

Slide 1

Name-Passing in an Ambient-Like Calculus and Its Proof Using Spatial Logic

Xudong GUAN, DI.FCT.UNL.PT

Slide 2

Part I. The Calculus

Background - Safe Ambients

Grave interference in Mobile Ambients:

$$a[in\ c\ |\ b[out\ a]\]\ |\ c[\]\ \longrightarrow^* c[b[\]\ |\ a[\]\]$$

Slide 3

$$a[in\ c\ |\ b[out\ a]\]\ |\ c[\]\ \longrightarrow^* b[\]\ |\ c[a[\]\]$$

Safe Ambients:

$$a[in\ c.\overline{out}\ a\ |\ b[out\ a]\]\ |\ c[\overline{in}\ c]\ \longrightarrow^* c[b[\]\ |\ a[\]\]$$

$$a[\overline{out}\ a.in\ c\ |\ b[out\ a]\]\ |\ c[\overline{in}\ c]\ \longrightarrow^* b[\]\ |\ c[a[\]\]$$

Background - ST/IM type system

Type system (ST/IM) is essential to eliminate GI in SA,
since $MA \subset SA$:

$$a[\overline{out}\ a\ |\ in\ c\ |\ b[out\ a]\]\ |\ c[\overline{in}\ c]$$

Slide 4

ST type - single-threaded and usually mobile

$$a[out\ s.in\ s'.\dots.\overline{open}\ a\ |\ a'[P]\ |\ s''[Q]\ |\ \dots]$$

IM type - immobile, persistent, and usually multi-threaded

$$s[!\overline{in}\ s\ |\ !\overline{out}\ s\ |\ !open\ a\ |\ a[P]\ |\ s'[Q]\ |\ \dots]$$

From Typed SA (ST agent, IM server) to Wagon

Agent-server interactions:

$$\text{SA} \left(\begin{array}{l} a[\text{in } s.M \mid P] \mid s[!\overline{\text{in}} s \mid Q] \longrightarrow s[a[M \mid P] \mid !\overline{\text{in}} s \mid Q] \\ s[!\overline{\text{out}} s \mid a[\text{out } s.M \mid P] \mid Q] \longrightarrow a[M \mid P] \mid s[!\overline{\text{out}} s \mid Q] \\ s[!\text{open } a \mid a[\overline{\text{open}} a \mid P] \mid Q] \longrightarrow s[!\text{open } a \mid P \mid Q] \end{array} \right)$$

Slide 5

$$\text{Wagon} \left(\begin{array}{l} s.M\langle P \rangle \mid s[Q] \longrightarrow s[M\langle P \rangle \mid Q] \\ s[\uparrow.M\langle P \rangle \mid Q] \longrightarrow M\langle P \rangle \mid s[Q] \\ s[\text{dis}\langle P \rangle \mid Q] \longrightarrow s[P \mid Q] \end{array} \right)$$

Agent-agent interaction:

$$\text{SA} \quad a[\text{in } b.\overline{\text{open}} a \mid P] \mid b[\overline{\text{in}} b.\text{open } a.M \mid Q] \longrightarrow^* b[M \mid P \mid Q]$$

$$\text{Wagon} \quad \text{put}\langle P \rangle \mid \text{get}.M\langle Q \rangle \longrightarrow M\langle P \mid Q \rangle$$

The Wagon Calculus

Syntax:

$$\begin{aligned} M, N &::= \text{dis} \mid \text{put} \mid \text{get}.M \mid \uparrow.M \mid a.M \\ P, Q &::= \mathbf{0} \mid (\nu a)P \mid (P \mid Q) \mid a[P] \mid M\langle P \rangle \mid !M\langle P \rangle \end{aligned}$$

Slide 6

Reduction rules:

$$\begin{aligned} a.M\langle P \rangle \mid a[Q] &\longrightarrow a[M\langle P \rangle \mid Q] \\ a[\uparrow.M\langle P \rangle \mid Q] &\longrightarrow M\langle P \rangle \mid a[Q] \\ \text{put}\langle P \rangle \mid \text{get}.M\langle Q \rangle &\longrightarrow M\langle P \mid Q \rangle \end{aligned}$$

Reduction context: $P \mid -, (\nu a)-, a[-]$, but not $M\langle - \rangle$

Dissolving (dis is often dropped): $\langle P \rangle \equiv P$

Example - Routing

Slide 7

$Marseille[toLisbon[! \text{get. } \uparrow. \uparrow.Lisbon\langle \rangle] | toLisbon.put\langle P \rangle] | Lisbon[Q]$
→ $Marseille[toLisbon[! \text{get. } \uparrow. \uparrow.Lisbon\langle \rangle | \text{put}\langle P \rangle]] | Lisbon[Q]$
→ $Marseille[toLisbon[! \text{get. } \uparrow. \uparrow.Lisbon\langle \rangle | \uparrow.Lisbon\langle P \rangle]] | Lisbon[Q]$
→ $Marseille[toLisbon[! \text{get. } \uparrow. \uparrow.Lisbon\langle \rangle] | \uparrow.Lisbon\langle P \rangle] | Lisbon[Q]$
→ $Marseille[toLisbon[! \text{get. } \uparrow. \uparrow.Lisbon\langle \rangle]] | Lisbon\langle P \rangle | Lisbon[Q]$
→ $Marseille[toLisbon[! \text{get. } \uparrow. \uparrow.Lisbon\langle \rangle]] | Lisbon[P | Q]$

Slide 8

Part II. The Encoding

State-of-the-Art of Name-Passing in Ambients

Name-passing in MA/SA *with* local communication
[CG98, LS00].

Slide 9

Name-passing in SA *without* local communication
[Zimmer00] – very tricky, and complex!

- channel servers (queues for I/O processes)
- I/O processes interacting inside channel servers
- creation of explicit substitution servers (after each communication)

Formal proof of [Zimmer00]:

- Encoding may follow source — yes!
- Encoding always follows source — conjecture.

Encoding Name-Passing – Overview

$\pi_a: a(b).p \mid a\langle c \rangle \longrightarrow p\{b := c\}$

\Downarrow

π_a with explicit substitution $[b := c]$ and channel $a[\dots]$

Slide 10

\Downarrow

Wagon implementation:

$[b := c] \Longrightarrow b[!get. \uparrow.c.put\langle \rangle]$

$a[\dots] \Longrightarrow a[!get\langle \rangle \mid \dots]$

$a\langle c \rangle, a(b).p \Longrightarrow a.put\langle \dots \rangle$

π_a with Explicit Channel and Substitution [Zimmer00]

Obtained from π_a by providing every free name a with an empty explicit channel $a[\]$, and breaking down $a(b).p \mid a\langle c \rangle \longrightarrow p\{b := c\}$ into the following 5 low-level rules:

Slide 11

$$\left(\begin{array}{ll} a(b).p \mid a[\dots] \longrightarrow a[(b).p \mid \dots] & \text{routing-1} \\ a\langle c \rangle \mid a[\dots] \longrightarrow a[\langle c \rangle \mid \dots] & \text{routing-2} \\ a[(b).p \mid \langle c \rangle \mid \dots] \longrightarrow a[\dots] \mid (\nu b)([b := c] \mid p) & \text{communication} \\ [b := c] \mid b\langle d \rangle.p \longrightarrow [b := c] \mid c\langle d \rangle.p & \text{routing-3} \\ [b := c] \mid b\langle d \rangle \longrightarrow [b := c] \mid c\langle d \rangle & \text{routing-4} \end{array} \right)$$

The Encoding $\langle\langle p \rangle\rangle$

Slide 12

$$\begin{array}{ll} \llbracket \mathbf{0} \rrbracket \triangleq \mathbf{0} & \mathbf{chn} \triangleq !\text{get}\langle \rangle \mid \text{comm}[\mathbf{0}] \\ \llbracket (\nu a)p \rrbracket \triangleq (\nu a)(a[\mathbf{chn}] \mid \llbracket p \rrbracket) & \mathbf{fwd } b \triangleq !\text{get}. \uparrow . b.\text{put}\langle \rangle \\ \llbracket p \mid q \rrbracket \triangleq \llbracket p \rrbracket \mid \llbracket q \rrbracket & \langle\langle p \rangle\rangle_x \triangleq \llbracket p \rrbracket \mid \prod_{a \in X} a[\mathbf{chn}] \\ \llbracket a\langle b \rangle \rrbracket \triangleq a.\text{put}\langle \text{comm}.\text{put}\langle \mathbf{fwd } b \rangle \rangle & \langle\langle p \rangle\rangle \triangleq \langle\langle p \rangle\rangle_{fn(p)} \\ \llbracket a(b).p \rrbracket \triangleq (\nu b)(b[\mathbf{0}] & \\ \quad \mid a.\text{put}\langle \text{comm}.\text{get}. \uparrow . \uparrow . b\langle \uparrow \langle \llbracket p \rrbracket \rangle \rangle \rangle) & \end{array}$$

Routing Into Explicit Channel

$$\begin{array}{ll}
 \llbracket 0 \rrbracket \triangleq \mathbf{0} & \mathbf{chn} \triangleq !\mathbf{get}\langle \rangle \mid \mathit{comm}[0] \\
 \llbracket (\nu a)p \rrbracket \triangleq (\nu a)(a[\mathbf{chn}] \mid \llbracket p \rrbracket) & \mathbf{fwd} \ b \triangleq !\mathbf{get}. \uparrow . b.\mathbf{put}\langle \rangle \\
 \llbracket p \mid q \rrbracket \triangleq \llbracket p \rrbracket \mid \llbracket q \rrbracket & \langle\langle p \rangle\rangle_X \triangleq \llbracket p \rrbracket \mid \prod_{a \in X} a[\mathbf{chn}] \\
 \llbracket a\langle b \rangle \rrbracket \triangleq a.\mathbf{put}\langle \mathit{comm}.\mathbf{put}\langle \mathbf{fwd} \ b \rangle \rangle & \langle\langle p \rangle\rangle \triangleq \langle\langle p \rangle\rangle_{fn(p)} \\
 \llbracket a(b).p \rrbracket \triangleq (\nu b)(b[0] & \\
 & \mid a.\mathbf{put}\langle \mathit{comm}.\mathbf{get}. \uparrow . \uparrow . b\langle \uparrow \langle \llbracket p \rrbracket \rangle \rangle \rangle)
 \end{array}$$

Slide 13

$$\begin{array}{l}
 a\langle c \rangle \mid a[\dots] \longrightarrow a[\langle c \rangle \mid \dots] \quad \mathbf{routing-2} \\
 \downarrow \\
 a.\mathbf{put}\langle \mathit{comm}.\mathbf{put}\langle \mathbf{fwd} \ c \rangle \rangle \mid a[!\mathbf{get}\langle \rangle \mid \dots] \longrightarrow^* a[!\mathbf{get}\langle \rangle \mid \mathit{comm}.\mathbf{put}\langle \mathbf{fwd} \ c \rangle \mid \dots]
 \end{array}$$

Routing With Explicit Substitution

$$\begin{array}{ll}
 \llbracket 0 \rrbracket \triangleq \mathbf{0} & \mathbf{chn} \triangleq !\mathbf{get}\langle \rangle \mid \mathit{comm}[0] \\
 \llbracket (\nu a)p \rrbracket \triangleq (\nu a)(a[\mathbf{chn}] \mid \llbracket p \rrbracket) & \mathbf{fwd} \ b \triangleq !\mathbf{get}. \uparrow . b.\mathbf{put}\langle \rangle \\
 \llbracket p \mid q \rrbracket \triangleq \llbracket p \rrbracket \mid \llbracket q \rrbracket & \langle\langle p \rangle\rangle_X \triangleq \llbracket p \rrbracket \mid \prod_{a \in X} a[\mathbf{chn}] \\
 \llbracket a\langle b \rangle \rrbracket \triangleq a.\mathbf{put}\langle \mathit{comm}.\mathbf{put}\langle \mathbf{fwd} \ b \rangle \rangle & \langle\langle p \rangle\rangle \triangleq \langle\langle p \rangle\rangle_{fn(p)} \\
 \llbracket a(b).p \rrbracket \triangleq (\nu b)(b[0] & \\
 & \mid a.\mathbf{put}\langle \mathit{comm}.\mathbf{get}. \uparrow . \uparrow . b\langle \uparrow \langle \llbracket p \rrbracket \rangle \rangle \rangle)
 \end{array}$$

Slide 14

$$\begin{array}{l}
 b[\mathbf{fwd} \ c] \mid b\langle d \rangle \longrightarrow b[\mathbf{fwd} \ c] \mid c\langle d \rangle \quad \mathbf{routing-4} \\
 \downarrow \\
 b[!\mathbf{get}. \uparrow . c.\mathbf{put}\langle \rangle] \mid b.\mathbf{put}\langle \dots \rangle \\
 \longrightarrow b[!\mathbf{get}. \uparrow . c.\mathbf{put}\langle \rangle \mid \mathbf{put}\langle \dots \rangle] \\
 \longrightarrow b[!\mathbf{get}. \uparrow . c.\mathbf{put}\langle \rangle \mid \uparrow . c.\mathbf{put}\langle \dots \rangle] \\
 \longrightarrow b[!\mathbf{get}. \uparrow . c.\mathbf{put}\langle \rangle] \mid c.\mathbf{put}\langle \dots \rangle
 \end{array}$$

Communication

$$\begin{array}{ll}
 \llbracket \mathbf{0} \rrbracket \triangleq \mathbf{0} & \mathbf{chn} \triangleq !\mathbf{get}\langle \rangle \mid \mathit{comm}[\mathbf{0}] \\
 \llbracket (\nu a)p \rrbracket \triangleq (\nu a)(a[\mathbf{chn}] \mid \llbracket p \rrbracket) & \mathbf{fwd} b \triangleq !\mathbf{get}.\uparrow.b.\mathbf{put}\langle \rangle \\
 \llbracket p \mid q \rrbracket \triangleq \llbracket p \rrbracket \mid \llbracket q \rrbracket & \langle\langle p \rangle\rangle_X \triangleq \llbracket p \rrbracket \mid \prod_{a \in X} a[\mathbf{chn}] \\
 \llbracket a\langle b \rangle \rrbracket \triangleq a.\mathbf{put}\langle \mathit{comm}.\mathbf{put}\langle \mathbf{fwd} b \rangle \rangle & \langle\langle p \rangle\rangle \triangleq \langle\langle p \rangle\rangle_{fn(p)} \\
 \llbracket a\langle b \rangle.p \rrbracket \triangleq (\nu b)(b[\mathbf{0}] & \\
 & \mid a.\mathbf{put}\langle \mathit{comm}.\mathbf{get}.\uparrow.\uparrow.b\langle \uparrow \llbracket p \rrbracket \rangle \rangle)
 \end{array}$$

Slide 15

$$\begin{aligned}
 a\langle b \rangle.p \mid \langle c \rangle \mid \dots &\longrightarrow a[\dots] \mid (\nu b)(b[\mathbf{fwd} c] \mid p) && \mathbf{communication} \\
 \downarrow & \\
 (\nu b)(b[\] \mid a[\mathit{comm}.\mathbf{get}.\uparrow.\uparrow.b\langle \uparrow \llbracket p \rrbracket \rangle \rangle \mid \mathit{comm}.\mathbf{put}\langle \mathbf{fwd} c \rangle \mid \mathit{comm}[\dots] \mid \dots]) & \\
 \longrightarrow^2 (\nu b)(b[\] \mid a[\mathit{comm}[\mathbf{get}.\uparrow.\uparrow.b\langle \uparrow \llbracket p \rrbracket \rangle \rangle \mid \mathbf{put}\langle \mathbf{fwd} c \rangle \mid \dots \mid \dots]) & \\
 \longrightarrow (\nu b)(b[\] \mid a[\mathit{comm}[\uparrow.\uparrow.b\langle \uparrow \llbracket p \rrbracket \rangle \mid \mathbf{fwd} c \rangle \mid \dots \mid \dots]) & \\
 \longrightarrow^3 (\nu b)(b[\uparrow \llbracket p \rrbracket \rangle \mid \mathbf{fwd} c \rangle \mid a[\mathit{comm}[\dots] \mid \dots]) & \\
 \longrightarrow a[\mathit{comm}[\dots] \mid \dots] \mid (\nu b)(b[\mathbf{fwd} c \rangle \mid \llbracket p \rrbracket) &
 \end{aligned}$$

Slide 16

Part III. The Proof

Overview

Goal: Finding the right equivalence \simeq s.t.

- **Follows:** $b[\mathbf{fwd} \ c] \mid b.\mathbf{put}\langle \dots \rangle \simeq b[\mathbf{fwd} \ c] \mid c.\mathbf{put}\langle \dots \rangle$
- **Always follows:** A stronger requirement: all auxiliary reduction implies \simeq .

Slide 17

Solution:

- Find the general shape of the encoding using logic formulas, then find its reduction-closed closure.
- Enumerating all possible interactions using the closure.

Chemical Representation

Motivation: Tighter name scope

Method: Breaking down $a[P \mid Q \mid \dots]$ (vertical structure) to $P@a \mid Q@a \mid \dots$ (horizontal structure) using the following heating/cooling rules:

$$a[A \mid B] \rightleftharpoons a[A] \mid B@a \quad a[0] \rightleftharpoons a$$

Slide 18

Syntax: $A ::= \mathbf{0} \mid (\nu a)A \mid (A \mid B) \mid a[A] \mid M\langle A \rangle \mid !M\langle A \rangle \mid a \mid A@a$

After fully heated, no $a[\dots]$ will appear in a solution. It is fully horizontal.

Each component is of the form $M\langle \dots \rangle@s$ or $a@s$.

Reduction of (fully heated) solutions:

$$\begin{aligned} a.M\langle A \rangle@s \mid a@s &\leftrightarrow M\langle A \rangle@a@s \mid a@s \\ \uparrow.M\langle A \rangle@a@s &\leftrightarrow M\langle A \rangle@s \\ \mathbf{get}.M\langle A \rangle@s \mid \mathbf{put}\langle B \rangle@s &\leftrightarrow M\langle A \mid B \rangle@s \end{aligned}$$

Spatial Logic

A spatial logic formula \mathbb{A} is a set of (\Rightarrow -closed) solutions.

$$\begin{aligned}
 u, v &::= x \mid X \mid u \cup v \\
 \mathbb{M}, \mathbb{N} &::= \mathbf{dis} \mid \mathbf{put} \mid \mathbf{get}.\mathbb{M} \mid \uparrow.\mathbb{M} \mid u.\mathbb{M} \\
 \mathbb{A}, \mathbb{B} &::= \mathbf{0} \mid (\nu a)\mathbb{A} \mid (\mathbb{A} \mid \mathbb{B}) \mid a[\mathbb{A}] \mid \mathbb{M}\langle\mathbb{A}\rangle \mid !\mathbb{M}\langle\mathbb{A}\rangle \mid a \mid \mathbb{A}@u \\
 &\quad \mid \mathbb{A}^* \mid \prod_{a \in u} \mathbb{A} \mid (\nu x)\mathbb{A} \mid \mathbb{A}(\vec{u})
 \end{aligned}$$

Slide 19

$$\begin{aligned}
 \mathbf{0} &\triangleq \{0\} \\
 \mathbf{put}\langle\mathbb{A}\rangle &\triangleq \{\mathbf{put}\langle A \rangle \mid A \in \mathbb{A}\} \\
 u.\mathbb{M}\langle\mathbb{A}\rangle &\triangleq \{a.M\langle A \rangle \mid a \in u \wedge A \in \mathbb{A}\} \\
 \mathbb{A}@u &\triangleq \{A@a \mid a \in u \wedge A \in \mathbb{A}\} \\
 \mathbb{A} \mid \mathbb{B} &\triangleq \{C \mid \exists A \in \mathbb{A}.\exists B \in \mathbb{B}.C \Rightarrow A \mid B\} \\
 (\nu a)\mathbb{A} &\triangleq \{(\nu a)A \mid A \in \mathbb{A}\}
 \end{aligned}$$

$$\begin{aligned}
 (\nu x)\mathbb{A} &\triangleq \{(\nu X)A \mid A \in \mathbb{A}\{x := X\}\} \\
 \mathbb{A}^* &\triangleq \{C \mid \exists k \geq 0.\exists A_1 \in \mathbb{A}.\dots.\exists A_k \in \mathbb{A}.C \Rightarrow A_1 \mid \dots \mid A_k\} \\
 &\dots
 \end{aligned}$$

Slide 20

A **closure** is a reduction-closed formula.

Finding the Closure for the Encoding

Start point:

$$X.\text{put}\langle \dots \rangle^* \mid ((\nu b)(b[] \mid X.\text{put}\langle \dots \rangle))^* \mid \prod_{a \in X} a[\text{comm}[]]$$

↓

Slide 21

$$(\nu Y)(XY.\text{put}\langle \dots \rangle^* \mid ((\nu b)(b[] \mid XY.\text{put}\langle \dots \rangle))^* \mid \prod_{a \in XY} a[\text{comm}[]])$$

↓

For all p with $fn(p) = X$

$\mathbb{O}_1^*, \mathbb{I}_1^*, \mathbb{C}$

$$\langle\langle p \rangle\rangle \in (\nu y)(Xy.\text{put}\langle \dots \rangle^* \mid ((\nu b)(b[] \mid Xy.\text{put}\langle \dots \rangle))^* \mid \prod_{a \in Xy} a[\text{comm}[]])$$

$$\triangleq (\nu y)(\mathbb{O}_1^* \mid \mathbb{I}_1^* \mid \mathbb{C})$$

Finding the Reduction Closure

$\mathbb{O}_1/\mathbb{I}_1$ -solutions move into channels:

$\mathbb{O}_2^*, \mathbb{I}_2^*$

$$\dots \mid (\text{put}\langle \dots \rangle @ Xy)^* \mid ((\nu b)(b[] \mid \text{put}\langle \dots \rangle @ Xy))^*$$

$\mathbb{O}_2/\mathbb{I}_2$ -solutions communicate with $!\text{get}\langle \rangle$, and dissolve:

$\mathbb{O}_3^*, \mathbb{I}_3^*$

$$\dots \mid (\text{comm}.\text{put}\langle \dots \rangle @ Xy)^* \mid ((\nu b)(b[] \mid \text{comm}.\text{get}.\uparrow.\uparrow.b\langle \dots \rangle @ Xy))^*$$

Slide 22

$\mathbb{O}_3/\mathbb{I}_3$ -solutions move into comm :

$\mathbb{O}_4^*, \mathbb{I}_4^*$

$$\dots \mid (\text{put}\langle \dots \rangle @ \text{comm} @ Xy)^* \mid ((\nu b)(b[] \mid \text{get}.\uparrow.\uparrow.b\langle \dots \rangle @ \text{comm} @ Xy))^*$$

$\mathbb{O}_4/\mathbb{I}_4$ -solutions interact inside comm :

\mathbb{I}_5^*

$$\dots \mid ((\nu b)(b[] \mid \uparrow.\uparrow.b\langle \dots \rangle @ \text{comm} @ Xy))^*$$

\mathbb{I}_5 -solutions leave comm :

\mathbb{I}_6^*

$$\dots \mid ((\nu b)(b[] \mid \uparrow.b\langle \dots \rangle @ Xy))^*$$

\mathbb{I}_6 -solutions leave channel

\mathbb{I}_7^*

$$\dots \mid ((\nu b)(b[] \mid b\langle \dots \rangle))^*$$

\mathbb{I}_7 -solutions move into b :

\mathbb{I}_8^*

$$\dots | ((\nu b)(b[] | \uparrow \langle \dots \rangle @b | \mathbf{fwd} \ Xy@b))^*$$

The collection of explicit substitutions generated by \mathbb{I}_8 -solutions:

\mathbb{V}

$$(\nu z)(\dots | \prod_{a \in z} a[! \mathbf{get}. \uparrow .Xyz.\mathbf{put} \langle \rangle])$$

Slide 23

$\mathbb{O}_1/\mathbb{I}_1$ -solutions route inside explicit substitutions:

$\mathbb{O}_0^* | \mathbb{I}_0^*$

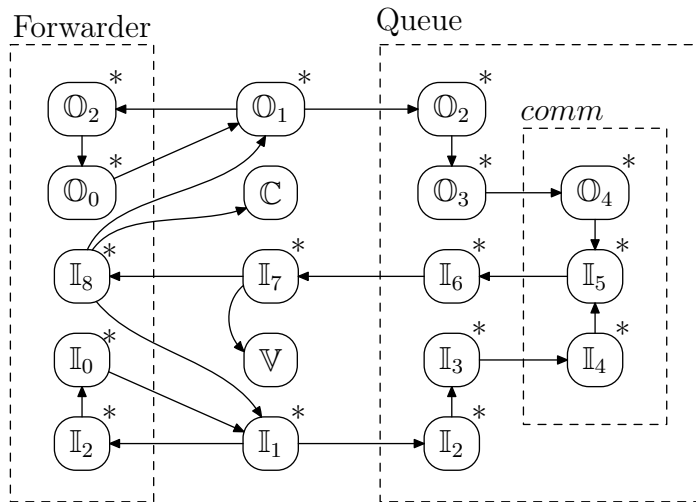
$$\dots | (\uparrow .Xyz.\mathbf{put} \langle \dots \rangle @z)^* | ((\nu b)(b[] | \uparrow .Xyz.\mathbf{put} \langle \dots \rangle @z))^*$$

The final closure \mathbf{CPI}_c are (roughly):

$$(\nu y)(\nu z)(\mathbf{C} | \mathbb{V} | \mathbb{O}_0^* | \dots | \mathbb{O}_4^* | \mathbb{I}_0^* | \dots | \mathbb{I}_8^*)$$

The Closure: \mathbf{CPI}_c

Slide 24



Results

Definition: May-testing congruence under \mathbb{CPI}_c : $\mathbb{CPI}_c \vdash A \simeq B$

For any observer \mathcal{C} , we require $\mathcal{C}(A) \in \mathbb{CPI}_c$ and $\mathcal{C}(B) \in \mathbb{CPI}_c$, i.e. the observer should behave like (the encoding of) a π -process.

Thus, possible interactions between \mathcal{C} and A, B are limited by the shape of \mathbb{CPI}_c .

Slide 25

Lemma: All auxiliary reductions of the encoding are may-testing congruent under \mathbb{CPI}_c (while the others are one-to-one correspond to reductions in the original process).

Theorem: The encoding follows, and always follows any reduction of the original process, up to the above equivalence.