Bisimulation techniques for the masses

Jean-Marie Madiot

École Normale Supérieure de Lyon
Università di Bologna

EJCP 2013, Dinard, 2013-05-25

General setting

Imagine any calculus with the following:

- some notion of reduction:
  - $\overline{a}b.P \mid a(x).Q \to P \mid Q[b/x]$ (π-calculus),
  - $(\lambda x.t)(u) \to t[u/x]$ (λ-calculus),
  - $\langle s \mid r := 5; c\rangle \to \langle s[r \mapsto 5] \mid c\rangle$ (imperative language);
- some notion of observation:
  - $\ldots \mid \overline{a}b.P \mid \ldots$,
  - $\lambda x.t$,
  - $\langle s \mid \mathsf{skip}\rangle$.
- some notion of context:
  - $(\boldsymbol{\nu} a)(\overline{a} \mid a.[\,])$,
  - $[\,](\lambda x.x)$,
  - $\langle b \mapsto \mathsf{false} \mid \mathsf{while(b)\{[\,]\}}\rangle$.

Example: fixed-point combinators

fixed point: x such that $x = f(x)$

fixed-point combinator: c such that $c(f)$ is a fixed point: $c(f) = f(c(f))$

$$Y = \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$$
$$\Theta = (\lambda x.\lambda y.(y(xxy)))(\lambda x.\lambda y.(y(xxy)))$$

$$Y \simeq \Theta \quad ?$$

What would it mean?

Behavioural equivalence

"P and Q behave the same in all circumstances."

$$P \simeq Q \quad \Leftrightarrow \quad \forall C,\ C[P] \mathrel{\dot{\sim}} C[Q]$$

Behavioural equivalence

"P and Q behave the same in all circumstances."

$$P \simeq Q \quad \Leftrightarrow \quad \forall C,\ C[P] \mathrel{\dot{\sim}} C[Q]$$

$$P \mathrel{\dot{\sim}} Q$$

$$\Downarrow$$

P is observable iff

Q is observable

$$P \quad \dot{\sim} \quad Q$$

$$\downarrow$$

$$P'$$

Behavioural equivalence

"P and Q behave the same in all circumstances."

$$P \simeq Q \quad \Leftrightarrow \quad \forall C,\ C[P] \mathrel{\dot{\sim}} C[Q]$$

$$P \mathrel{\dot{\sim}} Q$$

$$\Downarrow$$

P is observable iff

Q is observable

$$\begin{array}{ccc} P & \dot{\sim} & Q \\ \downarrow & & \downarrow \\ P' & \dot{\sim} & Q' \end{array}$$

Equivalence

$$P \simeq Q \quad \Leftrightarrow \quad \forall C,\ C[P] \mathrel{\dot{\sim}} C[Q]$$

Equivalence

$$P \simeq Q \quad \Leftrightarrow \quad \forall C,\ C[P] \mathrel{\dot{\sim}} C[Q]$$

Equivalence

$$P \simeq Q \quad \Leftrightarrow \quad \forall C,\ C[P] \mathrel{\dot{\sim}} C[Q]$$
$$\Leftrightarrow \quad P \sim Q$$

Equivalence

$$P \simeq Q \quad \Leftrightarrow \quad \forall C,\ C[P] \mathrel{\dot{\sim}} C[Q]$$
$$\Leftrightarrow \quad P \sim Q$$

$P \sim Q$ has to be easier to prove and compute

Bisimilarity

- Remove the $\forall C$ quantification,
- more complex notion of reduction $P \xrightarrow{\alpha} P'$ instead of $P \longrightarrow P'$.

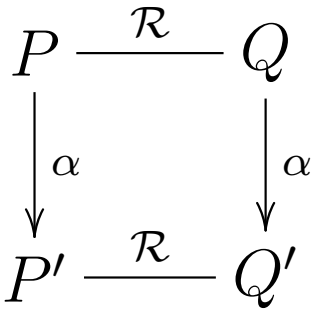
$$\begin{array}{ccc} P & \sim & Q \\ \alpha\downarrow & & \\ P' & & \end{array}$$

Bisimilarity

- Remove the $\forall C$ quantification,
- more complex notion of reduction $P \xrightarrow{\alpha} P'$ instead of $P \longrightarrow P'$.

$$\begin{array}{ccc} P & \sim & Q \\ \alpha\downarrow & & \alpha\downarrow \\ P' & \sim & Q' \end{array}$$

Techniques



$+$ up-to techniques

Using a large library of up-to techniques [Pous 2008, PhD thesis].

- works very well calculi with simple α's (graphs, CCS, automata, ...),
- we adapt it for calculi with complex α's,
  - π-calculi (π, Aπ, HOπ),
  - λ-calculi (cbn, cbv, cbv + ref, other effects),
  - your calculus here.

How: we simplify the α's and the library comes free.

Example: fixed-point combinators

$$Y = \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$$
$$\Theta = (\lambda x.\lambda y.(y(xxy)))(\lambda x.\lambda y.(y(xxy)))$$

Theorem

$Y \sim \Theta$

Proof: $\mathcal{R} = \{(Y, \Theta), (Y, (\lambda y.y(\Theta y)), (\delta_f \delta_f, \Theta f)\}$ is a bisimulation up to reduction, congruence and bisimilarity $(\delta_f = \lambda x.f(xx))$.

Example: fixed-point combinators

$$Y = \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$$
$$\Theta = (\lambda x.\lambda y.(y(xxy)))(\lambda x.\lambda y.(y(xxy)))$$

Theorem

$Y \sim \Theta$

Proof: $\mathcal{R} = \{(Y, \Theta), (Y, (\lambda y.y(\Theta y)), (\delta_f \delta_f, \Theta f)\}$ is a bisimulation up to reduction, congruence and bisimilarity $(\delta_f = \lambda x.f(xx))$.

$$Y \sim (\lambda x.xxxxxxxxxxxxxxxxxxxxxxxxxx)$$
$$(\lambda abcdefghijklmnopqstuvwxyzr.$$
$$(r(this\ is\ a\ fixed\ point\ combinator)))$$