

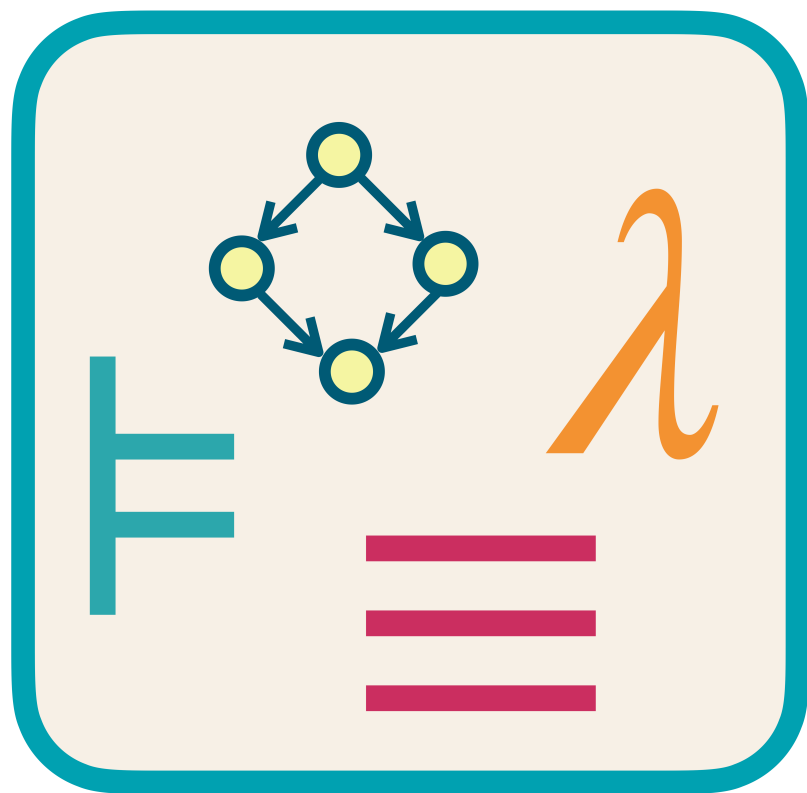
MPP 2025/26 (0077A, 9CFU)

Models for Programming Paradigms

Roberto Bruni

Filippo Bonchi

<http://www.di.unipi.it/~bruni/>



<https://didawiki.di.unipi.it/doku.php/magistraleinformatica/mpp/start>

18b - CCS strong bisimulation

CCS

Strong bisimulation

CCS syntax

p, q	$::=$	nil	inactive process
		x	process variable (for recursion)
		$\mu.p$	action prefix
		$p \backslash \alpha$	restricted channel
		$p[\phi]$	channel relabelling
		$p + q$	nondeterministic choice (sum)
		$p q$	parallel composition
		rec $x. p$	recursion

(operators are listed in order of precedence)

CCS op. semantics

$$\begin{array}{c}
 \text{Act)} \frac{}{\mu.p \xrightarrow{\mu} p} \qquad \text{Res)} \frac{p \xrightarrow{\mu} q \quad \mu \notin \{\alpha, \bar{\alpha}\}}{p \setminus \alpha \xrightarrow{\mu} q \setminus \alpha} \qquad \text{Rel)} \frac{p \xrightarrow{\mu} q}{p[\phi] \xrightarrow{\phi(\mu)} q[\phi]} \\
 \\
 \text{SumL)} \frac{p_1 \xrightarrow{\mu} q}{p_1 + p_2 \xrightarrow{\mu} q} \qquad \text{SumR)} \frac{p_2 \xrightarrow{\mu} q}{p_1 + p_2 \xrightarrow{\mu} q} \\
 \\
 \text{ParL)} \frac{p_1 \xrightarrow{\mu} q_1}{p_1 | p_2 \xrightarrow{\mu} q_1 | p_2} \qquad \text{Com)} \frac{p_1 \xrightarrow{\lambda} q_1 \quad p_2 \xrightarrow{\bar{\lambda}} q_2}{p_1 | p_2 \xrightarrow{\tau} q_1 | q_2} \qquad \text{ParR)} \frac{p_2 \xrightarrow{\mu} q_2}{p_1 | p_2 \xrightarrow{\mu} p_1 | q_2} \\
 \\
 \text{Rec)} \frac{p[\mathbf{rec} \ x. \ p / x] \xrightarrow{\mu} q}{\mathbf{rec} \ x. \ p \xrightarrow{\mu} q}
 \end{array}$$

Strong bisimulation

the notion of **bisimulation** is not restricted to CCS processes
it applies to any LTS

in the following we recall Milner's original definition of
strong bisimulation relation

to keep in mind

there are many strong bisimulation relations

we are interested in the largest such relation,
called *strong bisimilarity*

to prove that two processes are strong bisimilar
it is enough to show they are related by a strong bisimulation

Strong bisimulation

$\mathcal{P}$ set of processes

$\mathbf{R} \subseteq \mathcal{P} \times \mathcal{P}$ a binary relation

we write $p \mathbf{R} q$ when $(p, q) \in \mathbf{R}$

$\mathbf{R}$ is a strong bisimulation if

$$\forall p, q. (p, q) \in \mathbf{R} \Rightarrow \begin{cases} \forall \mu, p'. p \xrightarrow{\mu} p' \Rightarrow \exists q'. q \xrightarrow{\mu} q' \wedge p' \mathbf{R} q' \\ \wedge \\ \forall \mu, q'. q \xrightarrow{\mu} q' \Rightarrow \exists p'. p \xrightarrow{\mu} p' \wedge p' \mathbf{R} q' \end{cases}$$

intuitively: if two processes are related, then for any move of Alice, Bob can find a move that leads to related processes i.e., Bob has a winning strategy

Example

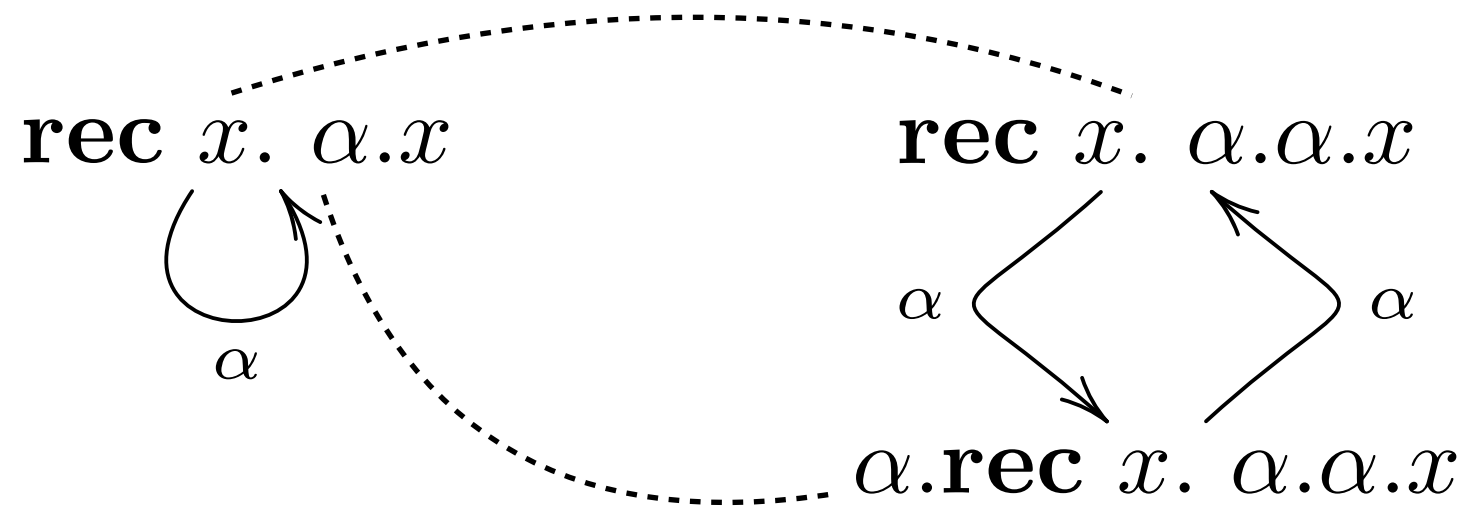
$\emptyset$ is a strong bisimulation

$Id \triangleq \{(p, p) \mid p \in \mathcal{P}\}$ is a strong bisimulation

any graph isomorphism defines a strong bisimulation

$$\mathbf{R}_f \triangleq \{(p, f(p))\}$$

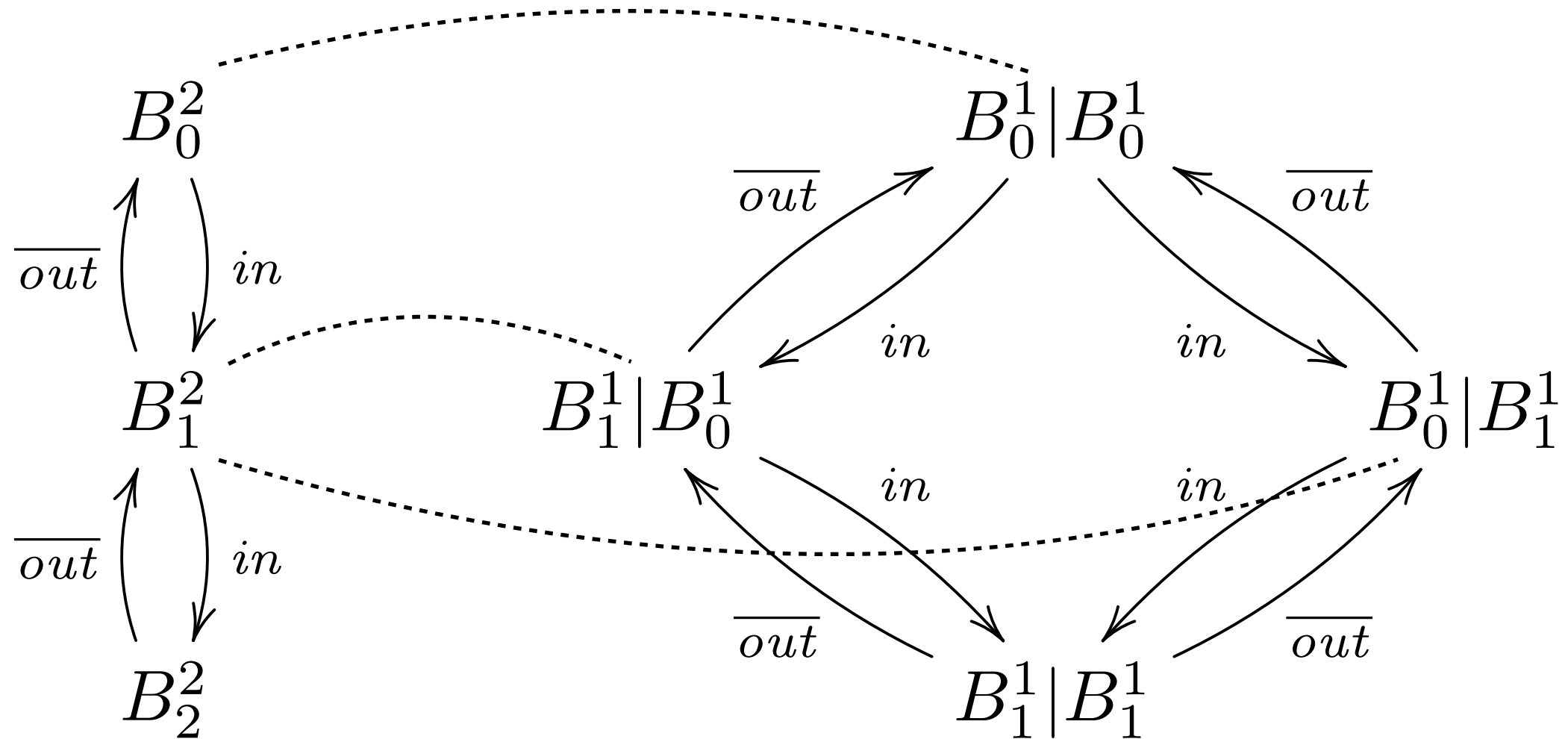
Example



$$\mathbf{R} \triangleq \left\{ \begin{array}{l} (\mathbf{rec} \ x. \ \alpha.x, \mathbf{rec} \ x. \ \alpha.\alpha.x), \\ (\mathbf{rec} \ x. \ \alpha.x, \alpha.\mathbf{rec} \ x. \ \alpha.\alpha.x) \end{array} \right\}$$

unlike graph isomorphisms,
the same process can be related to many processes

Example



$$\mathbf{R} \triangleq \left\{ \begin{array}{l} (B_0^2, B_0^1 | B_0^1), \\ (B_1^2, B_1^1 | B_0^1), \\ (B_1^2, B_0^1 | B_1^1), \\ (B_2^2, B_1^1 | B_1^1) \end{array} \right\}$$

Union

Lemma If $\mathbf{R}_1$ and $\mathbf{R}_2$ are strong bisimulations, then $\mathbf{R}_1 \cup \mathbf{R}_2$ is a strong bisimulation

proof. take $(p, q) \in \mathbf{R}_1 \cup \mathbf{R}_2$

take $p \xrightarrow{\mu} p'$ we want to find $q \xrightarrow{\mu} q'$ with $(p', q') \in \mathbf{R}_1 \cup \mathbf{R}_2$

since $(p, q) \in \mathbf{R}_1 \cup \mathbf{R}_2$ we have $p \mathbf{R}_i q$ for some $i \in \{1, 2\}$

since $\mathbf{R}_i$ is a strong bisimulation and $p \xrightarrow{\mu} p'$

we have $q \xrightarrow{\mu} q'$ with $p' \mathbf{R}_i q'$ and hence $(p', q') \in \mathbf{R}_1 \cup \mathbf{R}_2$

take $q \xrightarrow{\mu} q'$ we want to find $p \xrightarrow{\mu} p'$ with $(p', q') \in \mathbf{R}_1 \cup \mathbf{R}_2$

analogous to the previous case

Inverse

Lemma If $\mathbf{R}$ is a strong bisimulation,
then $\mathbf{R}^{-1} \triangleq \{(q, p) \mid p \mathbf{R} q\}$ is a strong bisimulation

proof. take $(q, p) \in \mathbf{R}^{-1}$

take $q \xrightarrow{\mu} q'$ we want to find $p \xrightarrow{\mu} p'$ with $(q', p') \in \mathbf{R}^{-1}$

since $(q, p) \in \mathbf{R}^{-1}$ we have $p \mathbf{R} q$

since $\mathbf{R}$ is a strong bisimulation and $q \xrightarrow{\mu} q'$

we have $p \xrightarrow{\mu} p'$ with $p' \mathbf{R} q'$ and hence $(q', p') \in \mathbf{R}^{-1}$

take $p \xrightarrow{\mu} p'$ we want to find $q \xrightarrow{\mu} q'$ with $(q', p') \in \mathbf{R}^{-1}$

analogous to the previous case

Composition

Lemma If $\mathbf{R}_1$ and $\mathbf{R}_2$ are strong bisimulations,
then $\mathbf{R}_2 \circ \mathbf{R}_1 \triangleq \{(p, q) \mid \exists r. p \mathbf{R}_1 r \wedge r \mathbf{R}_2 q\}$
is a strong bisimulation

proof. take $(p, q) \in \mathbf{R}_2 \circ \mathbf{R}_1$

take $p \xrightarrow{\mu} p'$ we want to find $q \xrightarrow{\mu} q'$ with $(p', q') \in \mathbf{R}_2 \circ \mathbf{R}_1$

since $(p, q) \in \mathbf{R}_2 \circ \mathbf{R}_1$ we have $p \mathbf{R}_1 r \wedge r \mathbf{R}_2 q$ for some r

since $\mathbf{R}_1$ is a strong bisimulation and $p \xrightarrow{\mu} p'$

we have $r \xrightarrow{\mu} r'$ with $p' \mathbf{R}_1 r'$

since $\mathbf{R}_2$ is a strong bisimulation and $r \xrightarrow{\mu} r'$

we have $q \xrightarrow{\mu} q'$ with $r' \mathbf{R}_2 q'$ and hence $(p', q') \in \mathbf{R}_2 \circ \mathbf{R}_1$

take $q \xrightarrow{\mu} q'$ we want to find $p \xrightarrow{\mu} p'$ with $(p', q') \in \mathbf{R}_2 \circ \mathbf{R}_1$

analogous to the previous case

Notation

$$\mathbf{R}_2 \circ \mathbf{R}_1 \triangleq \{(p, q) \mid \exists r. p \mathbf{R}_1 r \wedge r \mathbf{R}_2 q\}$$

sometimes written

$$\mathbf{R}_1 \mathbf{R}_2$$

CCS

Strong bisimilarity

Strong bisimilarity

$\simeq$ often denoted $\sim$ in the literature
we use $\simeq$ to remark it is a congruence relation

$p \simeq q$ iff $\exists \mathbf{R}$ a strong bisimulation with $(p, q) \in \mathbf{R}$

i.e. Bob has a winning strategy

i.e. $\simeq \triangleq \bigcup_{\mathbf{R} \text{ s.b.}} \mathbf{R}$

a strong bisimulation is not necessarily an equivalence
is strong bisimilarity an equivalence relation?

Equivalence relation

≡

Reflexive

$$\forall p \in \mathcal{P}$$

$$p \equiv p$$

Symmetric

$$\forall p, q \in \mathcal{P}$$

$$p \equiv q \Rightarrow q \equiv p$$

Transitive

$$\forall p, q, r \in \mathcal{P}$$

$$p \equiv q \wedge q \equiv r \Rightarrow p \equiv r$$

Induced equivalence

Any relation R induces an equivalence relation $\equiv_R$

$\equiv_R$ is the smallest equivalence that contains R

$$\frac{p \mathbf{R} q}{p \equiv_R q}$$

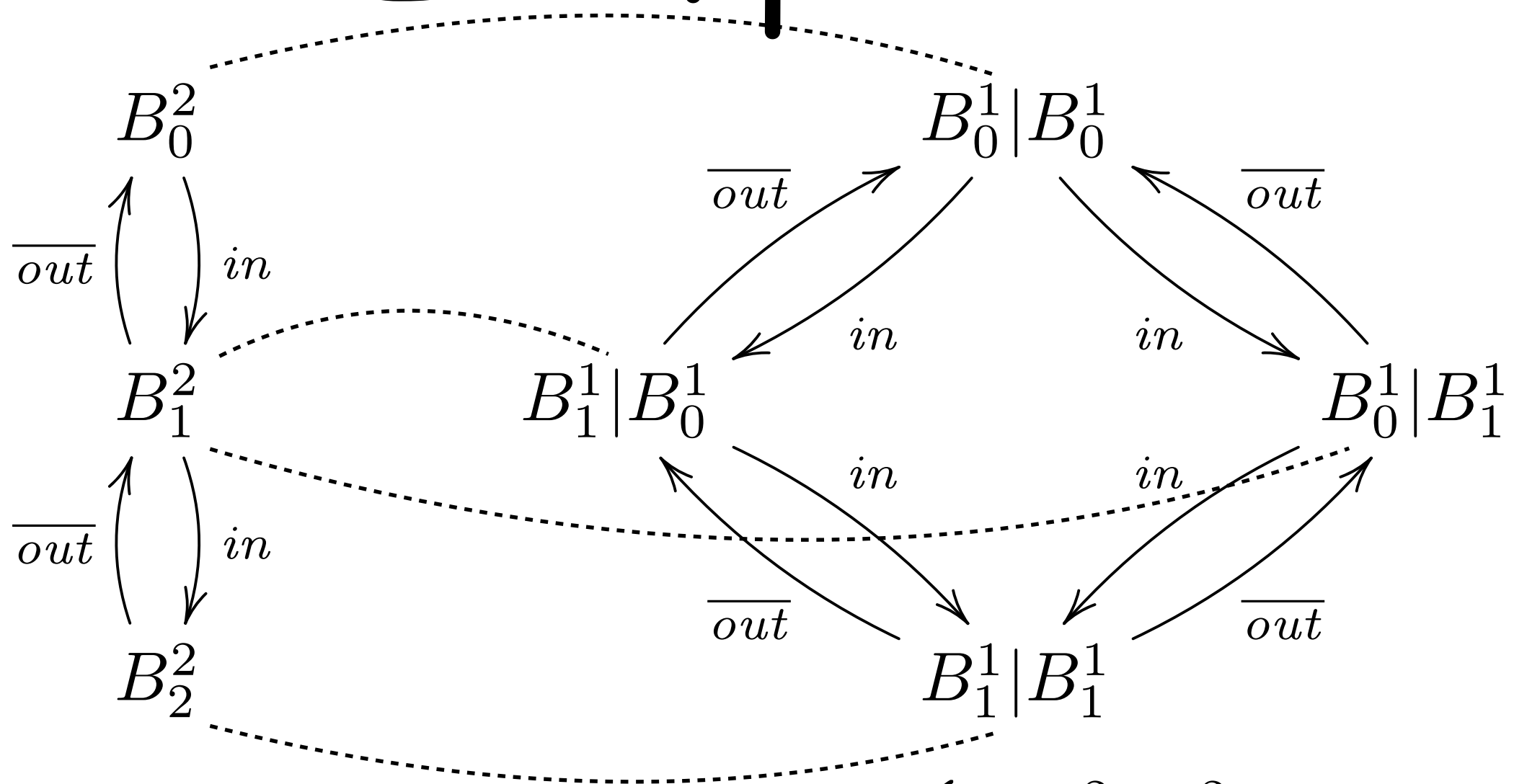
$$\frac{}{p \equiv_R p}$$

$$\frac{p \equiv_R q}{q \equiv_R p}$$

$$\frac{p \equiv_R q \quad q \equiv_R r}{p \equiv_R r}$$

Lemma if R is a strong bisimulation,
then $\equiv_R$ is a strong bisimulation

Example



$$\mathbf{R} \triangleq \left\{ \begin{array}{l} (B_0^2, B_0^1|B_0^1), \\ (B_1^2, B_1^1|B_0^1), \\ (B_1^2, B_0^1|B_1^1), \\ (B_2^2, B_1^1|B_1^1) \end{array} \right\} \quad \equiv_{\mathbf{R}} \triangleq \left\{ \begin{array}{l} (B_0^2, B_0^2), \\ (B_0^2, B_0^1|B_0^1), \\ (B_0^1|B_0^1, B_0^2), \\ (B_0^1|B_0^1, B_0^1|B_0^1), \\ (B_1^2, B_1^2), \\ \dots \end{array} \right\}$$

Induced partition

Any equivalence relation induces a partition of processes into equivalence classes

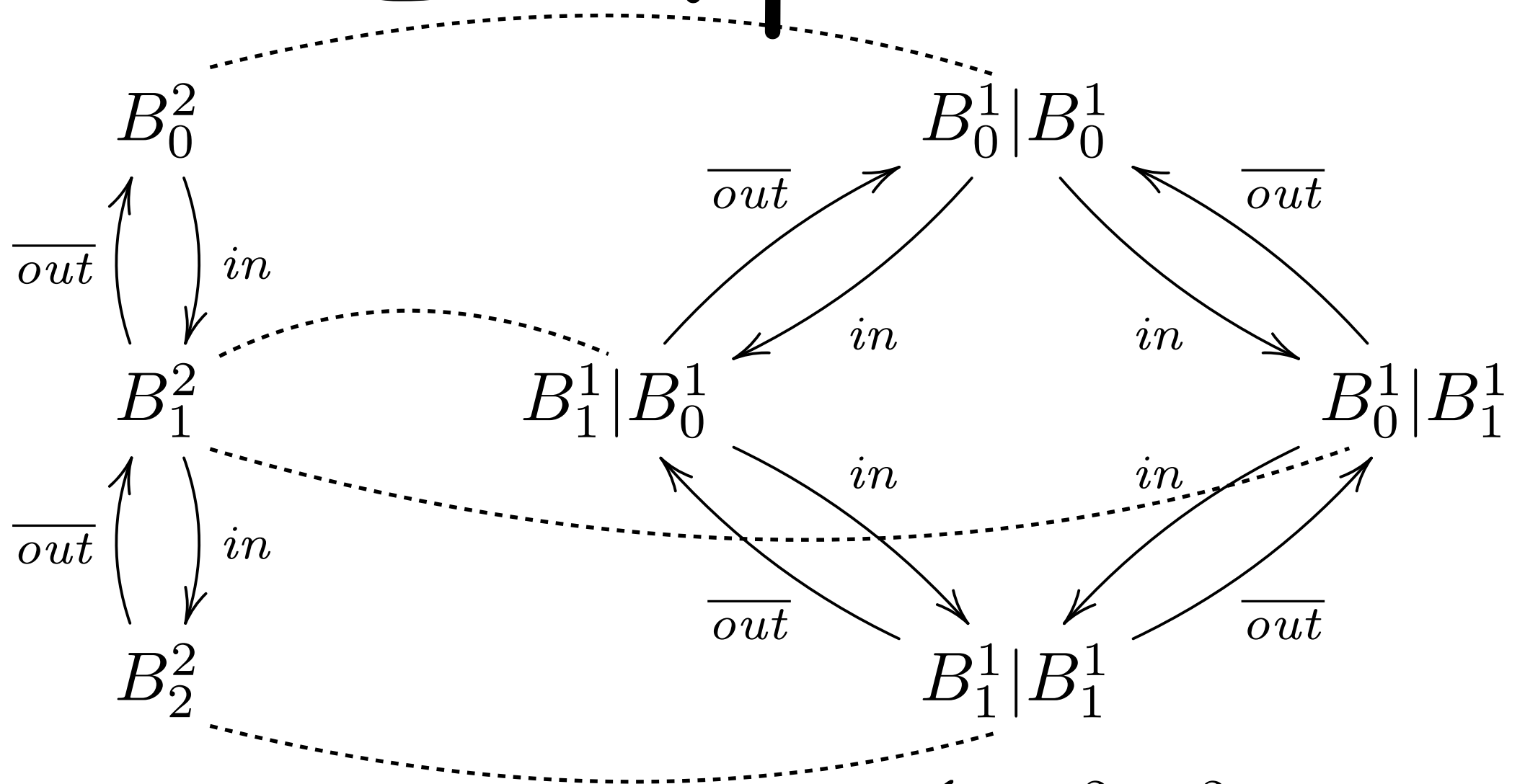
$$[p]_{\equiv} = \{q \mid p \equiv q\}$$

if $\equiv_{\mathbf{R}}$ is a strong bisimulation

$$q \in [p]_{\equiv_{\mathbf{R}}} \wedge p \xrightarrow{\mu} p' \Rightarrow \exists q' \in [p']_{\equiv_{\mathbf{R}}} \cdot q \xrightarrow{\mu} q'$$

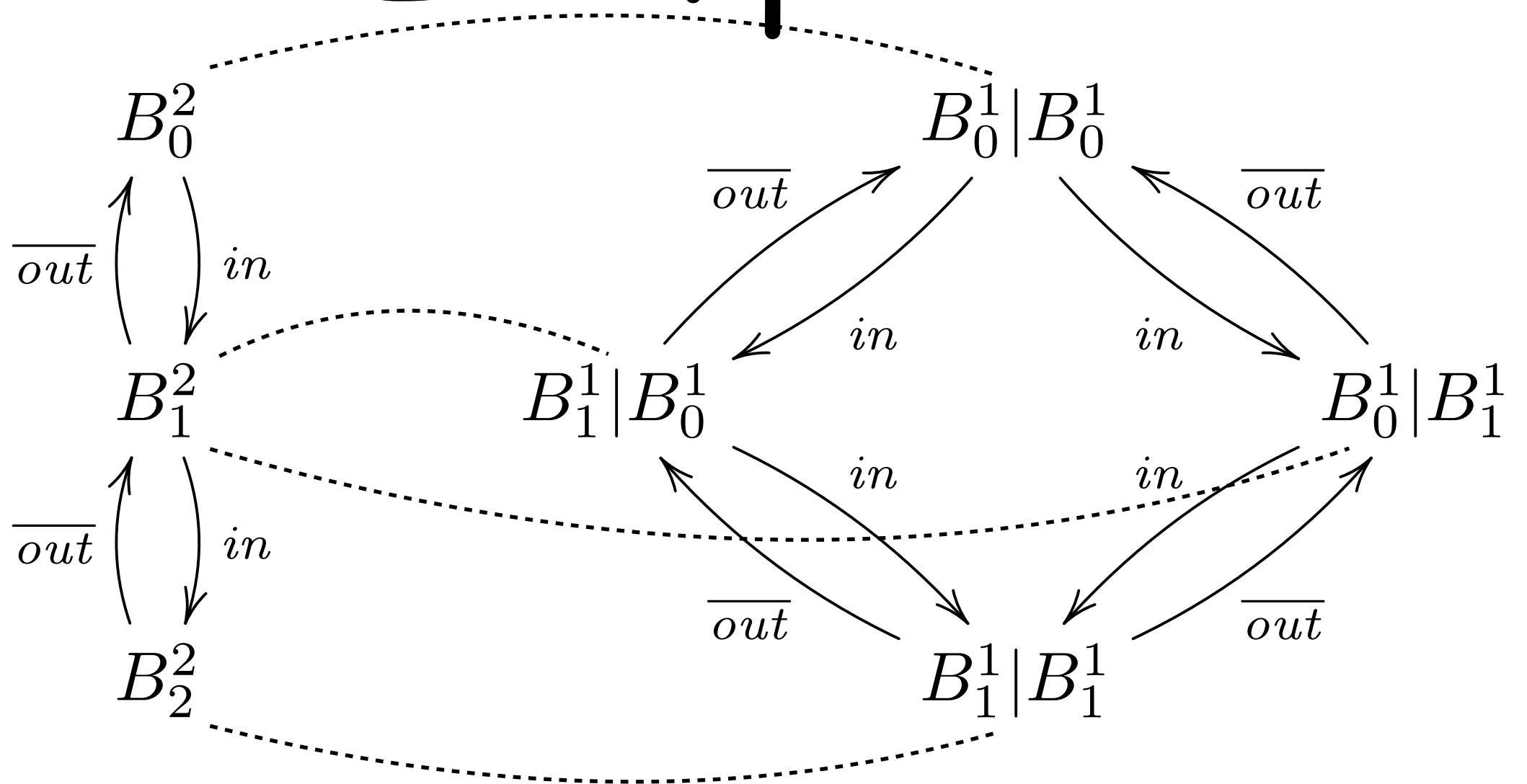
instead of listing all pairs of $\equiv_{\mathbf{R}}$
we list only its equivalence classes

Example



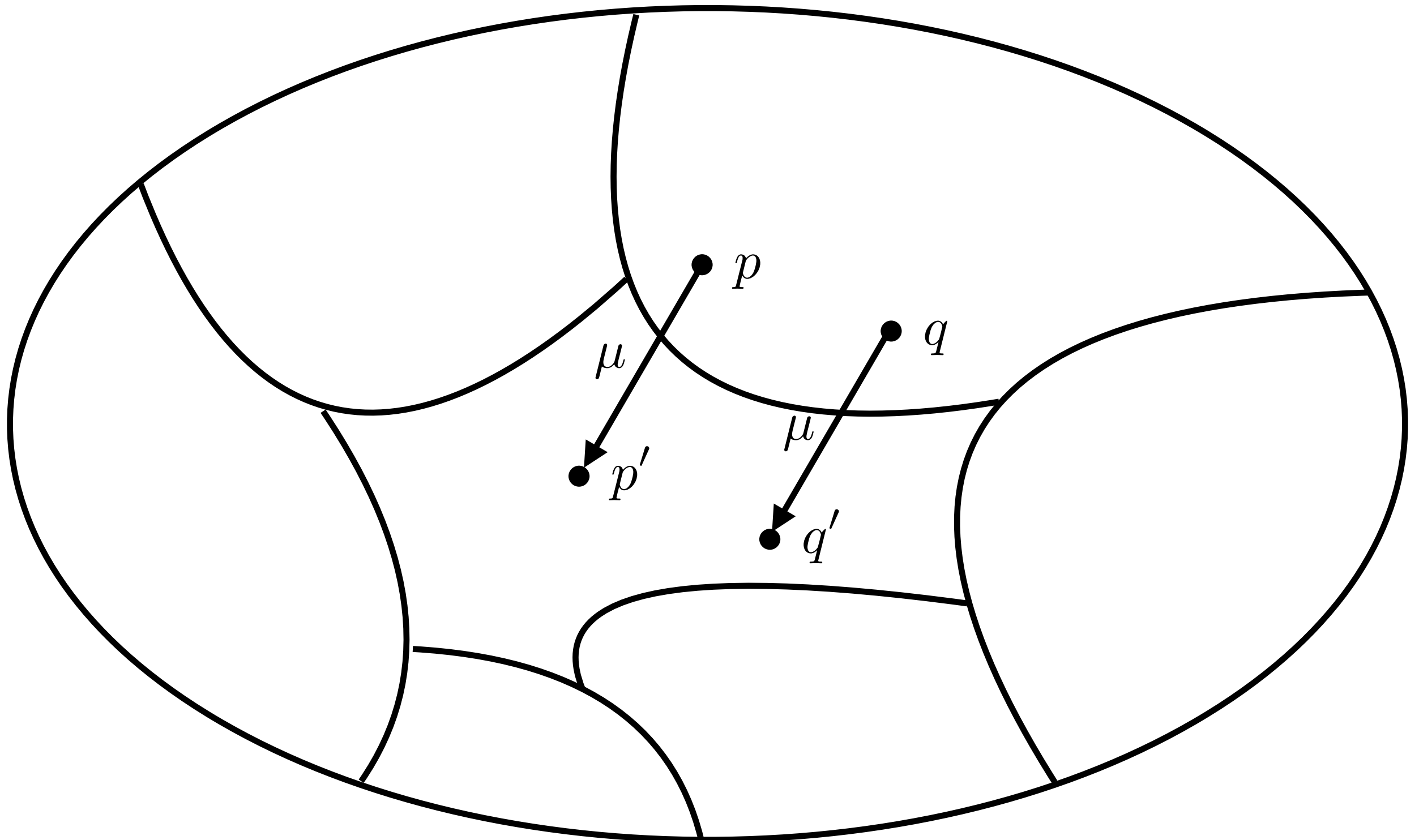
$$\mathbf{R} \triangleq \left\{ \begin{array}{l} (B_0^2, B_0^1|B_0^1), \\ (B_1^2, B_1^1|B_0^1), \\ (B_1^2, B_0^1|B_1^1), \\ (B_2^2, B_1^1|B_1^1) \end{array} \right\} \quad \equiv_{\mathbf{R}} \triangleq \left\{ \begin{array}{l} (B_0^2, B_0^2), \\ (B_0^2, B_0^1|B_0^1), \\ (B_0^1|B_0^1, B_0^2), \\ (B_0^1|B_0^1, B_0^1|B_0^1), \\ (B_1^2, B_1^2), \\ \dots \end{array} \right\}$$

Example

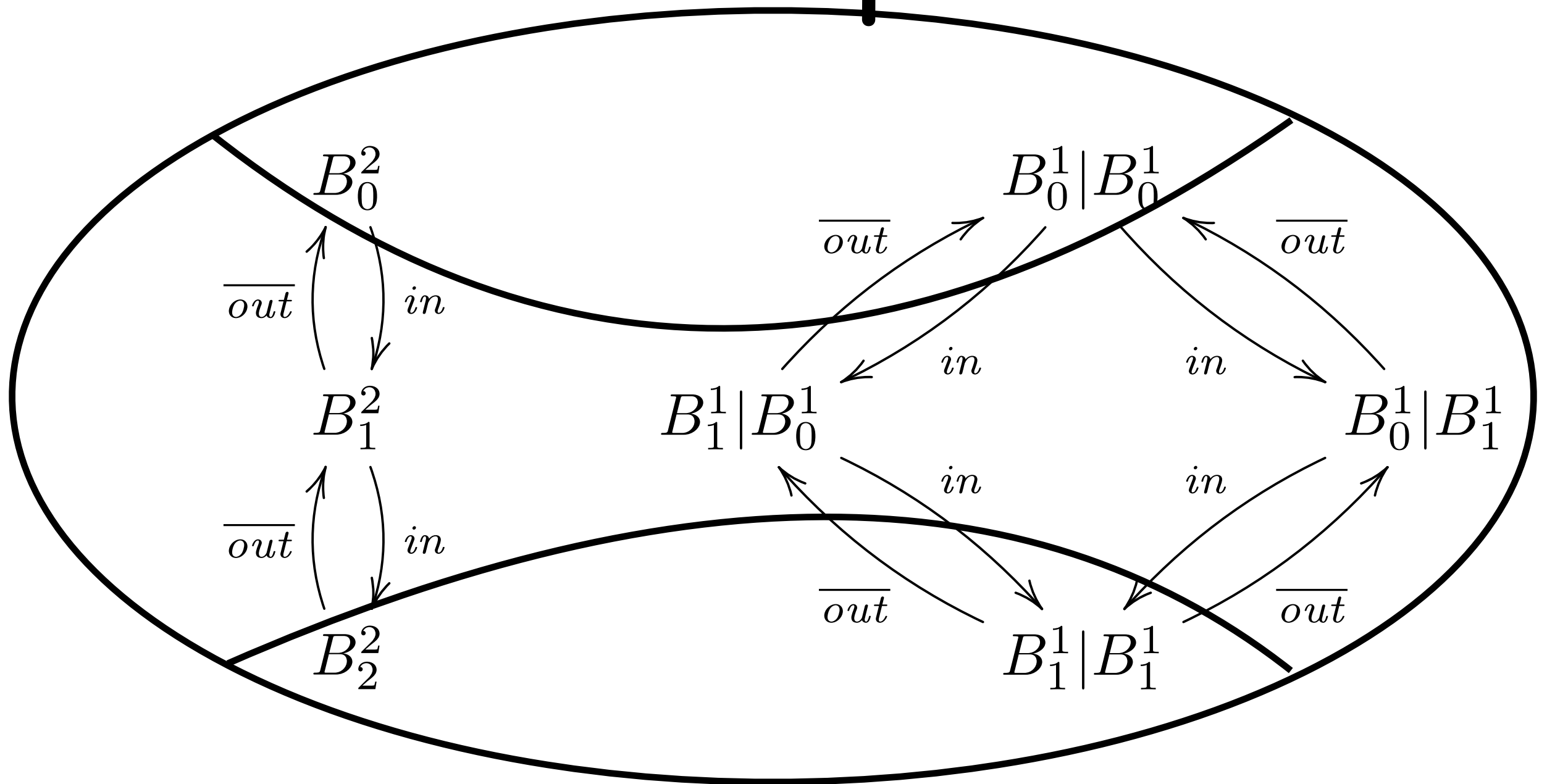


$$\mathbf{R} \triangleq \left\{ \begin{array}{l} (B_0^2, B_0^1 | B_0^1), \\ (B_1^2, B_1^1 | B_0^1), \\ (B_1^2, B_0^1 | B_1^1), \\ (B_2^2, B_1^1 | B_1^1) \end{array} \right\} \quad \equiv_{\mathbf{R}} \triangleq \left\{ \begin{array}{l} \{B_0^2, B_0^1 | B_0^1\}, \\ \{B_1^2, B_0^1 | B_1^1, B_1^1 | B_0^1\}, \\ \{B_2^2, B_1^1 | B_1^1\} \end{array} \right\}$$

Bisimulation check



Example



$$\equiv_{\mathbf{R}} \triangleq \left\{ \begin{array}{l} \{B_0^2, B_0^1 | B_0^1\}, \\ \{B_1^2, B_0^1 | B_1^1, B_1^1 | B_0^1\}, \\ \{B_2^2, B_1^1 | B_1^1\} \end{array} \right\}$$

TH. Strong bisimilarity is an equivalence relation

proof.

reflexive $Id \subseteq \simeq$

symmetric assume $p \simeq q$ we want to prove $q \simeq p$

$p \simeq q$ means there is a s.b. $\mathbf{R}$ with $(p, q) \in \mathbf{R}$

then $(q, p) \in \mathbf{R}^{-1}$ and $\mathbf{R}^{-1}$ is a s.b.

thus $(q, p) \in \mathbf{R}^{-1} \subseteq \simeq$ i.e. $q \simeq p$

transitive assume $p \simeq q$ $q \simeq r$ we want to prove $p \simeq r$

$p \simeq q$ means there is a s.b. $\mathbf{R}_1$ with $(p, q) \in \mathbf{R}_1$

$q \simeq r$ means there is a s.b. $\mathbf{R}_2$ with $(q, r) \in \mathbf{R}_2$

then $(p, r) \in \mathbf{R}_2 \circ \mathbf{R}_1$ and $\mathbf{R}_2 \circ \mathbf{R}_1$ is a s.b.

thus $(p, r) \in \mathbf{R}_2 \circ \mathbf{R}_1 \subseteq \simeq$ i.e. $p \simeq r$

TH. Strong bisimilarity is a strong bisimulation

proof.

take $p \simeq q$

take $p \xrightarrow{\mu} p'$ we want to find $q \xrightarrow{\mu} q'$ with $p' \simeq q'$

$p \simeq q$ means there is a s.b. $\mathbf{R}$ with $(p, q) \in \mathbf{R}$

since $\mathbf{R}$ is a strong bisimulation and $p \xrightarrow{\mu} p'$

we have $q \xrightarrow{\mu} q'$ with $(p', q') \in \mathbf{R}$

since $\mathbf{R} \subseteq \simeq$ we have $p' \simeq q'$

take $q \xrightarrow{\mu} q'$ we want to find $p \xrightarrow{\mu} p'$ with $p' \simeq q'$

follows from previous case (strong bisimilarity is symmetric)

Cor. Strong bisimilarity is the **largest** strong bisimulation

proof.

strong bisimilarity is a strong bisimulation (previous TH.)

by definition

$$\simeq \triangleq \bigcup_{\mathbf{R} \text{ s.b.}} \mathbf{R}$$

any other strong bisimulation is included in $\simeq$

TH. Recursive definition of strong bisimilarity

$$\forall p, q. p \simeq q \Leftrightarrow \begin{cases} \forall \mu, p'. p \xrightarrow{\mu} p' \Rightarrow \exists q'. q \xrightarrow{\mu} q' \wedge p' \simeq q' \\ \wedge \\ \forall \mu, q'. q \xrightarrow{\mu} q' \Rightarrow \exists p'. p \xrightarrow{\mu} p' \wedge p' \simeq q' \end{cases}$$

proof.

$\Rightarrow$) follows immediately because $\simeq$ is a strong bisimulation

$\Leftarrow$) take p, q s.t. $\begin{cases} \forall \mu, p'. p \xrightarrow{\mu} p' \Rightarrow \exists q'. q \xrightarrow{\mu} q' \wedge p' \simeq q' \\ \wedge \\ \forall \mu, q'. q \xrightarrow{\mu} q' \Rightarrow \exists p'. p \xrightarrow{\mu} p' \wedge p' \simeq q' \end{cases}$

we want to prove $p \simeq q$

this is done by proving that $\mathbf{R} \triangleq \{(p, q)\} \cup \simeq$ is a s.b.

(see next slide)

TH. Recursive definition of strong bisimilarity (continue)

$\mathbf{R} \triangleq \{(p, q)\} \cup \simeq$ is a s.b.

take $(r, s) \in \mathbf{R}$

take $r \xrightarrow{\mu} r'$ we want to find $s \xrightarrow{\mu} s'$ with $(r', s') \in \mathbf{R}$

if $r \simeq s$ then we can find $s \xrightarrow{\mu} s'$ with $(r', s') \in \simeq \subseteq \mathbf{R}$

because $\simeq$ is a strong bisimulation

if $(r, s) = (p, q)$ then $p \xrightarrow{\mu} r'$ and $\begin{cases} \forall \mu, p'. p \xrightarrow{\mu} p' \Rightarrow \exists q'. q \xrightarrow{\mu} q' \wedge p' \simeq q' \\ \wedge \\ \forall \mu, q'. q \xrightarrow{\mu} q' \Rightarrow \exists p'. p \xrightarrow{\mu} p' \wedge p' \simeq q' \end{cases}$

thus we can find $q \xrightarrow{\mu} s'$ with $(r', s') \in \simeq \subseteq \mathbf{R}$

take $s \xrightarrow{\mu} s'$ we want to find $r \xrightarrow{\mu} r'$ with $(r', s') \in \mathbf{R}$

analogous to the previous case

CCS

Bisimilarity as a fixpoint

Strong bis as fix

$$\forall p, q. (p, q) \in \mathbf{R} \Rightarrow \left\{ \begin{array}{l} \forall \mu, p'. p \xrightarrow{\mu} p' \Rightarrow \exists q'. q \xrightarrow{\mu} q' \wedge p' \mathbf{R} q' \\ \wedge \\ \forall \mu, q'. q \xrightarrow{\mu} q' \Rightarrow \exists p'. p \xrightarrow{\mu} p' \wedge p' \mathbf{R} q' \end{array} \right.$$

$\Phi : \wp(\mathcal{P} \times \mathcal{P}) \rightarrow \wp(\mathcal{P} \times \mathcal{P})$ maps relations to relations

$$\Phi(\mathbf{R}) \triangleq \left\{ (p, q) \left| \begin{array}{l} \forall \mu, p'. p \xrightarrow{\mu} p' \Rightarrow \exists q'. q \xrightarrow{\mu} q' \wedge p' \mathbf{R} q' \\ \wedge \\ \forall \mu, q'. q \xrightarrow{\mu} q' \Rightarrow \exists p'. p \xrightarrow{\mu} p' \wedge p' \mathbf{R} q' \end{array} \right. \right\}$$

$$\mathbf{R} \subseteq \Phi(\mathbf{R})$$

a strong bisimulation

Strong bis as fix

$$\forall p, q. p \simeq q \Leftrightarrow \left\{ \begin{array}{l} \forall \mu, p'. p \xrightarrow{\mu} p' \Rightarrow \exists q'. q \xrightarrow{\mu} q' \wedge p' \simeq q' \\ \wedge \\ \forall \mu, q'. q \xrightarrow{\mu} q' \Rightarrow \exists p'. p \xrightarrow{\mu} p' \wedge p' \simeq q' \end{array} \right.$$

$\Phi : \wp(\mathcal{P} \times \mathcal{P}) \rightarrow \wp(\mathcal{P} \times \mathcal{P})$ maps relations to relations

$$\Phi(\mathbf{R}) \triangleq \left\{ (p, q) \left| \begin{array}{l} \forall \mu, p'. p \xrightarrow{\mu} p' \Rightarrow \exists q'. q \xrightarrow{\mu} q' \wedge p' \mathbf{R} q' \\ \wedge \\ \forall \mu, q'. q \xrightarrow{\mu} q' \Rightarrow \exists p'. p \xrightarrow{\mu} p' \wedge p' \mathbf{R} q' \end{array} \right. \right\}$$

$$\simeq = \Phi(\simeq)$$

strong bisimilarity is a fixpoint

Fixpoint: which CPO?

Can we reuse Kleene's fix point theorem?

we want to find the **coarsest** relation,
not the **least** relation

Idea: reverse the usual order (inclusion)!

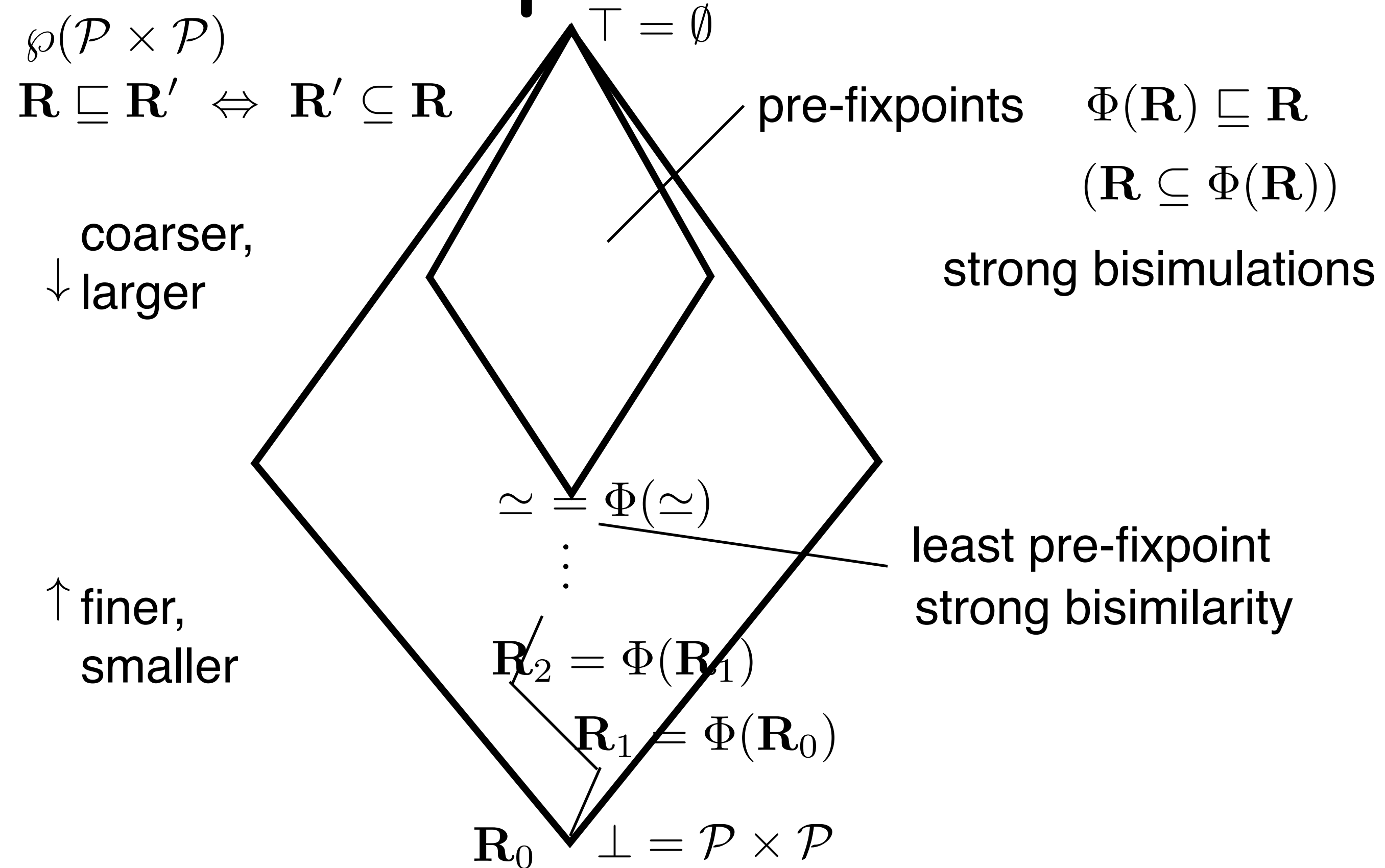
a relation with more pairs is
smaller than one with less pairs

$$(\wp(\mathcal{P} \times \mathcal{P}), \sqsubseteq)$$

$$\mathbf{R} \sqsubseteq \mathbf{R}' \quad \Leftrightarrow \quad \mathbf{R}' \subseteq \mathbf{R}$$

$$\perp = \mathcal{P} \times \mathcal{P}$$

Least fixpoint... reversed



Computing fixpoints

can we reuse Kleene's fix point theorem to compute $\simeq$?

$$\simeq \stackrel{?}{=} \bigsqcup \Phi^n(\mathcal{P} \times \mathcal{P})$$

intersection

start from the universal relation (all pairs, a unique partition)

all processes are equivalent

we apply Φ to distinguish more and more processes

$\mathbf{R}_1$ distinguishable in one step

$\mathbf{R}_2$ distinguishable in two steps

$\vdots$

the number of partitions increases at each step

TH. Φ is monotone

proof.

take $\mathbf{R}_1 \sqsubseteq \mathbf{R}_2$ we need to prove $\Phi(\mathbf{R}_1) \sqsubseteq \Phi(\mathbf{R}_2)$

$$\mathbf{R}_2 \subseteq \mathbf{R}_1 \qquad \Phi(\mathbf{R}_2) \subseteq \Phi(\mathbf{R}_1)$$

take $(p, q) \in \Phi(\mathbf{R}_2)$ we need to prove $(p, q) \in \Phi(\mathbf{R}_1)$

take $p \xrightarrow{\mu} p'$ we want to find $q \xrightarrow{\mu} q'$ with $(p', q') \in \mathbf{R}_1$

since $(p, q) \in \Phi(\mathbf{R}_2)$ we have $q \xrightarrow{\mu} q'$ with $(p', q') \in \mathbf{R}_2 \subseteq \mathbf{R}_1$

take $q \xrightarrow{\mu} q'$ we want to find $p \xrightarrow{\mu} p'$ with $(p', q') \in \mathbf{R}_1$

analogous to the previous case

hence $(p, q) \in \Phi(\mathbf{R}_1)$

TH. Φ is continuous (for finitely branching processes)

proof.

take a chain $\{\mathbf{R}_n\}_{n \in \mathbb{N}}$

$$\mathbf{R}_0 \sqsubseteq \mathbf{R}_1 \sqsubseteq \dots \sqsubseteq \mathbf{R}_n \sqsubseteq \dots$$

$$\mathbf{R}_0 \supseteq \mathbf{R}_1 \supseteq \dots \supseteq \mathbf{R}_n \supseteq \dots$$

we need to prove $\Phi \left(\bigsqcup_{n \in \mathbb{N}} \mathbf{R}_n \right) = \bigsqcup_{n \in \mathbb{N}} \Phi(\mathbf{R}_n)$

$$\Phi \left(\bigsqcup_{n \in \mathbb{N}} \mathbf{R}_n \right) \supseteq \bigsqcup_{n \in \mathbb{N}} \Phi(\mathbf{R}_n)$$

follows from monotonicity

$$\Phi \left(\bigsqcup_{n \in \mathbb{N}} \mathbf{R}_n \right) \sqsubseteq \bigsqcup_{n \in \mathbb{N}} \Phi(\mathbf{R}_n)$$

$$\Phi \left(\bigcap_{n \in \mathbb{N}} \mathbf{R}_n \right) \supseteq \bigcap_{n \in \mathbb{N}} \Phi(\mathbf{R}_n)$$

take $(p, q) \in \bigcap_{n \in \mathbb{N}} \Phi(\mathbf{R}_n)$ we want to prove $(p, q) \in \Phi \left(\bigcap_{n \in \mathbb{N}} \mathbf{R}_n \right)$
 $\forall n. (p, q) \in \Phi(\mathbf{R}_n)$ (continue)

TH. Φ is continuous (for finitely branching processes)

proof. (continue) $(p, q) \in \bigcap_{n \in \mathbb{N}} \Phi(\mathbf{R}_n)$

$$\forall n. (p, q) \in \Phi(\mathbf{R}_n) \Rightarrow (p, q) \in \Phi \left(\bigcap_{n \in \mathbb{N}} \mathbf{R}_n \right)$$

take $p \xrightarrow{\mu} p'$ we want to find $q \xrightarrow{\mu} q'$ with $(p', q') \in \bigcap_{n \in \mathbb{N}} \mathbf{R}_n$

$$\forall n. (p', q') \in \mathbf{R}_n$$

since $\forall n. (p, q) \in \Phi(\mathbf{R}_n)$ then $\forall n. \exists q_n. q \xrightarrow{\mu} q_n$ with $(p', q_n) \in \mathbf{R}_n$

$$\mathbf{R}_0 \supseteq \mathbf{R}_1 \supseteq \dots \supseteq \mathbf{R}_n \supseteq \dots \quad \forall k \leq n. (p', q_n) \in \mathbf{R}_k$$

q is finitely branching: $\{q' \mid q \xrightarrow{\mu} q'\}$ is finite

thus $\exists m \in \mathbb{N}$ such that $\{n \mid q_n = q_m\}$ is infinite

hence $\forall n. (p', q_m) \in \mathbf{R}_n$ and we take $q' = q_m$

take $q \xrightarrow{\mu} q'$ we want to find $p \xrightarrow{\mu} p'$ with $(p', q') \in \bigcap_{n \in \mathbb{N}} \mathbf{R}_n$

analogous to the previous case

Strong bis as fix

$\mathcal{P}_f$ finitely branching processes

$$\simeq = \bigsqcup_{n \in \mathbb{N}} \Phi^n(\mathcal{P}_f \times \mathcal{P}_f)$$

how do we know a process is finitely branching?

we can restrict the syntax: guarded processes

CCS

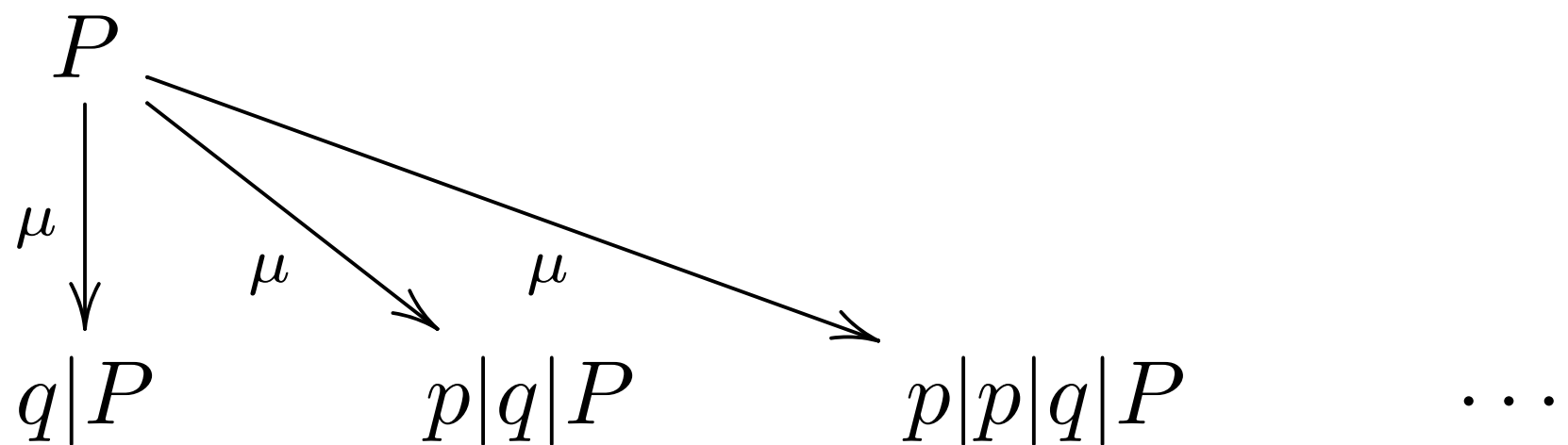
guarded processes

CCS: guarded processes

the allowed form of recursion is very general

there are processes with infinitely many outgoing transitions

suppose $p \xrightarrow{\mu} q$ take $P \triangleq \mathbf{rec} \ x. p|x$ $P \triangleq p|P$



such processes are called *infinitely branching* and are BAD

CCS: guarded processes

guarded processes guarantee that process variables occur under a prefix (recursion is guarded by some action)

let X be a set of process variables

$G(p, X)$ all recursively defined names are guarded in p
if a name in X occurs free in p it is prefixed by an action

$$G(\mathbf{nil}, X) \triangleq \mathbf{true}$$

$$G(p[\phi], X) \triangleq G(p, X)$$

$$G(x, X) \triangleq x \notin X$$

$$G(p + q, X) \triangleq G(p, X) \wedge G(q, X)$$

$$G(\mu.p, X) \triangleq G(p, \emptyset)$$

$$G(p|q, X) \triangleq G(p, X) \wedge G(q, X)$$

$$G(p \setminus \alpha, X) \triangleq G(p, X) \quad G(\mathbf{rec} \ x. \ p, X) \triangleq G(p, X \cup \{x\})$$

a closed process p is *guarded* if $G(p, \emptyset)$ holds true



Exercise: guarded?

$$R \triangleq \mathbf{rec} \ x. \ \alpha.x + \beta$$

$$G(R, \emptyset)?$$



$$G(\mathbf{rec} \ x. \ \alpha.x + \beta, \emptyset) = G(\alpha.x + \beta, \{x\})$$

$$= G(\alpha.x, \{x\}) \wedge G(\beta, \{x\})$$

$$= G(x, \emptyset) \wedge G(\mathbf{nil}, \emptyset)$$

$$= x \notin \emptyset \wedge \mathbf{true}$$

$$= \mathbf{true}$$



Exercise: guarded?

$$T \triangleq \mathbf{rec} \ x. (\alpha|x) + \beta$$

$$G(T, \emptyset)? \quad \text{✗}$$

$$\begin{aligned} G(\mathbf{rec} \ x. (\alpha|x) + \beta, \emptyset) &= G((\alpha|x) + \beta, \{x\}) \\ &= G(\alpha|x, \{x\}) \wedge G(\beta, \{x\}) \\ &= G(\alpha, \{x\}) \wedge G(x, \{x\}) \wedge G(\mathbf{nil}, \emptyset) \\ &= G(\mathbf{nil}, \emptyset) \wedge x \notin \{x\} \wedge \mathbf{true} \\ &= \mathbf{true} \wedge \mathbf{false} \\ &= \mathbf{false} \end{aligned}$$



Exercise: guarded?

$$U \triangleq \mathbf{rec} \ x. \ \alpha | \beta.x$$

$$G(U, \emptyset)?$$



$$\begin{aligned} G(\mathbf{rec} \ x. \ \alpha | \beta.x, \emptyset) &= G(\alpha | \beta.x, \{x\}) \\ &= G(\alpha, \{x\}) \wedge G(\beta.x, \{x\}) \\ &= G(\mathbf{nil}, \emptyset) \wedge G(x, \emptyset) \\ &= \mathbf{true} \wedge x \notin \emptyset \\ &= \mathbf{true} \end{aligned}$$



Exercise: guarded?

rec $x. x$



unguarded

rec $x. \alpha. \mathbf{rec} \ y. x$



guarded

rec $x. \alpha. \mathbf{rec} \ y. x + y$



unguarded

rec $x. \alpha. \mathbf{rec} \ y. x|y$



unguarded

rec $x. \alpha. \mathbf{rec} \ y. x|\beta.y$



guarded