MPRI Concurrency (course number 2-3) 2004-2005: π-calculus

9 December 2004

http://pauillac.inria.fr/$\sim$leifer/teaching/mpri-concurrency-2004/

James J. Leifer
INRIA Rocquencourt

James.Leifer@inria.fr

Today's plan

- exercises from last week
- review: barbed bisimilarity
- two natural congruences
- a family portrait
- weak barbed congruence and weak labelled bisimilarity correspond

Weak barbed bisimulation

Recall that a process P has a strong barb x, written $P{\downarrow}x$ iff there exists P_0, P_1, and $\vec{y}$ such that $P \equiv \boldsymbol{\nu}\vec{y}.(\overline{x}u.P_0 \mid P_1)$ and $x \notin \vec{y}$.

A process P has a weak barb x, written $P{\Downarrow}x$ iff there exists P' such that $P \longrightarrow^* P'$ and $P'{\downarrow}x$.

A relation $\mathcal{R}$ is a weak barbed bisimulation if it is symmetric and for all $(P, Q) \in \mathcal{R}$

- if $P \longrightarrow P'$, there exists Q' such that $Q \longrightarrow^* Q'$ and $(P', Q') \in \mathcal{R}$;
- if $P{\downarrow}x$ then $Q{\Downarrow}x$.

Weak barbed bisimilarity, written $\dot{\approx}$, is the largest such relation.

Two possible equivalences (non-input congruences)

We write "equivalence" for "non input-prefixing congruence".
Clearly $\dot{\approx}$ isn't an equivalence: $\overline{x}y \;\dot{\approx}\; \overline{x}z$ but $- \mid x(u).\overline{u}w$ can distinguish them. There are two ways of building an equivalence:

- Close up at the end: weak barbed equivalence, $\dot{\approx}^{\circ}$, is the largest equivalence included in $\dot{\approx}$. Concretely, $P \;\dot{\approx}^{\circ}\; Q$ iff for all contexts $C \in \mathcal{E}$ we have $C[P] \;\dot{\approx}\; C[Q]$. Check!

- Close up at every step: weak barbed reduction equivalence, $\approx$, is the largest relation $\mathcal{R}$ such that $\mathcal{R}$ is a weak barbed bisimulation and an equivalence. Concretely, $\approx$ is the largest symmetric relation $\mathcal{R}$ such that for all $(P, Q) \in \mathcal{R}$,
  - if $P \longrightarrow P'$, there exists Q' such that $Q \longrightarrow^{*} Q'$ and $(P', Q') \in \mathcal{R}$;
  - if $P{\downarrow}x$ then $Q{\Downarrow}x$;
  - for all $C \in \mathcal{E}$, we have $(C[P], C[Q]) \in \mathcal{R}$.

  Check!

An extended family portrait

	strong	
	labelled	barbed
not an equivalence		“bisimilarity” $\dot{\sim}$
equivalence	“bisimilarity” $\sim_\ell$	“equivalence” $\dot{\sim}^\circ$ “reduction equivalence” $\sim$
congruence	“full bisimilarity” $\simeq_\ell$	“congruence” $\dot{\simeq}^\circ$ “reduction congruence” $\simeq$

	weak	
	labelled	barbed
not an equivalence		“bisimilarity” $\dot{\approx}$
equivalence	“bisimilarity” $\approx_\ell$	“equivalence” $\dot{\approx}^\circ$ “reduction equivalence” $\approx$
congruence	“full bisimilarity” $\cong_\ell$	“congruence” $\dot{\cong}^\circ$ “reduction congruence” $\cong$

A detailed family portrait

<table>
<tr><th></th><th>labelled</th><th colspan="2">barbed</th></tr>
<tr><td>not an equivalence</td><td></td><td colspan="2">$\dot{\approx}$: largest $\mathcal{R}$ st
$$\begin{tikzcd} P \arrow[r] \arrow[d, no head, "\mathcal{R}"'] & P' \arrow[d, dashed, no head, "\mathcal{R}"] \\ Q \arrow[r, dashed, "*"] & Q' \end{tikzcd}$$
$P{\downarrow}x$ implies $Q{\Downarrow}x$</td></tr>
<tr><td>equivalence</td><td>$\approx_{\ell}$: largest $\mathcal{R}$ st
$$\begin{tikzcd} P \arrow[r, "\alpha"] \arrow[d, no head, "\mathcal{R}"'] & P' \arrow[d, dashed, no head, "\mathcal{R}"] \\ Q \arrow[r, dashed, "\tau^*\hat{\alpha}\tau^*"] & Q' \end{tikzcd}$$</td><td>$\approx$: largest $\mathcal{R}$ st
$$\begin{tikzcd} P \arrow[r] \arrow[d, no head, "\mathcal{R}"'] & P' \arrow[d, dashed, no head, "\mathcal{R}"] \\ Q \arrow[r, dashed, "*"] & Q' \end{tikzcd}$$
$P{\downarrow}x$ implies $Q{\Downarrow}x$
$\forall D \in \mathcal{E}.(D[P], D[Q]) \in \mathcal{R}$</td><td>$\dot{\approx}^{\circ}$:
$\{(P, Q) \;/\; (\forall D \in \mathcal{E}.D[P] \dot{\approx} D[Q]\}$</td></tr>
</table>

What's the difference between $\approx$ and $\dot{\approx}^\circ$?

- $\approx \subseteq \dot{\approx}^\circ$: Yes, trivially.
- $\approx \supseteq \dot{\approx}^\circ$: Not necessarily.

  Two difficult results due to Cédric Fournet and Georges Gonthier. "A hierarchy of equivalences for asynchronous cacluli". ICALP 1998. Journal version:

  http://research.microsoft.com/~fournet/papers/a-hierarchy-of-equivalences-for-asynchronous-calculi.pdf

  – In general they're not the same. $\dot{\approx}^\circ$ is not even guaranteed to be a weak barbed bisimulation:

$$\begin{tikzcd} P \arrow[d, dash, "\dot{\approx}^\circ"'] & C[P] \arrow[r] \arrow[d, dash, "\dot{\approx}"'] & P' \arrow[d, dash, dashed, "\dot{\approx}"] \\ Q & C[Q] \arrow[r, dashed, "*"] & Q' \end{tikzcd}$$

  – But for π-calculus, they coincide.

Comparing labels and barbs

- $\approx_\ell \subseteq \approx$: Yes, easy.
- $\approx_\ell \supseteq \approx$: Yes, provided we have name matching. The result is subtle.

Name matching

Motivation: Which context can detect that $P \xrightarrow{\overline{x}y} P'$? It's easy to tell P can output on x; we just check $P{\downarrow}x$. If we want to test that this transition leads to P', we can take the context $C = - \mid x(u).k \mid \overline{k}$ for k fresh. Now

$$C[P] \longrightarrow\longrightarrow P'$$

where $P' \not\downarrow k$.

But how do we detect that the message is y? We could try

$$C = - \mid x(u).(\overline{u} \mid y.k) \mid \overline{k}$$

but this risks having the $\overline{u}$ and the y interact with the process in the hole.

Thus, we introduce a simple new construct called name matching:

$$\begin{aligned} P ::= & \ \ldots \\ & [x = y].P \end{aligned}$$

Reductions: $[x = x].P \longrightarrow P$

Labelled transitions: $[x = x].P \xrightarrow{\tau} P$

Barbed equivalence is a weak labelled bisimulation

Theorem: $\approx_\ell \supseteq \approx$.

Proof: Consider $P \approx Q$ and suppose $P \xrightarrow{\alpha} P'$. (For simplicity, ignore structural congruence.)

case $\alpha = \tau$**:** Then $P \longrightarrow P'$. By definition, there exists Q' such that $Q \longrightarrow^* Q'$ and $P' \approx Q'$. Thus $Q \xrightarrow{\tau}{}^* Q'$ as desired.

case $\alpha = xy$**:** Let $C = - \mid \overline{x}y.k \mid \overline{k}$, where k is fresh. Then $C[P] \longrightarrow\longrightarrow P'$. Therefore, there exists Q such that $C[Q] \longrightarrow^* Q'$ and $P' \approx Q'$. Since $P' \not\downarrow_k$, we have $Q' \not\downarrow_k$, therefore $Q \xrightarrow{\tau}{}^* \xrightarrow{xy} \xrightarrow{\tau}{}^* Q'$, as desired.

case $\alpha = \overline{x}y$**:** Let $C = - \mid x(u).[u = y].k \mid \overline{k}$, where k is fresh. Then $C[P] \longrightarrow\longrightarrow\longrightarrow P'$. Therefore, there exists Q such that $C[Q] \longrightarrow^* Q'$ and $P' \approx Q'$. Since $P' \not\Downarrow k$, we have $Q' \not\Downarrow k$, therefore $Q \xrightarrow{\tau}{}^{*} \xrightarrow{\overline{x}y} \xrightarrow{\tau}{}^{*} Q'$, as desired.

case $\alpha = \overline{x}(y)$ **and** $y \notin \mathsf{fn}(Q)$**:** Let

$$C = - \mid x(u).\Big(\overline{z}u \mid k \mid \prod_{w \in \mathsf{fn}(P)} [u = w].\overline{k}\Big) \mid \overline{k}$$

where k and z are fresh. Then $C[P] \longrightarrow\longrightarrow H_{z,y}[P']$ where

$$H_{z,y} = \boldsymbol{\nu} y.(\overline{z}y \mid -)$$

Therefore, there exists Q'' such that $C[Q] \longrightarrow^* Q''$ and $H_{z,y}[P'] \approx Q''$. Since $H_{z,y}[P'] \not\Downarrow k$, we have $Q'' \not\Downarrow k$. Thus there exists Q' such that $Q'' \equiv C'[Q']$ and $Q \xrightarrow{\tau}{}^{*} \xrightarrow{\overline{x}(y)} \xrightarrow{\tau}{}^{*} Q'$. Do we know $P' \approx Q'$?

Exercises for next lecture

1. Since the last lecture, the proof has been fixed by using $\not\Downarrow k$ everywhere. Prove from the definition of $\approx$ that for $P \approx Q$ if $P{\Downarrow}x$ then $Q{\Downarrow}x$, and thus the contrapositive: if $Q \not\Downarrow x$ then $P \not\Downarrow x$.

   Answer: Suppose $P \approx Q$. If $P{\Downarrow}x$ then there exists P' such that $P \longrightarrow^* P'$ and $P'{\downarrow}x$. Thus there exists Q' such that $Q \longrightarrow^* Q'$ and $P' \approx Q'$. Since $P'{\downarrow}x$, we have $Q'{\Downarrow}x$. Hence there exists Q'' such that $Q' \longrightarrow^* Q''$ and $Q''{\downarrow}x$. Thus $Q{\Downarrow}x$, as desired.

2. The last case of the proof relies on the following lemma: $H_{z,y}[P] \approx H_{z,y}[Q]$ implies $P \approx Q$, where $z \notin \mathsf{fn}(P) \cup \mathsf{fn}(Q)$. In the updated version of the proof you will find the definition $H_{z,y} = \boldsymbol{\nu} y.(\overline{z}y \mid -)$.

Hints...

In order to prove this, consider

$$\mathcal{R} = \{(P, Q) \ / \ z \notin \mathsf{fn}(P) \cup \mathsf{fn}(Q) \text{ and } H_{z,y}[P] \approx H_{z,y}[Q]\}.$$

Our goal (as usual) is to prove that $\mathcal{R}$ satisfies the same properties as $\approx$, and thus deduce that $\mathcal{R} \subseteq \approx$. Assume $(P, Q) \in \mathcal{R}$.

- $\mathcal{R}$ is a bisimulation: Show that $P \longrightarrow P'$ implies that there exists Q' such that $Q \longrightarrow^* Q'$ and $(P', Q') \in \mathcal{R}$.

  Answer: Suppose $P \longrightarrow P'$. Since $H_{z,y}$ is an evaluation context, $H_{z,y}[P] \longrightarrow H_{z,y}[P']$. Since $H_{z,y}[P] \approx H_{z,y}[Q]$, there exists Q'' such that $H_{z,y}[Q] \longrightarrow^* Q''$ and $H_{z,y}[P'] \approx Q''$. Since $z \notin \mathsf{fn}(Q)$, these reductions could not be due to any interaction between $H_{z,y}$ and Q, nor can $H_{z,y}$ itself reduce, thus there exists Q' such that $Q \longrightarrow^* Q'$ and $Q'' \equiv H_{z,y}[Q']$. Thus (ignoring structural congruence), $(P', Q') \in \mathcal{R}$, as desired.

- $\mathcal{R}$ preserves barbs: Show that $P{\downarrow}w$ implies $Q{\Downarrow}w$.

  Answer: Suppose $P{\downarrow}w$. Since $z \notin \mathsf{fn}(P)$, we know that $w \neq z$. We distinguish two cases.

case $w \neq y$: We have $P{\downarrow}w$ implies $H_{z,y}[P]{\downarrow}w$. Since $H_{z,y}[P] \approx H_{z,y}[Q]$, we have $H_{z,y}[Q]{\Downarrow}w$. Since there is no interaction between $H_{z,y}$ and Q and the former can't be responsible for the barb because $w \neq z$, we have $Q{\Downarrow}w$, as desired.

case $w = y$: We don't have $H_{z,y}[P]{\downarrow}w$ since the new binder in $H_{z,y}$ prevents the w from being seen. However, we can let $C = z(u).u(v).\overline{k} \mid -$ where u, k are fresh. By extrusion we can see $C[H_{z,y}[P]]{\Downarrow}k$. Since $H_{z,y}[P] \approx H_{z,y}[Q]$, we have $C[H_{z,y}[P]] \approx C[H_{z,y}[Q]]$, thus $C[H_{z,y}[Q]]{\Downarrow}k$. The only way this can be is if $Q{\Downarrow}w$, as desired.

- $\mathcal{R}$ is an equivalence: It is sufficient to show that $(C[P], C[Q]) \in \mathcal{R}$ where $C = \boldsymbol{\nu}\vec{w}.(- \mid S)$. Hint: try to find a context C' and name z' such that $H_{z',y}[C[P]] \approx C'[H_{z,y}[P]]$ and the same for Q (perhaps using a labelled bisimilarity since we know $\approx_\ell \subseteq \approx$). You may have to distinguish between the cases $y \in \vec{w}$ and $y \notin \vec{w}$.

  Answer: We first need the following lemma, which, holds for all kinds of equivalences.

  Lemma (injective renaming): Let ρ be an injective function on the universe of names. If $P \approx Q$ then $\rho P \approx \rho Q$.

  Note that this lemma is quite different from a typical "substitution" lemma since ρ is not making two names equal that weren't previously equal. Now returning to the main problem...

  By hypothesis, we know that there exists z, y such that $z \notin \mathsf{fn}(P) \cup \mathsf{fn}(Q)$ and $H_{z,y}[P] \approx H_{z,y}[Q]$. By injective renaming, we can assume without loss of generality that z is fresh. As a consequences $z \notin \mathsf{fn}(P) \cup \mathsf{fn}(S) \cup y, \vec{w}$. Let z' be fresh. We distinguish two cases:

case $y \notin \vec{w}$**:**

$$
\begin{aligned}
H_{z',y}[C[P]] &\equiv \boldsymbol{\nu} y.(\overline{z'}y \mid \boldsymbol{\nu}\vec{w}.(P \mid S)) \\
&\equiv \boldsymbol{\nu}\vec{w}.\boldsymbol{\nu} y.(\overline{z'}y \mid P \mid S) \quad \text{since } y, z' \notin \vec{w} \\
&\approx_\ell \boldsymbol{\nu}\vec{w},z.(\boldsymbol{\nu} y.(\overline{z}y \mid P) \mid z(y).(\overline{z'}y \mid S)) \quad \text{since } z \notin \mathsf{fn}(P) \cup \mathsf{fn}(S) \cup y,\vec{w} \\
&= \boldsymbol{\nu}\vec{w}, z.(H_{z,y}[P] \mid z(y).(\overline{z'}y \mid S))
\end{aligned}
$$

Thus we let $C' = \boldsymbol{\nu}\vec{w}, z.(- \mid z(y).(\overline{z'}y \mid S))$

case $y \in \vec{w}$**:** Then $y, \vec{v} = \vec{w}$ for some $\vec{v}$. Now

$$
\begin{aligned}
H_{z,y}[C[P]] &\equiv \boldsymbol{\nu} y.(\overline{z}y \mid \boldsymbol{\nu}\vec{w}.(P \mid S)) \\
&\equiv \boldsymbol{\nu} y.(\overline{z}y) \mid \boldsymbol{\nu}\vec{w},z.(\overline{z}y \mid P \mid S) \quad \text{since } z \notin \mathsf{fn}(P) \cup \mathsf{fn}(S) \\
&\approx_\ell \boldsymbol{\nu} y.(\overline{z}y) \mid \boldsymbol{\nu}\vec{v},z.(\boldsymbol{\nu} y.(\overline{z}y \mid P) \mid z(y).(\overline{z'}y \mid S)) \quad \text{since } z \notin \mathsf{fn}(P) \cup \mathsf{fn}(S) \cup \vec{w} \\
&= \boldsymbol{\nu} y.(\overline{z}y) \mid \boldsymbol{\nu}\vec{v}, z.(H_{z,y}[P] \mid z(y).(\overline{z'}y \mid S))
\end{aligned}
$$

Thus we let $C' = \boldsymbol{\nu} y.(\overline{z}y) \mid \boldsymbol{\nu}\vec{v}, z.(- \mid z(y).(\overline{z'}y \mid S))$