# On substitution closure and congruence in $\pi$
## and axiomatisations of bisimilarity in absence of sum

Daniel Hirschkoff, ENS de Lyon
Damien Pous, CNRS, Grenoble

Dublin, 15.o4.2o11

# Open question

In which fragments of $\pi$ is bisimilarity ($\sim$) a congruence?

- $\pi$ is a process calculus ($P ::= \mathbf{0} \mid P \mid P \mid \alpha.P \mid (\nu a)P \mid \ldots$);
- equipped with a labelled transition semantics $P \xrightarrow{\alpha} P'$;
- yielding (labelled) bisimilarity:

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

- congruence property: does $P \sim Q$ entail $C[P] \sim C[Q]$ ?
- substitution closure: does $P \sim Q$ entail $P\sigma \sim Q\sigma$ ?

# Some known answers

- With sum, substitution closure and congruence fail:

$$\overline{a} \mid b \qquad \sim \qquad \overline{a}.b + b.\overline{a}$$

but

$$\begin{tikzcd}
c(a).\ (\overline{a} \mid b) \arrow[d, "c(b)"'] & \not\sim & c(a).\ (\overline{a}.b + b.\overline{a}) \arrow[d, "c(b)"] \\
(\overline{a} \mid b)\ \{b/a\} \arrow[d, "\tau"'] & \not\sim & (\overline{a}.b + b.\overline{a})\ \{b/a\} \arrow[d, "\tau"', "\times"{marking}] \\
{} & & {}
\end{tikzcd}$$

# Some known answers

- With sum, substitution closure and congruence fail:

$$\overline{a} \mid b \quad \sim \quad \overline{a}.b + b.\overline{a}$$

but

$$c(a).\,(\overline{a} \mid b) \quad \not\sim \quad c(a).\,(\overline{a}.b + b.\overline{a})$$

$$\begin{array}{ccc} c(a).\,(\overline{a} \mid b) & \not\sim & c(a).\,(\overline{a}.b + b.\overline{a}) \\ \Big\downarrow{\scriptstyle c(b)} & & \Big\downarrow{\scriptstyle c(b)} \\ (\overline{a} \mid b)\ \{b/a\} & \not\sim & (\overline{a}.b + b.\overline{a})\ \{b/a\} \\ \Big\downarrow{\scriptstyle \tau} & & \not\Big\downarrow{\scriptstyle \tau} \end{array}$$

- In the asynchronous $\pi$-calculus, they hold.

# Some answers, cont.

- In the synchronous case, without sum:

# Some answers, cont.

- In the synchronous case, without sum: no
  see [Sangiorgi & Walker 2001],
  counter-example with replication (!) and name restriction ($\nu$):

$$\begin{array}{rrcl} & !a.\overline{b}.\tau.\overline{t} \mid !\overline{b}.a.\tau.\overline{t} & \sim & !(\nu z)\,(a.z.\overline{t} \mid \overline{b}.\overline{z}) \\ \text{but} & !a.\overline{a}.\tau.\overline{t} \mid !\overline{a}.a.\tau.\overline{t} & \not\sim & !(\nu z)\,(a.z.\overline{t} \mid \overline{a}.\overline{z}) \end{array}$$

# Some answers, cont.

- In the synchronous case, without sum: no
  see [Sangiorgi & Walker 2001],
  counter-example with replication (!) and name restriction ($\nu$):

$$\begin{array}{lccc} & !a.\overline{b}.\tau.\overline{t} \mid !\overline{b}.a.\tau.\overline{t} & \sim & !(\nu z)\,(a.z.\overline{t} \mid \overline{b}.\overline{z}) \\ \text{but} & !a.\overline{a}.\tau.\overline{t} \mid !\overline{a}.a.\tau.\overline{t} & \not\sim & !(\nu z)\,(a.z.\overline{t} \mid \overline{a}.\overline{z}) \end{array}$$

- Our work: removing either replication or name restriction.

  Outline of the presentation:

  1. $\mu$CCS: neither replication nor name restriction [FoSSaCS 2007]
  2. finite $\pi$: no replication [FoSSaCS 2007]
  3. top-level replications (without name restriction) [ICALP 2010]

# $\mu$CCS

Consider the following tiny fragment of CCS:

$$E, F ::= \mathbf{0} \;\Big|\; F \mid F \;\Big|\; a.F$$

- no sum,
- no name restriction or relabelling,
- no replication or recursion,
- no synchronisation.

What does bisimilarity look like? Is it substitution closed?

# Bisimilarity in $\mu$CCS

- bisimilarity is a congruence;
- $(|, \mathbf{0})$ is an abelian monoid:

$$E \mid \mathbf{0} \sim E \qquad E \mid F \sim F \mid E \qquad E \mid (F \mid G) \sim (E \mid F) \mid G$$

- a distribution law relates prefixed processes and parallel composition:

$$a.(F \mid a.F) \sim a.F \mid a.F$$

# Bisimilarity in $\mu$CCS

- bisimilarity is a congruence;
- $(|, \mathbf{0})$ is an abelian monoid:

$$E \mid \mathbf{0} \sim E \qquad E \mid F \sim F \mid E \qquad E \mid (F \mid G) \sim (E \mid F) \mid G$$

- a distribution law relates prefixed processes and parallel composition:

$$a.(F \mid a.F) \sim a.F \mid a.F$$

more generally, $$a.(F \mid (a.F)^n) \sim (a.F)^{n+1}$$

# Bisimilarity in $\mu$CCS

- bisimilarity is a congruence;
- $(|, \mathbf{0})$ is an abelian monoid:

$$E \mid \mathbf{0} \sim E \qquad E \mid F \sim F \mid E \qquad E \mid (F \mid G) \sim (E \mid F) \mid G$$

- a distribution law relates prefixed processes and parallel composition:

$$a.(F \mid a.F) \sim a.F \mid a.F$$

more generally, $$a.(F \mid (a.F)^n) \sim (a.F)^{n+1}$$

Theorem: *the above laws axiomatise bisimilarity in $\mu$CCS.*

# Bisimilarity in $\mu$CCS

- bisimilarity is a congruence;
- $(|, \mathbf{0})$ is an abelian monoid:

$$E \mid \mathbf{0} \sim E \qquad E \mid F \sim F \mid E \qquad E \mid (F \mid G) \sim (E \mid F) \mid G$$

- a distribution law relates prefixed processes and parallel composition:

$$a.(F \mid a.F) \sim a.F \mid a.F$$

more generally, $\quad a.(F \mid (a.F)^n) \sim (a.F)^{n+1}$

Theorem: *the above laws axiomatise bisimilarity in $\mu$CCS.*
Corollary: *bisimilarity is substitution closed in $\mu$CCS.*

# Overview of the proof

1. Use the distribution law to normalise processes:

$$a.(F \mid (a.F)^n) \to (a.F)^{n+1}$$

(so that normal forms express the maximal degree of parallelism.)

2. Show that bisimilarity coincides with structural congruence on normal forms

# Overview of the proof

1. Use the distribution law to normalise processes:

$$a.(F \mid (a.F)^n) \to (a.F)^{n+1}$$

(so that normal forms express the maximal degree of parallelism.)

2. Show that bisimilarity coincides with structural congruence on normal forms

Two key lemmas:

- [Milner&Moller 1993] Any process admits a unique decomposition into prime factors

  (a process is prime if it cannot be decomposed as a parallel composition)

- If $a.F \sim E_1 \mid E_2$ (with $E_1, E_2 \not\sim \mathbf{0}$)
  then $a.F \sim (a.F')^k$ (with $k > 1$ and $F'$ in normal form)

# Outline

$\mu$CCS

finite $\pi$

Top-level replications

# Adding name restriction

- The previous strategy fails when adding name restriction: we found no axiomatisation

  (remember that the calculus lacks sum).

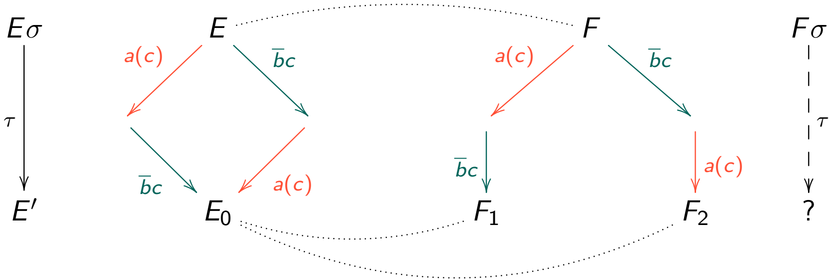
- We can however exploit the previous results.

# Proving substitution closure in $\pi$

- Show that $\{(E\sigma, F\sigma) \mid E \sim F\}$ is a bisimulation:
  - if $E\sigma \xrightarrow{\alpha} E'$, can $F\sigma$ answer?
  - easy except when $\alpha = \tau$, $\xleftarrow{a(c)} E \xrightarrow{\bar{b}c}$, with $\sigma(a) = \sigma(b)$:

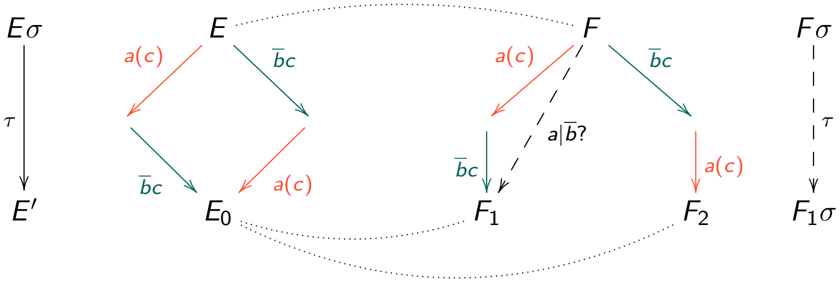
# Proving substitution closure in $\pi$

- Show that $\{(E\sigma, F\sigma) \mid E \sim F\}$ is a bisimulation:
  - if $E\sigma \xrightarrow{\alpha} E'$, can $F\sigma$ answer?
  - easy except when $\alpha = \tau$, $\xleftarrow{a(c)} E \xrightarrow{\overline{b}c}$, with $\sigma(a) = \sigma(b)$:



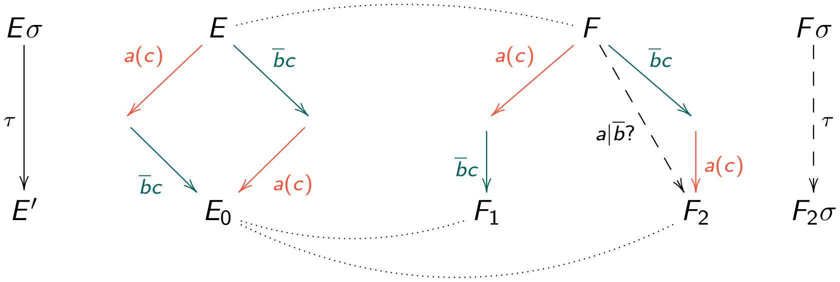
# Proving substitution closure in $\pi$

- Show that $\{(E\sigma, F\sigma) \mid E \sim F\}$ is a bisimulation:
  - if $E\sigma \xrightarrow{\alpha} E'$, can $F\sigma$ answer?
  - easy except when $\alpha = \tau$, $\xleftarrow{a(c)} E \xrightarrow{\overline{b}c}$, with $\sigma(a) = \sigma(b)$:



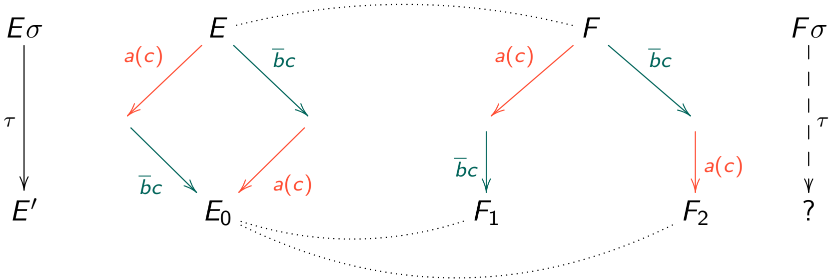
# Proving substitution closure in $\pi$

- Show that $\{(E\sigma, F\sigma) \mid E \sim F\}$ is a bisimulation:
  - if $E\sigma \xrightarrow{\alpha} E'$, can $F\sigma$ answer?
  - easy except when $\alpha = \tau$, $\xleftarrow{a(c)} E \xrightarrow{\overline{b}c}$, with $\sigma(a) = \sigma(b)$:



# Proving substitution closure in $\pi$

- Show that $\{(E\sigma, F\sigma) \mid E \sim F\}$ is a bisimulation:
  - if $E\sigma \xrightarrow{\alpha} E'$, can $F\sigma$ answer?
  - easy except when $\alpha = \tau$, $\xleftarrow{a(c)} E \xrightarrow{\overline{b}c}$, with $\sigma(a) = \sigma(b)$:




  - the remaining case is when both answers use sequential prefixes; call this a mutual desynchronisation.

# Using $\mu$CCS to finish the proof

$$
\begin{tikzcd}
 & F \arrow[dl, "a(c)"'] \arrow[dr, "\overline{b}c"] & \\
{} \arrow[d, "\overline{b}c"'] & \pi & {} \arrow[d, "a(c)"] \\
F_1 & \sim & F_2
\end{tikzcd}
$$

# Using $\mu$CCS to finish the proof

$$\begin{tikzcd}
 & F \arrow[dl, "a(c)"'] \arrow[dr, "\overline{b}c"] & \\
 {} \arrow[d, "\overline{b}c"'] & \pi & {} \arrow[d, "a(c)"] \\
F_1 & \sim & F_2
\end{tikzcd}$$

- Port this mutual desynchronisation to $\mu$CCS by using an erasing function;

# Using $\mu$CCS to finish the proof

$$
\begin{tikzcd}
& F \arrow[dl, "a(c)"'] \arrow[dr, "\overline{b}c"] & \\
{} \arrow[d, "\overline{b}c"'] & \pi & {} \arrow[d, "a(c)"] \\
F_1 & \sim & F_2
\end{tikzcd}
$$

*a and b are fixed*

$$\langle c(x).E\rangle = \begin{cases} a.\langle E\rangle & \text{if } c = a \\ \mathbf{0} & \text{otherwise} \end{cases}$$

$$\langle \overline{c}d.E\rangle = \begin{cases} \overline{b}.\langle E\rangle & \text{if } c = b \\ \mathbf{0} & \text{otherwise} \end{cases}$$

$$\langle (\nu c)E\rangle = \langle E\rangle \qquad (c \neq a, b)$$

- Port this mutual desynchronisation to $\mu$CCS by using an erasing function;

$$\langle (\nu c)(a(x).(\overline{b}x \mid \overline{x}c.\overline{a}c) \mid \overline{b}c.a(y).\overline{y}c)\rangle = a.\overline{b} \mid \overline{b}.a$$

# Using $\mu$CCS to finish the proof

$$
\begin{tikzcd}
 & F \arrow[dl, "a(c)"'] \arrow[dr, "\overline{b}c"] & \\
 {} \arrow[d, "\overline{b}c"'] & \pi & {} \arrow[d, "a(c)"] \\
 F_1 & \sim & F_2
\end{tikzcd}
$$

*$a$ and $b$ are fixed*

$$\langle c(x).E\rangle = \begin{cases} a.\langle E\rangle & \text{if } c = a \\ \mathbf{0} & \text{otherwise} \end{cases}$$

$$\langle \overline{c}d.E\rangle = \begin{cases} \overline{b}.\langle E\rangle & \text{if } c = b \\ \mathbf{0} & \text{otherwise} \end{cases}$$

$$\langle (\nu c)E\rangle = \langle E\rangle \qquad (c \neq a, b)$$

- Port this mutual desynchronisation to $\mu$CCS by using an erasing function;
$$\langle (\nu c)(a(x).(\overline{b}x \mid \overline{x}c.\overline{a}c) \mid \overline{b}c.a(y).\overline{y}c)\rangle = a.\overline{b} \mid \overline{b}.a$$
- Proposition: *$E \sim F$ in $\pi$ implies $\langle E\rangle \sim \langle F\rangle$ in $\mu$CCS.*

# Using $\mu$CCS to finish the proof

$$\begin{tikzcd}
& \langle F\rangle \arrow[dl, "a"'] \arrow[dr, "\overline{b}"] & \\
{} \arrow[d, "\overline{b}"'] & \mu\mathrm{CCS} & {} \arrow[d, "a"] \\
\langle F_2\rangle & \sim & \langle F_2\rangle
\end{tikzcd}$$

*a and b are fixed*

$$\langle c(x).E\rangle = \begin{cases} a.\langle E\rangle & \text{if } c = a \\ \mathbf{0} & \text{otherwise} \end{cases}$$

$$\langle \overline{c}d.E\rangle = \begin{cases} \overline{b}.\langle E\rangle & \text{if } c = b \\ \mathbf{0} & \text{otherwise} \end{cases}$$

$$\langle (\nu c)E\rangle = \langle E\rangle \qquad (c \neq a, b)$$

- Port this mutual desynchronisation to $\mu$CCS by using an erasing function;
$$\langle (\nu c)(a(x).(\overline{b}x \mid \overline{x}c.\overline{a}c) \mid \overline{b}c.a(y).\overline{y}c)\rangle = a.\overline{b} \mid \overline{b}.a$$
- Proposition: *$E \sim F$ in $\pi$ implies $\langle E\rangle \sim \langle F\rangle$ in $\mu$CCS.*

# Using $\mu$CCS to finish the proof

$$\begin{tikzcd}
 & \langle F \rangle \arrow[dl, "a"'] \arrow[dr, "\overline{b}"] & \\
{} \arrow[d, "\overline{b}"'] & \mu\text{CCS} & {} \arrow[d, "a"] \\
\langle F_2 \rangle & \sim & \langle F_2 \rangle
\end{tikzcd}$$

*a and b are fixed*

$$\langle c(x).E \rangle = \begin{cases} a.\langle E \rangle & \text{if } c = a \\ \mathbf{0} & \text{otherwise} \end{cases}$$

$$\langle \overline{c}d.E \rangle = \begin{cases} \overline{b}.\langle E \rangle & \text{if } c = b \\ \mathbf{0} & \text{otherwise} \end{cases}$$

$$\langle (\nu c)E \rangle = \langle E \rangle \qquad (c \neq a, b)$$

- Port this mutual desynchronisation to $\mu$CCS by using an erasing function;
$$\langle (\nu c)(a(x).(\overline{b}x \mid \overline{x}c.\overline{a}c) \mid \overline{b}c.a(y).\overline{y}c) \rangle = a.\overline{b} \mid \overline{b}.a$$
- Proposition: *$E \sim F$ in $\pi$ implies $\langle E \rangle \sim \langle F \rangle$ in $\mu$CCS.*
- The axiomatisation of $\sim$ on $\mu$CCS tells us that a mutual desynchronisation cannot happen in $\mu$CCS.

# Corollary:

*On finite $\pi$ without sum,*
*ground, early, late and open bisimilarity*
*coincide and are congruences.*

# Outline

$\mu$CCS

finite $\pi$

Top-level replications

# mCCS

- mCCS is $\mu$CCS with top-level replications:

$$E, F ::= \mathbf{0} \;\big|\; F \mid F \;\big|\; a.F$$
$$P, Q ::= F \;\big|\; P \mid P \;\big|\; !a.F \qquad\qquad (\text{with } !a.F \xrightarrow{a} !a.F \mid F)$$

- What does bisimilarity look like? Is it substitution closed?

# Bisimilarity in mCCS

- Same laws as in $\mu$CCS (abelian monoid, distribution law);
- Standard laws for replication:

$$!a.F \mid !a.F \sim !a.F \qquad\qquad !a.F \mid a.F \sim !a.F$$

- Other phenomena:

$$!a \mid b.a \sim !a \mid b \qquad\qquad !a.a \sim !a$$

# Bisimilarity in mCCS

- Same laws as in $\mu$CCS (abelian monoid, distribution law);
- Standard laws for replication:

$$!a.F \mid !a.F \sim !a.F \qquad\qquad !a.F \mid a.F \sim !a.F$$

- Other phenomena:

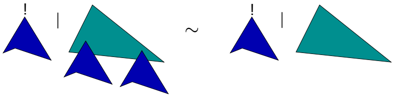
$$!a \mid b.a \sim !a \mid b \qquad\qquad !a.a \sim !a$$

- Note: there are mutual desynchronisations

$$\begin{tikzcd}[column sep=small]
 & & !a.b \mid !b.a \arrow[dll, "a"'] \arrow[drr, "b"] & & \\
{} \arrow[d, "b"'] & & & & {} \arrow[d, "a"'] \\
!a.b \mid !b.a & & \sim & & !a.b \mid !b.a
\end{tikzcd}$$

# Erasing subterms

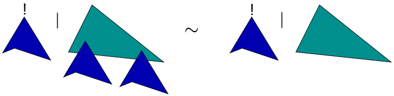
- $!a.F \mid a.F \sim !a.F$ generalises to $!a.F \mid C[a.F] \sim !a.F \mid C[\mathbf{0}]$:




$\rightarrow$ in particular, $!a \mid !b.a \mid c.a \sim !a \mid !b \mid c$.

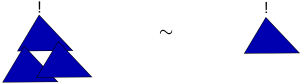
# Erasing subterms

- $!a.F \mid a.F \sim !a.F$ generalises to $!a.F \mid C[a.F] \sim !a.F \mid C[\mathbf{0}]$:



$\rightarrow$ in particular, $!a \mid !b.a \mid c.a \sim !a \mid !b \mid c$.

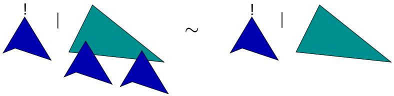
- $!a.a \sim !a$ generalises to $!a.C[a.C[\mathbf{0}]] \sim !a.C[\mathbf{0}]$:



$\rightarrow$ in particular, $!a.(b.a.(b \mid c) \mid c) \sim !a.(b \mid c)$.

# Erasing subterms

- $!a.F \mid a.F \sim\, !a.F$ generalises to $!a.F \mid C[a.F] \sim\, !a.F \mid C[\mathbf{0}]$:



$\rightarrow$ in particular, $!a \mid\, !b.a \mid c.a \sim\, !a \mid\, !b \mid c$.

- $!a.a \sim\, !a$ generalises to $!a.C[a.C[\mathbf{0}]] \sim\, !a.C[\mathbf{0}]$:

$\rightarrow$ in particular, $!a.(b.a.(b \mid c) \mid c) \sim\, !a.(b \mid c)$.

- Simultaneous, mutual erasing: $!a.b \mid\, !b.a \sim\, !a \mid\, !b$

# Axiom schemes?

- $\prod_{i \leq n} !a_i.C_i[C_0[\mathbf{0}], \ldots, C_n[\mathbf{0}]] \sim \prod_{i \leq n} !a_i.C_i[\mathbf{0}, \ldots, \mathbf{0}]$
- $!a.C[a.C[\ldots a.C[\mathbf{0}] \ldots]] \sim !a.C[\mathbf{0}]$
- $+$ combinations of these laws

$\rightarrow$ hard to reason about, not really informative.

# Axiom schemes?

- $\prod_{i \le n} !a_i.C_i[C_0[\mathbf{0}], \dots, C_n[\mathbf{0}]] \sim \prod_{i \le n} !a_i.C_i[\mathbf{0}, \dots, \mathbf{0}]$
- $!a.C[a.C[\dots a.C[\mathbf{0}] \dots]] \sim !a.C[\mathbf{0}]$
- $+$ combinations of these laws

$\rightarrow$ hard to reason about, not really informative.

Lemma: the following inference rule is sound

$$\frac{C[\mathbf{0}] \sim !a.F \mid P}{C[\mathbf{0}] \sim C[a.F]}$$

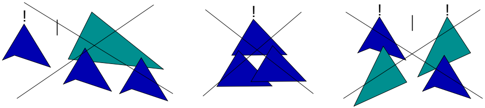
$\rightarrow$ not an equational rule.

# Another approach

- A seed for a process $P$ is a process of minimal size which is bisimilar to $P$.

$$\begin{aligned} !a.a\,,\quad !a \mid !a \quad &\rightsquigarrow \quad !a \\ !a.b \mid !b.a\,,\quad !a \mid !b.a \quad &\rightsquigarrow \quad !a \mid !b \\ !a \mid !b.c \mid d.b.c.a \quad &\rightsquigarrow \quad !a \mid !b.c \mid d \end{aligned}$$

  - any process has a seed;
  - seeds do not contain redundant subterms:

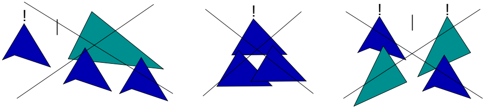


# Another approach

- A seed for a process $P$ is a process of minimal size which is bisimilar to $P$.

$$
\begin{array}{rcl}
!a.a\,, \quad !a \mid !a & \rightsquigarrow & !a \\
!a.b \mid !b.a\,, \quad !a \mid !b.a & \rightsquigarrow & !a \mid !b \\
!a \mid !b.c \mid d.b.c.a & \rightsquigarrow & !a \mid !b.c \mid d
\end{array}
$$

  - any process has a seed;
  - seeds do not contain redundant subterms:




- Uniqueness: if $P \sim Q$ and $P, Q$ are seeds, then $P \equiv Q$.

(difficult, technical proof)

# A rewriting system

- Guess the seed of a process, and use it to clean the process:

(modulo structural congruence and distribution law)

$$P \xrightarrow{S} P' \xrightarrow{S} P'' \xrightarrow{S} \ldots$$

# A rewriting system

- Guess the seed of a process, and use it to clean the process:

(modulo structural congruence and distribution law)

$$\frac{}{C[a.F] \xrightarrow{!a.F|P} C[\mathbf{0}]} \qquad \frac{}{!a.F \mid !a.F \mid P \xrightarrow{Q} !a.F \mid P}$$

# A rewriting system

- Guess the seed of a process, and use it to clean the process:

  (modulo structural congruence and distribution law)

$$\frac{}{C[a.F] \xrightarrow{!a.F|P} C[\mathbf{0}]} \qquad \frac{}{!a.F \mid !a.F \mid P \xrightarrow{Q} !a.F \mid P}$$

- Examples:

$$!a.b \mid !b \mid b.a \xrightarrow{!a|!b} !a.b \mid !b \mid b \xrightarrow{!a|!b} !a.b \mid !b \xrightarrow{!a|!b} !a \mid !b$$

$$!a.(b \mid a.b) \xrightarrow{!a.b} !a.b$$

$$!a.b \mid !b.a \xrightarrow{!a|!b} !a.b \mid !b \xrightarrow{!a|!b} !a \mid !b$$

# A rewriting system

- Guess the seed of a process, and use it to clean the process:

  (modulo structural congruence and distribution law)

$$\frac{}{C[a.F] \xrightarrow{!a.F|P} C[\mathbf{0}]} \qquad \frac{}{!a.F \mid !a.F \mid P \xrightarrow{Q} !a.F \mid P}$$

- Examples:

$$!a.b \mid !b \mid b.a \xrightarrow{!a|!b} !a.b \mid !b \mid b \xrightarrow{!a|!b} !a.b \mid !b \xrightarrow{!a|!b} !a \mid !b$$

$$!a.(b \mid a.b) \xrightarrow{!a.b} !a.b$$

$$!a.b \mid !b.a \xrightarrow{!a|!b} !a.b \mid !b \xrightarrow{!a|!b} !a \mid !b$$

$$!a \mid !a.b \xrightarrow{!a|!b} !a \mid !a \xrightarrow{!a|!b} !a$$

# A rewriting system, cont.

- Correctness: if $P \xrightarrow{S}^{\star} S$, then $P \sim S$. (easy)
- Completeness: $P \xrightarrow{\mathsf{seed}(P)}^{\star} \mathsf{seed}(P)$. (technical)

# A rewriting system, cont.

- Correctness: if $P \xrightarrow{S}^{\star} S$, then $P \sim S$. (easy)
- Completeness: $P \xrightarrow{\mathsf{seed}(P)}^{\star} \mathsf{seed}(P)$. (technical)

- Consequences:
  - $P \sim Q$ iff $P \xrightarrow{S}^{\star} S$ and $Q \xrightarrow{S}^{\star} S$ for some $S$;
  - $\sim$ is decidable [Christensen, Hirshfeld & Moller 1994].
  - $\sim$ is substitution closed, and hence, a congruence (in $\pi$);

# Conclusions

- Bisimilarity is a congruence in:
  - finite $\pi$ [FoSSaCS 2007];
  - public $\pi$ with top-level replications [ICALP 2010].

- Methodology:
  characterise bisimilarity in sum-free fragments of CCS/$\pi$
  - equational axiomatisations
  - rewriting systems
  - transfer property
  - seeds (minimal processes)

# Future work

- Richer fragments with replication
  - "deep" replications $a.(F \mid !a.F) \sim !a.F$
  - "nested" replications $!a.!b \sim a.(!a \mid !b)$
  - name restriction but prefixed replication $\cancel{!(\nu a)}$
- Weak bisimilarity?

$$!\overline{a}.a \mid a.b \;\approx\; !\overline{a}.a \mid a \mid b$$

$$!\overline{a} \mid !a.b \;\approx\; !\overline{a} \mid !a \mid !b$$

# Future work

- Richer fragments with replication
  - "deep" replications $a.(F \mid !a.F) \sim !a.F$
  - "nested" replications $!a.!b \sim a.(!a \mid !b)$
  - name restriction but prefixed replication $\cancel{!(\nu a)}$

- Weak bisimilarity?

$$!\overline{a}.a \mid a.b \;\approx\; !\overline{a}.a \mid a \mid b$$

$$!\overline{a} \mid !a.b \;\approx\; !\overline{a} \mid !a \mid !b$$

Thanks!