)} A \wedge \forall z^{ref(\tau)} \in \mathsf{dom}(\xi).\xi \models^{\mathcal{I}} A[z/x]$.
- $\xi \models^{\mathcal{I}} \exists x^{ref(\tau)}.A$ iff $\exists \alpha^{\tau}.\xi \models^{I\cdot x:\mathsf{Ref}(\alpha)} A \vee \exists z^{ref(\tau)} \in \mathsf{dom}(\xi).\xi \models^{\mathcal{I}} A[z/x]$.

In the first clause, a reference is interpreted as a distinction [12] rather than its name-free denotation. The second clause extracts a stored value from a reference. In the third/fourth clauses, we take off $\mathsf{dom}(\xi_i)$ from ξ because A is about a hypothetical state. In the fourth clause, B is interpreted by ξ together with the new value for z, not using ξ_i: the scope of a hypothetical state is restricted to the Hoare triple associated with application (consistently with effect typing). Note also $\beta_i \neq \omega$ is assumed in the third line (by totality). In the final two clauses, quantified reference names range over (not only fresh ones but also) those of existing references, which reflects the nature of proper names.

4.3 Sequent and Interpretation

The following classification of sequentially typed processes already exists in the original linear/affine typing. First, a process with a free linear output is called **thread**. Typewise such a process can be written $\vdash P \triangleright \Gamma \cdot x : \tau^{\uparrow}$. By typing, Γ contains no linear output channel and at most a single linear input channel. Among threads, we have further classification:

- *Pure thread*, which does not contain a free replicated input (i.e. Γ above does not contain $!!_{rw}$-channels).
- *Active thread*, which does not contain a free linear input (i.e. Γ above does not contain a $\downarrow$-channel).
- *Passive thread*, which does contain a free linear input (i.e. Γ above contains a $\downarrow$-channel).

A process without a free linear output is called **non-thread**. Typewise P is a non-thread iff $\vdash P \triangleright \Gamma$ such that $\mathsf{md}(\Gamma) \subset \{!, ?, !_{rw}, ?_{rw}\}$.

The distinction between threads and non-threads is reflected in the shape of sequents as follows. For non-threads, we use the same sequent as before:

$$P^{\Gamma;\Delta} \models \mathbf{rely}\, A\, \mathbf{guar}\, B,$$

which is used only when $P^{\Gamma;\Delta}$ is a non-thread. Note this means Γ does not contain $?_{rw}$-channels. As before, we require A is typed by $\Gamma \cdot \Theta$ while B is typed by $\Delta \cdot \Theta$, for some typing of auxiliary names Θ.

For threads, we use the sequent in the shape of:

$$P^{\Gamma;\Delta} \models \{C\}\, \mathbf{rely}\, A\, \mathbf{guar}\, B\, \{C'\}$$

which is used when P is a thread. For simplicity and without loss of practical expressiveness, we exclude the case when P contains *both* a reference *and* a linear input (which requires a sequent of a different shape). Let $\Gamma = ?_{rw}\Gamma_0 \cdot ? \downarrow \Gamma_1$. Then we require: (1) C and C' are typed by $\Gamma_0 \cdot \Theta$, (2) A is typed by $\Gamma_1 \cdot \Theta$, and (2) B is typed by $\Delta \cdot \Theta$, for a typing Θ of auxiliary names. *From now on we always assume all occurring judgements (including these sequents) are well-typed.*

We can now introduce the semantics of these two forms of sequents. As before, $P \models^{\mathcal{I}} A$ stands for $P \models_{\mathsf{b}} \xi$ and $\xi \cdot I \models A$ for some ξ.

Definition 2. (semantics of assertions)

1. Let $\vdash P \triangleright \Gamma; \Delta$ be a non-thread. With $\mathsf{fn}(\Gamma) = \{\vec{x}\}$:

$$P^{\overline{\Gamma};\Delta} \models^{\mathcal{I}} \mathbf{rely}\, A\, \mathbf{guar}\, B \;\equiv\; R^{\Gamma} \models^{\mathcal{I}} A \supset (\boldsymbol{\nu}\, \vec{x})(P|R) \models^{\mathcal{I}} B$$

2. Let $\vdash P \triangleright \Gamma; \Delta$ be a thread and $\Gamma = ?_{rw}\Gamma_0, ? \downarrow \Gamma_1$. With $\mathsf{fn}(\Gamma_0) = \{\vec{x}\}$:

$$P^{\overline{\Gamma};\Delta} \models^{\mathcal{I}} \{C\}\, \mathbf{rely}\, A\, \mathbf{guar}\, B\, \{C'\} \;\equiv\; R^{\Gamma} \models^{\mathcal{I}} C \wedge A \supset (\boldsymbol{\nu}\, \vec{x})(P|R) \models^{\mathcal{I}} B \wedge C'$$

4.4 Properties of Interpretations

In the following we list basic properties of the open stateful logic. Since there are quite a few changes from the statements in Sections 2.3 and 2.4, we restate the corresponding properties below for avoiding confusion, as well as augmenting them with what are genuinely new in stateful computation. As before, we always assume well-typedness and well-formedness of processes, formulae, sequents, etc.

Lemma 19. (1) $\equiv, \longrightarrow, \approx$ *are subrelations of* $\cong$.

(2) $!x(\vec{y}).R|P_1|P_2 \cong !x(\vec{y}).R|P_1|(\boldsymbol{\nu}\, x)(!x(\vec{y}).R|P_2)$. *Also we have* $(\boldsymbol{\nu}\, x)!x(\vec{y}).R \cong (\boldsymbol{\nu}\, x)\mathsf{Ref}\langle xy\rangle \cong \mathbf{0}$.

(3) (a) *If* $\vdash P \triangleright !!_{rw}\Gamma \cdot !\Delta$, *then* $P \cong Q|R$ *such that* $\vdash Q \triangleright \Gamma$ *and* $\vdash R \triangleright \Delta$.
(b) *If* $\vdash P \triangleright !\Gamma \cdot !_{rw}\Delta \cdot x : \tau^{\uparrow}$, *then* $P \cong Q|R$ *such that* $\vdash Q \triangleright \Gamma \cdot \Delta$ *and* $\vdash R \triangleright ?_{rw}\overline{\Delta} \cdot x{:}\tau$. *Further if* $P \not\longrightarrow$ *then we can set* $\vdash R \triangleright x{:}\tau$.

(4) *Let* $\vdash P_{1,2} \triangleright !!_{rw}\Gamma \cdot x{:}\rho^{\uparrow}$ *and* $P_i \overset{\overline{x}\mathtt{in}(\vec{y})}{\Longrightarrow} P_i'$. *Then* $P_1 \cong P_2$ *iff* $P_1' \cong P_2'$.

(5) *Let* $\vdash P_{1,2} \triangleright\ !\Gamma \cdot x\!:\!(\vec{\tau}\rho)^{!}_{\Delta}$ *and* $P_i \stackrel{x(\vec{y}z)}{\Longrightarrow} P_i'$. *Then* $P_1 \cong P_2$ *iff* $P_1' \cong P_2'$ *iff, for each* $\vdash R \triangleright \vec{y}\!:\!\vec{\tau} \cdot \overline{\Delta}$, *we have* $(\boldsymbol{\nu}\, \vec{y})(R|P_1) \cong (\boldsymbol{\nu}\, \vec{y})(R|P_2)$.

(6) *Let* $\vdash P_{1,2} \triangleright\ !!_{rw}\Gamma \cdot x : ref(\tau)$ *and* $P_i \stackrel{x\mathsf{read}(c)}{\Longrightarrow} P_i'$. *Then, setting* $P_i'' \stackrel{def}{=} C_i[\mathbf{0}]$ *with* $P_i' \stackrel{def}{=} C_i[\mathsf{Ref}\langle xz\rangle]$, $P_1 \cong P_2$ *iff* $P_1' \cong P_2'$ *iff* $P_1'' \cong P_2''$.

Notation 1. Below and henceforth we write $\mathsf{Ref}\langle x, (c)R\rangle$ for $(\boldsymbol{\nu}\, c)(\mathsf{Ref}\langle xc\rangle|R)$ where R is a replicated input with $\mathsf{fn}(R) = \{c\}$ (intuitively $\mathsf{Ref}\langle x, (c)R\rangle$ denotes a reference with subject x whose content is R with its subject abstracted).

Proof. (1), (2), (3-a) and (4) are as in Lemma 1 (note (2) holds even in stateful environment). (3-b) needs a separate treatment from (3-a) since a thread can have effects in its action type: however if the process has already converged, it can be typed without effects. Note (5) adds arbitrary references of the given type, in comparison with Lemma 1 (5). For (6), which is new, we first set $P_i \cong R_i|\mathsf{Ref}\langle x, (z)S_i\rangle$ by (3-a). Since $\mathsf{Ref}\langle x, (z)S_1\rangle \cong \mathsf{Ref}\langle x, (z)S_2\rangle$ iff $S_1 \cong S_2$ (cf. Lemma 23 below) and noting $C_i[\mathbf{0}] \cong R_i|\overline{c}(z)S_i$, we reason:

$$\begin{array}{lcl} P_1 \cong P_2 & \equiv & R_1|\mathsf{Ref}\langle x, (z)S_1\rangle \cong R_2|\mathsf{Ref}\langle x, (z)S_1\rangle \\ & \equiv & R_1 \cong R_2 \text{ and } \mathsf{Ref}\langle x, (z)S_1\rangle \cong \mathsf{Ref}\langle x, (z)S_1\rangle \\ & \equiv & R_1 \cong R_2 \text{ and } S_1 \cong S_2 \\ & \equiv & R_1 \cong R_2 \text{ and } \overline{c}(z)S_1 \cong \overline{c}(z)S_2 \\ & \equiv & R_1 \cong R_2 \text{ and } C_1[\mathbf{0}] \cong C_2[\mathbf{0}]. \quad \square \end{array}$$

Lemma 20. (1) $P^{\Gamma}_{1,2} \models_{\mathsf{b}} \xi$ *implies* $P_1 \cong P_2$. (2) $P^{\Gamma}_1 \models_{\mathsf{b}} \xi$ *and* $P_1 \cong P_2$ *then* $P^{\Gamma}_2 \models_{\mathsf{b}} \xi$. (3) *Let* $\xi_{1,2}$ *be definable. Then* $\xi_1 \simeq_{\mathsf{b}} \xi_2$ *iff* $P_1 \cong P_2$ *for some* $P_{1,2}$ *defining* $\xi_{1,2}$ *respectively iff* $P_1 \cong P_2$ *for any* $P_{1,2}$ *defining* $\xi_{1,2}$ *respectively.*

Proof. The proofs of (2) and (3) are as for Lemma 2. (1) adds the case when $\vdash P_{1,2} \triangleright\ !!_{rw}\Gamma \cdot x : ref(\tau)$. Assume so, and let $P_{1,2} \models_{\mathsf{b}} !!_{rw}\xi \cdot x : \mathsf{Ref}(\alpha)$. By assumption we have $P_{1,2} \stackrel{x\mathsf{read}(c)}{\longrightarrow} C_{1.2}[\mathsf{Ref}\langle xz_{1,2}\rangle]$ s.t. $C_{1,2}[\mathbf{0}] \models_{\mathsf{b}} \xi \cdot c : (\alpha)^{\uparrow}$. By induction, we have $C_1[\mathbf{0}] \cong C_2[\mathbf{0}]$. Using Lemma 19 (6) we are done. $\square$

The proofs of Corollary 3 and Lemma 21 are the same as those for Corollary 1 and Lemma 3 (the clause (1) of the latter given a refinement for divergence as in Lemma 14), hence are omitted.

Corollary 3. (1) *If* $P_1 \models^{\mathcal{I}} A$ *and* $P_1 \cong P_2$ *then* $P_2 \models^{\mathcal{I}} A$. (2) *If* $P \models^{\mathcal{I}} A$ *and* $P \models_{\mathsf{b}} \xi$ *then* $\xi \models^{\mathcal{I}} A$.

Lemma 21. (1) *Let* $\mathsf{fn}(\Gamma) \cap \mathsf{fn}(\Delta) = \emptyset$. *Then* $P \models_{\mathsf{b}} \xi$ *implies* $P^{\Gamma}|Q^{\Delta} \models_{\mathsf{b}} \xi^{\Gamma} \cdot \xi'^{\Delta}$ *for some* ξ'. *Further if* $\omega \notin \mathsf{ran}(\xi \cup \xi')$ *then* $P^{\Gamma}|Q^{\Delta} \models_{\mathsf{b}} \xi^{\Gamma} \cdot \xi'^{\Delta}$ *iff* $P \models_{\mathsf{b}} \xi$ *and* $Q \models_{\mathsf{b}} \xi'$. (2) *If* $P \models_{\mathsf{b}} \xi \cdot x\!:\!\alpha^{!}$ *then* $(\boldsymbol{\nu}\, x)P \models_{\mathsf{b}} \xi$.

Below the operation $\bullet$ is understood as a function application whose result is a pair of a behavioural constant and a model of reference types.

Lemma 22. *Let* $\alpha_1 \simeq_{\mathsf{b}} \alpha_2$, $\vec{\beta}_1 \simeq_{\mathsf{b}} \vec{\beta}_2$, *and* $\xi_1 \cong \xi_2$. *If* $\{\xi_1\}\alpha_1 \bullet \vec{\beta}_1 = \langle \gamma_1,\ \xi_1' \rangle$ *and* $\{\xi_2\}\alpha_2 \bullet \vec{\beta}_2 = \langle \gamma_2,\ \xi_2' \rangle$ *then:* (1) $\gamma_1 \simeq_{\mathsf{b}} \gamma_2$; *and* (2) *if* $\gamma_1 \neq \omega$, *then* $\xi_1' \simeq_{\mathsf{b}} \xi_2'$.

Proof. Below let α_i be monadic for brevity, i range over $\{1,2\}$, and $C[P_i][R_i][S_i]$ stand for $(\boldsymbol{\nu}\, x)(P_i|\overline{x}(yz)(R_i|[z \to w]))|S_i$ ($[z \to w]$ is the standard copy-cat).

$$
\begin{array}{lll}
P_i \models_{\mathsf{b}} x{:}\alpha_i, R_i \models_{\mathsf{b}} y{:}\beta_i, S_i \models_{\mathsf{b}} \xi_i, \alpha_1 \simeq_{\mathsf{b}} \alpha_2, \beta_1 \simeq_{\mathsf{b}} \beta_2, \xi_1 \simeq_{\mathsf{b}} \xi_2 & & \\
\quad \supset \quad C[P_i][R_i][S_i] \models_{\mathsf{b}} \xi_i' \cdot z : \gamma_i \quad \wedge & & \text{(def. of } \models_{\mathsf{b}}) \\
\quad\quad\ \ C[P_1][R_1][S_1] \ \cong\ C[P_2][R_2][S_2] & & \text{(Lemma 20 (3))} \\
\quad \supset \quad \xi_1' \cdot z : \gamma_1 \simeq_{\mathsf{b}} \xi_2' \cdot z : \gamma_2 & & \text{(Lemma 20 (3))} \\
\quad \supset \quad \gamma_1 \simeq_{\mathsf{b}} \gamma_2 \ \wedge\ (\gamma_i \neq \omega \supset \xi_1' \simeq_{\mathsf{b}} \xi_2') & & \text{(Lemma 21 (1)).} \quad \Box
\end{array}
$$

Lemma 23. $\mathsf{Ref}(\alpha_1) \simeq_{\mathsf{b}} \mathsf{Ref}(\alpha_2)$ *iff* $\alpha_1 \simeq_{\mathsf{b}} \alpha_2$.

Proof. We only show "only if" direction. The other direction is simpler. Let $C[\cdot] \stackrel{\text{def}}{=} [\cdot] \,|\, !w(\vec{u}v).\overline{x}\mathsf{read}(c)c(w).\overline{w}(\vec{u'}v')(\Pi![u_i' \to u_i]|[v' \to v])$. Note:

$$(*) \qquad C[\mathsf{Ref}\langle x, (y)R_i\rangle] \approx \mathsf{Ref}\langle x, (y)R_i\rangle | R_i.$$

Further let $\mathsf{Ref}\langle x, (y)R_i\rangle \models_{\mathsf{b}} x : \mathsf{Ref}(\alpha_i)$. We infer:

$$
\begin{array}{lll}
\mathsf{Ref}(\alpha_1) \simeq_{\mathsf{b}} \mathsf{Ref}(\alpha_2) \quad \supset \quad \mathsf{Ref}\langle x, (y)R_1\rangle \cong \mathsf{Ref}\langle x, (y)R_2\rangle & & \text{(Lem. 20 (3))} \\
\quad\quad\quad\quad\quad\quad\ \ \supset \quad C[\mathsf{Ref}\langle x, (y)R_1\rangle] \cong C[\mathsf{Ref}\langle x, (y)R_2\rangle] & & \text{(congruency)} \\
\quad\quad\quad\quad\quad\quad\ \ \supset \quad \mathsf{Ref}\langle x, (y)R_1\rangle | R_1 \cong \mathsf{Ref}\langle x, (y)R_2\rangle | R_2 & & (*) \\
\quad\quad\quad\quad\quad\quad\ \ \supset \quad R_1 \cong R_2 & & \text{(Lem. 19 (2))} \\
\quad\quad\quad\quad\quad\quad\ \ \supset \quad \alpha_1 \simeq_{\mathsf{b}} \alpha_2 & & \text{(Lem. 20 (3)).} \ \Box
\end{array}
$$

Lemma 24. *If* $P \models_{\mathsf{b}} \xi_{1,2}$ *then* $[\![a]\!]_{\mathcal{I}\cdot\xi_1} \simeq_{\mathsf{b}} [\![a]\!]_{\mathcal{I}\cdot\xi_2}$.

Proof. We add the case of $a = !x$ to the proof of Lemma 5, which is immediate since if $\xi_1 \simeq_{\mathsf{b}} \xi_2$ and $\xi_i(x) = \mathsf{Ref}(\alpha_i)$ $(i = 1,2)$ then $\mathsf{Ref}(\alpha_1) \simeq_{\mathsf{b}} \mathsf{Ref}(\alpha_2)$ which implies $\alpha_1 \simeq_{\mathsf{b}} \alpha_2$ by Lemma 22. $\Box$

Lemma 25. *Let* $P \models_{\mathsf{b}} \xi_{1,2}$. *Then* $\xi_1 \models^{\mathcal{I}} A$ *iff* $\xi_2 \models^{\mathcal{I}} A$.

Proof. Identical with the proof of Lemma 6 except for treating $\forall x^{ref(\tau)}.A$ and $\exists x^{ref(\tau)}.A$, which is immediate from the induction hypothesis (since the definition of $\models^{\mathcal{I}}$ for these constructs involve name substitution on A, we use induction on the size of A instead of structural induction). $\Box$

The proofs of Lemma 26, Corollary 4, Lemma 27 and Lemma 28 below precisely follow those for Lemma 7, Corollary 2, Lemma 8 and Lemma 9, hence omitted.

Lemma 26. *If* $\xi \models^{\mathcal{I}} A$ *and* $A \supset B$ *then* $\xi \models^{\mathcal{I}} B$.

Corollary 4. *If* $P \models^{\mathcal{I}} A$ *and* $A \supset B$ *then* $P \models^{\mathcal{I}} B$.

Lemma 27. *Let* $\mathsf{dom}(\xi') \cap \mathsf{fn}(A) = \emptyset$. *Then* $\xi \cdot \xi' \models^{\mathcal{I}} A$ *iff* $\xi \models^{\mathcal{I}} A$.

Lemma 28. (monotonicity of $\boldsymbol{\nu}$) *Let* $x \notin \mathsf{fn}(A)$. *Then* $P \models^{\mathcal{I}} A$ *iff* $(\boldsymbol{\nu}\, x)P \models^{\mathcal{I}} A$.

Lemma 29. (cut in $\models^{\mathcal{I}}$) *Let* $\Gamma_0 \subset \Gamma$ *and* $\Gamma';;\Theta \vdash A$ *with* $!\Gamma' \subset \Gamma$ *and* $\Theta \vdash I$.

(1) $P^{\overline{\Gamma_0};\Delta} \models \mathbf{rely}\ A_0\ \mathbf{guar}\ B$ *and* $R^{\Gamma} \models^{\mathcal{I}} A \wedge A_0$ *imply* $P|R \models^{\mathcal{I}} A \wedge B$.
(2) $P^{\overline{\Gamma_0}\cdot\overline{\Gamma_1};\Delta} \models \{C\}\ \mathbf{rely}\ A_0\ \mathbf{guar}\ B\ \{C'\}$ *and* $R^{\Gamma\cdot\Gamma_1} \models^{\mathcal{I}} C \wedge A \wedge A_0$ *with* $?_{rw}\overline{\Gamma_1};;\Theta \vdash C$ *imply* $P|R \models^{\mathcal{I}} C' \wedge A \wedge B$.

Remark 3. The essence of (2) is that the formula A continues to hold even after a state change at Γ.

Proof. The proof of (1) is identical with Lemma 10. For (2), letting $\Gamma = \Gamma_0 \cdot \Gamma_2$:

$$
\begin{array}{lll}
R^{\Gamma\cdot\Gamma_1} \models^{\mathcal{I}} C \wedge A \wedge A_0 & & \\
\quad \supset \quad R^{\Gamma\cdot\Gamma_1} \models^{\mathcal{I}} A \quad \wedge \quad R^{\Gamma\cdot\Gamma_1} \models^{\mathcal{I}} C \wedge A_0 & & (\wedge) \\
\quad \supset \quad R^{\Gamma\cdot\Gamma_1} \models^{\mathcal{I}} A \quad \wedge \quad (\boldsymbol{\nu}\,\mathsf{fn}(\Gamma_2))R \models^{\mathcal{I}} C \wedge A_0 & & (\text{Lemma 28}) \\
\quad \supset \quad R^{\Gamma\cdot\Gamma_1} \models^{\mathcal{I}} A \quad \wedge \quad (\boldsymbol{\nu}\,\mathsf{fn}(\Gamma))(P|R) \models^{\mathcal{I}} B \wedge C' & & (\text{IH}) \\
\quad \supset \quad R^{\Gamma_0\cdot\Gamma_1} \models^{\mathcal{I}} A \quad \wedge \quad P|R \models^{\mathcal{I}} B \wedge C' & & (\text{Lemma 28}) \\
\quad \supset \quad P|R \models^{\mathcal{I}} A \quad \wedge \quad P|R \models^{\mathcal{I}} B \wedge C' & & (\text{Lemma 27}) \\
\quad \supset \quad P|R \models^{\mathcal{I}} A \wedge B \wedge C' & & (\wedge). \qquad \square
\end{array}
$$

The proofs of the following two Lemmas are identical with those of Lemmas 11 and 12, hence are omitted.

Lemma 30. *Let* $x \notin \mathsf{fn}(A, a)$ *and* $[\![a]\!]_{\mathcal{I}\cdot\xi} \neq \omega$. *Then* $\xi \models^{\mathcal{I}} A[a/x]$ *iff* $\xi \models^{\mathcal{I}} \exists x.(A \wedge x = a)$.

Lemma 31. $\xi \cdot x{:}\mathtt{in}_i(\vec{\alpha})^{\uparrow} \models^{\mathcal{I}} A$ *iff* $\xi \cdot \vec{y}{:}\vec{\alpha} \models^{\mathcal{I}} A[\mathtt{in}_i(\vec{y})^{\uparrow}/x]$.

Lemma 32. *Let* $\mathsf{fn}(B) \cap \{x\vec{y}\} = \emptyset$ *and* $\tau = (\vec{\rho}\tau)^{!}_{\Delta}$ *with* $\mathsf{dom}(\Delta) = \vec{w}$. *Then having* $\xi \cdot x^{\tau} : \wedge_{i\in J}\{\xi_i\}(\vec{\alpha}_i\beta_i)^{!}\{\xi_i'\} \cdot \vec{y} : \vec{\alpha} \models^{\mathcal{I}} \langle\{C\}x \bullet \vec{y}\{C'\}/z\rangle B$ *is logically equivalent to having* $\xi \cdot z : \beta_i \models^{\mathcal{I}} B$ *and* $\xi/\vec{w} \cdot \xi_i' \models^{\mathcal{I}} C'$ *whenever* $\vec{\alpha}_i = \vec{\alpha}$ *and* $\xi/\vec{w} \cdot \xi_i \models^{\mathcal{I}} C$. $\square$

Proof. Direct from the definition. $\square$

4.5 Refined Typing Rules

By the distinction between threads and non-threads in the sequent, we need to use a refined form of typing rules, especially for parallel composition. We shall use the following observations for organising the typing and proof rules.

P1: Two threads can only be composed at a compensating linear channel to generate another thread, because of the sequentiality constraint on parallel composition (note this means at least one of them should be passive).
P2: Without losing generality, composition of threads can always be done first between pure threads then between a pure thread and a non-thread, up to associativity of parallel composition.
P3: By this the application of (Rec) is in fact only needed for non-threads up to strong bisimilarity.

This leads to the refined rules which give clear articulation for the proof rules of the present logic. We give the complete set of the refined typing rules in Figure 6, which can type the same set of processes as we have presented in Section 4.1 except for the case when a process contains both a linear input and a reference (which does not jeopardise the derivation of typed processes of other forms).

4.6 Soundness of Proof Rules

The proof rules are given in Figure 7 (composition rules and consequence rules) and Figure 8 (prefix rules). We assume each proof rule is applied following the typing rule of the corresponding name, even though we almost always omit the typing annotations for legibility. As before, we assume that, in each rule, all processes, formulae, and sequents are well-typed and follow the standard binding convention over processes and formulae. These rules reduce to, when restricted to affinely typed processes, those of Figures 2 and 4 (in the refined form based on thread/non-thread distinction). In (Write), $\mathsf{prim}(A)$ is the primary names used in A (induced from the underlying typing). Thus "$A \supset^x B$" means A specifies more than B concerning x, but not other primary names. We now prove:

Theorem 3. (soundness of proof rules for logic with open state) *Let P be a process with open state. Then $P^{\Gamma;\Delta} \vdash \mathbf{rely}\, A\, \mathbf{guar}\, B$ implies $P^{\Gamma;\Delta} \models \mathbf{rely}\, A\, \mathbf{guar}\, B$. Also $P^{\Gamma;\Delta} \vdash \{C\}\, \mathbf{rely}\, A\, \mathbf{guar}\, B\, \{C'\}$ implies $P^{\Gamma;\Delta} \models \{C\}\, \mathbf{rely}\, A\, \mathbf{guar}\, B\, \{C'\}$.*

Proof. The proofs are quite similar to those which we have carried out so far except for the treatment of state change. The proofs of:

$$(\mathsf{Zero})\ \frac{-}{\mathbf{0}^{\emptyset} \vdash \mathbf{rely}\, A\, \mathbf{guar}\, A} \qquad (\mathsf{Res\text{-}nonthread})\ \frac{P \vdash \mathbf{rely}\, A\, \mathbf{guar}\, B^{\text{-}x}}{(\boldsymbol{\nu}\, x)P \vdash \mathbf{rely}\, A\, \mathbf{guar}\, B}$$

remain identical. For (Res-thread) we only treat hiding of a replicated name.

$$(\mathsf{Res\text{-}thread})\ \frac{P \vdash \{C\}\, \mathbf{rely}\, A\, \mathbf{guar}\, B^{\text{-}x}\, \{C'\}}{(\boldsymbol{\nu}\, x)P \vdash \{C\}\, \mathbf{rely}\, A\, \mathbf{guar}\, B\, \{C'\}}$$

Let $P^{\overline{\Gamma};\Delta\cdot x:\tau}$, A^{Γ} and B^{Δ} with $\mathsf{md}(\tau) = !$. Note $x \notin \mathsf{fn}(C, C')$ since x is a primary name of a $!$-type.

$$\begin{array}{lll} R^{\Gamma} \models^{\mathcal{I}} A \wedge C & \supset & (\boldsymbol{\nu}\, \mathsf{fn}(\Gamma))(P|R) \models^{\mathcal{I}} B \wedge C' \qquad \text{(IH)} \\ & \supset & (\boldsymbol{\nu}\, x)(\boldsymbol{\nu}\, \mathsf{fn}(\Gamma))(P|R) \models^{\mathcal{I}} B \wedge C' \qquad \text{(Lemma 28)}. \end{array}$$

For the refinement of (Par) for sequential composition:

$$(\mathsf{Par}\text{-}\downuparrows)\ \frac{P_1 \vdash \{C\}\, \mathbf{rely}\, A_1\, \mathbf{guar}\, E\, \{C''\} \quad P_1 \vdash \{C''\}\, \mathbf{rely}\, A_2 \wedge E\, \mathbf{guar}\, B\, \{C'\}}{P_1|P_2 \vdash \{C\}\, \mathbf{rely}\, A_1 \wedge A_2\ \mathbf{guar}\, B\, \{C'\}}$$

is treated as, assuming $P_1^{\overline{\Gamma_1};\Delta_1}$ and $P_2^{\overline{\Gamma_2};\Delta_2}$:

$$\begin{array}{lll} R^{\Gamma} \models^{\mathcal{I}} C \wedge A_1 \wedge A_2 & & \\ \quad \supset & P_1|R \models^{\mathcal{I}} C'' \wedge A_2 \wedge (B_1 \wedge E) & \text{(IH, Lem. 29)} \\ \quad \supset & P_1|P_2|R \models^{\mathcal{I}} C' \wedge B_1 \wedge B_2 & \text{(IH, Lem. 29)} \\ \quad \supset & (\boldsymbol{\nu}\, \mathsf{fn}(\Gamma_1 \cup \Gamma_2))(P_1|P_2|R) \models^{\mathcal{I}} C' \wedge B_1 \wedge B_2 & \text{(Lem. 28).} \end{array}$$

(Zero)

$$\frac{-}{\vdash \mathbf{0} \triangleright \_}$$

(Par-⇵)

$$\frac{\vdash P_1 \triangleright \Gamma_1 ;\ x{:}\rho^{\uparrow} \qquad \vdash P_2 \triangleright \Gamma_2, x{:}\overline{\rho}^{\downarrow} ;\ y : \tau^{\uparrow}}{\vdash P_1 | P_2 \triangleright \Gamma_1 \cup \Gamma_2,\ x{:}\updownarrow ;\ y{:}\tau}$$

(Par-⁉)

$$\frac{\vdash P_1 \triangleright \Gamma_1 ;\ \mathbin{!!}_{rw}\Delta_1, \mathbf{!}\Theta \qquad \vdash P_2 \triangleright \Gamma_2, \mathbf{?}\overline{\Theta} ;\ \mathbin{!!}_{rw}\Delta_2}{\vdash P_1 | P_2 \triangleright \Gamma_1 \cup \Gamma_2 ;\ \Delta_1, \Delta_2, \Theta}$$

(Par-⁉-thread)

$$\frac{\vdash P_1 \triangleright \Gamma_1 ;\ \mathbin{!!}_{rw}\Delta, \mathbin{!!}_{rw}\Theta \qquad \vdash P_2 \triangleright \Gamma_2, \mathbin{??}_{rw}\overline{\Theta} ;\ x{:}\tau^{\uparrow}}{\vdash P_1 | P_2 \triangleright \Gamma_1 \cup \Gamma_2 ;\ \Delta, \Theta, x{:}\tau}$$

(Rec)

$$\frac{\vdash P \triangleright \mathbf{?}\Gamma,\ y{:}\overline{\tau}^{?},\ \mathbf{!}\Delta,\ x{:}\tau^{!}}{\vdash \mu y{=}x.P \triangleright \Gamma,\ \Delta,\ x{:}\tau}$$

(Res-thread) $(\uparrow \in \mathsf{md}(\Gamma))$

$$\frac{\vdash P \triangleright \Gamma, x{:}\tau \quad \mathsf{md}(\tau) \in \{\updownarrow, \mathbf{!}\}}{\vdash (\boldsymbol{\nu}\, x)P \triangleright \Gamma}$$

(Res-nonthread) $(\uparrow \notin \mathsf{md}(\Gamma))$

$$\frac{\vdash P \triangleright \Gamma, x{:}\tau \quad \mathsf{md}(\tau) \in \{\updownarrow, \mathbf{!}\}}{\vdash (\boldsymbol{\nu}\, x)P \triangleright \Gamma}$$

(Weak)

$$\frac{\vdash P \triangleright \Gamma, z{:}\rho^{\uparrow} \quad \mathsf{md}(\tau) \in \{\updownarrow, \mathbf{?}\}}{\vdash P \triangleright \Gamma, z{:}\rho^{\uparrow}, x{:}\tau}$$

(Bra$^{\downarrow}$)

$$\frac{\forall i.\ \vdash P_i \triangleright \mathbin{??}_{rw}\Gamma^{\text{-}x},\ \vec{y}_i{:}\vec{\tau}_i,\ v{:}\rho^{\uparrow}}{\vdash x[\&_i(\vec{y}_i).P_i] \triangleright \Gamma,\ x{:}[\&_i\vec{\tau}_i]^{\downarrow},\ v{:}\rho}$$

(Sel$^{\uparrow}$)

$$\frac{\vdash P \triangleright \mathbin{??}_{rw}\Gamma,\ \vec{y}{:}\vec{\tau}_i}{\vdash \overline{x}\mathtt{in}_i(\vec{y})P \triangleright \Gamma,\ x{:}[\oplus_{i\in I}\vec{\tau}_i]^{\uparrow}}$$

(In$^{!}$)

$$\frac{\vdash P \triangleright \mathbf{?}\Gamma^{\text{-}x},\ \mathbf{?}_{rw}\Delta^{\text{-}x},\ \vec{y}z{:}\vec{\tau}\rho}{\vdash !\, x(\vec{y}z).P \triangleright \Gamma,\ x{:}(\vec{\tau}\rho)^{!}_{\Delta}}$$

(Out$^{?}$) $(\Gamma(x){=}(\vec{\tau}\rho)^{?}_{\Delta},\ \Delta \subset \Theta)$

$$\frac{\vdash R \triangleright \mathbf{?}\Gamma,\ \vec{y}{:}\vec{\tau} \qquad \vdash P \triangleright \mathbf{?}\Gamma,\ \mathbf{?}_{rw}\Theta,\ z{:}\rho^{\downarrow},\ v{:}\sigma^{\uparrow}}{\vdash \overline{x}(\vec{y}z)(R|P) \triangleright \Gamma,\ \Theta,\ v{:}\sigma}$$

(Ref)

$$\frac{-}{\vdash \mathsf{Ref}\langle xy\rangle \triangleright x{:}\mathit{ref}(\tau), y{:}\overline{\tau}}$$

(Read) $(\Gamma(x) = rw(\tau))$

$$\frac{\vdash P \triangleright \uparrow \mathbin{??}_{rw}\Gamma,\ c{:}(\overline{\tau})^{\downarrow}}{\vdash \overline{x}\mathsf{read}(c)P \triangleright \Gamma}$$

(Write) $(\Gamma(x) = rw(\tau))$

$$\frac{\vdash P \triangleright \uparrow \mathbin{??}_{rw}\Gamma,\ v{:}\tau,\ c{:}()^{\downarrow}}{\vdash \overline{x}\mathsf{write}(vc)P \triangleright \Gamma}$$

Fig. 6. Refined Typing Rules for Open State

(Zero)
$$\frac{-}{\mathbf{0}^{\emptyset} \vdash \mathbf{rely}\ A\ \mathbf{guar}\ A}$$

(Par-⇵)
$$\frac{P_1 \vdash \{C\}\ \mathbf{rely}\ A_1\ \mathbf{guar}\ E\ \{C''\} \quad P_1 \vdash \{C''\}\ \mathbf{rely}\ A_2 \wedge E\ \mathbf{guar}\ B\ \{C'\}}{P_1|P_2 \vdash \{C\}\ \mathbf{rely}\ A_1 \wedge A_2\ \mathbf{guar}\ B\ \{C'\}}$$

(Par-!?)
$$\frac{P_1 \vdash \mathbf{rely}\ A_1\ \mathbf{guar}\ B_1 \wedge E \quad P_2 \vdash \mathbf{rely}\ A_2 \wedge E\ \mathbf{guar}\ B_2}{P_1|P_2 \vdash \mathbf{rely}\ A_1 \wedge A_2\ \mathbf{guar}\ B_1 \wedge B_2}$$

(Par-$!!_{rw}??_{rw}$)
$$\frac{P_1 \vdash \mathbf{rely}\ A_1\ \mathbf{guar}\ B_1 \wedge D \wedge E \quad P_2 \vdash \{C \wedge D\}\ \mathbf{rely}\ A_2 \wedge E\ \mathbf{guar}\ B_2\ \{C' \wedge D'\}}{P_1|P_2 \vdash \{C\}\ \mathbf{rely}\ A_1 \wedge A_2\ \mathbf{guar}\ B_1 \wedge B_2 \wedge D'\ \{C'\}}$$

(Rec)
$$\frac{P \vdash \mathbf{rely}\ A^{-y}\ \mathbf{guar}\ B(0) \quad P \vdash \mathbf{rely} A \wedge B(i)[y/x]\ \mathbf{guar}\ B(i+1)}{\mu y = x.P \vdash \mathbf{rely}\ A\ \mathbf{guar}\ B}$$

(Res-thread)
$$\frac{P \vdash \{C\}\ \mathbf{rely}\ A\ \mathbf{guar}\ B^{-x}\ \{C'\}}{(\boldsymbol{\nu}\, x)P \vdash \{C\}\ \mathbf{rely}\ A\ \mathbf{guar}\ B\ \{C'\}}$$

(Res-nonthread)
$$\frac{P \vdash \mathbf{rely}\ A\ \mathbf{guar}\ B^{-x}}{(\boldsymbol{\nu}\, x)P \vdash \mathbf{rely}\ A\ \mathbf{guar}\ B}$$

(Weak-thread)
$$\frac{P^{\Gamma} \vdash \{C\}\ \mathbf{rely}\ A\ \mathbf{guar}\ B\ \{C'\}}{P^{\Gamma, x:\tau} \vdash \{C\}\ \mathbf{rely}\ A\ \mathbf{guar}\ B\ \{C'\}}$$

(Consequence-thread)
$$\frac{(C \supset C_0,\ A \supset A_0,\ B_0 \supset B,\ C_0' \supset C') \quad P \vdash \{C_0\}\ \mathbf{rely}\ A_0\ \mathbf{guar}\ B_0\ \{C_0'\}}{P \vdash \{C\}\ \mathbf{rely}\ A\ \mathbf{guar}\ B\ \{C'\}}$$

(Consequence-nonthread) $(A \supset A_0,\ B_0 \supset B)$
$$\frac{P \vdash \mathbf{rely}\ A_0\ \mathbf{guar}\ B_0}{P \vdash \mathbf{rely}\ A\ \mathbf{guar}\ B}$$

Fig. 7. Proof Rules for Processes with Open State (composition/consequence)

The second (Par) rule:

$$\text{(Par-!?)}\ \frac{P_1 \vdash \mathbf{rely}\ A_1\ \mathbf{guar}\ B_1 \wedge E \quad P_2 \vdash \mathbf{rely}\ A_2 \wedge E\ \mathbf{guar}\ B_2}{P_1|P_2 \vdash \mathbf{rely}\ A_1 \wedge A_2\ \mathbf{guar}\ B_1 \wedge B_2}$$

is reasoned precisely as in Theorem 1. The third (Par) rule:

$$\text{(Par-}!!_{rw}??_{rw}\text{)}\ \frac{P_1 \vdash \mathbf{rely}\ A_1\ \mathbf{guar}\ B_1 \wedge D \wedge E \quad P_2 \vdash \{C \wedge D\}\ \mathbf{rely}\ A_2 \wedge E\ \mathbf{guar}\ B_2\ \{C' \wedge D'\}}{P_1|P_2 \vdash \{C\}\ \mathbf{rely}\ A_1 \wedge A_2\ \mathbf{guar}\ B_1 \wedge B_2 \wedge D'\ \{C'\}}$$

is a simpler case of (Par-⇵). The recursion rule:

$$\text{(Rec)}\ \frac{P \vdash \mathbf{rely}\ A^{-y}\ \mathbf{guar}\ B(0) \quad P \vdash \mathbf{rely} A \wedge B(i)[y/x]\ \mathbf{guar}\ B(i+1)}{\mu y = x.P \vdash \mathbf{rely}\ A\ \mathbf{guar}\ B}$$

$$(\mathsf{Bra}^{\downarrow})\quad \frac{\forall i.\ P_i \vdash \{C\}\ \mathbf{rely}\ A[\mathtt{in}_i(\vec{y}_i)^{\uparrow}/x]\ \mathbf{guar}\ B\ \{C'\}}{x[\&_i(\vec{y}_i).P_i] \vdash \{C\}\ \mathbf{rely}\ A\ \mathbf{guar}\ B\ \{C'\}}$$

$$(\mathsf{Sel}^{\uparrow})\quad \frac{P \vdash \mathbf{rely}\ A\ \mathbf{guar}\ B[\mathtt{in}_i(\vec{y})^{\uparrow}/x]}{\overline{x}\mathtt{in}_i(\vec{y})P \vdash \{C\}\ \mathbf{rely}\ A\ \mathbf{guar}\ B\ \{C\}}$$

$$(\mathsf{In}^{!})\quad \frac{(\forall \vec{y}\vec{l}.(B_1 \supset \langle\{C\}x\bullet\vec{y}\{C'\}/z\rangle B_2) \supset B)\quad P \vdash \{C\}\ \mathbf{rely}\ A^{\vec{y}\vec{l}} \wedge B_1^{\vec{y}\vec{l}}\ \mathbf{guar}\ B_2\ \{C'\}}{!x(\vec{y}z).P \vdash \mathbf{rely}\ A\ \mathbf{guar}\ B}$$

$$(\mathsf{Out}^{?})\quad \frac{(A \supset A_1 \supset \langle\{C\}x\bullet\vec{y}\{C'\}/z\rangle A_2)\quad R \vdash \mathbf{rely}\ A\ \mathbf{guar}\ A_1\quad P \vdash \{C\}\ \mathbf{rely}\ A \wedge A_2\ \mathbf{guar}\ B\ \{C'\}}{\overline{x}(\vec{y}z)(R|P) \vdash \{C\}\ \mathbf{rely}\ A\ \mathbf{guar}\ B\ \{C'\}}$$

$$(\mathsf{Ref})\quad \frac{(A[!x/y] \supset B)\quad -}{\mathsf{Ref}\langle xy\rangle \vdash \mathbf{rely}\ A\ \mathbf{guar}\ B}$$

$$(\mathsf{Read})\quad \frac{(C \supset A'[(!x)^{\uparrow}/c])\quad P \vdash \{C\}\ \mathbf{rely}\ A \wedge A'\ \mathbf{guar}\ B\ \{C'\}}{\overline{x}\mathsf{read}(c)P \vdash \{C\}\ \mathbf{rely}\ A\ \mathbf{guar}\ B\ \{C'\}}$$

$$(\mathsf{Write})\quad \frac{(E[!x/y] \supset^{x} C)\quad R \vdash \mathbf{rely}\ A^{-c}\ \mathbf{guar}\ E\quad P \vdash \{C\}\ \mathbf{rely}\ A^{-c}\ \mathbf{guar}\ B\ \{C'\}}{\overline{x}\mathsf{write}(yc)(R|P) \vdash \{\exists x.(C \wedge E[!x/y])\}\ \mathbf{rely}\ A\ \mathbf{guar}\ B\ \{C'\}}$$

$\star$ In (Write), $A \supset^{x} B \stackrel{\text{def}}{=} A \supset \exists \vec{y}.B$ where $\{\vec{y}\} = \mathsf{prim}(B)\backslash\{x\}$.

Fig. 8. Proof Rules for Processes with Open State (Prefix)

is reasoned as before. For the weakening:

$$(\mathsf{Weak\text{-}thread})\ \frac{P^{\Gamma} \vdash \{C\}\ \mathbf{rely}\ A\ \mathbf{guar}\ B\ \{C'\}}{P^{\Gamma\cdot x:\tau} \vdash \{C\}\ \mathbf{rely}\ A\ \mathbf{guar}\ B\ \{C'\}}$$

Let $\mathsf{md}(\tau) = \mathbf{?}_{rw}$ as before and $\Gamma = \Delta;\Theta$. We set:

$$R^{\overline{\Delta}\cdot x:\overline{\tau}} \stackrel{\text{def}}{=} R'^{\Gamma}|R_0^{x:\tau} \qquad \text{such that } R' \models_{\mathsf{b}} \xi \text{ and } R_0 \models_{\mathsf{b}} x : \mathsf{Ref}(\alpha).$$

The following reasoning is identical with Theorem 1 except having states.

$$\begin{array}{lll} R \models^{\mathcal{I}} C \wedge A & \supset & R \models_{\mathsf{b}} \xi \cdot x{:}\mathsf{Ref}(\alpha) \models^{\mathcal{I}} C \wedge A \qquad \text{(Corollary 3)} \\ & \supset & R' \models_{\mathsf{b}} \xi \models^{I\cdot x:\mathsf{Ref}(\alpha)} C \wedge A \qquad \text{(Definition of } \models^{\mathcal{I}}) \\ & \supset & (\boldsymbol{\nu}\,\mathsf{fn}(\Gamma))(P|R') \models^{I\cdot x:\alpha} B \wedge C' \qquad \text{(IH)} \\ & \supset & (\boldsymbol{\nu}\,\mathsf{fn}(\Gamma))(P|R')|(\boldsymbol{\nu}\,x)R_0 \models^{I\cdot x:\alpha} B \wedge C' \qquad \text{(Corollary 3)} \\ & \supset & (\boldsymbol{\nu}\,\mathsf{fn}(\Gamma), x)(P|R) \models^{\mathcal{I}} B \wedge C' \qquad \text{(Lemma 27).} \end{array}$$

Similarly:

$$(\mathsf{Bra}^{\downarrow})\ \frac{\forall i.\ P_i \vdash \{C\}\ \mathbf{rely}\ A[\mathtt{in}_i(\vec{y}_i)^{\uparrow}/x]\ \mathbf{guar}\ B\ \{C'\}}{x[\&_i(\vec{y}_i).P_i] \vdash \{C\}\ \mathbf{rely}\ A\ \mathbf{guar}\ B\ \{C'\}}$$

is reasoned precisely as before except we incorporate state:

$$\begin{array}{lll} R \models_{\mathsf{b}} \vec{w}{:}\vec{\gamma}\cdot x{:}\mathtt{in}_i(\vec{\alpha})^{\uparrow} \models^{\mathcal{I}} A & & \\ \quad \supset \quad R'|R_0 \models_{\mathsf{b}} \vec{w}{:}\vec{\gamma}\cdot\vec{y}{:}\vec{\alpha} \models^{\mathcal{I}} A[\mathtt{in}_i(\vec{y})^{\uparrow}/x] & & \text{(Def. of } \models_{\mathsf{b}}\text{, Lemma 31)} \\ \quad \supset \quad (\boldsymbol{\nu}\,\vec{y}\vec{w})(P_i|R'|R_0) \models^{\mathcal{I}} B & & \text{(IH)} \\ \quad \supset \quad (\boldsymbol{\nu}\,\vec{w}x)(x[\&(\vec{y}_i).P_i]|R'|\overline{x}\mathtt{in}_i(\vec{y})R_0) \models_{\mathsf{b}} B & & \text{(Corollary 3),} \end{array}$$

where we let, w.l.o.g., $R^{\Gamma} \equiv R'|\overline{x}\mathtt{in}_i(\vec{y})R_0$ with $\mathsf{dom}(\Gamma) = \{\vec{w}x\}$. Similarly:

$$(\mathsf{Sel}^{\uparrow})\ \frac{P \vdash \mathbf{rely}\ A\ \mathbf{guar}\ B[\mathtt{in}_i(\vec{y})^{\uparrow}/x]}{\overline{x}\mathtt{in}_i(\vec{y})P \vdash \{C\}\,\mathbf{rely}\,A\,\mathbf{guar}\,B\,\{C\}}$$

Assume the first sequent is under the typing $\vdash P \triangleright \Gamma_1; \vec{y}{:}\vec{\tau}^{\,\sharp}$, while the second sequent is under the typing $\vdash \overline{x}\mathtt{in}_i(\vec{y})P \triangleright ?\Gamma_1 \cdot ?_{rw}\,\Gamma_2; x : \rho^{\uparrow}$ (note Γ_2 is weakened). Further let $\vdash R \triangleright \overline{\Gamma_1}\cdot\overline{\Gamma_2}$ such that $R \cong T|S$ with $\vdash T \triangleright \overline{\Gamma_1}$ and $\vdash S \triangleright \overline{\Gamma_2}$.

$$\begin{array}{lll} R \models_{\mathsf{b}} \vec{w}{:}\vec{\gamma}\cdot !_{rw}\eta \models^{\mathcal{I}} A \wedge C & & \\ \quad \supset \quad (T \models_{\mathsf{b}} \vec{w}{:}\vec{\gamma} \models^{\mathcal{I}} A) \ \wedge\ (S \models_{\mathsf{b}} \eta \models^{\mathcal{I}} C) & & \\ \quad \supset \quad (\boldsymbol{\nu}\,\vec{w})(P|T) \models_{\mathsf{b}} \vec{y}{:}\vec{\alpha} \models^{\mathcal{I}} B[\mathtt{in}_i(\vec{y})^{\uparrow}/x] & & \text{(IH)} \\ \quad \supset \quad \overline{x}\mathtt{in}_i(\vec{y})(\boldsymbol{\nu}\,\vec{w})(P|T) \models_{\mathsf{b}} x{:}\mathtt{in}_i(\vec{\alpha}) \models^{\mathcal{I}} B & & \text{(Def. of } \models_{\mathsf{b}}\text{, Lemma 31)} \\ \quad \supset \quad (\boldsymbol{\nu}\,\vec{w})(\overline{x}\mathtt{in}_i(\vec{y})P \mid T) \models^{\mathcal{I}} B & & \text{(Corollary 3)} \\ \quad \supset \quad (\boldsymbol{\nu}\,\vec{w})(\overline{x}\mathtt{in}_i(\vec{y})P \mid T) \mid S \models^{\mathcal{I}} B \wedge C & & \text{(Lemma 21)} \\ \quad \supset \quad (\boldsymbol{\nu}\,\vec{w})(\overline{x}\mathtt{in}_i(\vec{y})P \mid R) \models^{\mathcal{I}} B \wedge C & & \text{(Corollary 3).} \end{array}$$

We turn to the replicated input.

$$(\mathsf{In}^{!})\ \frac{P \vdash \{C\}\,\mathbf{rely}\,A^{\vec{y}\vec{l}} \wedge B_1^{\vec{y}\vec{l}}\,\mathbf{guar}\,B_2\,\{C'\}, \quad \forall\vec{y}\vec{l}.(B_1 \supset B_2[\{C\}x\bullet\vec{y}\{C'\}/z]) \supset B}{!x(\vec{y}z).P \vdash \mathbf{rely}\ A\ \mathbf{guar}\ B}$$

Let $R^{\Gamma} \models^{\mathcal{I}} A$ with $\mathsf{dom}(\Gamma) = \{\vec{w}\}$. As before, we first construct the behavioural constant $\theta \stackrel{\text{def}}{=} \wedge_i \eta_i(\vec{\alpha}_i\beta_i)^{!}\eta_i'$, this time incorporating state.

$$(\star) \qquad \begin{array}{l} \text{For each } \vec{\alpha}_i^{\vec{\tau}} \text{ and } \eta_i\text{, we choose } S_i \models_{\mathsf{b}} \eta_i\cdot\vec{y}{:}\vec{\alpha}\text{, and let} \\ (\boldsymbol{\nu}\,\vec{w}\vec{y})(P|R|S_i) \models_{\mathsf{b}} z{:}\beta_i\cdot\eta_i'. \end{array}$$

For such θ, we show $(\vec{w})(!x(\vec{y}z).P|R) \models_{\mathsf{b}} x{:}\theta \models^{\mathcal{I}} B$. As before, the reasoning is decomposed into the part for $\models_{\mathsf{b}}$ and the part for $\models^{\mathcal{I}}$.

(1) We first show $(\boldsymbol{\nu}\,\vec{w})(!x(\vec{y}z).P|R) \models_{\mathsf{b}} x{:}\theta$. Since

$$(\boldsymbol{\nu}\,\vec{w})(!x(\vec{y}z).P|R) \xrightarrow{x(\vec{y}z)} (\boldsymbol{\nu}\,\vec{w})(!x(\vec{y}z).P|R)|(\boldsymbol{\nu}\,\vec{w})(P|R),$$

it suffices to show the following statement for each $S \models_{\mathsf{b}} \vec{y}{:}\vec{\alpha}_i\cdot\eta_i$:

$$(\boldsymbol{\nu}\,\vec{y})((\boldsymbol{\nu}\,\vec{w})(P|R) \mid S) \equiv (\boldsymbol{\nu}\,\vec{y}\vec{w})(P|R|S) \models_{\mathsf{b}} z{:}\beta_i\cdot\eta_i'\ .$$

Below S_i is the process used for constructing θ in $(\star)$.

$$\begin{array}{lll} S \models_{\mathsf{b}} \vec{y}{:}\vec{\alpha}_i\cdot\eta_i & & \\ \quad \supset \quad S_i \cong S & & (\star\text{, Lemma 20 (1)}) \\ \quad \supset \quad (\boldsymbol{\nu}\,\vec{y}\vec{w})(P|R|S) \cong (\boldsymbol{\nu}\,\vec{y}\vec{w})(P|R|S_i) \models_{\mathsf{b}} z{:}\beta_i & & \text{(congruency, } \star\text{)} \\ \quad \supset \quad (\boldsymbol{\nu}\,\vec{y}\vec{w})(P|R|S) \models_{\mathsf{b}} z{:}\beta_i\cdot\eta_i' & & \text{(Corollary 3).} \quad \square \end{array}$$

(2) Next we show $x{:}\theta \models^{\mathcal{I}} B$. By $\forall \vec{y}\vec{l}.(B_1 \supset \langle\{C\}x \bullet \vec{y}\{C'\}/z\rangle B_2) \supset B$ it suffices to show $x{:}\theta \models^{\mathcal{I}} \forall \vec{y}\vec{l}.(B_1 \supset \langle\{C\}x \bullet \vec{y}\{C'\}/z\rangle B_2)$ (by Lemma 26). Below we let J be an arbitrary well-typed assignment to $\vec{l}$. S_i is the process used in $(\star)$.

$$
\begin{array}{ll}
x{:}\theta \cdot \vec{y}{:}\vec{\alpha}_i \cdot J \models^{\mathcal{I}} B_1,\ \ \eta_i \models^{\mathcal{I}} C & \\
\equiv\ \vec{y}{:}\vec{\alpha}_i \models^{I\cdot J} B_1,\ \ \eta_i \models^{\mathcal{I}} C & \text{(Lem. 27)} \\
\supset\ (\boldsymbol{\nu}\, \vec{w}\vec{y})(P|S_i|R) \models^{I\cdot J} B_2 \wedge C',\ (\boldsymbol{\nu}\, \vec{w}\vec{y})(P|S_i|R) \models_{\mathsf{b}} z{:}\beta_i \cdot \eta_i' & \text{(IH, } \star\text{)} \\
\supset\ z{:}\beta_i \cdot \eta_i' \models^{I\cdot J} B_2 \wedge C' & \text{(Cor. 3 (2))} \\
\supset\ z{:}\beta_i \models^{I\cdot J} B_2,\ \ \eta_i' \models^{I\cdot J} C' & (\wedge) \\
\supset\ x{:}\theta \cdot \vec{y}{:}\vec{\alpha}_i \cdot J \models^{\mathcal{I}} B_2[x \bullet \vec{y}/z] & \text{(Lem. 32).}
\end{array}
$$

Next we treat the dual case.

$$
(\mathsf{Out}^{?})\ \frac{S \vdash \mathbf{rely} A\ \mathbf{guar} A_1,\ P \vdash \{C''\}\,\mathbf{rely} A \wedge A_2 \mathbf{guar} B\{C'\},\ A \supset A_1 \supset A_2[\{C\}x \bullet \vec{y}\{C''\}/z]}{\overline{x}(\vec{y}z)(S|P) \vdash \{C\}\,\mathbf{rely}\ A\,\mathbf{guar}\,B\ \{C'\}}
$$

Let $\vdash S \triangleright ?\overline{\Gamma}; \Delta$ and $\vdash P \triangleright ?\overline{\Gamma}\cdot ?_{rw}\,\overline{\Theta}\cdot z{:}\rho^{\downarrow}; u{:}\sigma^{\uparrow}$ where $\mathsf{dom}(\Gamma) = \{x\vec{w}\}$, $\mathsf{dom}(\Theta) = \{\vec{u}\}$ and $\mathsf{dom}(\Delta) = \{\vec{y}\}$. By the binder convention, $\mathsf{fn}(\{\vec{y}z\}) \wedge \mathsf{fn}(A, B) = \emptyset$.

Assume $\vdash R \triangleright \Gamma \cdot \Theta$ such that (w.l.o.g. by Lemma 19 (3) and Lemma 20) we have $R \equiv (!x(\vec{y}z).R_0) \mid R' \mid S$ with: (i) $!x(\vec{y}z).R_0 \models_{\mathsf{b}} x{:}\theta$ with $\theta = \wedge_i \eta_i(\vec{\alpha}_i\beta_i)^{!}\eta_i'$, (ii) $R' \models_{\mathsf{b}} \vec{w}{:}\vec{\gamma}$, and (iii) $S \models_{\mathsf{b}} \eta_i$ (note η_i ranges over all states of type Θ, so this loses no generality). Below we let $\xi \stackrel{\text{def}}{=} x{:}\theta \cdot \vec{w}{:}\vec{\gamma}$ and write $\ddagger$ for the condition $A \supset A_1 \supset \langle\{C\}x \bullet \vec{y}\{C'\}/z\rangle A_2$.

$$
\begin{array}{ll}
R \models_{\mathsf{b}} \xi \cdot \eta_i \models^{\mathcal{I}} A \wedge C & \\
\supset\ S|R \models_{\mathsf{b}} \xi \cdot \eta_i \cdot \vec{y}{:}\vec{\alpha} \models^{\mathcal{I}} A \wedge A_1 \wedge C & \text{(IH, Corollary 3)} \\
\supset\ S|R \models_{\mathsf{b}} \xi \cdot \eta_i \cdot \vec{y}{:}\vec{\alpha}_i \models^{\mathcal{I}} A \wedge \langle\{C\}x \bullet \vec{y}\{C'\}/z\rangle A_2 & (\ddagger\text{, Corollary 4}) \\
\supset\ S|R|R_0 \models_{\mathsf{b}} \xi \cdot \eta_i' \cdot \vec{y}{:}\vec{\alpha}_i \cdot z{:}\beta_i, & \text{(Lemma 21 (1))} \\
\quad \xi \cdot \eta_i' \cdot \vec{y}{:}\vec{\alpha}_i \cdot z{:}\beta_i \models_{\mathsf{b}} A \wedge A_2 \wedge C'' & \text{(Lemma 32)} \\
\supset\ (\boldsymbol{\nu}\, \vec{y})(S|R|R_0) \models_{\mathsf{b}} \xi \cdot z{:}\beta_i \models^{\mathcal{I}} A \wedge A_2 \wedge C'' & \text{(Cor. 4, Lem. 21 (2))} \\
\supset\ (\boldsymbol{\nu}\, \vec{w}xz)(P|(\boldsymbol{\nu}\, \vec{y})(S|R|R_0)) \models^{\mathcal{I}} B \wedge C' & \text{(IH)} \\
\supset\ (\boldsymbol{\nu}\, \vec{w}x)(\overline{x}(\vec{y}z)(S|P) \mid R) \models^{\mathcal{I}} B \wedge C' & \text{(Corollary 3)}
\end{array}
$$

In the last step, we use $(\boldsymbol{\nu}\, \vec{w}xz)(\overline{x}(\vec{y}z)(S|P) \mid R) \longrightarrow (\boldsymbol{\nu}\, \vec{w}\vec{y}xz)(S|P|R|R_0)$ as well as $\longrightarrow \subset \cong$ (Lemma 19-1).

For the reference rule:

$$
(\mathsf{Ref})\ \frac{A[!x/y] \supset B}{\mathsf{Ref}\langle xy\rangle \vdash \mathbf{rely}\, A\, \mathbf{guar}\, B}
$$

Let $\vdash R \triangleright y{:}\tau^{!}$ and $A[!x/y] \supset B$ below.

$$
\begin{array}{ll}
R \models^{\mathcal{I}} A & \\
\supset\ \mathsf{Ref}\langle xy\rangle | R \models_{\mathsf{b}} x{:}\mathsf{Ref}(\alpha) \cdot y{:}\alpha \models^{\mathcal{I}} A \wedge !x = y & \text{(Lem. 23/27, } \wedge\text{)} \\
\supset\ (\boldsymbol{\nu}\, y)(\mathsf{Ref}\langle xy\rangle | R) \models_{\mathsf{b}} x{:}\mathsf{Ref}(\alpha) \models^{\mathcal{I}} \exists y.\,(A \wedge !x = y) & \text{(Res-Nonthread, } \exists\text{)} \\
\supset\ (\boldsymbol{\nu}\, y)(\mathsf{Ref}\langle xy\rangle | R) \models_{\mathsf{b}} x{:}\mathsf{Ref}(\alpha) \models^{\mathcal{I}} A[!x/y] & \text{(Lemma 30)} \\
\supset\ (\boldsymbol{\nu}\, y)(\mathsf{Ref}\langle xy\rangle | R) \models_{\mathsf{b}} x{:}\mathsf{Ref}(\alpha) \models^{\mathcal{I}} B & \text{(Corollary 4).}
\end{array}
$$

For the read rule:

$$(\mathsf{Read})\ \frac{P \vdash \{C\}\,\mathbf{rely}\ A \wedge A'\ \mathbf{guar}\ B\,\{C'\}, \quad C \supset A'[(!x)^{\uparrow}/c]}{\overline{x}\mathsf{read}(c)P \vdash \{C\}\,\mathbf{rely}\ A\,\mathbf{guar}\ B\,\{C'\}}$$

Assume $\vdash P \triangleright ?\overline{\Gamma} \cdot ?_{rw}\overline{\Delta} \cdot c\!:\!(\tau)^{\downarrow}; v\!:\!\rho^{\uparrow}$ and $\vdash R \triangleright \Gamma \cdot \Delta$. By Lemma 19 (1), we can set $R \cong \mathsf{Ref}\langle x,(y)S\rangle|R'$. Further let $C \supset A'[(!x)^{\uparrow}/c]$.

$$\begin{array}{ll}
R \models_{\mathsf{b}} x\!:\!\mathsf{Ref}(\alpha)\cdot\xi \models^{\mathcal{I}} C \wedge A & \\
\supset\ R \models_{\mathsf{b}} x\!:\!\mathsf{Ref}(\alpha)\cdot\xi \models^{\mathcal{I}} C \wedge A \wedge A'[(!x)^{\uparrow}/c] & (\text{Cor. 4})\\
\supset\ R \models_{\mathsf{b}} x\!:\!\mathsf{Ref}(\alpha)\cdot\xi \models^{\mathcal{I}} \exists c.\,(C \wedge A \wedge A' \wedge c = (!x)^{\uparrow}) & (\text{Lem. 30})\\
\supset\ R|\overline{c}(y)S \models_{\mathsf{b}} x\!:\!\mathsf{Ref}(\alpha)\cdot c\!:\!(\alpha)^{\uparrow}\cdot\xi \models^{\mathcal{I}} C \wedge A \wedge A' \wedge c = (!x)^{\uparrow} & (\text{Lem. 27})\\
\supset\ R|\overline{c}(y)S \models^{\mathcal{I}} C \wedge A \wedge A' & (\text{Lem. 27})\\
\supset\ P|R|\overline{c}(y)S \models^{\mathcal{I}} B \wedge C' & (\text{IH})\\
\supset\ (\boldsymbol{\nu}\, c)(P|R|\overline{c}(y)S) \models^{\mathcal{I}} B \wedge C' & (\text{Res-thread})\\
\supset\ \overline{x}\mathsf{read}(c)P|R \models^{\mathcal{I}} B \wedge C' & (\text{Cor. 4}).
\end{array}$$

In the last line we used $\overline{x}\mathsf{read}(c)P|R \longrightarrow\!\!\cong (\boldsymbol{\nu}\, c)(P|R|\overline{c}(y)S)$.

The consequence rules are reasoned precisely as in the proof of Theorem 1. Finally we attack the write rule.

$$(\mathsf{Write})\ \frac{S \vdash \mathbf{rely}\ A^{-c}\ \mathbf{guar}\ E, \quad P \vdash \{C\}\,\mathbf{rely}\ A\,\mathbf{guar}\ B\,\{C'\}, \quad E[!x/y] \supset^{x} C}{\overline{x}\mathsf{write}(yc)(S|P) \vdash \{\exists x.(C \wedge E[!x/y])\}\,\mathbf{rely}\ A\,\mathbf{guar}\ B\,\{C'\}}$$

We first observe:

Claim. Let $\vdash P_1 \triangleright x : \sigma$ and $\vdash P_2 \triangleright \vec{y} : \vec{\rho}$ s,t, $\mathsf{md}(\sigma\vec{\rho}) \subset \{!, !_{rw}\}$. Further let $A^{x:\sigma} \supset^{x} B^{x\vec{y}:\sigma\vec{\rho}}$ and $R \models^{\mathcal{I}} \exists x.(A \wedge B)$. Then $P \models^{\mathcal{I}} A$ implies $P|R \models^{\mathcal{I}} B$.

For the proof let $R \models_{\mathsf{b}} \xi$ and assume $P \models_{\mathsf{b}} x\!:\!\alpha \models^{\mathcal{I}} A$. Then $x\!:\!\alpha\cdot\xi \models^{\mathcal{I}} B$ from $A \supset^{x} B$ (by definition), that is $P|R \models^{\mathcal{I}} B$, concluding the proof of the claim.

Let $\vdash S \triangleright ?\overline{\Gamma}; y\!:\!\tau^{!}$, $\vdash R \triangleright\, !\Gamma \cdot !_{rw}\Delta$ and $\vdash P \triangleright ?\overline{\Gamma}\cdot ?_{rw}\overline{\Delta}\cdot c\!:\!()^{\downarrow}\ ;\ v\!:\!\rho^{\uparrow}$ with $x \in \mathsf{dom}(\Delta)$. By Lemma 19 (2), let $R \cong \mathsf{Ref}\langle x,(c)T\rangle|U|V$ where $\vdash \mathsf{Ref}\langle x,(c)T\rangle|U \triangleright \Delta$ and $\vdash V \triangleright \Gamma$, and let $S|R \cong S'|R$ s.t. $\vdash S' \triangleright y\!:\!\tau^{!}$. We infer:

$$\begin{array}{ll}
R \models_{\mathsf{b}} x\!:\!\mathsf{Ref}(\alpha)\cdot !_{rw}\eta_i \cdot !\xi \models^{\mathcal{I}} C \wedge A & \\
\supset\ S|R \models^{\mathcal{I}} E \quad\wedge\quad V \models^{\mathcal{I}} A & (\text{IH, Lem. 27})\\
\supset\ S'|R \models^{\mathcal{I}} E \quad\wedge\quad V \models^{\mathcal{I}} A & (\text{Cor. 3})\\
\supset\ S' \models^{\mathcal{I}} E \quad\wedge\quad V \models^{\mathcal{I}} A & (\text{Lem. 27})\\
\supset\ \mathsf{Ref}\langle x,(y)S'\rangle \models^{\mathcal{I}} \exists y.(E \wedge !x = y) \quad\wedge\quad V \models^{\mathcal{I}} A & (\exists, \text{Res-nonthread})\\
\supset\ \mathsf{Ref}\langle x,(y)S'\rangle \models^{\mathcal{I}} E[!x/y] \quad\wedge\quad V \models^{\mathcal{I}} A & (\text{Lem. 30})\\
\supset\ \mathsf{Ref}\langle x,(y)S'\rangle|U \models^{\mathcal{I}} C \quad\wedge\quad V \models^{\mathcal{I}} A & (\text{Claim above})\\
\supset\ \mathsf{Ref}\langle x,(y)S'\rangle|U|V|\overline{c} \models^{\mathcal{I}} A \wedge C & (\text{Lem. 29})\\
\supset\ P|(\mathsf{Ref}\langle x,(y)S'\rangle|U|V|\overline{c}) \models^{\mathcal{I}} B \wedge C' & (\text{IH})\\
\supset\ (\boldsymbol{\nu}\, c)(P|(\mathsf{Ref}\langle x,(y)S'\rangle|U|V|\overline{c})) \models^{\mathcal{I}} B \wedge C' & (\text{Res-Thread})\\
\supset\ \overline{x}\mathsf{write}(yc)(S|P)|(\mathsf{Ref}\langle x,(y)T\rangle|U|V) \models^{\mathcal{I}} B \wedge C' & (\text{Cor. 3})\\
\supset\ \overline{x}\mathsf{write}(yc)(S|P)|R \models^{\mathcal{I}} B \wedge C' & (\text{Cor. 3}).
\end{array}$$

In the second line to the last, we used $\overline{x}\mathsf{write}(yc)(S|P)|(\mathsf{Ref}\langle x,(y)T\rangle|U|V) \cong \overline{x}\mathsf{write}(yc)(S'|P)|(\mathsf{Ref}\langle x,(y)T\rangle|U|V) \longrightarrow (\boldsymbol{\nu}\, c)(P|(\mathsf{Ref}\langle x,(y)S'\rangle|U|V|\overline{c}))$. □

$$\frac{\vdash P \triangleright \Gamma \cdot x{:}\tau^{\natural w}}{\vdash (\boldsymbol{\nu}\, x)P \triangleright \Gamma/x}\ \text{(Hide)}$$

Fig. 9. Typing Rule for Local Reference

5 Logic for Sequential Processes with Local State

5.1 Affine Behaviour with Local State

Types, action types and (untyped) processes are as in Section 3.3. To the typing rules, we add the rule which hides a reference channel, given in Figure 9. In the rule, Γ/x takes off x from the effect channels used in Γ.

5.2 Logic for Local State (1): Language

The logic with local state we study in the following enjoys full compatibility with the logic with open state (hence the preceding two logics). This is made possible by the use of two kinds of names of (co-)replicated types, one called *opaque* and another called *non-opaque* or *open*. Some notations follow, with $\mathsf{md}(\tau) = \mathbf{!}$.

- When we write $x^{[\tau]}$, this means x is an opaque name.
- When we write $\underline{x}^{\tau}$, this means $\underline{x}$ is an opened name.

We often omit type annotations. x and $\underline{x}$ are considered as distinct names. All free auxiliary names of $\mathbf{!}$-types are non-opaque (without loss of expressiveness).

The grammar of terms follows, which is identical with Section 3.3 except for the distinction between opaque and non-opaque terms of replicated types.

$$a^{\uparrow} ::= x^{\tau^{\uparrow}} \mid \mathtt{in}_e(\vec{a}^{\vec{\tau}^{!}})^{\uparrow} \quad a^{!} ::= \underline{x}^{\tau^{!}} \mid !(x^{ref(\tau)}) \quad a^{[!]} ::= x^{[\tau^{!}]} \quad a^{\natural w} ::= x^{\tau^{\natural w}}\ .$$

Terms in the category $a^{!}$ are *non-opaque* (or *open*); while terms in $a^{[!]}$, with types of the form $[\tau]$, are *opaque*. $!(x^{ref(\tau)})$ has type τ. Equations are only considered between terms of the same type (so an opaque name and an opened name cannot be compared). Formulae are given by, in addition to the well-typed equations and omitting obvious type annotations:

$$A \ ::= \ \ldots \mid \{A\}\,\underline{x} \bullet \underline{\vec{y}}\{A'\} \mid \langle\{A\}\underline{x} \bullet \underline{\vec{y}}/z\rangle B \mid \mathsf{open}\ \vec{x}^{[\vec{\tau}]}\ \mathsf{as}\ (\boldsymbol{\nu}\vec{y}^{\vec{\delta}})\underline{\vec{x}}^{\vec{\rho}}\ \mathsf{in}\ A$$

We already encountered the first two constructs in Section 4.2: we now restrict their operands to non-opaque names. $\mathsf{open}\ \vec{a}^{[\vec{\tau}]}\ \mathsf{as}\ (\boldsymbol{\nu}\vec{y}^{\vec{\delta}})\underline{\vec{x}}^{\vec{\rho}}\ \mathsf{in}\ A$ is called the *opening formula for* $\vec{a}$. The rule is simple: *all opaque names should be opened before they can be used with the operators as given in the open state logic.* In $\mathsf{open}\ \vec{x}^{[\vec{\tau}]}\ \mathsf{as}\ (\boldsymbol{\nu}\vec{y}^{\vec{\delta}})\underline{\vec{x}}^{\vec{\rho}}\ \mathsf{in}\ A$, we always assume:

– $\vec{x}$ (resp. $\underline{\vec{x}}$, resp. $\vec{y}$) is a vector of pairwise distinct names such that $\vec{x}$ and $\underline{\vec{x}}$ have the same non-zero length and $\{\vec{x}\} \cap \mathsf{fn}(A) = \emptyset$. $\vec{x}$ (resp. $\vec{y}$) is called the *opened opaque names* (resp. *local references*) of the formula.
– $\vec{y}$ (resp. $\underline{\vec{x}}$) in $\vec{\rho}$ and A (resp. in A) are bound in the formula, while $\vec{x}$ occur free in the formula. Further each name in $\vec{y}$ should occur in $\vec{\rho}$ as an effect name s.t. $\vec{\rho}/\vec{y} = \vec{\tau}$.

The following notion smoothly integrates the open state logic into the local one.

Convention 1. (trivially opaque name) *A **trivially opaque name**, or **TON** for short, is an opaque name which is opened with the empty local reference (for example, x in "$\mathsf{open}\ x\ \mathsf{as}\ (\boldsymbol{\nu}\varepsilon)\underline{x}\ \mathsf{in}\ A$" is a TON). When an opaque name occurs in a place where a non-opaque name should occur, we regard it as a TON (e.g. "$\langle\{A\}x \bullet \vec{y}/z\rangle B$" in fact means "$\mathsf{open}\ x\vec{y}\ \mathsf{as}\ (\boldsymbol{\nu}\varepsilon)\underline{x\vec{y}}\ \mathsf{in}\ \langle\{A\}\underline{x} \bullet \underline{\vec{y}}/z\rangle B$").*

In essence, a TON is an opaque name without its local state, hence can be used in the same way we treated replicated names in the open state logic.

5.3 Logic for Local State (2): Model

For behavioural constants, we refine (co-)replicated behaviours.

$$\alpha \ ::= \ \ldots \mid \wedge_{i\in I}\{\xi_i\}(\vec{\alpha}_i(\boldsymbol{\nu}\,\eta_i)\beta_i)^{!}\{\xi_i'\} \mid \vee_{i\in I}\{\xi_i\}(\vec{\alpha}_i(\boldsymbol{\nu}\,\eta_i)\beta_i)^{?}\{\xi_i'\}$$

where η_i in $\wedge_{i\in I}\{\xi_i\}(\vec{\alpha}_i(\boldsymbol{\nu}\,\eta_i)\beta_i)^{!}\{\xi_i'\}$ (resp. $\vee_{i\in I}\{\xi_i\}(\vec{\alpha}_i(\boldsymbol{\nu}\,\eta_i)\beta_i)^{?}\{\xi_i'\}$) is a finite map from names to reference (resp. co-reference) behaviours, where $\mathsf{dom}(\eta_i)$ acts as binders. We always assume the standard naming convention. When η_i is empty, $\wedge_{i\in I}\{\xi_i\}(\vec{\alpha}_i(\eta_i)\beta_i)^{!}\{\xi_i'\}$ is simply written as $\wedge_{i\in I}\{\xi_i\}(\vec{\alpha}_i\beta_i)^{!}\{\xi_i'\}$.

Models now include hiding, and are generated by the following clauses. Below u ranges over all names including opaque and non-opaque names.

– $\emptyset$ is a model, with its subjects $\mathsf{sbj}(\emptyset)$ being $\emptyset$.
– $\xi \cdot u : \alpha$ is a model if $u \notin \mathsf{sbj}(\xi)$. Then $\mathsf{sbj}(\xi \cdot u : \alpha) = \mathsf{sbj}(\xi) \cup \{u\}$.
– $(\boldsymbol{\nu}\,x)\xi$ is a model if $x^{!_{rw}} \in \mathsf{sbj}(\xi)$ and if, whenever $\xi \equiv \underline{y} : \alpha^{!} \cdot \xi'$, we have $x \notin \mathsf{fn}(\alpha)$. Then $\mathsf{sbj}((\boldsymbol{\nu}\,x)\xi) = \mathsf{sbj}(\xi)\setminus\{x\}$.

In the third clause, $\xi_1 \equiv \xi_2$ is given by the α-equality together with: (1) $\xi{\cdot}\eta \equiv \eta{\cdot}\xi$, (2) $(\boldsymbol{\nu}\,x)(\xi \cdot x : \tau^{!_{rw}}) \equiv \xi$ if $x \notin \mathsf{fn}(\xi)$, and (3) $(\boldsymbol{\nu}\,x)\xi_1 \equiv (\boldsymbol{\nu}\,x)\xi_2$ if $\xi_1 \equiv \xi_2$. We always consider models via $\equiv$. By the third clause, an effect name of the behaviour of a non-opaque name is never hidden, i.e. never becomes local.

For the definition of $\models_{\mathsf{b}}$, we take a quickest way to reach the required relation.

1. $P \models_{\mathsf{b}} !!_{rw}\xi \cdot x : \mathtt{in}_i(\vec{\alpha})^{\uparrow}$ when $P \stackrel{\overline{x}\mathtt{in}_i(\vec{y})}{\Longrightarrow} P'$ s.t. $P' \models_{\mathsf{b}} \xi \cdot \vec{y} : \vec{\alpha}$. Further $P \models_{\mathsf{b}} \xi \cdot x{:}\omega$ when $P \Uparrow$.
2. $P \models_{\mathsf{b}} !\xi \cdot x{:}\wedge_{i\in I}\{\xi_i\}(\vec{\alpha}_i(\vec{w}_i{:}\vec{\gamma}_i)\beta_i)^{!}\{\xi_i'\}$ when $P \stackrel{x(\vec{y}z)}{\longrightarrow} P'$ such that, for each $i \in I$, $R \models_{\mathsf{b}} \vec{y}{:}\vec{\alpha}_i \cdot \xi_i$ implies $(\boldsymbol{\nu}\,x\vec{y})(P'|R) \models_{\mathsf{b}} (\boldsymbol{\nu}\,\vec{w}_i)(\xi \cdot \xi_i' \cdot z{:}\beta_i \cdot \vec{w}_i{:}\vec{\gamma}_i)$.
3. $P \models_{\mathsf{b}} !!_{rw}\xi \cdot x{:}\mathsf{Ref}(\alpha)^x$ when $P \stackrel{x\,\mathsf{read}(c)}{\longrightarrow} C[\mathsf{Ref}\langle xz\rangle]$ s.t. $C[\mathbf{0}] \models_{\mathsf{b}} \xi \cdot c{:}(\alpha)^{\uparrow}$.
4. $P \models_{\mathsf{b}} (\boldsymbol{\nu}\,x)\xi$ when $P \cong (\boldsymbol{\nu}\,x)P'$ such that $P' \models_{\mathsf{b}} \xi$.

$\cong$ in the fourth clause makes $\models_{\mathsf{b}}$ intractable. For the soundness proof this is enough. $\simeq_{\mathsf{b}}$ is defined as before.

5.4 Logic for Local State (3): Sequents and Interpretation

For the function $[\![a]\!]_{\mathcal{I}\cdot\xi}$ and the relation $\xi \models^{\mathcal{I}} A$, we add the following clauses to the defining clauses for the logic for open state, assuming the type correctness of formulae and models.

- $[\![x^{[\tau^!]}]\!]_{\mathcal{I}\cdot\xi} \stackrel{\text{def}}{=} x$.
- $[\![\underline{x}^{\tau^!}]\!]_{\mathcal{I}\cdot\xi} \stackrel{\text{def}}{=} \alpha$ when $I\cdot\xi = \xi'\cdot x{:}\alpha$. Similarly for $[\![x^{\tau^\uparrow}]\!]_{\mathcal{I}\cdot\xi}$.
- $\xi \models^{\mathcal{I}} \mathsf{open}\ \vec{x}\ \mathsf{as}\ (\boldsymbol{\nu}\,\vec{y})\underline{\vec{x}}\ \mathsf{in}\ A$ iff $\xi \simeq_{\mathsf{b}} (\boldsymbol{\nu}\,\vec{y})(\vec{x}{:}\vec{\alpha}\cdot\xi')$ s.t. $\underline{\vec{x}}{:}\vec{\alpha}\cdot\xi' \models^{\mathcal{I}} A$.
- $\xi \models^{\mathcal{I}} \forall x^{[\tau]}.A$ iff $\forall\alpha^{\tau}.\xi \models^{I\cdot x:\alpha} A \ \wedge\ \forall z^{[\tau]}\in\mathsf{dom}(\xi).\xi \models^{\mathcal{I}} A[z/x]$.
- $\xi \models^{\mathcal{I}} \exists x^{[\tau]}.A$ iff $\exists\alpha^{\tau}.\xi \models^{I\cdot x:\alpha} A \ \vee\ \exists z^{[\tau]}\in\mathsf{dom}(\xi).\xi \models^{\mathcal{I}} A[z/x]$.

In the first clause, an opaque name is interpreted simply as a distinction. The second clause is precisely as in the logic for open state. The third clause says that, after opening, the concerned channels should indeed get opened. In the forth and fifth clauses, opaque names are treated just as references.

We use the sequents $P \vdash \{C\}\,\mathbf{rely}\,A\,\mathbf{guar}\,B\,\{C'\}$ and $P \vdash \mathbf{rely}\,A\,\mathbf{guar}\,B$ from Section 3.3, whose well-typedness and validity are defined in the identical way.

5.5 Properties of interpretation

A small difference in the following Lemma from Lemma 19 is in the need for having local references in clause (3).

Lemma 33. (1) $\equiv, \longrightarrow, \approx$ *are subrelations of* $\cong$.

(2) $!x(\vec{y}).R|P_1|P_2 \cong !x(\vec{y}).R|P_1|(\boldsymbol{\nu}\,x)(!x(\vec{y}).R|P_2)$. *Also we have* $(\boldsymbol{\nu}\,x)!x(\vec{y}).R \cong (\boldsymbol{\nu}\,x)\mathsf{Ref}\langle xy\rangle \cong \mathbf{0}$.

(3) (a) *If* $\vdash P \triangleright \mathbin{!!}_{rw}\Gamma\cdot!\Delta$, *then* $P \cong (\boldsymbol{\nu}\,\vec{w})(Q|R|S)$ *such that* $\vdash Q \triangleright \Gamma'$, $\vdash R \triangleright \Delta'$ *and* $\vdash S \triangleright \vec{w}{:}\vec{\tau}^{\,!_{rw}}$ *where* $\Gamma'/\vec{w} = \Gamma$ *and* $\Delta'/\vec{w} = \Delta$.
(b) *If* $\vdash P \triangleright !\Gamma\cdot!_{rw}\Delta\cdot x{:}\rho^{\uparrow}$, *then* $P \cong (\boldsymbol{\nu}\,\vec{w})(Q|R|S)$ *such that* $\vdash Q \triangleright \Gamma'\cdot\Delta'$, $\vdash R \triangleright ?_{rw}\overline{\Delta'}\cdot x:\rho'$ *and* $\vdash S \triangleright \vec{w}:\vec{\tau}^{\,!_{rw}}$ *where* $\Gamma'/\vec{w} = \Gamma$, $\Delta'/\vec{w} = \Delta$ *and* $\rho'/\vec{w} = \rho$. *Further if* $P \not\longrightarrow$ *then we can set* $\vdash R \triangleright x{:}\rho'$.

(4) *Let* $\vdash P_{1,2} \triangleright \mathbin{!!}_{rw}\Gamma, x{:}\rho^{\uparrow}$ *and* $P_i \stackrel{\overline{x}\mathtt{in}(\vec{y})}{\Longrightarrow} P_i'$. *Then* $P_1 \cong P_2$ *iff* $P_1' \cong P_2'$.

(5) *Let* $\vdash P_{1,2} \triangleright !\Gamma\cdot x{:}(\vec{\tau}\rho)^{!}_{\Delta}$ *and* $P_i \stackrel{x(\vec{y}z)}{\Longrightarrow} P_i'$. *Then* $P_1 \cong P_2$ *iff* $P_1' \cong P_2'$ *iff, for each* $\vdash R \triangleright \vec{y}{:}\vec{\tau}\cdot\overline{\Delta}$, *we have* $(\boldsymbol{\nu}\,\vec{y})(R|P_1) \cong (\boldsymbol{\nu}\,\vec{y})(R|P_2)$.

(6) *Let* $\vdash P_{1,2} \triangleright \mathbin{!!}_{rw}\Gamma\cdot x : \mathit{ref}(\tau)$ *and* $P_i \stackrel{x\mathsf{read}(c)}{\Longrightarrow} P_i'$. *Then, setting* $P_i'' \stackrel{def}{=} C_i[\mathbf{0}]$ *with* $P_i' \stackrel{def}{=} C_i[\mathsf{Ref}\langle xz\rangle]$, $P_1 \cong P_2$ *iff* $P_1' \cong P_2'$ *iff* $P_1'' \cong P_2''$.

Proof. (1) and (2) are standard. For (3-a), let P be typed as given. Then $P \equiv (\boldsymbol{\nu}\,\vec{w}\vec{v})(\Pi P_i)$ where (i) ΠP_i is the concurrent composition of references and prime (i.e. prefixed) replicated processes and (ii) $\vec{w}$ and $\vec{v}$ respectively restrict references and replications. By (1) above, we have $(\boldsymbol{\nu}\,\vec{v})(\Pi P_i) \cong (\Pi Q_j)|(\Pi R_l)$ where (i) each Q_j is a prime replicated process without no free **?** names and (ii) each R_l is a

reference without no free **?** name. By setting $Q \stackrel{\text{def}}{=} \Pi Q_j$ and $R \stackrel{\text{def}}{=} \Pi R_l$ we are done. (3-b) is by the same reasoning as (3-a). (4), (5) and (6) are reasoned as in Lemmas 1 and 19, using (1), (2) and (3) above. We only show (3) below. Let $\vdash P_{1,2} \triangleright \mathbin{!!}_{rw}\Gamma, x:\rho^{\uparrow}$ and $P_i \stackrel{\overline{x}\mathtt{in}(\vec{y})}{\Longrightarrow} P_i'$. By (3-b) above, we can assume, without loss of precision, $P_i \cong (\boldsymbol{\nu}\,\vec{w})(\overline{x}\mathtt{in}_i(\vec{y})Q_i|R_i|S_i)$, hence $P_i' \cong (\boldsymbol{\nu}\,\vec{w})(Q_i|R_i|S_i)$. Let $C[\ \cdot\ ] \stackrel{\text{def}}{=} (\boldsymbol{\nu}\,x\vec{y})(x\mathtt{in}_i\langle\vec{y}\rangle|[\ \cdot\ ])$ where the free output in $x\mathtt{in}_i\langle\vec{y}\rangle$ is given by copycats. Then:

$$P_1' \cong P_2' \ \supset\ P_1 \cong C[P_1'] \cong C[P_2'] \cong P_2.$$

Conversely, let $\Delta = \vec{y}:\vec{\rho}$ where $\vec{\rho}^{\bullet}$ are the types of $\vec{y}$, and assume $\vdash U \triangleright \mathbf{?}\overline{\Delta} \cdot \mathbf{??}_{rw}\,\overline{\Gamma} \cdot e:()^{\uparrow}$. By the standard context lemma, $P_1' \cong P_2'$ iff $P_1|U \cong P_2|U$. Let $C'[\ \cdot\ ] \stackrel{\text{def}}{=} x[...\&(\vec{y}).U\&...]$ where $(\vec{y}).U$ is the i-th summand while ... indicates arbitrary summands. Then:

$$P_1 \cong P_2 \ \supset\ C'[P_1] \cong C'[P_2] \ \supset\ P_1'|U \cong P_2'|U \ \supset\ P_1' \cong P_2'\,. \qquad \square$$

Because $\models_{\mathsf{b}}$ is defined via $\cong$, the following two properties are trivial.

Lemma 34. (1) $P_{1,2}^{\Gamma} \models_{\mathsf{b}} \xi$ *implies* $P_1 \cong P_2$. (2) $P_1^{\Gamma} \models_{\mathsf{b}} \xi$ *and* $P_1 \cong P_2$ *then* $P_2^{\Gamma} \models_{\mathsf{b}} \xi$. (3) *Let* $\xi_{1,2}$ *be definable. Then* $\xi_1 \simeq_{\mathsf{b}} \xi_2$ *iff* $P_1 \cong P_2$ *for some* $P_{1,2}$ *defining* $\xi_{1,2}$ *respectively iff* $P_1 \cong P_2$ *for any* $P_{1,2}$ *defining* $\xi_{1,2}$ *respectively.*

Corollary 5. (1) *If* $P_1 \models^{\mathcal{I}} A$ *and* $P_1 \cong P_2$ *then* $P_2 \models^{\mathcal{I}} A$. (2) *If* $P \models^{\mathcal{I}} A$ *and* $P \models_{\mathsf{b}} \xi$ *then* $\xi \models^{\mathcal{I}} A$.

The following Lemma refines Lemma 21

Lemma 35. (1) *Let* $\mathsf{fn}(\Gamma) \cap \mathsf{fn}(\Delta) = \emptyset$. *Then* $P \models_{\mathsf{b}} \xi$ *implies* $P^{\Gamma}|Q^{\Delta} \models_{\mathsf{b}} \xi^{\Gamma} \cdot \xi'^{\Delta}$ *for some* ξ'. *Further if* $\omega \notin \mathsf{ran}(\xi \cup \xi')$ *then* $P^{\Gamma}|Q^{\Delta} \models_{\mathsf{b}} \xi^{\Gamma} \cdot \xi'^{\Delta}$ *iff* $P \models_{\mathsf{b}} \xi$ *and* $Q \models_{\mathsf{b}} \xi'$. (2) (a) $P \models_{\mathsf{b}} \xi \cdot x:\alpha^{!}$ *implies* $(\boldsymbol{\nu}\,x)P \models_{\mathsf{b}} \xi$. (b) $P \models_{\mathsf{b}} (\boldsymbol{\nu}\,\vec{w})(\xi \cdot x:\alpha^{!})$ *implies* $(\boldsymbol{\nu}\,x)P \models_{\mathsf{b}} (\boldsymbol{\nu}\,\vec{w})\xi$. (3) $P^{\Gamma\cdot x:\delta} \models_{\mathsf{b}} \xi$ *implies* $(\boldsymbol{\nu}\,x)P \models_{\mathsf{b}} (\boldsymbol{\nu}\,x)\xi$.

Proof. (1) and (2-a) are as before. For (2-b):

$$\begin{array}{lll} P \models_{\mathsf{b}} (\boldsymbol{\nu}\,\vec{w})(\xi \cdot x:\alpha^{!}) & \supset & P \equiv (\boldsymbol{\nu}\,\vec{w})P' \ \wedge\ P' \models_{\mathsf{b}} \xi \cdot x:\alpha^{!} \qquad (\text{def. of } \models_{\mathsf{b}}) \\ & \supset & (\boldsymbol{\nu}\,x)P \equiv (\boldsymbol{\nu}\,\vec{w}x)P' \ \wedge\ (\boldsymbol{\nu}\,x)P' \models_{\mathsf{b}} \xi \qquad (2\text{-a}) \\ & \supset & (\boldsymbol{\nu}\,x)P \models_{\mathsf{b}} (\boldsymbol{\nu}\,\vec{w})\xi \qquad (\text{def. of } \models_{\mathsf{b}})\,. \end{array}$$

(3) is immediate from the definition. $\square$

Lemmas 36, 37 and 38 are proved just as before. In Lemma 36 below, note the value given as the result of invocation would in general accompany hidden references.

Lemma 36. *Let* $\alpha_1 \simeq_{\mathsf{b}} \alpha_2$, $\vec{\beta}_1 \simeq_{\mathsf{b}} \vec{\beta}_2$, *and* $\xi_1 \cong \xi_2$, *and let* $\zeta_i = \vec{w}_i : \vec{\alpha}^{!_{rw}}$. *Further let* $\{\xi_1\}\alpha_1 \bullet \vec{\beta}_1 = \langle(\boldsymbol{\nu}\,\zeta_1)\gamma_1,\ \xi_1'\rangle$ *and* $\{\xi_2\}\alpha_2 \bullet \vec{\beta}_2 = \langle(\boldsymbol{\nu}\,\zeta_2)\gamma_2,\ \xi_2'\rangle$. *Then:* (1) $(\boldsymbol{\nu}\,\vec{w}_1)(y:\gamma_1,\zeta_1) \simeq_{\mathsf{b}} (\boldsymbol{\nu}\,\vec{w}_2)(y:\gamma_2,\zeta_2)$ *and* (2) *if* $\gamma_1 \neq \omega$, *then* $\xi_1' \simeq_{\mathsf{b}} \xi_2'$.

Lemma 37. $\mathsf{Ref}(\alpha_1) \simeq_{\mathsf{b}} \mathsf{Ref}(\alpha_2)$ *iff* $\alpha_1 \simeq_{\mathsf{b}} \alpha_2$.

Lemma 38. *If* $P \models_{\mathsf{b}} \xi_{1,2}$ *then* $[\![a]\!]_{\mathcal{I}\cdot\xi_1} \simeq_{\mathsf{b}} [\![a]\!]_{\mathcal{I}\cdot\xi_2}$.

Lemma 39. *Let* $P \models_{\mathsf{b}} \xi_{1,2}$. *Then* $\xi_1 \models^{\mathcal{I}} A$ *iff* $\xi_2 \models^{\mathcal{I}} A$.

Proof. There are two additional cases to the proof of Lemma 39. First the quantifications over opaque names are immediate from the respective induction hypothesis. Second, the case of the opening formula reasoned as follows:

$$\begin{array}{lll}
\xi_1 \models^{\mathcal{I}} \mathsf{open}\ \vec{x}\ \mathsf{as}\ (\boldsymbol{\nu}\vec{y})\underline{\vec{x}}\ \mathsf{in}\ A & & \\
\quad \equiv \quad \xi_1 \cong (\boldsymbol{\nu}\,\vec{y})(\vec{x}{:}\vec{\alpha}\cdot\xi') \ \wedge\ \xi'[\underline{\vec{x}}/\vec{x}] \models^{\mathcal{I}} A & \text{(Def. of } \models^{\mathcal{I}}) \\
\quad \supset \quad \xi_2 \cong (\boldsymbol{\nu}\,\vec{y})(\vec{x}{:}\vec{\alpha}\cdot\xi') \ \wedge\ \xi'[\underline{\vec{x}}/\vec{x}] \models^{\mathcal{I}} A & (P \models_{\mathsf{b}} \xi_{1,2}, \text{ Lemma 34 (3)}) \\
\quad \supset \quad \xi_2 \models^{\mathcal{I}} \mathsf{open}\ \vec{x}\ \mathsf{as}\ (\boldsymbol{\nu}\vec{y})\underline{\vec{x}}\ \mathsf{in}\ A & \text{(Def. of } \models^{\mathcal{I}})\ . \qquad \square
\end{array}$$

Lemma 40 and Corollary 6 are proved as before.

Lemma 40. *If* $\xi \models^{\mathcal{I}} A$ *and* $A \supset B$ *then* $\xi \models^{\mathcal{I}} B$.

Corollary 6. *If* $P \models^{\mathcal{I}} A$ *and* $A \supset B$ *then* $P \models^{\mathcal{I}} B$.

Lemma 41. *Let* $\mathsf{sbj}(\xi') \cap (\mathsf{fn}(A) \cup \{\vec{w}\}) = \emptyset$. *Then* $(\boldsymbol{\nu}\,\vec{w})(\xi\cdot\xi') \models^{\mathcal{I}} A$ *iff* $(\vec{w})\xi \models^{\mathcal{I}} A$.

Proof. Assume as given. We use induction on A. The only non-trivial cases are an equation on behavioural expressions and an opening formula.

Case $a_1 = a_2$. The case when both are linear is reduced to the case when both are replicated. If it is $x_1 = x_2$ (i.e. both are opaque), by definition this means they are identical, so no difference comes about. If it is $\underline{x_1} = \underline{x_2}$ (i.e. both are opened), then, by the definition of models, $I \cdot (\boldsymbol{\nu}\,\vec{w})\xi = \underline{x_1} : \alpha_1 \cdot \underline{x_2} : \alpha_2 \cdot \xi''$ so again no difference comes about. Similarly when a_i is of the form $!x$.

Case $\mathsf{open}\ \vec{x}\ \mathsf{as}\ (\boldsymbol{\nu}\vec{y})\underline{\vec{x}}\ \mathsf{in}\ A$. By assumption $\vec{x}$ only occur in $(\boldsymbol{\nu}\,\vec{w})\xi$.

$$\begin{array}{lll}
(\boldsymbol{\nu}\,\vec{w})\xi \models^{\mathcal{I}} \mathsf{open}\ \vec{x}\ \mathsf{as}\ (\boldsymbol{\nu}\vec{y})\underline{\vec{x}}\ \mathsf{in}\ A & & \\
\quad \equiv \quad (\boldsymbol{\nu}\,\vec{w})\xi \equiv (\boldsymbol{\nu}\,\vec{y})\xi_0 \ \wedge\ \xi_0 \models^{\mathcal{I}} A & & \text{(def. of } \models^{\mathcal{I}}) \\
\quad \equiv \quad (\boldsymbol{\nu}\,\vec{w})(\xi\cdot\xi') \equiv (\boldsymbol{\nu}\,\vec{y})(\xi_0\cdot\xi'') \ \wedge\ \xi_0\cdot\xi'' \models^{\mathcal{I}} A & & (\equiv, \text{ IH}) \\
\quad \equiv \quad (\boldsymbol{\nu}\,\vec{w})(\xi\cdot\xi') \models^{\mathcal{I}} \mathsf{open}\ \vec{x}\ \mathsf{as}\ (\boldsymbol{\nu}\vec{y})\underline{\vec{x}}\ \mathsf{in}\ A & & \text{(def. of } \models^{\mathcal{I}}).
\end{array}$$

Other cases are direct from the respective induction hypothesis. $\square$

Lemma 42. (monotonicity of $\boldsymbol{\nu}$) *Let* $\vdash \Gamma \cdot x : \tau^{!} \triangleright P$ *and* $x \notin \mathsf{fn}(A)$. *Then* $P \models^{\mathcal{I}} A$ *iff* $(\boldsymbol{\nu}\,x)P \models^{\mathcal{I}} A$.

Proof. Assume as given. Letting $x \notin \{\vec{w}\}$ and α^{τ}:

$$\begin{array}{lll}
P \models_{\mathsf{b}} (\boldsymbol{\nu}\,\vec{w})(\xi\cdot x{:}\alpha) \models^{\mathcal{I}} A & & \\
\quad \equiv \quad (\boldsymbol{\nu}\,x)P \models_{\mathsf{b}} (\boldsymbol{\nu}\,\vec{w})\xi \ \wedge\ (\boldsymbol{\nu}\,\vec{w})(\xi\cdot x{:}\alpha) \models^{\mathcal{I}} A & & \text{(Lemma 35 (2))} \\
\quad \equiv \quad (\boldsymbol{\nu}\,x)P \models_{\mathsf{b}} (\boldsymbol{\nu}\,\vec{w})\xi \ \wedge\ (\boldsymbol{\nu}\,\vec{w})\xi \models^{\mathcal{I}} A & & \text{(Lemma 41)}\ . \qquad \square
\end{array}$$

To study properties of local references, it is useful to consider a stricter notion of typing. First we stipulate:

Convention 2. *Henceforth we assume all names in processes occur explicitly typed, and that the effect typing for each (co-)replicated types is an ordered sequence rather than a set.*

A formula (resp. sequent, resp. model) is *strictly typed* w.r.t. a process iff its all primary free opened names are typable with their explicit types in the process. Using strictly typed formulae:

- $P \models'_{\mathsf{b}} \xi$ is a strictly typed version of $P \models'_{\mathsf{b}} \xi$, in the sense that we replace $\cong$ in the fourth defining clause with $\equiv$.
- $\xi \models'^{\mathcal{I}} A$ is a strictly typed version of $\xi \models^{\mathcal{I}} A$, in the sense that we replace $\simeq_{\mathsf{b}}$ in the third defining clause with $\equiv$.
- $P \models'^{\mathcal{I}} \mathbf{rely} A \mathbf{guar} B$ is given using $\models'_{\mathsf{b}}$ and $\models'^{\mathcal{I}}$ above.

The following conventions on strictly typed formulae does not lose generality.

Convention 3. *In strictly typed formulae/sequents:*

1. *In each occurrence of form* $\mathsf{open}\ \vec{w}\ \mathsf{as}\ (\boldsymbol{\nu}\vec{y})\underline{\vec{w}^{\vec{\tau}}}\ \mathsf{in}\ A$, *we assume* $\vec{y}$ *is fixed once* $\vec{w}$ *is given.*
2. *Whenever an identical name is opened in two places in a formula, the effect types of the opened name coincide.*

The clause 1 above does not lose generality since the effect names are a sequence in strict typing, so that we can uniquely determine which binding name corresponds to which effect name.

Lemma 43. $P \models^{\mathcal{I}} A$ *iff* $P' \models'^{\mathcal{I}} A$ *for some* P' *such that* $P \cong P'$.

Proof. Write $\xi' \succ^I A$ for ξ' conforms to A in the standard typing under $\mathcal{I}$, and $\xi' \succ^I_s A$ for ξ' conforms to A in the strict typing under $\mathcal{I}$. For brevity we omit the mention of $\mathcal{I}$ from now on. Since $P \models^{\mathcal{I}} \xi$ is closed under $\cong$, the statement is equivalent to: $\xi \models^{\mathcal{I}} A$ iff $\xi' \models^{\mathcal{I}} A$ for some (any) ξ' such that $\xi \simeq_{\mathsf{b}} \xi'$ and $\xi' \succ_s A$. Since $\models^{\mathcal{I}}$ is closed under $\simeq_{\mathsf{b}}$ by definition, this is further equivalent to saying (just): for any $\xi \succ A$, there exists some ξ' such that $\xi' \simeq_{\mathsf{b}} \xi$ and $\xi' \succ_s A$. We argue by induction on the size of A. We only consider $\wedge$, $\neg$ and $\forall$ as logical connectives. Below we omit the existential quantifier on ξ' and ξ'' for brevity.

Case $A \stackrel{\mathbf{def}}{=} e_1 = e_2$**.** Vacuous since ξ, ξ' do not affect $e_{1,2}$.

Case $A \stackrel{\mathbf{def}}{=} a_1 = a_2$**.** Immediate from Lemma 34.

Case $A \stackrel{\mathbf{def}}{=} A_1 \wedge A_2$**.** Because:

$$\xi \succ A_1 \wedge A_2 \ \equiv\ \xi \succ A_{1,2} \ \supset\ \xi \simeq_{\mathsf{b}} \xi' \succ_s A_{1,2} \ \equiv\ \xi \simeq_{\mathsf{b}} \xi' \succ_s A_1 \wedge A_2$$

Case $A \stackrel{\mathbf{def}}{=} \neg A'$**.** Because:

$$\xi \succ \neg A' \ \equiv\ \xi \succ A' \ \equiv\ \xi \simeq_{\mathsf{b}} \xi' \succ_s A' \ \equiv\ \xi \simeq_{\mathsf{b}} \xi' \succ_s \neg A' \ .$$

Case $A \stackrel{\mathbf{def}}{=} \forall x.A'$**.** Because (treating the case x is not opaque):

$$\xi \succ \forall x.A' \;\equiv\; \xi \succ A' \;\equiv\; \xi \simeq_{\mathsf{b}} \xi' \succ_s A' \;\equiv\; \xi \simeq_{\mathsf{b}} \xi' \succ_s \forall x.A' \;.$$

The case when x is opaque or a reference is the same.

Case $A \stackrel{\mathbf{def}}{=} \mathsf{open}\ \vec{w}\ \mathsf{as}\ (\boldsymbol{\nu}\vec{y})\underline{\vec{w}}\ \mathsf{in}\ A'$**.** Because:

$$\begin{array}{rcl} \xi \succ \mathsf{open}\ \vec{w}\ \mathsf{as}\ (\boldsymbol{\nu}\vec{y})\underline{\vec{w}}\ \mathsf{in}\ A' & \equiv & \xi \simeq_{\mathsf{b}} (\boldsymbol{\nu}\,\vec{y})\xi' \ \wedge\ \xi'[\underline{\vec{w}}/\vec{w}] \succ A' \\ & \equiv & \xi \simeq_{\mathsf{b}} (\boldsymbol{\nu}\,\vec{y})\xi' \ \wedge\ \xi' \simeq_{\mathsf{b}} \xi''[\underline{\vec{w}}/\vec{w}] \succ_s A' \\ & \equiv & \xi \simeq_{\mathsf{b}} (\boldsymbol{\nu}\,\vec{y})\xi'' \succ_s \mathsf{open}\ \vec{w}\ \mathsf{as}\ (\boldsymbol{\nu}\vec{y})\underline{\vec{w}}\ \mathsf{in}\ A' \;. \end{array}$$

Case $A \stackrel{\mathbf{def}}{=} \{C\}x\bullet\vec{y}\{C'\}$**.** Below let η be a model for the concerned effect names (associated with x).

$$\xi \succ \{C\}x\bullet\vec{y}zC' \;\supset\; \eta \succ C, C' \;\equiv\; \eta \simeq_{\mathsf{b}} \eta' \succ_s C, C' \;\equiv\; \xi \simeq_{\mathsf{b}} \xi' \succ_s \{C\}x\bullet\vec{y}zC' \;.$$

Case $A \stackrel{\mathbf{def}}{=} \{C\}\langle x\bullet\vec{y}/z\rangle A'$**.** Because:

$$\xi \succ \langle\{C\}x\bullet\vec{y}/z\rangle A' \;\supset\; z{:}\beta\cdot\xi \simeq_{\mathsf{b}} z{:}\beta'\cdot\xi' \succ_s A' \;\equiv\; \xi \simeq_{\mathsf{b}} \xi' \succ_s \langle\{C\}x\bullet\vec{y}/z\rangle A'. \;\square$$

Below we say x occurs in P^{Γ} with type τ if the occurring type of x is τ (this may differ from the type of x in Γ by hiding).

Lemma 44. (1) *Let $\vec{w}$ occur in P with type $\vec{\tau}$. Then $P \models'^{\mathcal{I}} \mathsf{open}\ \vec{w}\ \mathsf{as}\ (\boldsymbol{\nu}\vec{y}\vec{z})\underline{\vec{w}}^{\vec{\tau}}\ \mathsf{in}\ A$ iff $(\boldsymbol{\nu}\,x)P \models'^{\mathcal{I}} \mathsf{open}\ \vec{w}\ \mathsf{as}\ (\boldsymbol{\nu}\vec{y}x\vec{z})\underline{\vec{w}}^{\vec{\tau}}\ \mathsf{in}\ A$, assuming both sequents are well-typed.* (2) *Assume $\vdash P \triangleright \Gamma\cdot x{:}\tau^{\natural_{rw}}$ and $x \notin \mathsf{fn}(A)$. Then $P \models^{\mathcal{I}} A$ iff $(\boldsymbol{\nu}\,x)P \models^{\mathcal{I}} A$.*

Proof. For (1), since $\vec{w}$ occur in P with type $\vec{\tau}$ which follows the opening formula, x in $(\boldsymbol{\nu}\,x)P$ and x in $(\boldsymbol{\nu}\,x)P \models'^{\mathcal{I}} \mathsf{open}\ \vec{w}\ \mathsf{as}\ (\boldsymbol{\nu}\vec{y}x\vec{z})\underline{\vec{w}}^{\vec{\tau}}\ \mathsf{in}\ A$ bind a unique effect name in $\vec{\tau}$ in precisely the same way. We now prove both ways. Below we observe that, by the condition on name occurrence we stipulated for opening formulae, A does not contain any name from $\vec{w}$. For the "only if" direction:

$$\begin{array}{rcll} & & P \models^{\mathcal{I}} \mathsf{open}\ \vec{w}\ \mathsf{as}\ (\boldsymbol{\nu}\vec{y}\vec{z})\underline{\vec{w}}\ \mathsf{in}\ A & \\ & \equiv & P \models'_{\mathsf{b}} (\boldsymbol{\nu}\,\vec{w}\vec{y})\xi \ \wedge\ \xi \models'^{\mathcal{I}} A & (\text{Def. of } \models^{\mathcal{I}}) \\ & \supset & (\boldsymbol{\nu}\,x)P \models'_{\mathsf{b}} (\boldsymbol{\nu}\,x\vec{w}\vec{y})\xi \ \wedge\ \xi \models^{\mathcal{I}} \mathsf{open}\ \vec{w}\ \mathsf{as}\ (\boldsymbol{\nu}\vec{y}\vec{z})\underline{\vec{w}}\ \mathsf{in}\ A & (\text{Def. of } \models_{\mathsf{b}}) \\ & \supset & (\boldsymbol{\nu}\,x)P \models'_{\mathsf{b}} (\boldsymbol{\nu}\,x\vec{w}\vec{y})\xi \models^{\mathcal{I}} \mathsf{open}\ \vec{w}\ \mathsf{as}\ (\boldsymbol{\nu}\vec{y}\vec{z})\underline{\vec{w}}\ \mathsf{in}\ A & (\text{Def. of } \models^{\mathcal{I}}) \;. \end{array}$$

For the other direction, strict typing is essential:

$$\begin{array}{rcll} & & (\boldsymbol{\nu}\,x)P \models'^{\mathcal{I}} \mathsf{open}\ \vec{w}\ \mathsf{as}\ (\boldsymbol{\nu}\vec{y}x\vec{z})\underline{\vec{w}}\ \mathsf{in}\ A & \\ & \equiv & (\boldsymbol{\nu}\,x)P \models'_{\mathsf{b}} \xi \models'^{\mathcal{I}} \mathsf{open}\ \vec{w}\ \mathsf{as}\ (\boldsymbol{\nu}\vec{y}x\vec{z})\underline{\vec{w}}\ \mathsf{in}\ A & \\ & \supset & (\boldsymbol{\nu}\,x)P \equiv (\boldsymbol{\nu}\,x\vec{y}\vec{z})P' \models'_{\mathsf{b}} (\boldsymbol{\nu}\,x\vec{y}\vec{z})\xi' \ \wedge\ P' \models'_{\mathsf{b}} \xi' & (\models'_{\mathsf{b}}) \\ & & \quad \wedge\ (\boldsymbol{\nu}\,x\vec{y}\vec{z})\xi' \models'^{\mathcal{I}} \mathsf{open}\ \vec{w}\ \mathsf{as}\ (\boldsymbol{\nu}\vec{y}x\vec{z})\underline{\vec{w}}\ \mathsf{in}\ A & \\ & \supset & (\boldsymbol{\nu}\,x)P \equiv (\boldsymbol{\nu}\,x\vec{y}\vec{z})P' \ \wedge\ P' \models'_{\mathsf{b}} \xi' \models'^{\mathcal{I}} A & (\models'^{\mathcal{I}}) \\ & \supset & (\boldsymbol{\nu}\,\vec{y}\vec{z})P' \models'_{\mathsf{b}} (\boldsymbol{\nu}\,\vec{y}\vec{z})P' \models^{\mathcal{I}} \mathsf{open}\ \vec{w}\ \mathsf{as}\ (\boldsymbol{\nu}\vec{y}\vec{z})\underline{\vec{w}}\ \mathsf{in}\ A & (\models'_{\mathsf{b}}, \models'^{\mathcal{I}}). \end{array}$$

For (2), we prove $\xi \models^{\mathcal{I}} A \equiv (\boldsymbol{\nu}\, x)\xi \models^{\mathcal{I}} A$ by induction on A (the reasoning is identical for $\models'^{\mathcal{I}}$). We show two non-trivial cases. Other cases are simpler. Let $A \stackrel{\text{def}}{=} \underline{y_1^\tau} = \underline{y_2^\tau}$ s.t. $x \notin \{y_1, y_2\} \cup \mathsf{fn}(\tau)$. Further let $\underline{y_1}, \underline{y_2} \in \mathsf{fn}(\xi)$ (when $\underline{y_i} \in \mathsf{dom}(I)$ is easier).

$$\begin{array}{lcll} \xi \models^{\mathcal{I}} \underline{y_1^\tau}{=}\underline{y_2^\tau} & \equiv & \xi \simeq_{\mathsf{b}} \underline{y_1^\tau}{:}\alpha \cdot \underline{y_2^\tau}{:}\alpha \cdot \xi' & \text{(Definition of } \models^{\mathcal{I}}) \\ & \equiv & (\boldsymbol{\nu}\, x)\xi \simeq_{\mathsf{b}} \underline{y_1^\tau}{:}\alpha \cdot \underline{y_2^\tau}{:}\alpha \cdot (\boldsymbol{\nu}\, x)\xi' & (x \notin \mathsf{fn}(\tau)) \\ & \equiv & (\boldsymbol{\nu}\, x)\xi \models^{\mathcal{I}} \underline{y_1^\tau} = \underline{y_2^\tau} & \text{(Definition of } \models^{\mathcal{I}}). \end{array}$$

Next let $A \stackrel{\text{def}}{=} \mathsf{open}\ \vec{w}^{[\vec{\tau}]}\ \mathsf{as}\ (\boldsymbol{\nu}\vec{y})\underline{\vec{w}}\ \mathsf{in}\ A'$ with $x \notin \mathsf{fn}(A) = \mathsf{fn}(A') \cup \mathsf{fn}(\vec{\tau}) \cup \{\vec{w}\}$.

$$\begin{array}{lll} \xi \models^{\mathcal{I}} \mathsf{open}\ \vec{w}\ \mathsf{as}\ (\boldsymbol{\nu}\vec{y})\underline{\vec{w}}\ \mathsf{in}\ A & & \\ \quad \equiv\ \xi \simeq_{\mathsf{b}} (\boldsymbol{\nu}\,\vec{y})(\underline{\vec{w}}{:}\vec{\alpha} \cdot \xi')\ \wedge\ \underline{\vec{w}}{:}\vec{\alpha} \cdot \xi' \models^{\mathcal{I}} A & \text{(Def. } \models^{\mathcal{I}}) \\ \quad \equiv\ (\boldsymbol{\nu}\, x)\xi \simeq_{\mathsf{b}} (\boldsymbol{\nu}\,\vec{y})(\underline{\vec{w}}{:}\vec{\alpha} \cdot (\boldsymbol{\nu}\, x)\xi')\ \wedge\ (\underline{\vec{w}}{:}\vec{\alpha} \cdot (\boldsymbol{\nu}\, x)\xi') \models^{\mathcal{I}} A & (x \notin \mathsf{fn}(\vec{\tau}) \cup \{\vec{w}\}) \\ \quad \equiv\ (\boldsymbol{\nu}\, x)\xi \models^{\mathcal{I}} \mathsf{open}\ \vec{w}\ \mathsf{as}\ (\boldsymbol{\nu}\vec{y})\underline{\vec{w}}\ \mathsf{in}\ A & \text{(Def. } \models^{\mathcal{I}}). \quad \square \end{array}$$

The proofs of the remaining lemmas are either identical with those of the corresponding lemmas in the preceding sections (Lemmas 45, 46 and 47) or direct from the definition (Lemma 48). We present the statement and proofs for $\models^{\mathcal{I}}$ instead of $\models'^{\mathcal{I}}$ since they do not need strict typing: we can easily check all proofs trivially extend to $\models'^{\mathcal{I}}$.

Lemma 45. (cut in $\models^{\mathcal{I}}$) *Let* $\Gamma_0 \subset \Gamma$ *and* $\Gamma' \cdot \Theta \vdash A$ *with* $!\Gamma' \subset \Gamma$ *and* $\Theta \vdash I$.

(1) $P^{\overline{\Gamma_0};\Delta} \models \mathbf{rely}\ A_0\ \mathbf{guar}\ B$ *and* $R^\Gamma \models^{\mathcal{I}} A \wedge A_0$ *imply* $P|R \models^{\mathcal{I}} A \wedge B$.
(2) $P^{\overline{\Gamma_0}\cdot\overline{\Gamma_1};\Delta} \models \{C\}\ \mathbf{rely}\ A_0\ \mathbf{guar}\ B\ \{C'\}$ *and* $R^{\Gamma\cdot\Gamma_1} \models^{\mathcal{I}} C \wedge A \wedge A_0$ *with* $?_{rw}\overline{\Gamma_1} \cdot \Theta \vdash C$ *imply* $P|R \models^{\mathcal{I}} C' \wedge A \wedge B$.

Lemma 46. *Let* $x \notin \mathsf{fn}(A, a)$ *and* $[\![a]\!]_{\mathcal{I}\cdot\xi} \neq \omega$. *Then* $\xi \models^{\mathcal{I}} A[a/x]$ *iff* $\xi \models^{\mathcal{I}} \exists x.(A \wedge x = a)$.

Lemma 47. $\xi \cdot x{:}\mathtt{in}_i(\vec{\alpha})^\uparrow \models^{\mathcal{I}} A$ *iff* $\xi \cdot \vec{y}{:}\vec{\alpha} \models^{\mathcal{I}} A[\mathtt{in}_i(\vec{y})^\uparrow/x]$.

Lemma 48. *Let* $\mathsf{fn}(B) \cap \{x\vec{y}\} = \emptyset$, $\tau = (\vec{\rho}\tau)^!_\Delta$ *with* $\mathsf{dom}(\Delta) = \vec{w}$ *and* $\eta_i = \vec{u}{:}\vec{\gamma}$. *Then having* $\xi \cdot \underline{x}^\tau : \wedge_{i\in J}\{\xi_i\}(\vec{\alpha}_i(\boldsymbol{\nu}\,\eta_i)\beta_i)^!\{\xi_i'\} \cdot \underline{\vec{y}}{:}\vec{\alpha} \models^{\mathcal{I}} \langle\{C\}\underline{x} \bullet \underline{\vec{y}}\{C'\}/z\rangle B$ *is logically equivalent to having* $(\boldsymbol{\nu}\,\vec{u})(\xi \cdot z{:}\beta_i \cdot \eta_i) \models^{\mathcal{I}} B$ *and* $\xi/\vec{w} \cdot \xi_i' \models^{\mathcal{I}} C'$ *whenever* $\vec{\alpha}_i = \vec{\alpha}$ *and* $\xi/\vec{w} \cdot \xi_i \models^{\mathcal{I}} C$.

Remark 4.

1. By Lemma 43, strictly typed sequents and standard sequents have precisely the same expressive power for representing process behaviour, as far as we take a process up to $\cong$.
2. Lemma 44 (1) is the only result in this subsection which is not valid (and, indeed, whose statement does not even make sense) if we do not consider strict typing. Lemma 44 (1) essentially says that, when a reference is hidden with an opening formula, we do not lose any essential information as far as we are strict about types, presenting a path from the local state logic to the open state logic. The property plays an essential role in the proof of soundness in the next subsection.

$$\frac{P \vdash \mathbf{rely}\ A\ \mathbf{guar}\ B[\boldsymbol{\nu}\vec{y}_i\vec{z}_i/\boldsymbol{\nu}\vec{y}_i x\vec{z}_i]_i}{(\boldsymbol{\nu}\, x)P \vdash \mathbf{rely}\ A\ \mathbf{guar}\ B}\ (\text{Hide}) \qquad \frac{P \vdash \{C\}\ \mathbf{rely}\ A\ \mathbf{guar}\ B[\boldsymbol{\nu}\vec{y}_i\vec{z}_i/\boldsymbol{\nu}\vec{y}_i x\vec{z}_i]_i\ \{C'\}}{(\nu x)P \vdash \{C\}\ \mathbf{rely}\ A\ \mathbf{guar}\ B\ \{C'\}}\ (\text{Hide-thread})$$

$$\frac{P \vdash \mathbf{rely}\ A[\boldsymbol{\nu}\,\vec{y}_i\vec{z}_i/\boldsymbol{\nu}\,\vec{y}_i x\vec{z}_i]_i\ \mathbf{guar}\ B}{P \vdash \mathbf{rely}\ A\ \mathbf{guar}\ B}\ (\text{Co-Hide}) \qquad \frac{(A[\boldsymbol{\nu}\,\vec{y}_i\vec{z}_i/\boldsymbol{\nu}\,\vec{y}_i x\vec{z}_i]_i \supset A_0 \wedge E) \quad P \vdash \{C \wedge E\}\ \mathbf{rely}\ A_0\ \mathbf{guar}\ B\ \{C'\}}{P \vdash \{C\}\ \mathbf{rely}\ A\ \mathbf{guar}\ B\ \{C'\}}\ (\text{Co-Hide-thread})$$

Fig. 10. Proof Rules for Local State

5.6 Soundness of Proof Rules

The proof rules for sequential processes with local state use all rules from those for the logic for open state in Figure 8 and Figure 7. In each of these rules which involve operations on replicated names (which in particular include all prefix rules), we regard the names mentioned in formulae as TONs (cf. Convention 1).

In addition to these rules, we have the proof rules for hiding and their dual, given in Figure 10, which include the hiding in a process (thread and non-thread) and the hiding in an environment (thread and non-thread). In all of these rules, we assume x has type $!_{rw}$. These rules use the substitution of the shape:

$$A[\boldsymbol{\nu}\,\vec{y}_i\vec{z}_i/\boldsymbol{\nu}\,\vec{y}_i x\vec{z}_i]_i,$$

which indicates multiple substitutions which replace each occurrence of shape $\boldsymbol{\nu}\,\vec{y}_i x\vec{z}_i$ (in an opening formula) with $\boldsymbol{\nu}\,\vec{y}_i\vec{z}_i$, taking off x from the binder. As an example, the substitution:

$$(\mathsf{open}\ u\ \mathsf{as}\ (\boldsymbol{\nu} wx)\underline{u}\ \mathsf{in}\ A \wedge \mathsf{open}\ uv\ \mathsf{as}\ (\boldsymbol{\nu} wxw')\underline{uv}\ \mathsf{in}\ B)[\boldsymbol{\nu}\,\vec{y}_i\vec{z}_i/\boldsymbol{\nu}\,\vec{y}_i x\vec{z}_i]_i$$

yields the following formula, assuming no further binding by x occurs in A.

$$\mathsf{open}\ u\ \mathsf{as}\ (\boldsymbol{\nu} w)\underline{u}\ \mathsf{in}\ A \wedge \mathsf{open}\ uv\ \mathsf{as}\ (\boldsymbol{\nu} ww')\underline{uv}\ \mathsf{in}\ B.$$

Note the first formula does not obey the binder convention in the standard sense. In fact, the idea is to start from B, and consider the "original" formula in which x is bound. As necessary, we may consider an appropriate α-conversion is done. Some observations on usage of names in these rules:

- In (Hide), whenever there is at least one occurrence of $\boldsymbol{\nu}\,\vec{y}_i x\vec{z}_i$ in B, x cannot occur in A by the bound name convention. If there is no occurrence of $\boldsymbol{\nu}\,\vec{y}_i x\vec{z}_i$ in B then there is no substitution, so the rule is identity.
- Similarly, in (Hide-Thread), we can assume $x \notin \mathsf{fn}(A, C, C')$; in (Co-Hide), $x \notin \mathsf{fn}(B)$; and, in (Co-Hide-Thread), $x \notin \mathsf{fn}(C, B, C')$.

We now prove:

Theorem 4. (soundness of proof rules for logic with local state) *Let P be a process with local state. Then $P^{\Gamma;\Delta} \vdash \mathbf{rely}\, A\, \mathbf{guar}\, B$ implies $P^{\Gamma;\Delta} \models \mathbf{rely}\, A\, \mathbf{guar}\, B$. Also $P^{\Gamma;\Delta} \vdash \{C\}\, \mathbf{rely}\, A\, \mathbf{guar}\, B\, \{C'\}$ implies $P^{\Gamma;\Delta} \models \{C\}\, \mathbf{rely}\, A\, \mathbf{guar}\, B\, \{C'\}$.*

Proof. We use the alternative proof system whose sequents are written $P \vdash' \mathbf{rely} A \mathbf{guar} B$ and $P \vdash' \{C\}\mathbf{rely} A \mathbf{guar} B\{C'\}$. In this system, we restrict sequents to strictly typed ones, as well as using, in the side condition (if any) of each rule, the validity with respect to $\models'^{\mathcal{I}}$. We observe:

Claim. Write $P \vdash_{\cong} \mathbf{rely} A \mathbf{guar} B$ and $P \vdash_{\cong} \{C\}\mathbf{rely} A \mathbf{guar} B\{C'\}$ if they are derivable by the following rules.

$$(\cong)\ \frac{P' \vdash \mathbf{rely} A \mathbf{guar} B \quad P' \cong P}{P \vdash_{\cong} \mathbf{rely} A \mathbf{guar} B} \qquad (\cong\text{-thread})\ \frac{P' \vdash \{C\}\mathbf{rely} A \mathbf{guar} B\{C'\} \quad P' \cong P}{P \vdash_{\cong} \{C\}\mathbf{rely} A \mathbf{guar} B\{C'\}}.$$

Similarly define $P \vdash'_{\cong} \mathbf{rely} A \mathbf{guar} B$ and $P \vdash'_{\cong} \{C\}\mathbf{rely} A \mathbf{guar} B\{C'\}$. Then $P \vdash_{\cong} \mathbf{rely} A \mathbf{guar} B$ (resp. $P \vdash_{\cong} \{C\}\mathbf{rely} A \mathbf{guar} B\{C'\}$) if and only if $P \vdash'_{\cong} \mathbf{rely} A \mathbf{guar} B$ (resp. $P \vdash'_{\cong} \{C\}\mathbf{rely} A \mathbf{guar} B\{C'\}$).

Proof of the Claim. By definition, $P \vdash'_{\cong} \mathbf{rely} A \mathbf{guar} B$ implies $P \vdash_{\cong} \mathbf{rely} A \mathbf{guar} B$, similarly for sequents for threads. For the converse direction, since $\models^{\mathcal{I}}$ (resp. $\models'^{\mathcal{I}}$) is closed under $\cong$ (resp. under $\cong$ as far as strictness is assumed), and because the rules strictly follow the (refined) typing rules[1], the provability $\vdash_{\cong}$ (resp. $\vdash'_{\cong}$) is identical with the provability resulting from integrating each rule for $\vdash$ (resp. $\vdash'$) with $\cong$ in the obvious way [because we can always postpone the application of $\cong$]. Hence it suffices, for each rule, that if a sequent is derivable in the standard rule, then we can obtain the same result by the strictly typed rule up to $\cong$, which is mechanical by Lemma 43 and by induction, using the closure of $\cong$ for $\models_{\mathsf{b}}$ in the case of opening formula. (end of the proof of the Claim).

We can now establish the soundness. For the rules from Figure 8 and Figure 7, the proofs are identical with those given in the proof of Theorem 3, using Lemmas in Section 5.5 and by noting, for prefix rules, that TONs can be treated just as names in the open state logic.

For the rules in Figure 10, we establish the soundness of the derivation in both the standard direction (from the antecedent to the conclusion) and the reverse one (from the conclusion to the antecedent), in each case assuming the well-typedness of the sequent(s) in both parts. We use $\vdash'$, which does not lose generality by by above Claim. The two directions are simultaneously established by induction on formula to which substitution is applied. Throughout we fix well-typed, arbitrary interpretation $\mathcal{I}$. By duality we only have to consider $\wedge$, $\neg$ and $\forall$ as logical connectives. First we treat:

$$(\mathsf{Hide})\ \frac{P \vdash' \mathbf{rely}\, A\, \mathbf{guar}\, B[\boldsymbol{\nu}\vec{y}_i\vec{z}_i/\boldsymbol{\nu}\vec{y}_i x\vec{z}_i]_i}{(\boldsymbol{\nu}\, x)P \vdash' \mathbf{rely}\, A\, \mathbf{guar}\, B},$$

[1] The refined typing rules are equivalent to the standard affine typing rules up to strong bisimilarity, except that the former cannot derive a process with a linear input and one or more replicated processes. Here we simply stipulate the typed congruence is considered under the generation of terms in the refined typing rules.

We show, by induction on B, that $P \models^{\mathcal{I}} \mathbf{rely}\, A\, \mathbf{guar}\, B\boldsymbol{\sigma}$ if and only if $(\boldsymbol{\nu}\, x)P \models^{\mathcal{I}} \mathbf{rely}\, A\, \mathbf{guar}\, B$ assuming (strict) typability of both formulae in each direction, writing $\boldsymbol{\sigma}$ for the involved substitution $[\vec{y}_i\vec{z}_i/\vec{y}_i x\vec{z}_i]_i$.

Case $B \stackrel{\text{def}}{=} e_1 = e_2$. x never occurs hence immediate from Lemma 44 (2).

Case $B \stackrel{\text{def}}{=} a_1 = a_2$. Same as above.

Case $B \stackrel{\text{def}}{=} \{C\}u \bullet \vec{v}\{C'\}$. In this case, x does not occur in C and C' by typing, hence immediate from Lemma 44 (2).

Case $B \stackrel{\text{def}}{=} \langle\{C\}u \bullet \vec{v}/z\rangle B'$. Note x can only occur in B' by typing. For legibility below we let primary (reference) names in C be disjoint from those in B' and let, w.l.o.g. $P \equiv U^u|V^{\vec{v}}|R$ where P^x means $\mathsf{fn}(P) = \{x\}$, and use the free output notation encoded by copy-cats in the standard way. Let $R \models'^{\mathcal{I}} A$.

$$
\begin{array}{lll}
P|R \models'^{\mathcal{I}} \langle\{C\}u \bullet \vec{v}/z\rangle B'\boldsymbol{\sigma} & & \\
\quad \equiv \quad \forall S \models'^{\mathcal{I}} C.\ (\boldsymbol{\nu}\, \vec{v}z)(U|\overline{u}\langle\vec{v}z\rangle|V)|S \models'^{\mathcal{I}} (B'\boldsymbol{\sigma}) & & (\text{Def. of } \models'^{\mathcal{I}}) \\
\quad \equiv \quad \forall S \models'^{\mathcal{I}} C.\ (\boldsymbol{\nu}\, x)((\boldsymbol{\nu}\, \vec{v}z)(U|\overline{u}\langle\vec{v}z\rangle|V)|S)|R \models'^{\mathcal{I}} B' & & (\text{IH}) \\
\quad \equiv \quad \forall S \models'^{\mathcal{I}} C.\ (\boldsymbol{\nu}\, \vec{v}z)(\boldsymbol{\nu}\, x)(U|\overline{u}\langle\vec{v}z\rangle|V)|S\eta \models'^{\mathcal{I}} B' & & (x \notin \mathsf{fn}(S)) \\
\quad \equiv \quad ((\boldsymbol{\nu}\, x)P)|R \models'^{\mathcal{I}} \langle\{C\}u \bullet \vec{v}/z\rangle B' & & (\text{Def. of } \models'^{\mathcal{I}})\ .
\end{array}
$$

Case $B \stackrel{\text{def}}{=} B_1 \wedge B_2$. We infer, again with $R \models'^{\mathcal{I}} A$:

$$
\begin{array}{lll}
P|R \models'^{\mathcal{I}} (B_1 \wedge B_2)\boldsymbol{\sigma} \quad \equiv \quad P|R \models'^{\mathcal{I}} B_1\boldsymbol{\sigma} \ \wedge\ P \models'^{\mathcal{I}} B_2\boldsymbol{\sigma} & & (\text{Def. of } \models'^{\mathcal{I}}) \\
\qquad\qquad\quad \equiv \quad ((\boldsymbol{\nu}\, x)P)|R \models'^{\mathcal{I}} B_1 \ \wedge\ (\boldsymbol{\nu}\, x)P \models'^{\mathcal{I}} B_2 & & (\text{IH}) \\
\qquad\qquad\quad \equiv \quad ((\boldsymbol{\nu}\, x)P)|R \models'^{\mathcal{I}} B_1 \wedge B_2 & & (\text{Def. of } \models'^{\mathcal{I}})\ .
\end{array}
$$

Case $B \stackrel{\text{def}}{=} \neg B'$. Again with $R \models'^{\mathcal{I}} A$:

$$
\begin{array}{lll}
P|R \models'^{\mathcal{I}} \neg B' \quad \equiv \quad \neg\, P|R \models'^{\mathcal{I}} B'\boldsymbol{\sigma} & & (\text{Def. of } \models'^{\mathcal{I}}) \\
\qquad\quad \equiv \quad \neg\, ((\boldsymbol{\nu}\, x)P)|R \models'^{\mathcal{I}} B' & & (\text{IH}) \\
\qquad\quad \equiv \quad ((\boldsymbol{\nu}\, x)P)|R \models'^{\mathcal{I}} \neg B' & & (\text{Def. of } \models'^{\mathcal{I}})\ .
\end{array}
$$

Case $B \stackrel{\text{def}}{=} \forall z^{\tau^{!,\uparrow}}.B'$. With $R \models'^{\mathcal{I}} A$:

$$
\begin{array}{lll}
P|R \models'^{\mathcal{I}} \forall z^{\tau^{!,\uparrow}}.B'\boldsymbol{\sigma} \quad \equiv \quad \forall \alpha.\ P|R \models'^{I\cdot z:\alpha}.B'\boldsymbol{\sigma} & & (\text{Def. of } \models'^{\mathcal{I}}) \\
\qquad\qquad\quad \equiv \quad \forall \alpha.((\boldsymbol{\nu}\, x)P)|R \models'^{I\cdot z:\alpha}.B' & & (\text{IH}) \\
\qquad\qquad\quad \equiv \quad ((\boldsymbol{\nu}\, x)P)|R \models'^{\mathcal{I}} \forall z^{\tau^{!,\uparrow}}.B' & & (\text{Def. of } \models'^{\mathcal{I}})\ .
\end{array}
$$

Case $B \stackrel{\text{def}}{=} \forall z^{[\tau]}.B'$ and $B \stackrel{\text{def}}{=} \forall z^{ref(\tau)}.B'$. As the case above.

Case $B \stackrel{\text{def}}{=} \mathsf{open}\ \vec{w}\ \mathsf{as}\ (\boldsymbol{\nu}\vec{y}x\vec{z})\underline{\vec{w}^{\vec{\tau}}}\ \mathsf{in}\ B'$. By appropriate α-conversion, we safely assume $\vec{w}$ occur in P with type $\vec{\tau}$. Further, by the condition on name occurrence for opening formulae, B' does not contain any name from $\vec{w}$, hence no hiding of the form $(\boldsymbol{\nu}\, \vec{y}_i x\vec{z}_i)$ can occur in B'. Thus:

$$
\begin{array}{lll}
P \models'^{\mathcal{I}} (\mathsf{open}\ \vec{w}\ \mathsf{as}\ (\boldsymbol{\nu}\vec{y}x\vec{z})\underline{\vec{w}}\ \mathsf{in}\ B')\boldsymbol{\sigma} & & \\
\qquad \equiv \quad P \models'^{\mathcal{I}} \mathsf{open}\ \vec{w}\ \mathsf{as}\ (\boldsymbol{\nu}\vec{y}\vec{z})\underline{\vec{w}}\ \mathsf{in}\ B' & & \\
\qquad \equiv \quad (\boldsymbol{\nu}\, x)P \models'^{\mathcal{I}} \mathsf{open}\ \vec{w}\ \mathsf{as}\ (\boldsymbol{\nu}\vec{y}x\vec{z})\underline{\vec{w}}\ \mathsf{in}\ B' & & (\text{Lemma 44 (1)}).
\end{array}
$$

(Hide-thread) is reasoned precisely the same way. For co-hiding: we first treat:

$$\text{(Co-Hide)}\ \frac{P \vdash \mathbf{rely}\ A[\boldsymbol{\nu}\,\vec{y}_i\vec{z}_i/\boldsymbol{\nu}\,\vec{y}_i x\vec{z}_i]_i\ \mathbf{guar}\ B}{P \vdash \mathbf{rely}\ A\ \mathbf{guar}\ B}$$

We again argue by induction on A. $x \notin \mathsf{fn}(B)$ is implicit in the rule, which is crucial for the reasoning. We only treat the case of opening formulae, i.e. $A \stackrel{\text{def}}{=} \mathsf{open}\ \vec{w}\ \mathsf{as}\ (\boldsymbol{\nu}\vec{y}x\vec{z})\underline{\vec{w}^{\vec{\tau}}}\ \mathsf{in}\ A'$. By appropriate α-conversion, we safely assume $\vec{w}$ occur in R with type $\vec{\tau}$. First we show the antecedent implies the conclusion:

$$\begin{array}{lll}
R^{\Gamma} \models'^{\mathcal{I}} \mathsf{open}\ \vec{w}\ \mathsf{as}\ (\boldsymbol{\nu}\vec{y}x\vec{z})\underline{\vec{w}^{\vec{\tau}}}\ \mathsf{in}\ A' & & \\
\supset \quad R \equiv (\boldsymbol{\nu}\,x)R' \quad \wedge \quad R' \models'^{\mathcal{I}} \mathsf{open}\ \vec{w}\ \mathsf{as}\ (\boldsymbol{\nu}\vec{y}\vec{z})\underline{\vec{w}^{\vec{\tau}}}\ \mathsf{in}\ A' & & \text{(Def. of } \models'^{\mathcal{I}}) \\
\supset \quad R \equiv (\boldsymbol{\nu}\,x)R' \quad \wedge \quad P|R' \models'^{\mathcal{I}} B & & \text{(IH)} \\
\supset \quad R \equiv (\boldsymbol{\nu}\,x)R' \quad \wedge \quad P|(\boldsymbol{\nu}\,x)R' \models'^{\mathcal{I}} B & & \text{(Lemma 44 (2)).}
\end{array}$$

The other direction (under appropriate α-conversion):

$$\begin{array}{lll}
R^{\Gamma} \models'^{\mathcal{I}} \mathsf{open}\ \vec{w}\ \mathsf{as}\ (\boldsymbol{\nu}\vec{y}\vec{z})\underline{\vec{w}^{\vec{\tau}}}\ \mathsf{in}\ A' & & \\
\supset \quad R \equiv (\boldsymbol{\nu}\,\vec{y}\vec{z})R' \quad \wedge \quad R' \models'^{\mathcal{I}} A' & & \text{(Def. of } \models'^{\mathcal{I}}) \\
\supset \quad (\boldsymbol{\nu}\,x)R \models'^{\mathcal{I}} \mathsf{open}\ \vec{w}\ \mathsf{as}\ (\boldsymbol{\nu}\vec{y}x\vec{z})\underline{\vec{w}^{\vec{\tau}}}\ \mathsf{in}\ A' & & \text{(Def. of } \models'^{\mathcal{I}}) \\
\supset \quad R \equiv (\boldsymbol{\nu}\,x)R' \quad \wedge \quad P|R' \models'^{\mathcal{I}} B & & \text{(IH)} \\
\supset \quad R \equiv (\boldsymbol{\nu}\,x)R' \quad \wedge \quad P|(\boldsymbol{\nu}\,x)R' \models'^{\mathcal{I}} B & & \text{(Lemma 44 (2)).} \quad \square
\end{array}$$

For its thread version, we consider the following essentially equivalent rule:

$$\text{(Co-Hide-thread)}\ \frac{P \vdash \{C \wedge E\}\ \mathbf{rely}\ A_0\ \mathbf{guar}\ B\ \{C'\} \quad A[\boldsymbol{\nu}\,\vec{y}_i\vec{z}_i/\boldsymbol{\nu}\,\vec{y}_i x\vec{z}_i]_i \equiv A_0 \wedge E}{P \vdash \{C\}\ \mathbf{rely}\ A\ \mathbf{guar}\ B\ \{C'\}}$$

We again only treat the case when A is the opening formula. First we show the standard direction. Assume $\mathsf{open}\ \vec{w}\ \mathsf{as}\ (\boldsymbol{\nu}\vec{y}x\vec{z})\underline{\vec{w}}\ \mathsf{in}\ (A \wedge B)$ is equivalent to $(\mathsf{open}\ \vec{w}\ \mathsf{as}\ (\boldsymbol{\nu}\vec{y}\vec{z})\underline{\vec{w}}\ \mathsf{in}\ A) \wedge B$ iff x is the only prime name of B.

$$\begin{array}{lll}
R^{\Gamma} \models'^{\mathcal{I}} C \wedge (\mathsf{open}\ \vec{w}\ \mathsf{as}\ (\boldsymbol{\nu}\vec{y}x\vec{z})\underline{\vec{w}^{\vec{\tau}}}\ \mathsf{in}\ A') & & \\
\supset \quad R \equiv Q|(\boldsymbol{\nu}\,x)(R'|S^x) \quad \wedge \quad Q \models'^{\mathcal{I}} C & & \\
\qquad R'|S \models'^{\mathcal{I}} (\mathsf{open}\ \vec{w}\ \mathsf{as}\ (\boldsymbol{\nu}\vec{y}\vec{z})\underline{\vec{w}^{\vec{\tau}}}\ \mathsf{in}\ A_0) \wedge E & & (\models'^{\mathcal{I}}) \\
\supset \quad R \equiv (\boldsymbol{\nu}\,x)(Q|R'|S^x) \quad \wedge \quad P|(Q|R'|S) \models'^{\mathcal{I}} B \wedge C' & & \text{(IH)} \\
\supset \quad R \equiv (\boldsymbol{\nu}\,x)(Q|R'|S^x) \quad \wedge \quad P|R \models'^{\mathcal{I}} B \wedge C' & & \text{(Lem. 44 (2)).}
\end{array}$$

For the reverse direction, we consider the side condition in the antecedent is the condition to be assumed. We can again assume the following equivalence (under the given typings): $\mathsf{open}\ \vec{w}\ \mathsf{as}\ (\boldsymbol{\nu}\vec{y}\vec{z})\underline{\vec{w}^{\vec{\tau}}}\ \mathsf{in}\ A' \equiv (\mathsf{open}\ \vec{w}\ \mathsf{as}\ (\boldsymbol{\nu}\vec{y}\vec{z})\underline{\vec{w}^{\vec{\tau}}}\ \mathsf{in}\ A'_0) \wedge E$ where, without loss of generality, $A_0 \equiv (\mathsf{open}\ \vec{w}\ \mathsf{as}\ (\boldsymbol{\nu}\vec{y}\vec{z})\underline{\vec{w}^{\vec{\tau}}}\ \mathsf{in}\ A'_0)$.

$$\begin{array}{lll}
R^{\Gamma} \models'^{\mathcal{I}} C \wedge E \wedge A_0 & & \\
\supset \quad R \equiv Q|(\boldsymbol{\nu}\,\vec{y}\vec{z})R'|S^x \quad \wedge \quad R'|S \models'^{\mathcal{I}} A' & & \text{(Def. of } \models'^{\mathcal{I}}) \\
\supset \quad R \equiv Q|(\boldsymbol{\nu}\,\vec{y}\vec{z})R'|S^x \quad \wedge \quad P|Q|(\boldsymbol{\nu}\,x\vec{y}\vec{z})R'|S \models'^{\mathcal{I}} B \wedge C' & & \text{(IH)} \\
\supset \quad (\boldsymbol{\nu}\,x)R \equiv Q|(\boldsymbol{\nu}\,\vec{y}x\vec{z})(R'|S^x) \quad \wedge \quad P|(\boldsymbol{\nu}\,x)R \models'^{\mathcal{I}} B & & \text{(Lemma 44 (2)).}
\end{array}$$

Other cases are mechanical from induction hypothesis hence omitted. $\square$

6 Partial Logic

6.1 Affine Logic for Partial Correctness

In this section we show the partial counterpart of the logic in Section 3 (pure affine logic). The stateful extensions should be treated in the same way. In fact, the only change from the total logic is in (co-)replication.

Because of the combination of partiality and higher-order feature, we need a subtle change in the language of the logic and its interpretation, in particular in expressions of the form $x \bullet \vec{y}$. While we can keep them by changing their interpretation, here we take them off and add the following form to formulae: in fact, we can define equations of the original shape with changed interpretation using the new construct, so this loses no generality.

$$A \quad ::= \quad ... \quad | \quad \langle\!\langle x \bullet \vec{y}/z \rangle\!\rangle A$$

We sometimes call this added formula, *partial substitution formula*. Intuitively, $\langle\!\langle x \bullet \vec{y}/z \rangle\!\rangle A$ means invoking x with $\vec{y}$ either diverges, or if it converges then we set the returned value to be z and demands it satisfies A. Using this construct, we can set, for example,

$$x \bullet \vec{y} = \mathtt{in}_i(\vec{u}) \quad \stackrel{\text{def}}{=} \quad \langle\!\langle x \bullet \vec{y}/z \rangle\!\rangle (z = \mathtt{in}_i(\vec{u})).$$

By the informal semantics noted above, this equation holds if $x \bullet \vec{y}$ diverges, substantiating safety-based specification for partial computation. The negation of such formulae is defined similarly

$$x \bullet \vec{y} \neq \mathtt{in}_i(\vec{u}) \quad \stackrel{\text{def}}{=} \quad \langle\!\langle x \bullet \vec{y}/z \rangle\!\rangle (z \neq \mathtt{in}_i(\vec{u})).$$

Note this is different from "$\neg x \bullet \vec{y} = \mathtt{in}_i(\vec{u})$", which does not give what we usually expect from the partial correctness.

We use the sequent of the same form $P^{\overline{\Gamma};\Delta} \models^{\mathcal{I}} \mathbf{rely} A \ \mathbf{guar} B$ as we used in Section 3. Interpretations of formulae and sequent are given as follows. Below and henceforth we always assume the well-typedness/formedness.

- We use the same set of models $(\xi, \xi', \ldots)$ as given in Section 3.2.
- $P \models_{\mathsf{b}} \xi$ is also given precisely as in Section 3.2.
- $\xi \models^{\mathcal{I}} A$ is defined by the following clauses plus those in Section 2.2 (omitting the behavioural expressions of the form $x \bullet \vec{y}$).
  - $\xi \cdot x\!:\!\omega \models^{\mathcal{I}} A$ for any A.
  - $\xi \models^{\mathcal{I}} \langle\!\langle x \bullet \vec{y}/z \rangle\!\rangle A$ if $\xi(x) : \xi(\vec{y}) \mapsto \gamma$ and $\xi \cdot z\!:\!\gamma \models^{\mathcal{I}} A$.
- $P \models^{\mathcal{I}} A$ is defined to be $\exists \xi. P \models_{\mathsf{b}} \xi \models_{\mathsf{b}} A$, just as before.
- The sequent $P^{\overline{\Gamma};\Delta} \models^{\mathcal{I}} \mathbf{rely} A \ \mathbf{guar} B$ is also interpreted in the same way: $P^{\overline{\Gamma};\Delta} \models^{\mathcal{I}} \mathbf{rely} A \ \mathbf{guar} B$ iff $R^{\Gamma} \models^{\mathcal{I}} A \supset (\boldsymbol{\nu}\, \mathsf{fn}(\Gamma))(P|R) \models^{\mathcal{I}} B$.

Note the definition of $\models^{\mathcal{I}}$ follows the standard idea of partial correctness, saying we do not care about the properties of divergent processes.

6.2 Admissible Formulae

We can represent a liveness property using the formulae given above. As a simple example, assuming x is a primary name, $P \models \neg(\langle\!\langle x \bullet \varepsilon/z \rangle\!\rangle \mathsf{F})$ says that P, when invoked at x, surely terminates. This point, as well as others, cause a problem for formulating a sound proof rule for recursion. This motivates the following restriction to formulae, called admissible. Below and henceforth we write $\Omega_x^\tau \stackrel{\text{def}}{=} (!x(\vec{y}z).\omega_z)^{x:\tau}$ (ω_z is a diverging process). We usually omit τ and simply write Ω_s. Note Ω_x diverges immediately after its initial invocation. $\mathsf{trace}(\mathsf{P}^\Gamma)$ is the set of well-typed traces of P^Γ [4], while $P^\Gamma \sqsubseteq Q^\Gamma$ denotes $\mathsf{trace}(\mathsf{P}^\Gamma) \subset \mathsf{trace}(\mathsf{Q}^\Gamma)$. We also say A is *I satisfiable* if we have $\xi \models^{\mathcal{I}} A$ for some $\Gamma \vdash \xi$.

Definition 3. Assume $!\Gamma;;\Theta \vdash A$ with $\mathsf{dom}(\Gamma) = \{\vec{x}\}$ and A is $\mathcal{I}$-satisfiable for some $\Theta \vdash I$. Then A is *admissible under $\mathcal{I}$*, or simply *admissible* if the following conditions hold.

1. $\Pi_i \Omega_i \models^{\mathcal{I}} A$ for each $\Theta \vdash I$.
2. Whenever $P^\Gamma \sqsubseteq Q^\Gamma$ and $Q \models^{\mathcal{I}} A$, we have $P \models^{\mathcal{I}} A$.
3. Given $\{P_i^\Gamma\}_{i\in\mathbb{N}}$ ($\mathbb{N}$ is the set of natural numbers) and P_ω^Γ such that $P_i \sqsubseteq P_{i+1}$ and $\mathsf{trace}(\mathsf{P}_\omega^\Gamma) = \cup_i \mathsf{trace}(\mathsf{P}_i^\Gamma)$, we have $P_\omega \models^{\mathcal{I}} A$ iff $P_i \models^{\mathcal{I}} A$ for each i.

The admissibility as given above is behavioural, using $\models^{\mathcal{I}}$: the following gives one possible sound syntactic characterisation.

Definition 4. Let $!\Gamma;;\Theta \vdash A$ with $\mathsf{dom}(\Gamma) = \{\vec{x}\}$. Then the set of *syntactically admissible formulae at $\vec{x}$*, or *s-admissible formulae at $\vec{x}$* for short, is inductively generated as follows. Below we assume formulae are typed with given names as part of its primary names.

- T is s-admissible at $\vec{x}$.
- $e_1 = e_2$ is s-admissible at $\vec{x}$.
- $a_1 = a_2$ is s-admissible at $\emptyset$.
- $\langle\!\langle x \bullet \vec{y}/z \rangle\!\rangle A$ is s-admissible at $\vec{w}$ with $x \in \{\vec{w}\}$ if A is.
- $A_1 \wedge A_2$ is a s-admissible at $\vec{x}$ when $A_{1,2}$ are.
- $A_1 \vee A_2$ is a s-admissible at $\vec{x}$ when $A_{1,2}$ are.
- $A_1 \supset A_2$ such that $\mathsf{fn}(A_1) \cap \{\vec{x}\} = \emptyset$ is a s-admissible at $\vec{x}$ when A_2 is.
- $\forall x.A$ is s-admissible at $\vec{y}$ with $x \notin \{\vec{y}\}$ when A is.

We say A is *syntactically admissible*, or *s-admissible* for short, if it is a s-admissible at $\mathsf{dom}(\Gamma)$.

We can further incorporate significant instances of the formulae of the forms $\exists i.A$ and $\exists x.A$. We observe:

Proposition 4. *Given $\Gamma;;\Delta \vdash A$, if A is syntactically admissible and $\mathcal{I}$-satisfiable for some $\Theta \vdash I$, then A is admissible under $\mathcal{I}$.*

Proof. We show We establish the three conditions by induction on the generation of safety formulae. For the first condition:

Case $A \stackrel{\mathbf{def}}{=} \mathsf{T}$**:** Vacuous.

Case $A \stackrel{\mathbf{def}}{=} e_1 = e_2$**:** Because $P \models^{\mathcal{I}} e_1 = e_2$ for any P^{Γ} by $\mathcal{I}$-satisfiability.

Case $A \stackrel{\mathbf{def}}{=} a_1 = a_2$**:** In this case if A is a safety formula then $\mathsf{fn}(a_1) = \mathsf{fn}(a_2) = \emptyset$ hence $P \models^{\mathcal{I}} A$ for any P^{Γ} by $\mathcal{I}$-satisfiability.

Case $\langle\!\langle x \bullet \vec{y}/z \rangle\!\rangle A$**:** Because $x : \Omega \cdot \xi \models^{\mathcal{I}} \langle\!\langle x \bullet \vec{y}/z \rangle\!\rangle A$, where we write Ω for the behavioural constant corresponding to this behaviour.

Case $A_1 \wedge A_2$**:** Because $\Pi_i \Omega_{x_i} \models^{\mathcal{I}} A_{1,2}$ by induction hypothesis.

Case $A_1 \vee A_2$**:** Because $\Pi_i \Omega_{x_i} \models^{\mathcal{I}} A_i$ for $i = 1$ or $i = 2$ by induction hypothesis.

Case $\forall i.A$**:** By $\mathcal{I}$-satisfiability, $\xi \models^{\mathcal{I}} \forall i.A$ for some ξ, hence setting $I' = I \cdot i : \alpha$ for an arbitrary well-typed α, we have $\xi \models^{\mathcal{I}'} A$, that is A is $\mathcal{I}'$-satisfiable. Hence A is a semantic safety property, that is $\Pi\Omega_{x_i} \models^{\mathcal{I}'} A$ for each I'. Hence $\Pi\Omega_{x_i} \models^{\mathcal{I}} \forall i.A$, as required. $\exists i.A$ is similar.

Case $\forall x.A$. As above.

The second condition is proved in the same way. Similarly for the third condition (approximation), for which we only have to show the direction $\forall i \in \mathbb{N}.\, P \models^{\mathcal{I}} A$ implies $P_\omega \models^{\mathcal{I}} A$, except for partial substitution. We list this case below.

Case $\langle\!\langle x \bullet \vec{y}/z \rangle\!\rangle A$**:** Assume $P_i \models^{\mathcal{I}} \langle\!\langle x \bullet \vec{y}/z \rangle\!\rangle A$ for each $i \in \mathbb{N}$. If (the defining process of) $x \bullet \vec{y}$ diverges for each P_i, then this is shown in the corresponding trace of P_i (i.e. has no corresponding linear answer), hence P_ω does not have it either, showing $P_\omega \langle\!\langle x \bullet \vec{y}/z \rangle\!\rangle A$. On the other hand, assume $x \bullet \vec{y}$ converges for some P_i, hence for each P_n with $n \geq i$, including P_ω. By definition, this means, with $P'_n | \overline{z}\mathtt{in}_j(\vec{u}) Q'_n$ being the resulting process, $P'_n | Q'_n \models^{\mathcal{I}} A[\mathtt{in}_j(\vec{u})/z]$. Noting Q'_j again forms a similar chain (since the induced trace equivalence is a congruence), we can conclude $P_\omega | Q_\omega \models^{\mathcal{I}} A[\mathtt{in}_j(\vec{u})/z]$ by induction hypothesis. Hence $P'_\omega | \overline{z}\mathtt{in}_j(\vec{u}) Q'_\omega \models^{\mathcal{I}} A$, from which we conclude $P_\omega \models^{\mathcal{I}} \langle\!\langle x \bullet \vec{y}/z \rangle\!\rangle A$. □

6.3 Properties of Recursion

The following unfolding of recursion is the key to the soundness of its proof rule, together with admissibility.

Lemma 49. *Let* $\vdash P \triangleright ?\overline{\Gamma} \cdot y : \overline{\tau}^{?}$; $x : \tau$ *and* $\vdash R \triangleright \Gamma$*, and define* $\mathbf{rec}^n(Q)$ *(*$n \geq 0$*) by the following induction.*

$$\mathbf{rec}^0(Q) \stackrel{def}{=} \Omega^{\tau}_x$$
$$\mathbf{rec}^{n+1}(Q) \stackrel{def}{=} (\boldsymbol{\nu}\, \mathsf{fn}(\Gamma), y)(P | R | \mathbf{rec}^n(Q_n[y/x]).$$

Further set $Q_\omega \stackrel{def}{=} (\boldsymbol{\nu}\, \mathsf{fn}(\Gamma))(\mu y = x.P | R)$*. Then we have:* (1) $\mathsf{trace}(\mathsf{Q_i}) \subset \mathsf{trace}(\mathsf{Q_{i+1}})$ *for each* i*; and* (2) $\mathsf{trace}(\mathsf{Q}_\omega) = \cup_{\mathsf{i}} \mathsf{trace}(\mathsf{Q_i})$.

Proof. We reduce the statement to a simpler case. Define $\mathbf{rec}^n(P)$ $(n \geq 0)$ by the following induction.

$$\mathbf{rec}^0(P) \stackrel{\text{def}}{=} \Omega_x^\tau$$
$$\mathbf{rec}^{n+1}(P) \stackrel{\text{def}}{=} (\boldsymbol{\nu}\, y)(P|\mathbf{rec}^n(P_n[y/x]).$$

We again set $P_\omega \stackrel{\text{def}}{=} \mu y = x.P$. We first show the stated two properties for this simplified chain of processes. For this purpose we unfold the behaviours of $\mathbf{rec}^n(P)$ and $\mathbf{rec}^n(P)$. First, by Lemma 17, we have:

$$\mathbf{rec}^\omega(P) \approx (\boldsymbol{\nu}\, y_1..y_n)(P|P[y_1y_2/xy]|..|P[y_{n-1}y_n/xy]|\mathbf{rec}^\omega(P)). \quad (1)$$

Note the explicit trace of the right-hand side involves P together with its interactions with its copy of P which would further invoke another copy of P, and so on. On the other hand, by simply expanding the definition, we obtain:

$$\mathbf{rec}^{n+1}(P) \stackrel{\text{def}}{=} (\boldsymbol{\nu}\, y_1..y_n)(P|P[y_1y_2/xy]|P[y_2y_3/xy]||..|P[y_{n-1}y_n/xy]|\Omega_{y_n}). \quad (2)$$

which again have the same behaviour except it is "terminated" at Ω_{y_n}. Clearly any explicit trace of the r.h.s. of (2) is precisely mimickable by the r.h.s. of (1), showing $\mathsf{trace}(\mathsf{P}_i) \subset \mathsf{trace}(\mathsf{P}_{i+1})$. On the other hand, an arbitrary explicit transition of r.h.s. of (1) can be mimicked by the r.h.s. of (2) by taking a sufficiently large n. Thus any trace in $\mathbf{rec}^\omega(P)$ is in $\mathbf{rec}^n(P)$ for some n, that is, together with the first property, we know $\cup_n \mathsf{trace}(\mathbf{rec}^{\mathsf{n}}(\mathsf{P})) = \mathsf{trace}(\mathbf{rec}^\omega(\mathsf{P}))$. Finally we observe $\mathbf{rec}^n(Q) \approx (\boldsymbol{\nu}\, \mathsf{fn}(\Gamma))(\mathbf{rec}^n(P)|R)$ by repeating Lemma 1 (2) (the replication theorem), from which we conclude the required properties by the traces of processes being congruence properties. □

6.4 Properties of Partial Affine Logic

The statement we gave for the total affine logic in Section 3 hold without change in its partial counterpart (the proofs are also identical line by line even though the interpretation is different, as well as the use of partial substitution).

Lemma 50. *Let* $P \models_{\mathsf{b}} \xi_{1,2}$. *Then* $\xi_1 \models^{\mathcal{I}} A$ *iff* $\xi_2 \models^{\mathcal{I}} A$.

Lemma 51. *If* $\xi \models^{\mathcal{I}} A$ *and* $A \supset B$ *then* $\xi \models^{\mathcal{I}} B$.

Corollary 7. *If* $P \models^{\mathcal{I}} A$ *and* $A \supset B$ *then* $P \models^{\mathcal{I}} B$.

Lemma 52. *Let* $\mathsf{dom}(\xi') \cap \mathsf{fn}(A) = \emptyset$. *Then* $\xi \cdot \xi' \models^{\mathcal{I}} A$ *iff* $\xi \models^{\mathcal{I}} A$.

Lemma 53. (monotonicity of $\boldsymbol{\nu}$) *Let* $x \notin \mathsf{fn}(A)$. *Then* $P \models^{\mathcal{I}} A$ *iff* $(\boldsymbol{\nu}\, x)P \models^{\mathcal{I}} A$.

Lemma 54. (cut in $\models^{\mathcal{I}}$) *Let* $!\Gamma' \subset \Gamma$, $\Gamma_0 \subset \Gamma$ *and* $\Gamma';;\Theta \vdash A$ *such that* $\Theta \vdash I$. *Then* $P^{\overline{\Gamma_0};\Delta} \models \mathbf{rely}\, A_0\, \mathbf{guar}\, B$ *and* $R^\Gamma \models^{\mathcal{I}} A \wedge A_0$ *imply* $P|R \models^{\mathcal{I}} A \wedge B$.

Lemma 55. *If* $x \notin \mathsf{fn}(A, a)$, *then* $\xi \models^{\mathcal{I}} A[a/x]$ *iff* $\xi \models^{\mathcal{I}} \exists x.(A \wedge x = a)$.

Lemma 56. $\xi \cdot x{:}\mathtt{in}_i(\vec{\alpha})^\uparrow \models^{\mathcal{I}} A$ *iff* $\xi \cdot \vec{y}{:}\vec{\alpha} \models^{\mathcal{I}} A[\mathtt{in}_i(\vec{y})^\uparrow/x]$.

Lemma 57. *Let* $\mathsf{fn}(B) \cap \{x\vec{y}\} = \emptyset$. *Then* $\xi \cdot x : \wedge_{i\in J}(\vec{\alpha}_i\beta_i)^! \cdot \vec{y}{:}\vec{\alpha}_i \models^{\mathcal{I}} \langle\!\langle x \bullet \vec{y}/z \rangle\!\rangle B$ *iff* $\xi \cdot z{:}\beta_i \models^{\mathcal{I}} B$.

$$\frac{\begin{array}{l}(\mathsf{Rec}) \quad (B \text{ s-admissible},\ A \supset \exists \mathsf{prim}(B).B) \\ P \vdash \mathbf{rely}\ A^{-y} \wedge B[y/x]\ \mathbf{guar}\ B\end{array}}{\mu y{=}x.P \vdash \mathbf{rely}\ A\ \mathbf{guar}\ B}$$

Fig. 11. Proof Rule for Recursion (partial correctness)

6.5 Soundness of Proof Rules

The proof system for the partial affine logic uses the rules from Figure 2 in which the substitution of the form $A[x \bullet \vec{y}/z]$ in $(\mathsf{In}^!)$ and $(\mathsf{Out}^?)$ is replaced by $\langle\!\langle x \bullet \vec{y}/z \rangle\!\rangle A$, as well as the new rule for recursion, given in Figure 11. The given condition is precisely what is needed for soundness. In essence,

$$A \supset \exists \mathsf{prim}(B).B$$

says that the constraints which A has on its auxiliary names should be stronger than that of B (since we can always assign least defined processes to $\mathsf{prim}(B)$ to validate the property, checking this property is usually not hard in practice). Without this side condition, we can easily derive inconsistent assertions. For example, we can derive $\mu y = x.[x \to y] \vdash$ **rely**T **guar**Fa from $[x \to y] \vdash$ **rely**F **guar**F. We also note the s-admissibility in the side condition can be replaced by admissibility: we only use the latter in the proof of soundness below. We now establish:

Theorem 5. *The proof system of the partial affine logic is sound.*

Proof. The rules from Figure 2 (with substitution replaced) are proved precisely in the same way, line by line, as in the proof of Theorem 1, replacing the lemmas employed with those for partial logic as needed (note, in particular, Lemma 57 for partial substitution has precisely the same shape as Lemma 13). Thus it suffices to prove the soundness of the recursion rule:

$$(\mathsf{Rec})\ \frac{P^{\overline{\Gamma}\cdot y:\overline{\tau}\,;\,x:\tau} \vdash \mathbf{rely}\ A^{-y} \wedge B[y/x]\ \mathbf{guar}\ B,\quad A \supset \exists \mathsf{prim}(B).B,\quad B \text{ s-admissible}}{\mu y{=}x.P \vdash \mathbf{rely}\ A\ \mathbf{guar}\ B}$$

Let:

$$\begin{array}{ll}(\star) & A \supset \exists \mathsf{prim}(B).B \\ (\star\star) & B \text{ s-admissible}\end{array}$$

and assume $\vdash R \triangleright \Gamma$ below. We infer:

$$\begin{array}{lllll} R^\Gamma \models^{\mathcal{I}} A & & & & \\ \quad \supset & R^\Gamma \models^{\mathcal{I}} A & \wedge & A\ \mathcal{I}\text{-satisfiable} & (\star) \\ \quad \supset & R^\Gamma \models^{\mathcal{I}} A & \wedge & B \text{ admissible under } \mathcal{I} & ((\star\star), \text{Proposition 4}) \\ \quad \supset & (\boldsymbol{\nu}\,\mathsf{fn}(\Gamma))(\mu y{=}x.P|R) \models^{\mathcal{I}} B & & & (\text{Lemma 49, Definition 3})\ . \quad \square \end{array}$$

At this point we tentatively close our investigation of the relationship between semantics and proof rules in sequential process logics.

End, November 2003.

Revised (Section 3), January 2004.

References

1. Abramsky, S., Jagadeesan, R. and Malacaria, P., Full Abstraction for PCF, 1994. *Info. & Comp.* 163 (2000), 409-470.
2. Apt, K.R. Ten Years of Hoare Logic: a survey. *TOPLAS*, 3, pp.431-483, 1981.
3. Ashcroft, E. A., Clint, M. Hoare C.A.R. Remarks on "Program proving: jumps and functions by M. Clint and C.A.R Hoare." *Acta Informatica*, vol 6. pp.317-318, 1976.
4. Berger, M., Honda, K. and Yoshida, N., Sequentiality and the π-Calculus, *TLCA01*, LNCS 2044, 29–45, Springer, 2001.
5. Hoare, C.A.R. An axiomatic basis of computer programming. *CACM*, 12:576-580, 1969.
6. Honda, K. *Sequential Process Logics: A Brief Overview.* November, 2003.
7. Honda, K. *Process Logic and Duality*, under preparation.
8. Hyland, M. and Ong, L., "On Full Abstraction for PCF": I, II and III. *Info. & Comp.* 163 (2000), 285-408.
9. Jones, C.B. Spefification and Design of (Parallel) Programs. *Proc. IFIP 9th World Computer Concgress.* North Holland, pp321-332, 1983.
10. Laurent, O., Polarized games, LICS 2002, 265-274, IEEE, 2002.
11. Milner, R., Functions as Processes, *MSCS.* 2(2):119–141, 1992,
12. Milner, R., Parrow, J. and Walker, D., A Calculus of Mobile Processes, *Info. & Comp.* 100(1):1–77, 1992.
13. Nickau, M., Hereditarily Sequential Functionals, LNCS 813, pp.253–264, Springer-Verlag, 1994.
14. Talpin, J-P, Jouvelot, P. The type and effect discpline. *LICS'92*, pp.162-173, 1992.
15. Yoshida, N., Berger, M. and Honda, K., Strong Normalisation in the π-Calculus, *LICS'01*, 311–322, IEEE, 2001. The full version: DoC Technical Report, Department of Computing, Imperial College London, 2003/01. 56 pages. To appear in *Information and Computation.* Available at www.doc.ic.ac.uk/~yoshida.

A Composition of Dual Action Types

The operation $\Theta \odot \overline{\Theta}$ (assuming Θ does not contain $\updownarrow$) is given by the following induction.

- $\emptyset \odot \emptyset \;=\; \emptyset$.
- $(\Theta \cdot x : \tau) \odot (\overline{\Theta} \cdot x : \overline{\tau}) \;=\; \Theta \odot \overline{\Theta} \cdot x : (\tau \odot \overline{\tau})$.

where we set $\tau \odot \overline{\tau} = \overline{\tau} \odot \tau = \begin{cases} \updownarrow & \text{if } \mathsf{md}(\tau) = \downarrow \\ \tau & \text{if } \mathsf{md}(\tau) = \mathbf{!} \end{cases}$.

B Copycat

$[x \to x']^{\tau}$ is given by the following induction.

$$[x \to x']^{(\vec{\tau})^!} \stackrel{\text{def}}{=} !x(\vec{y}).\overline{x'}(\vec{y'})\Pi_i[y'_i \to y_i]^{\overline{\tau_i}}$$
$$[x \to x']^{[\&_i \vec{\tau_i}]^{\downarrow}} \stackrel{\text{def}}{=} x[\&_i(\vec{y}_i).\overline{x'}\mathtt{in}_i(\vec{y'}_i)\Pi_{ij}[y'_{ij} \to y_{ij}]^{\overline{\tau_{ij}}}]$$