Coinduction All the Way Up

Damien Pous

Plume group, CNRS, ENS Lyon

Gallium, Paris, 12.o9.2o16

Program Equivalence

Through equational reasoning

$$
\begin{aligned}
!a.\, a &= !(a \mid a) \\
&= !a \mid !a \\
&= !a
\end{aligned}
$$

$$
\begin{aligned}
(a.\, a &= a \mid a) \\
(!(P \mid Q) &= !P \mid !Q) \\
(!P \mid !P &= !P)
\end{aligned}
$$

(Axiomatic, syntax-based, inductive)

Program Equivalence

Through bisimulations

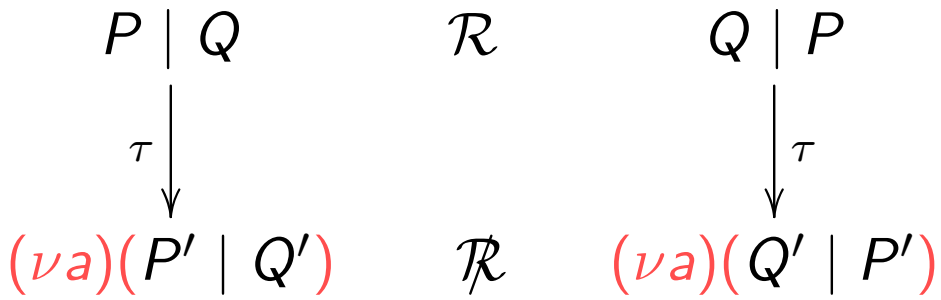
$$\mathcal{R} \triangleq \{\langle P \mid Q,\ Q \mid P\rangle \ \mid\ \forall P, Q\}$$

(Behavioural, LTS-based, coinductive)

Program Equivalence

Through bisimulations

$$\mathcal{R} \triangleq \{\langle P \mid Q,\ Q \mid P\rangle \ \mid\ \forall P, Q\}$$



Program Equivalence

Through bisimulations

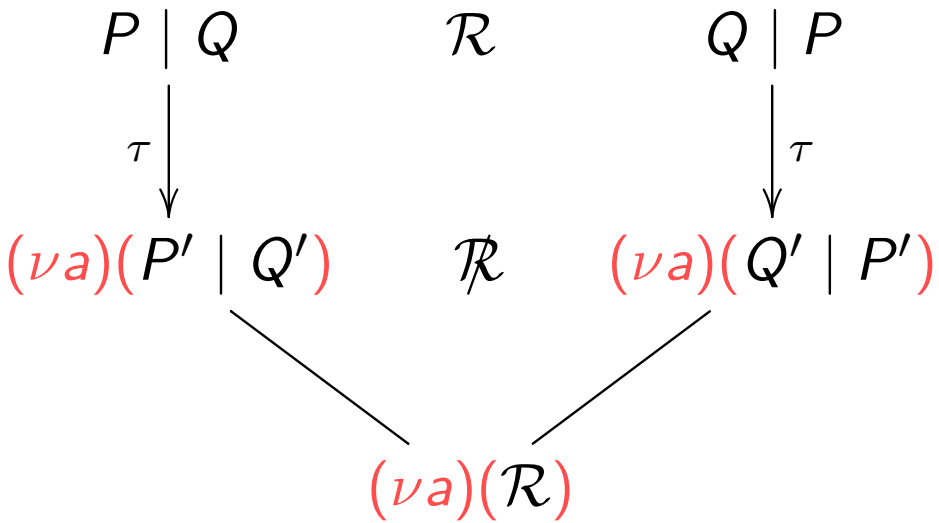
$$\mathcal{R} \triangleq \{\langle (\nu \tilde{a})(P \mid Q), (\nu \tilde{a})(Q \mid P) \mid \forall \tilde{a}, P, Q \rangle\}$$

$$\begin{array}{ccccc} P \mid Q & & \mathcal{R} & & Q \mid P \\ \Big\downarrow{\scriptstyle \tau} & & & & \Big\downarrow{\scriptstyle \tau} \\ (\nu a)(P' \mid Q') & & \mathcal{R} & & (\nu a)(Q' \mid P') \end{array}$$

Program Equivalence

Through bisimulations up-to

$$\mathcal{R} \triangleq \{\langle P \mid Q,\ Q \mid P\rangle \ \mid\ \forall P, Q\}$$



Program Equivalence

Through bisimulations up-to

$$\mathcal{R} \triangleq \{\langle !(P \mid Q),\ !P \mid !Q \rangle\}$$

Program Equivalence

Through bisimulations up-to

$$\mathcal{R} \triangleq \{\langle !(P \mid Q),\ !P \mid !Q \rangle\}$$

$$\begin{array}{ccc} !(P \mid Q) & \mathcal{R} & !P \mid !Q \\ {\scriptstyle\alpha}\downarrow & & \downarrow{\scriptstyle\alpha} \\ !(P \mid Q) \mid (P' \mid Q) & \not{\mathcal{R}} & (!P \mid P') \mid !Q \end{array}$$

Program Equivalence

Through bisimulations up-to

$$\mathcal{R} \triangleq \{\langle !(P \mid Q),\ !P \mid !Q \rangle\}$$

$$\begin{array}{ccc} !(P \mid Q) & \mathcal{R} & !P \mid !Q \\ {\scriptstyle\alpha}\downarrow & & \downarrow{\scriptstyle\alpha} \\ !(P \mid Q) \mid (P' \mid Q) & \mathcal{R}\!\!\!/ & (!P \mid P') \mid !Q \\ & & \sim \\ & & (!P \mid P') \mid (!Q \mid Q) \end{array}$$

Program Equivalence

Through bisimulations up-to

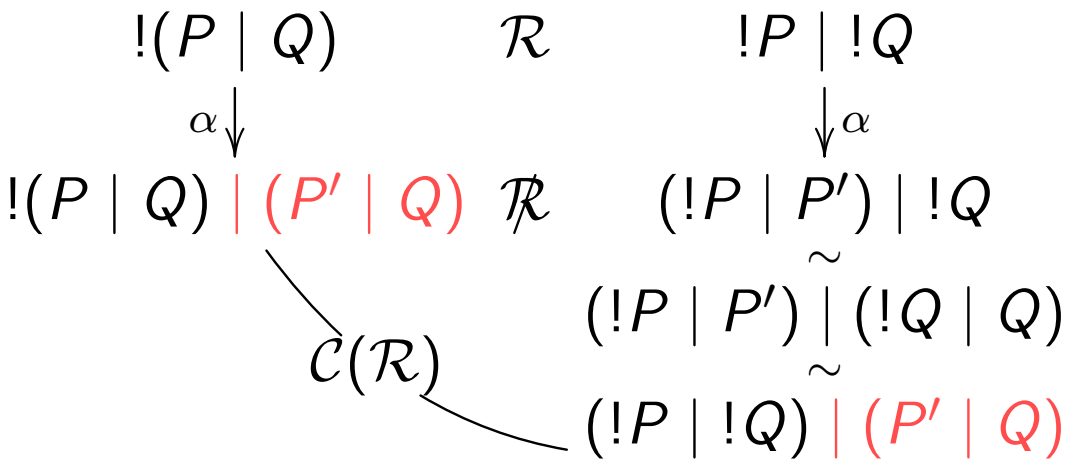
$$\mathcal{R} \triangleq \{\langle !(P \mid Q),\ !P \mid !Q \rangle\}$$

$$\begin{array}{ccc}
!(P \mid Q) & \mathcal{R} & !P \mid !Q \\
{\scriptstyle\alpha}\downarrow & & \downarrow{\scriptstyle\alpha} \\
!(P \mid Q) \mid (P' \mid Q) & \not{\mathcal{R}} & (!P \mid P') \mid !Q \\
 & & (!P \mid P') \mid (!Q \mid Q) \\
 & & (!P \mid !Q) \mid (P' \mid Q)
\end{array}$$

Program Equivalence

Through bisimulations up-to

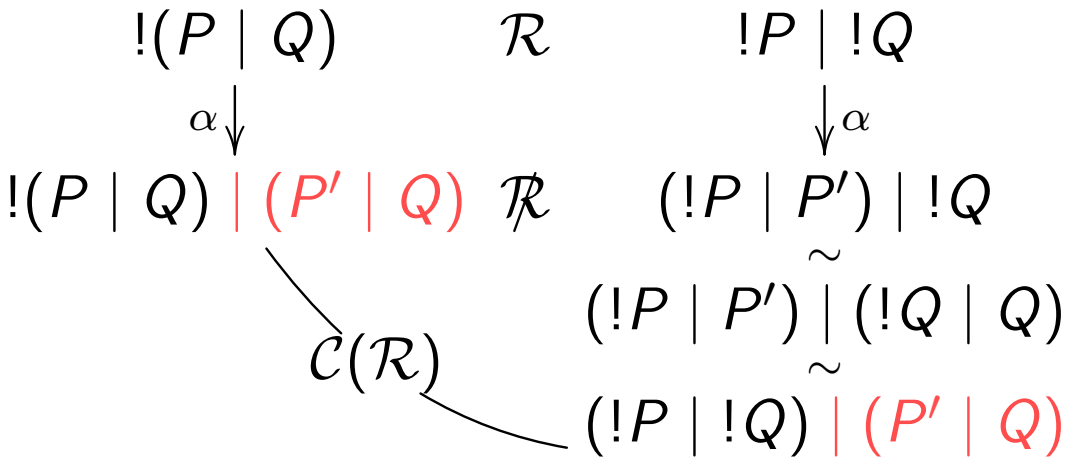
$$\mathcal{R} \triangleq \{\langle !(P \mid Q),\ !P \mid !Q \rangle\}$$



Program Equivalence

Through bisimulations up-to

$$\mathcal{R} \triangleq \{\langle !(P \mid Q),\ !P \mid !Q \rangle\}$$



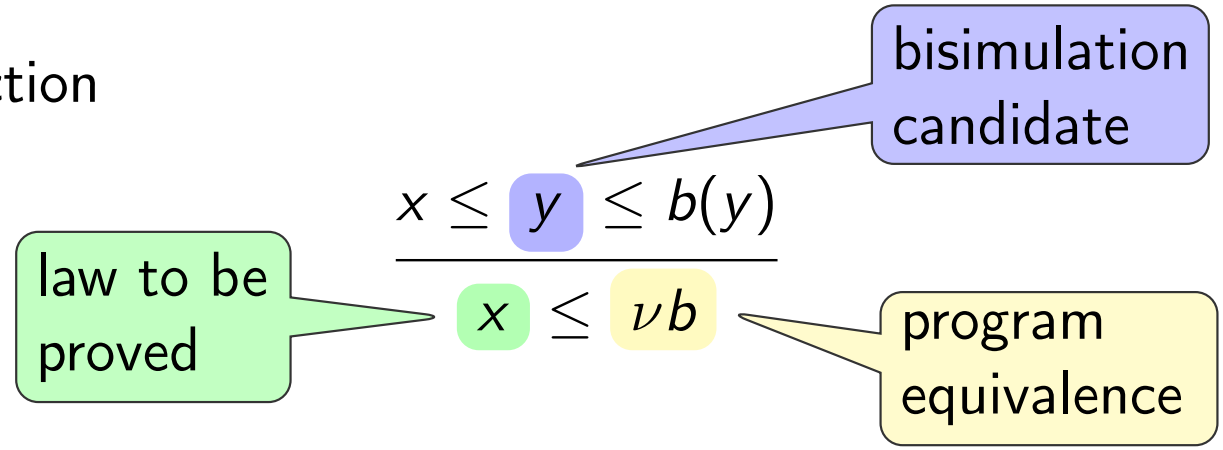
A mix of

- behavioural, LTS-based, coinductive
- axiomatic, syntax-based, inductive

Abstract coinduction

Let b a monotone function on a complete lattice

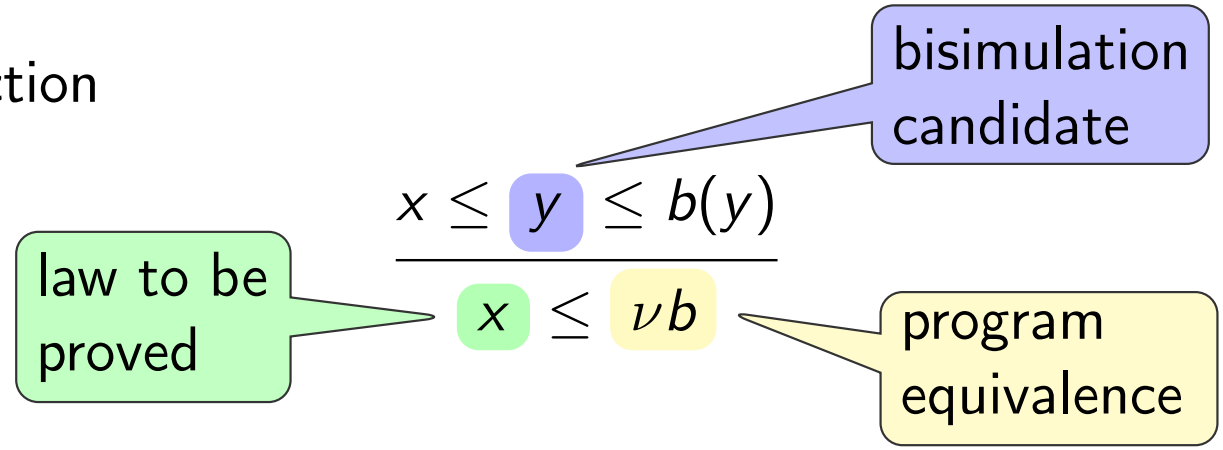
- Coinduction



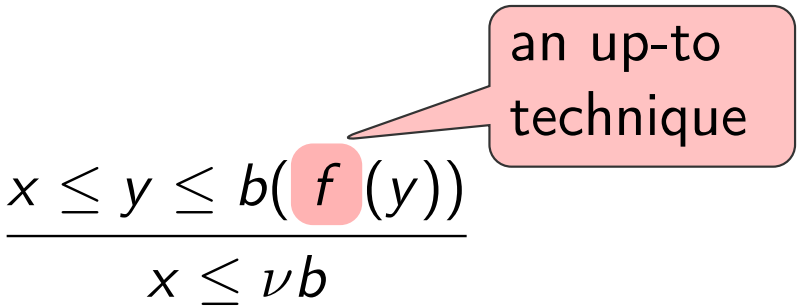
Abstract coinduction

Let b a monotone function on a complete lattice

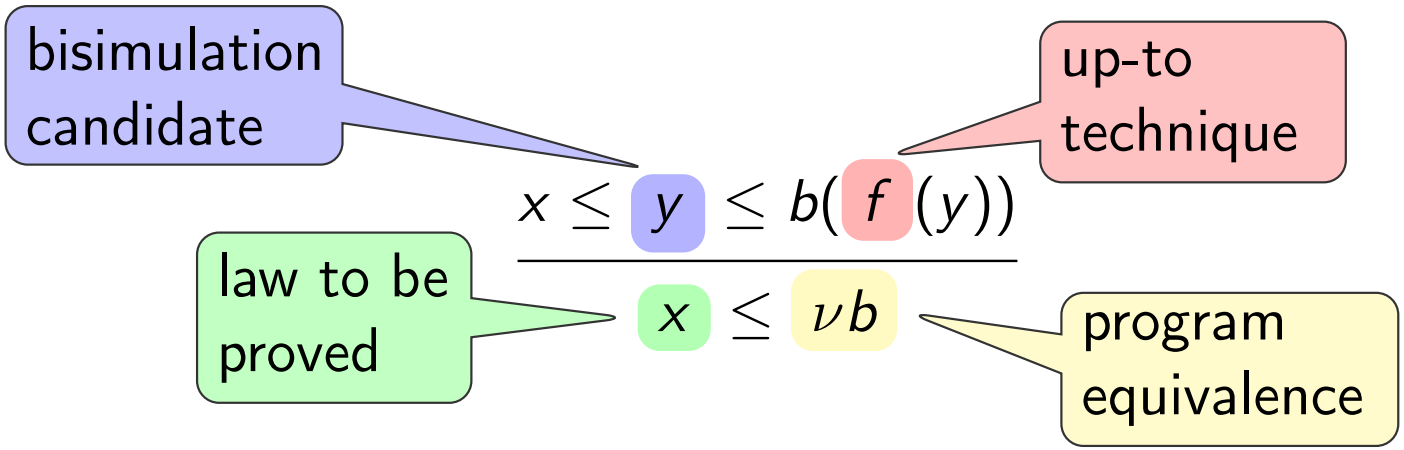
- Coinduction



- Coinduction up to



Up-to techniques



The function f can be

- sound: it just makes the rule valid
- respectful: $f \circ (b \cap id) \leq b \circ f$ [Sangiorgi'94]
- compatible: $f \circ b \leq b \circ f$ [me'07]

The latter two classes are closed under union and composition

The companion

Let t be the largest compatible function

- We have
  1. $id \leq t$ and $t \circ t \leq t$, i.e., t is a closure
  2. $\nu b = t(\bot)$
  3. $\nu b = t(\nu b)$
  4. $\nu b = \nu(b \circ t)$
  5. t coincides with the largest respectful

- Intuitively $t(x)$ is "what can be deduced assuming x"

- Leads to a new presentation of parameterized coinduction

Parameterized Coinduction (Up-to)

$$\frac{y \leq t(\bot)}{y \leq \nu b}\ \text{Init}$$

law to be proved

program equivalence

$$\frac{y \leq \mathrm{f}(t(x)) \qquad f \leq t}{y \leq t(x)}\ \text{Up to } f$$

reason inductively, using some valid up-to technique

$$\frac{y \leq \mathrm{x}}{y \leq t(x)}\ \text{Done}$$

use coinduction hypotheses

$$\frac{y \leq b(t(\mathrm{y} \cup x))}{y \leq t(x)}\ \text{CoInd}$$

assume y by coinduction

Second order techniques

The companion is itself a coinductive object

$$t = \nu B$$

(for some $B : (X \to X) \to (X \to X)$ whose post-fixpoints are the compatible functions)

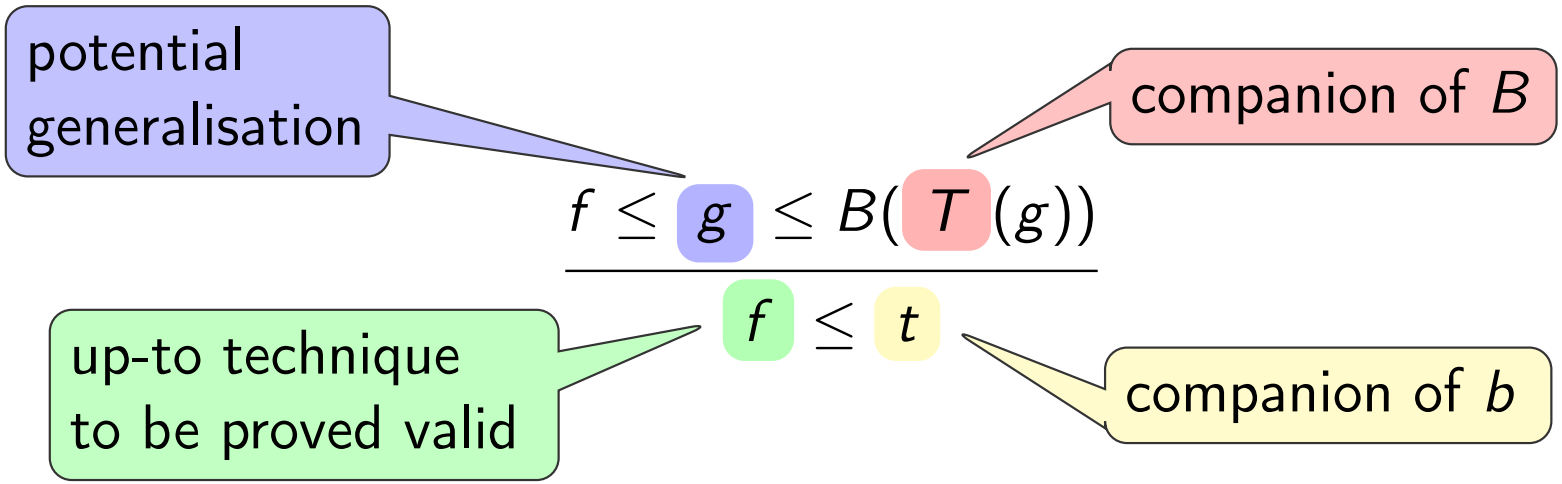
Second order techniques

The companion is itself a coinductive object

$$t = \nu B$$

(for some $B : (X \to X) \to (X \to X)$ whose post-fixpoints are the compatible functions)

We can apply the previous results



Back to process algebra

[Standard approach] The following function is respectful:

$$\mathcal{C} : \mathcal{R} \mapsto \{\langle C[\tilde{P}],\, C[\tilde{Q}]\rangle \mid \tilde{P}\ \mathcal{R}\ \tilde{Q}\}$$

Back to process algebra

[Standard approach] The following function is respectful:

$$\mathcal{C} : \mathcal{R} \mapsto \{\langle C[\tilde{P}],\, C[\tilde{Q}]\rangle \mid \tilde{P}\ \mathcal{R}\ \tilde{Q}\}$$

[New approach] The following functions are below t:

$$\begin{aligned} c_{(\nu a)} &: \mathcal{R} \mapsto \{\langle (\nu a)P,\, (\nu a)Q\rangle \mid P\ \mathcal{R}\ Q\} \\ c_{|} &: \mathcal{R} \mapsto \{\langle P_1 \mid P_2,\, Q_1 \mid Q_2\rangle \mid P_i\ \mathcal{R}\ Q_i\} \\ c_{!} &: \mathcal{R} \mapsto \{\langle !P,\, !Q\rangle \mid P\ \mathcal{R}\ Q\} \\ &\ldots \end{aligned}$$

(A routine check in each case, thanks to the second order companion)

Back to process algebra

```
Lemma rep_t: unary_ctx rep <= t.
Proof.
  apply Coinduction, by_Symmetry. apply unary_sym.
  intro R. apply (leq_unary_ctx rep). intros p q Hpq l p0 Hp0.
  apply rep_trans in Hp0 as [p1 ppp1 p0p1].
  assert (H: b (t R) (par p p) (par q q)).
   apply (compat_t b). apply par_t. now apply in_binary_ctx.
  destruct (proj1 H _ _ ppp1) as [q1 qqq1 p1q1].
  apply rep_trans' in qqq1 as [q0 Hq0 q0q1].
  eexists. eassumption.
  rewrite p0p1, q0q1.
  apply (ftT_T _ par_t). apply (in_binary_ctx par).
   apply (fTf_Tf b). apply (in_unary_ctx rep). now apply (b_T b).
   now apply (t_T b).
Qed.
```

[Standard approach] The foll

$$\mathcal{C} : \mathcal{R} \mapsto \{\langle C$$

[New approach] The following functions are below t:

$$c_{(\nu a)} : \mathcal{R} \mapsto \{\langle (\nu a)P, (\nu a)Q\rangle \mid P\ \mathcal{R}\ Q\}$$

$$c_{|} : \mathcal{R} \mapsto \{\langle P_1 \mid P_2, Q_1 \mid Q_2\rangle \mid P_i\ \mathcal{R}\ Q_i\}$$

$$c_{!} : \mathcal{R} \mapsto \{\langle !P, !Q\rangle \mid P\ \mathcal{R}\ Q\}$$

$$\ldots$$

(A routine check in each case, thanks to the second order companion)

Summary

Huge simplification of the theory of up-to techniques...

...just by focusing on the largest one

- parameterized coinduction through the companion
- second order techniques for the companion
- straightforward formalisation in Coq

In progress: categorical account