Observing Success in the π-Calculus

David Sabel and **Manfred Schmidt-Schauß**

Goethe-University, Frankfurt, Germany

WPTE'15

Warsaw, Poland

2 July, 2015

Motivation

π-**calculus** ([Milner et al.'92, Milner'99])

- model for **concurrent processes with message passing**
- **several** process equivalences (see [Sangiorgi & Walker'01]), mainly bisimulations which **observe input and output capabilities**
- bisimulations are usually **very fine-grained**

Motivation

π-calculus ([Milner et al.'92, Milner'99])

- model for **concurrent processes with message passing**
- **several** process equivalences (see [Sangiorgi & Walker'01]), mainly bisimulations which **observe input and output capabilities**
- bisimulations are usually **very fine-grained**

(Concurrent) λ-calculi

- canonical, **very coarse-grained**
  Morris' style contextual equivalence
- based on **observing success**ful (may- and should-) termination
- **no notion** of channels and input / output

Motivation

π-**calculus** ([Milner et al.'92, Milner'99])

- model for **concurrent processes with message passing**
- **several** process equivalences (see [Sangiorgi & Walker'01]), mainly bisimulations which **observe input and output capabilities**
- bisimulations are usually **very fine-grained**

(Concurrent) λ**-calculi**

- canonical, **very coarse-grained**
  Morris' style contextual equivalence
- based on **observing success**ful (may- and should-) termination
- **no notion** of channels and input / output

→ hard to **compare both worlds**

(e.g. required for expressivity results)

Our Contributions

- Morris' style contextual equivalence $\sim_c$ **for the π-calculus**
    - ➽ requires a **notion of success**
    - ➽ **extend** the π-calculus by a **constant** Stop to denote success (similar as [Gorla'10, Peters et al.'14])

Our Contributions

- Morris' style contextual equivalence $\sim_c$ **for the π-calculus**
  - ➽ requires a **notion of success**
  - ➽ **extend** the π-calculus by a **constant** Stop to denote success (similar as [Gorla'10, Peters et al.'14])
- **tools and techniques** to prove contextual equivalences
  - ➽ a context lemma
  - ➽ a sound similarity

Our Contributions

- Morris' style contextual equivalence $\sim_c$ **for the π-calculus**
  - ➽ requires a **notion of success**
  - ➽ **extend** the π-calculus by a **constant** Stop to denote success (similar as [Gorla'10, Peters et al.'14])
- **tools and techniques** to prove contextual equivalences
  - ➽ a context lemma
  - ➽ a sound similarity
- compare $\sim_c$ with process equivalences in the π-calculus

Syntax of the Π_{Stop}-Calculus

Processes:	P	$::=$	$\pi.P$	(action)
		$\mid$	$P_1 \mid P_2$	(parallel composition)
		$\mid$	$!\,P$	(replication)
		$\mid$	$\mathbf{0}$	(silent process)
		$\mid$	$\nu x.P$	(name restriction)
		$\mid$	Stop	**(constant denoting success)**
Action prefixes:	π	$::=$	$x(y)$	input
		$\mid$	$\overline{x}\langle y\rangle$	output

where x, y are names

Contexts: $C ::= [\cdot] \mid \pi.C \mid C \mid P \mid P \mid C \mid !\,C \mid \nu x.C$

"processes with a single hole"

Operational Semantics

Reduction rule for **interaction**:

$$\underbrace{\boldsymbol{x}(y).P}_{\text{receiver}} \mid \underbrace{\overline{\boldsymbol{x}}\langle \boldsymbol{v} \rangle.Q}_{\text{sender}} \quad \xrightarrow{ia} \quad P[\boldsymbol{v}/y] \mid Q$$

"message $\boldsymbol{v}$ is sent along the channel $\boldsymbol{x}$"

Operational Semantics

Reduction rule for **interaction**:

$$\underbrace{x(y).P}_{\text{receiver}} \mid \underbrace{\overline{x}\langle v\rangle.Q}_{\text{sender}} \quad \xrightarrow{ia} \quad P[v/y] \mid Q$$

"message v is sent along the channel x"

Reduction contexts: $\mathbf{D} \in \mathcal{D} ::= [\cdot] \mid \mathbf{D} \mid P \mid P \mid \mathbf{D} \mid \nu x.\mathbf{D}$

Structural congruence $\equiv$:

$$\begin{array}{ccc} P \equiv Q, \text{ if } P =_\alpha Q & P \mid \mathbf{0} \equiv P & \nu x.\mathsf{Stop} \equiv \mathsf{Stop} \\ P_1 \mid (P_2 \mid P_3) \equiv (P_1 \mid P_2) \mid P_3 & P \mid Q \equiv Q \mid P & \nu z.\nu w.P \equiv \nu w.\nu z.P \\ \nu z.(P_1 \mid P_2) \equiv P_1 \mid \nu z.P_2, \text{ if } z \notin \mathsf{fn}(P_1) & \nu z.\mathbf{0} \equiv \mathbf{0} & !P \equiv P \mid !P \end{array}$$

Standard reduction

$$\frac{P \equiv \mathbf{D}[P'] \quad \wedge \quad P' \xrightarrow{ia} Q' \quad \wedge \quad \mathbf{D}[Q'] \equiv Q}{P \xrightarrow{sr} Q}$$

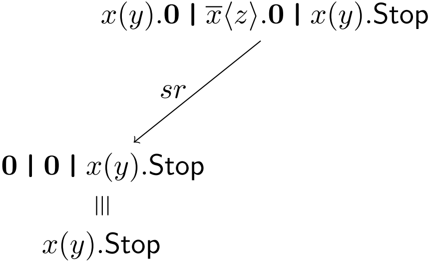
Example

$$x(y).\mathbf{0} \mid \overline{x}\langle z\rangle.\mathbf{0} \mid x(y).\mathsf{Stop}$$

Example

$$\begin{tikzcd}
 & x(y).\mathbf{0} \mid \overline{x}\langle z\rangle.\mathbf{0} \mid x(y).\mathsf{Stop} \arrow[dl, "sr"'] \\
\mathbf{0} \mid \mathbf{0} \mid x(y).\mathsf{Stop} &
\end{tikzcd}$$

Example



$$x(y).\mathbf{0} \mid \overline{x}\langle z\rangle.\mathbf{0} \mid x(y).\mathsf{Stop}$$

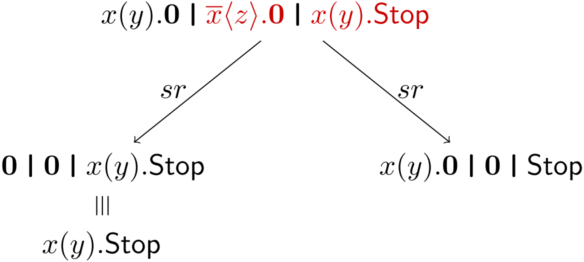
$$\xrightarrow{sr}$$

$$\mathbf{0} \mid \mathbf{0} \mid x(y).\mathsf{Stop}$$

$$\equiv$$

$$x(y).\mathsf{Stop}$$

Example



$x(y).\mathbf{0} \mid \overline{x}\langle z\rangle.\mathbf{0} \mid x(y).\mathsf{Stop}$

$\mathbf{0} \mid \mathbf{0} \mid x(y).\mathsf{Stop}$

$x(y).\mathbf{0} \mid \mathbf{0} \mid \mathsf{Stop}$

$\equiv$

$x(y).\mathsf{Stop}$

Example

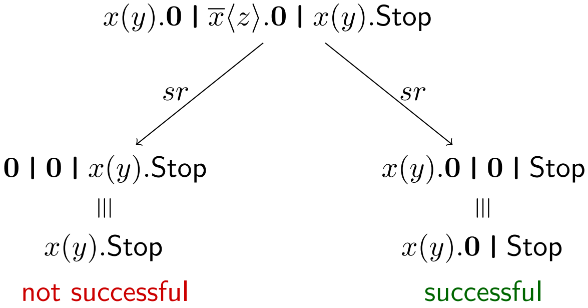
$$\begin{tikzcd}
 & x(y).\mathbf{0} \mid \overline{x}\langle z\rangle.\mathbf{0} \mid x(y).\mathsf{Stop} \arrow[dl, "sr"'] \arrow[dr, "sr"] & \\
\mathbf{0} \mid \mathbf{0} \mid x(y).\mathsf{Stop} \arrow[d, phantom, sloped, "\equiv"] & & x(y).\mathbf{0} \mid \mathbf{0} \mid \mathsf{Stop} \arrow[d, phantom, sloped, "\equiv"] \\
x(y).\mathsf{Stop} & & x(y).\mathbf{0} \mid \mathsf{Stop}
\end{tikzcd}$$

Example

$$\begin{tikzcd}
 & x(y).\mathbf{0} \mid \overline{x}\langle z\rangle.\mathbf{0} \mid x(y).\mathsf{Stop} \arrow[dl, "sr"] \arrow[dr, "sr"] & \\
\mathbf{0} \mid \mathbf{0} \mid x(y).\mathsf{Stop} \arrow[d, phantom, sloped, "\equiv"] & & x(y).\mathbf{0} \mid \mathbf{0} \mid \mathsf{Stop} \arrow[d, phantom, sloped, "\equiv"] \\
x(y).\mathsf{Stop} & & x(y).\mathbf{0} \mid \mathsf{Stop}
\end{tikzcd}$$

- **Process P is successful** (stop(P)) iff $P \equiv \mathsf{Stop} \mid P'$

Example



$x(y).\mathbf{0} \mid \overline{x}\langle z\rangle.\mathbf{0} \mid x(y).\mathsf{Stop}$

$\xrightarrow{sr}$ $\mathbf{0} \mid \mathbf{0} \mid x(y).\mathsf{Stop}$ $\equiv$ $x(y).\mathsf{Stop}$

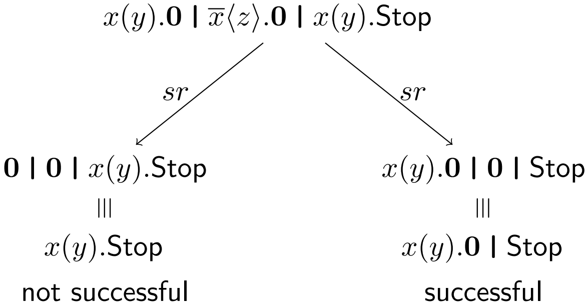
not successful

$\xrightarrow{sr}$ $x(y).\mathbf{0} \mid \mathbf{0} \mid \mathsf{Stop}$ $\equiv$ $x(y).\mathbf{0} \mid \mathsf{Stop}$

successful

- **Process** P **is successful** ($\mathsf{stop}(P)$) iff $P \equiv \mathsf{Stop} \mid P'$

Example



$x(y).\mathbf{0} \mid \overline{x}\langle z\rangle.\mathbf{0} \mid x(y).\mathsf{Stop}$

sr sr

$\mathbf{0} \mid \mathbf{0} \mid x(y).\mathsf{Stop}$ $\equiv$ $x(y).\mathsf{Stop}$

not successful

$x(y).\mathbf{0} \mid \mathbf{0} \mid \mathsf{Stop}$ $\equiv$ $x(y).\mathbf{0} \mid \mathsf{Stop}$

successful

- **Process** P **is successful** ($\mathsf{stop}(P)$) iff $P \equiv \mathsf{Stop} \mid P'$
- **May-convergence** $P{\downarrow}$ iff $\exists P' : P \xrightarrow{sr,*} P' \wedge \mathsf{stop}(P')$
- **Should-convergence** $P{\Downarrow}$ iff $\forall P' : P \xrightarrow{sr,*} P' \implies P'{\downarrow}$.
- may-divergence $P{\uparrow}$ iff $\neg P{\Downarrow}$
- must-divergence $P{\Uparrow}$ iff $\neg P{\downarrow}$

Example

$$\begin{tikzcd}
 & x(y).\mathbf{0} \mid \overline{x}\langle z\rangle.\mathbf{0} \mid x(y).\mathsf{Stop}\ \downarrow \text{ and } \uparrow \arrow[dl, "sr"'] \arrow[dr, "sr"] & \\
\mathbf{0} \mid \mathbf{0} \mid x(y).\mathsf{Stop} \arrow[d, phantom, sloped, "\equiv"] & & x(y).\mathbf{0} \mid \mathbf{0} \mid \mathsf{Stop} \arrow[d, phantom, sloped, "\equiv"] \\
x(y).\mathsf{Stop}\Uparrow \arrow[d, phantom, ""] & & x(y).\mathbf{0} \mid \mathsf{Stop}\Downarrow \arrow[d, phantom, ""] \\
\text{not successful} & & \text{successful}
\end{tikzcd}$$

- **Process** P **is successful** (stop(P)) iff $P \equiv \mathsf{Stop} \mid P'$
- **May-convergence** $P{\downarrow}$ iff $\exists P' : P \xrightarrow{sr,*} P' \land \mathsf{stop}(P')$
- **Should-convergence** $P{\Downarrow}$ iff $\forall P' : P \xrightarrow{sr,*} P' \implies P'{\downarrow}$.
- may-divergence $P{\uparrow}$ iff $\neg P{\Downarrow}$
- must-divergence $P{\Uparrow}$ iff $\neg P{\downarrow}$

Contextual Equivalence

Contextual Equivalence

$P \sim_c Q$ iff $\forall C \in \mathcal{C} : C[P]\downarrow \iff C[Q]\downarrow$ and $C[P]\Downarrow \iff C[Q]\Downarrow$

Contextual Equivalence

Contextual Equivalence

$P \sim_c Q$ iff $\forall C \in \mathcal{C} : C[P]\downarrow \iff C[Q]\downarrow$ and $C[P]\Downarrow \iff C[Q]\Downarrow$

Details

Contextual may-preorder	$P \leq_{c,\downarrow} Q$ iff $\forall C : C[P]\downarrow \implies C[Q]\downarrow$
Contextual should-preorder	$P \leq_{c,\Downarrow} Q$ iff $\forall C : C[P]\Downarrow \implies C[Q]\Downarrow$
Contextual preorder	$\leq_c \quad := \quad \leq_{c,\downarrow} \cap \leq_{c,\Downarrow}$
Contextual equivalence	$\sim_c \quad := \quad \leq_c \cap \geq_c$

Context Lemma

Context Lemma

If for all name substitutions σ and processes R:

- $(\sigma(P) \mid R)\downarrow \implies (\sigma(Q) \mid R)\downarrow$ and
- $(\sigma(P) \mid R)\Downarrow \implies (\sigma(Q) \mid R)\Downarrow$

then $P \leq_c Q$.

"it suffices to consider contexts $\sigma([\cdot]) \mid R$"

“Applicative” May-Similarity

Full applicative $\downarrow$-similarity $P \precsim^{\sigma}_{b,\downarrow} Q$ iff $\forall\sigma : \sigma(P) \precsim_{b,\downarrow} \sigma(Q)$
where $\precsim_{b,\downarrow}$ is the greatest fixpoint of $\boldsymbol{F_{b,\downarrow}}$ and
$\boldsymbol{F_{b,\downarrow}}$ is the operator binary relations η on processes, s.t. $P\ \boldsymbol{F_{b,\downarrow}}(\eta)\ Q$ iff

1. If P is successful, then $Q{\downarrow}$.
2. If $P \xrightarrow{sr} P'$, then $\exists Q'$ with $Q \xrightarrow{sr,*} Q'$ and $P'\ \eta\ Q'$.
3. If P is not successful, then:

"Applicative" May-Similarity

Full applicative ↓-similarity $P \precsim^{\sigma}_{b,\downarrow} Q$ iff $\forall \sigma : \sigma(P) \precsim_{b,\downarrow} \sigma(Q)$
where $\precsim_{b,\downarrow}$ is the greatest fixpoint of $\boldsymbol{F_{b,\downarrow}}$ and
$\boldsymbol{F_{b,\downarrow}}$ is the operator binary relations η on processes, s.t. $P\ \boldsymbol{F_{b,\downarrow}}(\eta)\ Q$ iff

1. If P is successful, then $Q\downarrow$.
2. If $P \xrightarrow{sr} P'$, then $\exists Q'$ with $Q \xrightarrow{sr,*} Q'$ and $P'\ \eta\ Q'$.
3. If P is not successful, then:
    - **Open input:** If $P \equiv D[x(y).P_1]$, then $(\forall z \in \mathcal{N} : \exists Q' : Q \xrightarrow{sr,*} Q' \equiv D'[x(y).Q_1] \wedge D[P_1[z/y]]\ \eta\ D'[Q_1[z/y]])$ where x is free in P and Q'

“Applicative” May-Similarity

Full applicative ↓-similarity $P \precsim^{\sigma}_{b,\downarrow} Q$ iff $\forall \sigma : \sigma(P) \precsim_{b,\downarrow} \sigma(Q)$
where $\precsim_{b,\downarrow}$ is the greatest fixpoint of $\boldsymbol{F_{b,\downarrow}}$ and
$\boldsymbol{F_{b,\downarrow}}$ is the operator binary relations η on processes, s.t. $P\ \boldsymbol{F_{b,\downarrow}}(\eta)\ Q$ iff

1. If P is successful, then $Q{\downarrow}$.
2. If $P \xrightarrow{sr} P'$, then $\exists Q'$ with $Q \xrightarrow{sr,*} Q'$ and $P'\ \eta\ Q'$.
3. If P is not successful, then:
    - **Open input:** If $P \equiv D[x(y).P_1]$, then
      $(\forall z \in \mathcal{N} : \exists Q' : Q \xrightarrow{sr,*} Q' \equiv D'[x(y).Q_1] \wedge D[P_1[z/y]]\ \eta\ D'[Q_1[z/y]])$
      where x is free in P and Q'
    - **Open output:** If $P \equiv D[\overline{x}\langle y\rangle.P_1]$ then
      $(Q \xrightarrow{sr,*} Q' \equiv D'[\overline{x}\langle y\rangle.Q_1] \wedge D[P_1]\ \eta\ D[Q_1])$
      where x, y are free in P and Q'

"Applicative" May-Similarity

Full applicative $\downarrow$-similarity $P \precsim^{\sigma}_{b,\downarrow} Q$ iff $\forall \sigma : \sigma(P) \precsim_{b,\downarrow} \sigma(Q)$
where $\precsim_{b,\downarrow}$ is the greatest fixpoint of $\boldsymbol{F_{b,\downarrow}}$ and
$\boldsymbol{F_{b,\downarrow}}$ is the operator binary relations η on processes, s.t. $P\ \boldsymbol{F_{b,\downarrow}}(\eta)\ Q$ iff

1. If P is successful, then $Q{\downarrow}$.
2. If $P \xrightarrow{sr} P'$, then $\exists Q'$ with $Q \xrightarrow{sr,*} Q'$ and $P'\ \eta\ Q'$.
3. If P is not successful, then:
   - **Open input:** If $P \equiv D[x(y).P_1]$, then
     $(\forall z \in \mathcal{N} : \exists Q' : Q \xrightarrow{sr,*} Q' \equiv D'[x(y).Q_1] \wedge D[P_1[z/y]]\ \eta\ D'[Q_1[z/y]])$
     where x is free in P and Q'
   - **Open output:** If $P \equiv D[\overline{x}\langle y\rangle.P_1]$ then
     $(Q \xrightarrow{sr,*} Q' \equiv D'[\overline{x}\langle y\rangle.Q_1] \wedge D[P_1]\ \eta\ D[Q_1])$
     where x, y are free in P and Q'
   - **Bound output:** If $P \equiv \nu y.D[\overline{x}\langle y\rangle.P_1]$ then
     $(Q \xrightarrow{sr,*} Q' \equiv \nu y.D'[\overline{x}\langle y\rangle.Q_1] \wedge D[P_1]\ \eta\ D[Q_1])$
     where x is free in P and Q'

“Applicative” Should-Similarity

Full applicative ↑-similarity $P \precsim^{\sigma}_{b,\uparrow} Q$ iff $\forall \sigma : \sigma(P) \precsim_{b,\downarrow} \sigma(Q)$
where $\precsim_{b,\uparrow}$ is the greatest fixpoint of $\boldsymbol{F_{b,\uparrow}}$ and
$\boldsymbol{F_{b,\uparrow}}$ is the operator binary relations η on processes, s.t. $P \ \boldsymbol{F_{b,\uparrow}}(\eta) \ Q$ iff

1. If ~~P is successful, then $Q{\downarrow}$~~.
2. If $P \xrightarrow{sr} P'$, then $\exists Q'$ with $Q \xrightarrow{sr,*} Q'$ and $P' \ \eta \ Q'$.
3. If ~~P is not successful~~, then:
   - **Open input:** If $P \equiv D[x(y).P_1]$, then
     $(\forall z \in \mathcal{N} : \exists Q' : Q \xrightarrow{sr,*} Q' \equiv D'[x(y).Q_1] \wedge D[P_1[z/y]] \ \eta \ D'[Q_1[z/y]])$
     where x is free in P and Q'
   - **Open output:** If $P \equiv D[\overline{x}\langle y\rangle.P_1]$ then
     $(Q \xrightarrow{sr,*} Q' \equiv D'[\overline{x}\langle y\rangle.Q_1] \wedge D[P_1] \ \eta \ D[Q_1])$
     where x, y are free in P and Q'
   - **Bound output:** If $P \equiv \nu y.D[\overline{x}\langle y\rangle.P_1]$ then
     $(Q \xrightarrow{sr,*} Q' \equiv \nu y.D'[\overline{x}\langle y\rangle.Q_1] \wedge D[P_1] \ \eta \ D[Q_1])$
     where x is free in P and Q'

“Applicative” Should-Similarity

Full applicative ↑-similarity $P \precsim^{\sigma}_{b,\uparrow} Q$ iff $\forall \sigma : \sigma(P) \precsim_{b,\downarrow} \sigma(Q)$
where $\precsim_{b,\uparrow}$ is the greatest fixpoint of $\boldsymbol{F_{b,\uparrow}}$ and
$\boldsymbol{F_{b,\uparrow}}$ is the operator binary relations η on processes, s.t. $P\ \boldsymbol{F_{b,\uparrow}}(\eta)\ Q$ iff

1. If $P{\Uparrow}$, then $Q{\uparrow}$.
2. If $P \xrightarrow{sr} P'$, then $\exists Q'$ with $Q \xrightarrow{sr,*} Q'$ and $P'\ \eta\ Q'$.
3. If ~~P is not successful~~, then:
    - **Open input:** If $P \equiv D[x(y).P_1]$, then
      $(\forall z \in \mathcal{N} : \exists Q' : Q \xrightarrow{sr,*} Q' \equiv D'[x(y).Q_1] \wedge D[P_1[z/y]]\ \eta\ D'[Q_1[z/y]])$
      where x is free in P and Q'
    - **Open output:** If $P \equiv D[\overline{x}\langle y\rangle.P_1]$ then
      $(Q \xrightarrow{sr,*} Q' \equiv D'[\overline{x}\langle y\rangle.Q_1] \wedge D[P_1]\ \eta\ D[Q_1])$
      where x, y are free in P and Q'
    - **Bound output:** If $P \equiv \nu y.D[\overline{x}\langle y\rangle.P_1]$ then
      $(Q \xrightarrow{sr,*} Q' \equiv \nu y.D'[\overline{x}\langle y\rangle.Q_1] \wedge D[P_1]\ \eta\ D[Q_1])$
      where x is free in P and Q'

“Applicative” Should-Similarity

Full applicative ↑-similarity $P \precsim^{\sigma}_{b,\uparrow} Q$ iff $\forall \sigma : \sigma(P) \precsim_{b,\downarrow} \sigma(Q)$
where $\precsim_{b,\uparrow}$ is the greatest fixpoint of $\boldsymbol{F_{b,\uparrow}}$ and
$\boldsymbol{F_{b,\uparrow}}$ is the operator binary relations η on processes, s.t. $P\ \boldsymbol{F_{b,\uparrow}}(\eta)\ Q$ iff

1. If $P\Uparrow$, then $Q\uparrow$.
2. If $P \xrightarrow{sr} P'$, then $\exists Q'$ with $Q \xrightarrow{sr,*} Q'$ and $P'\ \eta\ Q'$.
3. If $\neg P\Uparrow$, then:
   - **Open input:** If $P \equiv D[x(y).P_1]$, then
     $(\forall z \in \mathcal{N} : \exists Q' : Q \xrightarrow{sr,*} Q' \equiv D'[x(y).Q_1] \wedge D[P_1[z/y]]\ \eta\ D'[Q_1[z/y]])$
     where x is free in P and Q'
   - **Open output:** If $P \equiv D[\overline{x}\langle y\rangle.P_1]$ then
     $(Q \xrightarrow{sr,*} Q' \equiv D'[\overline{x}\langle y\rangle.Q_1] \wedge D[P_1]\ \eta\ D[Q_1])$
     where x, y are free in P and Q'
   - **Bound output:** If $P \equiv \nu y.D[\overline{x}\langle y\rangle.P_1]$ then
     $(Q \xrightarrow{sr,*} Q' \equiv \nu y.D'[\overline{x}\langle y\rangle.Q_1] \wedge D[P_1]\ \eta\ D[Q_1])$
     where x is free in P and Q'

"Applicative" Should-Similarity

Full applicative ↑-similarity $P \precsim^{\sigma}_{b,\uparrow} Q$ iff $\forall \sigma : \sigma(P) \precsim_{b,\downarrow} \sigma(Q)$
where $\precsim_{b,\uparrow}$ is the greatest fixpoint of $\boldsymbol{F_{b,\uparrow}}$ and
$\boldsymbol{F_{b,\uparrow}}$ is the operator binary relations η on processes, s.t. $P\ \boldsymbol{F_{b,\uparrow}}(\eta)\ Q$ iff

1. If $P{\Uparrow}$, then $Q{\uparrow}$.
2. If $P \xrightarrow{sr} P'$, then $\exists Q'$ with $Q \xrightarrow{sr,*} Q'$ and $P'\ \eta\ Q'$.
3. If $\neg P{\Uparrow}$, then:
   - **Open input:** If $P \equiv D[x(y).P_1]$, then
     $(\forall z \in \mathcal{N} : \exists Q' : Q \xrightarrow{sr,*} Q' \equiv D'[x(y).Q_1] \wedge D[P_1[z/y]]\ \eta\ D'[Q_1[z/y]])$
     where x is free in P and Q'
   - **Open output:** If $P \equiv D[\overline{x}\langle y\rangle.P_1]$ then
     $(Q \xrightarrow{sr,*} Q' \equiv D'[\overline{x}\langle y\rangle.Q_1] \wedge D[P_1]\ \eta\ D[Q_1])$
     where x, y are free in P and Q'
   - **Bound output:** If $P \equiv \nu y.D[\overline{x}\langle y\rangle.P_1]$ then
     $(Q \xrightarrow{sr,*} Q' \equiv \nu y.D'[\overline{x}\langle y\rangle.Q_1] \wedge D[P_1]\ \eta\ D[Q_1])$
     where x is free in P and Q'
4. $Q \precsim_{b,\downarrow} P$

Applicative Similarity: Soundness

Theorem (Soundness)

$$(P \precsim_{b,\downarrow}^{\sigma} Q \land Q \precsim_{b,\uparrow}^{\sigma} P) \quad \Longrightarrow \quad P \leq_c Q$$

Proof (outline):

- if $(P \precsim_{b,\downarrow} Q)$ then $((P \mid R)\downarrow \Longrightarrow (Q \mid R)\downarrow)$
- if $(Q \precsim_{b,\uparrow} P)$ then $((P \mid R)\Downarrow \Longrightarrow (Q \mid R)\Downarrow)$

Applicative Similarity: Soundness

Theorem (Soundness)

$$(P \precsim^{\sigma}_{b,\downarrow} Q \wedge Q \precsim^{\sigma}_{b,\uparrow} P) \quad \Longrightarrow \quad P \leq_c Q$$

Proof (outline):

- if $(P \precsim^{\sigma}_{b,\downarrow} Q)$ then $((\sigma(P) \mid R)\downarrow \Longrightarrow (\sigma(Q) \mid R)\downarrow)$
- if $(Q \precsim^{\sigma}_{b,\uparrow} P)$ then $((\sigma(P) \mid R)\Downarrow \Longrightarrow (\sigma(Q) \mid R)\Downarrow)$

Applicative Similarity: Soundness

Theorem (Soundness)

$$(P \precsim^{\sigma}_{b,\downarrow} Q \wedge Q \precsim^{\sigma}_{b,\uparrow} P) \quad \Longrightarrow \quad P \leq_c Q$$

Proof (outline):

- if $(P \precsim^{\sigma}_{b,\downarrow} Q)$ then $((\sigma(P) \mid R) \downarrow \Longrightarrow (\sigma(Q) \mid R) \downarrow)$
- if $(Q \precsim^{\sigma}_{b,\uparrow} P)$ then $((\sigma(P) \mid R) \Downarrow \Longrightarrow (\sigma(Q) \mid R) \Downarrow)$
- apply the context lemma

Applicative Similarity: Soundness

Theorem (Soundness)

$$(P \precsim^{\sigma}_{b,\downarrow} Q \wedge Q \precsim^{\sigma}_{b,\uparrow} P) \quad \Longrightarrow \quad P \leq_c Q$$

Proof (outline):

- if $(P \precsim^{\sigma}_{b,\downarrow} Q)$ then $((\sigma(P) \mid R)\downarrow \Longrightarrow (\sigma(Q) \mid R)\downarrow)$
- if $(Q \precsim^{\sigma}_{b,\uparrow} P)$ then $((\sigma(P) \mid R)\Downarrow \Longrightarrow (\sigma(Q) \mid R)\Downarrow)$
- apply the context lemma

Remark: let $\oplus$ be an encoded choice-operator and

$$A := a(x).\mathbf{0} \qquad B := b(x).\mathbf{0} \qquad C := c(x).\mathbf{0}$$

then $(A \oplus B) \oplus C \sim_{c} A \oplus (B \oplus C)$ but $(A \oplus B) \oplus C \not\precsim^{\sigma}_{b,\uparrow} A \oplus (B \oplus C)$
(main reason: 4th condition)

Applicative Similarity: Soundness

Theorem (Soundness)

$$(P \precsim^{\sigma}_{b,\downarrow} Q \wedge Q \precsim^{\sigma}_{b,\uparrow} P) \quad \implies \quad P \leq_c Q$$

Proof (outline):

- if $(P \precsim^{\sigma}_{b,\downarrow} Q)$ then $((\sigma(P) \mid R)\downarrow \implies (\sigma(Q) \mid R)\downarrow)$
- if $(Q \precsim^{\sigma}_{b,\uparrow} P)$ then $((\sigma(P) \mid R)\Downarrow \implies (\sigma(Q) \mid R)\Downarrow)$
- apply the context lemma

Remark: let $\oplus$ be an encoded choice-operator and

$$A := a(x).\mathbf{0} \qquad B := b(x).\mathbf{0} \qquad C := c(x).\mathbf{0}$$

then $(A \oplus B) \oplus C \boldsymbol{\sim_c} A \oplus (B \oplus C)$ but $(A \oplus B) \oplus C \not\precsim^{\boldsymbol{\sigma}}_{\boldsymbol{b,\uparrow}} A \oplus (B \oplus C)$ (main reason: 4^{th} condition)

Open problem: find a **coarser** sound similarity for ↑

Tools at Work: A Correct Program Transformation

Correctness of Deterministic Interaction

For all processes P, Q the following equation holds:

$$\nu x.(x(y).P \mid \overline{x}\langle z\rangle.Q)) \sim_c \nu x.(P[z/y] \mid Q)$$

Proof (outline):

- $\mathcal{S} := \equiv \cup \{(\sigma(\nu x.(x(y).P \mid \overline{x}\langle z\rangle.Q)), \sigma(\nu x.(P[z/y] \mid Q))) \mid \text{for all } x, y, z, P, Q, \sigma\}$
- $\mathcal{S}$ and $\mathcal{S}^{-1}$ are $F_{b,\downarrow}$-dense and $F_{b,\uparrow}$-dense
  - $\mathcal{S} \subseteq F_{b,\downarrow}(\mathcal{S})$ and thus $\mathcal{S} \subseteq \precsim_{b,\downarrow}$
  - $\mathcal{S} \subseteq F_{b,\uparrow}(\mathcal{S})$ and thus $\mathcal{S} \subseteq \precsim_{b,\uparrow}$
  - $\mathcal{S}^{-1} \subseteq F_{b,\downarrow}(\mathcal{S}^{-1})$ and thus $\mathcal{S}^{-1} \subseteq \precsim_{b,\downarrow}$
  - $\mathcal{S}^{-1} \subseteq F_{b,\uparrow}(\mathcal{S}^{-1})$ and thus $\mathcal{S}^{-1} \subseteq \precsim_{b,\uparrow}$

Tools at Work: Some Laws

Theorem

For all processes P, Q the following equivalences hold:

1. $!\,P \sim_c \,!\,!\,P$.
2. $!\,P \mid !\,P \sim_c \,!\,P$.
3. $!\,(P \mid Q) \sim_c \,!\,P \mid !\,Q$.
4. $!\,\mathbf{0} \sim_c \mathbf{0}$.
5. $!\,\mathsf{Stop} \sim_c \mathsf{Stop}$.
6. $!\,(P \mid Q) \sim_c \,!\,(P \mid Q) \mid P$.
7. $x(y).\nu z.P \sim_c \nu z.x(y).P$ if $z \notin \{x, y\}$.
8. $\overline{x}\langle y\rangle.\nu z.P \sim_c \nu z.\overline{x}\langle y\rangle.P$ if $z \notin \{x, y\}$.

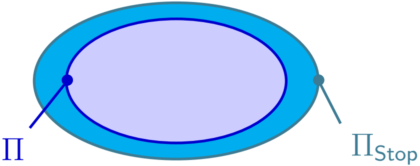
Analyzing the Contextual Ordering

Theorem

1. If P, Q are two successful processes, then $P \sim_c Q$.
2. If P, Q are two processes with $P{\downarrow}, Q{\downarrow}$, then $P \sim_{c,\downarrow} Q$.
3. There are may-convergent processes P, Q with $P \not\sim_c Q$.
4. Stop is the greatest process w.r.t. $\leq_c$.
5. $\mathbf{0}$ is the smallest process w.r.t. $\leq_{c,\downarrow}$.
6. There is no smallest process w.r.t. $\leq_c$.

The Stop-free Calculus Π

$\Pi =$ subcalculus of $\Pi_{\textsf{Stop}}$
without constant Stop
(in processes, contexts, ...)



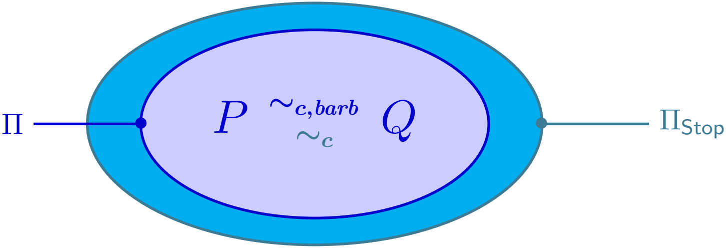
Barbed Testing in Π

- $P\Rsh^x = P$ **has a barb on input x**: $P \equiv D[x(y).P']$ (where x is free)
- May-testing: $P\downharpoonright_x$ iff $\exists P' : P \xrightarrow{sr,*} P' \wedge P\Rsh^x$
- Should-testing: $P \downdownharpoons_x$ iff $\forall P' : P \xrightarrow{sr,*} P' \implies P'\downharpoonright_x$
- **Barbed may- and should-testing equivalence**
  [Fournet & Gonthier'05]
  $P \sim_{c,barb} Q$ iff $\forall C : C[P]\downharpoonright_x \iff C[Q]\downharpoonright_x \wedge C[P] \downdownharpoons_x \iff C[Q] \downdownharpoons_x$

Conservativity

Theorem

For all Stop-free processes P, Q: $P \sim_{c,barb} Q \iff P \sim_c Q$.



Consequences:

- $\langle \Pi_{\mathsf{Stop}}, \sim_c \rangle$ conservatively extends $\langle \Pi, \sim_{c,barb} \rangle$
- the shown Stop-free equivalences also hold in $\langle \Pi, \sim_{c,barb} \rangle$

Conclusion & Future Work

Conclusion

- Morris' style contextual equivalence w.r.t **may- and should-convergence** in a π-calculus with Stop
- tools: context lemma, sound applicative similarities
- several proved process equivalences using the tools
- conservatively extends barbed may- and should-testing in the π-calculus, s.t. results can be transferred back

Conclusion & Future Work

Conclusion

- Morris' style contextual equivalence w.r.t **may- and should-convergence** in a π-calculus with Stop
- tools: context lemma, sound applicative similarities
- several proved process equivalences using the tools
- conservatively extends barbed may- and should-testing in the π-calculus, s.t. results can be transferred back

Future work

- extend the results to variants of the π-calculus (guarded sums, matching prefixes, . . .)
- analyze encodings between concurrent lambda-calculi and the π-calculus w.r.t. contextual equivalence
- find a better sound should-similarity