If you are in a hurry

SPOILER WARNING: Plot and/or ending details follow.

$$\frac{-}{\emptyset \vdash \mathbf{0}}\ (\textsc{Nil}) \qquad \frac{A \vdash_\pi P \quad A \leq A'}{A' \vdash_\pi P \quad A' \vdash_{\mathrm{p}} P}\ (\textsc{Weak}) \qquad \frac{A \vdash_\pi P}{!\,A \vdash_\pi !\,P}\ (\textsc{Rep})$$

$$\frac{i = 1,2 : A_i \vdash_\pi P_i}{A_1 \odot A_2 \vdash_\pi P_1 \mid P_2}\ (\textsc{Par}) \qquad \frac{A \vdash_\pi P}{(\boldsymbol{\nu} x)\, A \vdash_\pi (\boldsymbol{\nu} x)\, P}\ (\textsc{Res})$$

$$\frac{A \vdash_\pi P \qquad \forall l : \mathbf{md}(\Sigma_A(l)) \notin \{\downarrow_1, \uparrow_1, \downarrow_{\omega_0}\}}{p.(\boldsymbol{\nu}\tilde{x})\ (p : ((\tilde{\sigma})^{\mathrm{p}}, \rho) + \rho(\tilde{x} : \tilde{\sigma}) \odot A) \vdash_{\mathrm{p}} p(\tilde{x}).P}\ (\textsc{Inp}_{\mathrm{p}})$$

$$\frac{A \vdash_\pi P}{(\boldsymbol{\nu}\tilde{x})\ \left(l : ((\tilde{\sigma})^{\downarrow_1}, \rho, \emptyset, (\tilde{x})) + l.\hat{l}.\rho(\tilde{x} : \tilde{\sigma}) \odot l.A\right) \vdash_\pi l(\tilde{x}).P}\ (\textsc{Inp}_1)$$

$$\frac{A \vdash P}{(\boldsymbol{\nu}\tilde{x})\ \left(u : ((\tilde{\sigma})^{\downarrow_{\omega_0}}, \rho, \emptyset, (\tilde{x})) + u.\rho(\tilde{x} : \tilde{\sigma}) \odot u.A\right) \vdash u(\tilde{x}).P}\ (\textsc{Inp}_\omega)$$

$$\frac{A \vdash_\pi P \quad \uparrow_\omega \notin \mathbf{md}(\tilde{\sigma}) \quad \forall l : \mathbf{md}(\Sigma_A(l)) \notin \{\downarrow_1, \uparrow_1, \downarrow_{\omega_0}\}}{p.\ (p : ((\bar{\tilde{\sigma}})^{\mathrm{p}}, \rho) + \bar{\rho}(\tilde{x} : \bar{\tilde{\sigma}}) \odot A) \vdash_{\mathrm{p}} \bar{p}\langle\tilde{x}\rangle.P}\ (\textsc{Out}_{\mathrm{p}})$$

$$\frac{A \vdash_\pi P}{l : ((\bar{\tilde{\sigma}})^{\uparrow_1}, \rho, \emptyset, \emptyset, (\tilde{x})) + l.\check{l}.\bar{\rho}(\tilde{x} : \bar{\tilde{\sigma}}) \odot l.A \vdash_\pi \bar{l}\langle\tilde{x}\rangle.P}\ (\textsc{Out}_1)$$

$$\frac{A \vdash_\pi P}{u : ((\tilde{\sigma})^{\uparrow_\omega}, \rho) + u.\check{u}.\bar{\rho}(\tilde{x} : \bar{\tilde{\sigma}}) \odot u.A \vdash_\pi \bar{u}\langle\tilde{x}\rangle.P}\ (\textsc{Out}_\omega)$$

Deciding Deterministic Responsiveness and Closeness in π-calculus

Maxime Gamboni

Insituto Superior Técnico

June 27, 2006

Teach Yourself Polyadic π-Calculus in 4 Minutes (I)

- Model for Communication & Concurrency
- Based around Named Channels

Teach Yourself Polyadic π-Calculus in 4 Minutes (I)

- Model for Communication & Concurrency
- Based around Named Channels

Two kinds of things are done in π.

- Sending something (ξ) over a channel (a): $\overline{a}\langle\xi\rangle.P$
- Receiving something on a channel (a), and referring to it as x afterwards: $a(x).P$

Teach Yourself Polyadic π-Calculus in 4 Minutes (II)

- Some other constructs: $P_1|P_2$, $(\boldsymbol{\nu} x)\, P$, $!\, P$, $\mathbf{0}$

E.g. $\bar{a}\langle s\rangle \mid a(x).\bar{x} \;\rightarrow \mathbf{0} \mid \bar{s}$

Teach Yourself Polyadic π-Calculus in 4 Minutes (II)

- Some other constructs: $P_1 | P_2$, $(\boldsymbol{\nu} x)\, P$, $!\, P$, $\mathbf{0}$

E.g. $\bar{a}\langle s \rangle \mid a(x).\bar{x} \;\rightarrow \mathbf{0} \mid \bar{s}$

- *POLY*-adic: More than one name can be moved around at a time

E.g. $\bar{a}\langle x, y, z \rangle.P$

Encodings

- Higher level languages can be *encoded* into π:

$$[\![\, \overline{a}\langle \xi \rangle \,]\!] \stackrel{\text{def}}{=} \overline{a}\langle u \rangle . \underbrace{!\, u(\tilde{r}). \cdots}_{\text{server for } \xi}$$

Encodings

- Higher level languages can be *encoded* into π:

$$[\![\, \overline{a}\langle \xi \rangle \,]\!] \stackrel{\text{def}}{=} \overline{a}\langle u \rangle . \underbrace{!\, u(\tilde{r}). \cdots}_{\text{server for } \xi}$$

- We want *Full Abstraction*:

$$(P \approx Q) \iff ([\![\, P \,]\!] \approx_{\mathsf{R}} [\![\, Q \,]\!])$$

$\approx_R$ is not a Regular Bisimulation

- These two (high level) processes are *bisimilar*

$$P = a(\mathfrak{b}).\mathrm{if}(\mathfrak{b})\big(\mathrm{if}(\neg\mathfrak{b})\ \mathrm{print}\ OOPS;\ \mathrm{else\ print}\ OK;\big)$$
$$Q = a(\mathfrak{b}).\mathrm{print}\ OK;$$

$\approx_R$ is not a Regular Bisimulation

- These two (high level) processes are *bisimilar*

$$P = a(\mathfrak{b}).\text{if}(\mathfrak{b})\big(\text{if}(\neg\mathfrak{b})\ \text{print}\ OOPS;\ \text{else print}\ OK;\big)$$
$$Q = a(\mathfrak{b}).\text{print}\ OK;$$

- Yet their encoded forms are not.
- We also need to enforce:

$\approx_R$ is not a Regular Bisimulation

- These two (high level) processes are *bisimilar*

$$P = a(\mathfrak{b}).\mathrm{if}(\mathfrak{b})\big(\mathrm{if}(\neg\mathfrak{b})\ \mathrm{print}\ \mathit{OOPS};\ \mathrm{else\ print}\ \mathit{OK};\big)$$
$$Q = a(\mathfrak{b}).\mathrm{print}\ \mathit{OK};$$

- Yet their encoded forms are not.
- We also need to enforce:

Determinism,

$\approx_R$ is not a Regular Bisimulation

- These two (high level) processes are *bisimilar*

$$P = a(\mathfrak{b}).\text{if}(\mathfrak{b})\big(\text{if}(\neg\mathfrak{b})\ \text{print } OOPS;\ \text{else print } OK;\big)$$
$$Q = a(\mathfrak{b}).\text{print } OK;$$

- Yet their encoded forms are not.
- We also need to enforce:

Determinism, *Closeness*,

$\approx_R$ is not a Regular Bisimulation

- These two (high level) processes are *bisimilar*

$$P = a(\mathfrak{b}).\mathrm{if}(\mathfrak{b})\big(\mathrm{if}(\neg\mathfrak{b})\ \mathrm{print}\ OOPS;\ \mathrm{else\ print}\ OK;\big)$$
$$Q = a(\mathfrak{b}).\mathrm{print}\ OK;$$

- Yet their encoded forms are not.
- We also need to enforce:

Determinism, *Closeness*, *Responsiveness*

$\approx_R$ is not a Regular Bisimulation

- These two (high level) processes are *bisimilar*

$$P = a(\mathfrak{b}).\mathrm{if}(\mathfrak{b})\big(\mathrm{if}(\neg\mathfrak{b})\ \mathrm{print}\ \mathit{OOPS};\ \mathrm{else\ print}\ \mathit{OK};\big)$$
$$Q = a(\mathfrak{b}).\mathrm{print}\ \mathit{OK};$$

- Yet their encoded forms are not.
- We also need to enforce:

Determinism, *Closeness*, *Responsiveness* and *Uniformity*.

Name Classes

Names in an encoded process (and its environment) are separated in three groups.

- For encoded data:

ω-names

Name Classes

Names in an encoded process (and its environment) are separated in three groups.

- For encoded data:

ω-names

- For responsiveness:

linear names

Name Classes

Names in an encoded process (and its environment) are separated in three groups.

- For encoded data:

ω-names

- For responsiveness:

linear names

- For the rest:

plain names

Templates and Observability

Two constructs are needed for defining bisimilarity:

Definition

Template Processes $\mathsf{L}_\sigma(a)$: Models ω-servers in the environment.

$$\mathsf{L}_{((\mathrm{p})^{\uparrow 1})^{\downarrow\omega}}(a) = \,!\, a(x).\overline{x}\langle a_1\rangle$$

Templates and Observability

Two constructs are needed for defining bisimilarity:

Definition

Template Processes $\mathsf{L}_{\sigma}(a)$: Models ω-servers in the environment.

$$\mathsf{L}_{((\mathrm{p})^{\uparrow 1})^{\downarrow\omega}}(a) = \,!\, a(x).\overline{x}\langle a_1\rangle$$

Definition

Observable Data $\Omega_P^{\Sigma}(a)$: Tests ω-servers in the process.

If $P = \,!\, a(x).\overline{x}\langle z\rangle$ then $\Omega_P^{\Sigma}(a) = \langle z\rangle$

Discreet Bisimulation

Symmetric $\mathcal{R}$ is a *discreet* bisimulation if $P \mathcal{R} Q$ implies:

Discreet Bisimulation

Symmetric $\mathcal{R}$ is a *discreet* bisimulation if $P\mathcal{R}Q$ implies:

1. If $P \xrightarrow{\mu} P'$ where μ is silent or on a plain/linear channel:
   - $Q \stackrel{\hat{\mu}}{\Longrightarrow} Q'$ and $P'\mathcal{R}Q'$.

Discreet Bisimulation

Symmetric $\mathcal{R}$ is a *discreet* bisimulation if $P \mathcal{R} Q$ implies:

1. If $P \xrightarrow{\mu} P'$ where μ is silent or on a plain/linear channel:
   - $Q \stackrel{\hat{\mu}}{\Longrightarrow} Q'$ and $P' \mathcal{R} Q'$.
2. $\forall u$ ω-input in P (say $P \xrightarrow{u(\tilde{x})} P'$)
   - $\Omega^{\Sigma}_{P}(u) = \Omega^{\Sigma}_{Q}(u)$,

Discreet Bisimulation

Symmetric $\mathcal{R}$ is a *discreet* bisimulation if $P\mathcal{R}Q$ implies:

1. If $P \xrightarrow{\mu} P'$ where μ is silent or on a plain/linear channel:
   - $Q \stackrel{\hat{\mu}}{\Longrightarrow} Q'$ and $P'\mathcal{R}Q'$.
2. $\forall u$ ω-input in P (say $P \xrightarrow{u(\tilde{x})} P'$)
   - $\Omega^{\Sigma}_{P}(u) = \Omega^{\Sigma}_{Q}(u)$,
   - Safety: $P'\mathcal{R}P'$,

Discreet Bisimulation

Symmetric $\mathcal{R}$ is a *discreet* bisimulation if $P \mathcal{R} Q$ implies:

1. If $P \xrightarrow{\mu} P'$ where μ is silent or on a plain/linear channel:
   - $Q \stackrel{\hat{\mu}}{\Longrightarrow} Q'$ and $P' \mathcal{R} Q'$.
2. $\forall u$ ω-input in P (say $P \xrightarrow{u(\tilde{x})} P'$)
   - $\Omega^{\Sigma}_{P}(u) = \Omega^{\Sigma}_{Q}(u)$,
   - Safety: $P' \mathcal{R} P'$,
   - Determinism: $\exists! \xi$ s.t. $\Omega^{\Sigma}_{P}(u) = \xi$,

Discreet Bisimulation

Symmetric $\mathcal{R}$ is a *discreet* bisimulation if $P\mathcal{R}Q$ implies:

1. If $P \xrightarrow{\mu} P'$ where μ is silent or on a plain/linear channel:
   - $Q \stackrel{\hat{\mu}}{\Longrightarrow} Q'$ and $P'\mathcal{R}Q'$.
2. $\forall u$ ω-input in P (say $P \xrightarrow{u(\tilde{x})} P'$)
   - $\Omega_P^\Sigma(u) = \Omega_Q^\Sigma(u)$,
   - Safety: $P'\mathcal{R}P'$,
   - Determinism: $\exists! \xi$ s.t. $\Omega_P^\Sigma(u) = \xi$,
   - Closeness: $P\mathcal{R}(\boldsymbol{\nu}\tilde{x})\, P'$.

Discreet Bisimulation

Symmetric $\mathcal{R}$ is a *discreet* bisimulation if $P \mathcal{R} Q$ implies:

1. If $P \xrightarrow{\mu} P'$ where μ is silent or on a plain/linear channel:
   - $Q \stackrel{\hat{\mu}}{\Longrightarrow} Q'$ and $P' \mathcal{R} Q'$.
2. $\forall u$ ω-input in P (say $P \xrightarrow{u(\tilde{x})} P'$)
   - $\Omega^{\Sigma}_{P}(u) = \Omega^{\Sigma}_{Q}(u)$,
   - Safety: $P' \mathcal{R} P'$,
   - Determinism: $\exists! \xi$ s.t. $\Omega^{\Sigma}_{P}(u) = \xi$,
   - Closeness: $P \mathcal{R} (\boldsymbol{\nu} \tilde{x})\, P'$.
3. $\forall u$ ω-output in P:
   - $(\mathsf{L}_{\sigma}(u) \mid P) \mathcal{R} Q$.

Channel Types

Definition

A Channel Type is a structure of the form:

$$a : ((\tilde{\sigma})^m, \rho, \tilde{\alpha}, \tilde{\beta}, \tilde{\gamma})$$

- $\tilde{\sigma}$: Parameters
- m: Action Mode
- ρ: Protocol
- $\tilde{\alpha}$: Receptiveness
- $\tilde{\beta}$: Input Responsiveness
- $\tilde{\gamma}$: Output Responsiveness

Inter-Class Interactions

Highly constrained ω and unreliable plain names can interact.

Inter-Class Interactions

Highly constrained ω and unreliable plain names can interact.

- ω over p:

$$(\boldsymbol{\nu} p)\left(\overline{p}\langle u\rangle.!\,u\cdots \mid \overline{p}\langle v\rangle.!\,v\cdots \mid p(x).\cdots \mid p(y).\cdots\right)$$

Inter-Class Interactions

Highly constrained ω and unreliable plain names can interact.

- ω over p:

$$(\boldsymbol{\nu} p)\ \big(\overline{p}\langle u\rangle.!\,u\cdots \mid \overline{p}\langle v\rangle.!\,v\cdots \mid p(x).\cdots \mid p(y).\cdots\big)$$

- p over ω:

$$\overline{u}\langle l, p, q\rangle \mid !\,u(x,y,z).\overline{x}\langle y\rangle$$

Inter-Class Interactions

Highly constrained ω and unreliable plain names can interact.

- ω over p:

$$(\boldsymbol{\nu} p)\left(\overline{p}\langle u\rangle.!\, u \cdots \mid \overline{p}\langle v\rangle.!\, v \cdots \mid p(x). \cdots \mid p(y). \cdots\right)$$

- p over ω:

$$\overline{u}\langle l, p, q\rangle \mid !\, u(x, y, z).\overline{x}\langle y\rangle$$

- Still, ω's discreetness guarantees are preserved.

Anatomy of one Rule

$$\frac{A \vdash_\pi P}{(\boldsymbol{\nu}\tilde{x})\ \left(l : (\downarrow_1)((\tilde{\sigma})^{\downarrow_1}, \rho, \emptyset, (\tilde{x})) + l.\hat{l}.\rho(\tilde{x} : \tilde{\sigma}) \odot l.A\right) \vdash_\pi l(\tilde{x}).P}\ (\textsc{Inp}_1)$$

Anatomy of one Rule

$$\frac{A \vdash_{\pi} P}{(\boldsymbol{\nu}\tilde{x})\ \left(l : (\downarrow_1)((\tilde{\sigma})^{\downarrow_1}, \rho, \emptyset, (\tilde{x})) + l.\hat{l}.\rho(\tilde{x} : \tilde{\sigma}) \odot l.A\right) \vdash_{\pi} l(\tilde{x}).P}\ (\text{INP}_1)$$

- l receptive now; responsive when parameters are ready

Anatomy of one Rule

$$\frac{A \vdash_\pi P}{(\boldsymbol{\nu}\tilde{x})\ \left(l : (\downarrow_1)((\tilde{\sigma})^{\downarrow_1}, \rho, \emptyset, (\tilde{x})) + l.\hat{l}.\rho(\tilde{x} : \tilde{\sigma}) \odot l.A\right) \vdash_\pi l(\tilde{x}).P}\ (\textsc{Inp}_1)$$

- l receptive now; responsive when parameters are ready
- Remote parameters

Anatomy of one Rule

$$\frac{A \vdash_\pi P}{(\boldsymbol{\nu}\tilde{x})\ \left(l : (\downarrow_1)((\tilde{\sigma})^{\downarrow_1}, \rho, \emptyset, (\tilde{x})) + l.\hat{l}.\rho(\tilde{x} : \tilde{\sigma}) \odot l.A \right) \vdash_\pi l(\tilde{x}).P} \ (\textsc{Inp}_1)$$

- l receptive now; responsive when parameters are ready
- Remote parameters
- Continuation

Anatomy of one Rule

$$\dfrac{A \vdash_\pi P}{(\boldsymbol{\nu}\tilde{x})\left(l : (\downarrow_1)((\tilde{\sigma})^{\downarrow_1}, \rho, \emptyset, (\tilde{x})) + l.\hat{l}.\rho(\tilde{x} : \tilde{\sigma}) \odot l.A\right) \vdash_\pi l(\tilde{x}).P} \ (\textsc{Inp}_1)$$

- l receptive now; responsive when parameters are ready
- Remote parameters
- Continuation
- P must provide resources specified in $\tilde{\sigma}$

(Expected) Results

Discreetness:

Theorem

$$(A \vdash_\pi P) \Rightarrow (P \approx_{\mathrm{R}} P)$$

(Expected) Results

Discreetness:

Theorem

$$(A \vdash_\pi P) \Rightarrow (P \approx_{\mathrm{R}} P)$$

Soundness:

Theorem

$$(A \vdash_\pi P) \wedge \Sigma_A(a) = (\cdots)^{\uparrow_1} \Rightarrow (P \xrightarrow{(\boldsymbol{\nu}\tilde{z})\,\overline{a}\langle\tilde{x}\rangle})$$

$$(A \vdash_\pi P) \wedge \Sigma_A(a) = (\cdots)^{\downarrow_1} \Rightarrow (P \xrightarrow{a(\tilde{x})})$$

(Expected) Results

Discreetness:

Theorem

$$(A \vdash_\pi P) \Rightarrow (P \approx_\mathrm{R} P)$$

Soundness:

Theorem

$$(A \vdash_\pi P) \wedge \Sigma_A(a) = (\cdots)^{\uparrow_1} \Rightarrow (P \xRightarrow{(\nu \tilde{z})\, \overline{a}\langle \tilde{x} \rangle})$$

$$(A \vdash_\pi P) \wedge \Sigma_A(a) = (\cdots)^{\downarrow_1} \Rightarrow (P \xRightarrow{a(\tilde{x})})$$

Safety:

Theorem

$$(A \vdash_\pi P) \wedge (P \to P') \Rightarrow (A \vdash_\pi P')$$

Thank You (Obrigado, Shukria, Kiitos, Merci)!

The paper can be found at `http://gamboni.org/maxime/`