PSC 2020/21 (375AA, 9CFU)

Principles for Software Composition

Roberto Bruni
http://www.di.unipi.it/~bruni/

http://didawiki.di.unipi.it/doku.php/magistraleinformatica/psc/

18c - bisimilarity as a fixpoint

CCS syntax

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

(operators are listed in order of precedence)

CCS op. semantics

Act) $$\frac{}{\mu.p \xrightarrow{\mu} p}$$

Res) $$\frac{p \xrightarrow{\mu} q \quad \mu \notin \{\alpha, \overline{\alpha}\}}{p\backslash\alpha \xrightarrow{\mu} q\backslash\alpha}$$

Rel) $$\frac{p \xrightarrow{\mu} q}{p[\phi] \xrightarrow{\phi(\mu)} q[\phi]}$$

SumL) $$\frac{p_1 \xrightarrow{\mu} q}{p_1 + p_2 \xrightarrow{\mu} q}$$

SumR) $$\frac{p_2 \xrightarrow{\mu} q}{p_1 + p_2 \xrightarrow{\mu} q}$$

ParL) $$\frac{p_1 \xrightarrow{\mu} q_1}{p_1|p_2 \xrightarrow{\mu} q_1|p_2}$$

Com) $$\frac{p_1 \xrightarrow{\lambda} q_1 \quad p_2 \xrightarrow{\overline{\lambda}} q_2}{p_1|p_2 \xrightarrow{\tau} q_1|q_2}$$

ParR) $$\frac{p_2 \xrightarrow{\mu} q_2}{p_1|p_2 \xrightarrow{\mu} p_1|q_2}$$

Rec) $$\frac{p[^{\mathbf{rec}\ x.\ p}/_{x}] \xrightarrow{\mu} q}{\mathbf{rec}\ x.\ p \xrightarrow{\mu} q}$$

Strong bisimilarity

$\mathcal{P}$ set of processes $\quad$ $\mathbf{R} \subseteq \mathcal{P} \times \mathcal{P}$ a binary relation

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

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

Alice plays $\qquad$ Bob replies

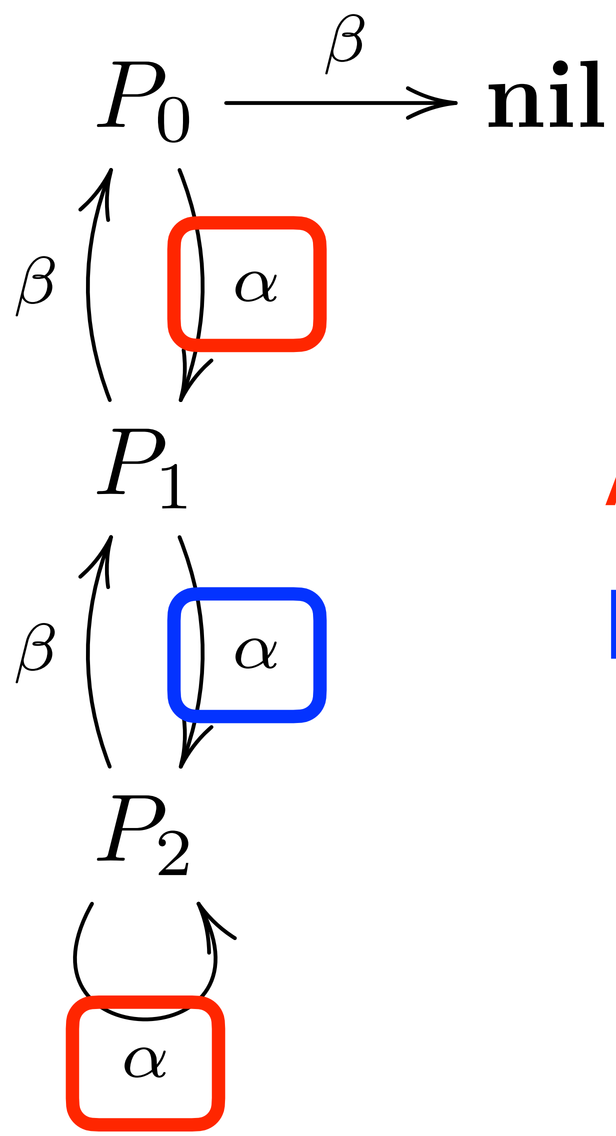
strong bisimilarity $\quad \simeq \ \triangleq \bigcup_{\mathbf{R}\ \text{s.b.}} \mathbf{R}$

is an equivalence

is a strong bisimulation

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

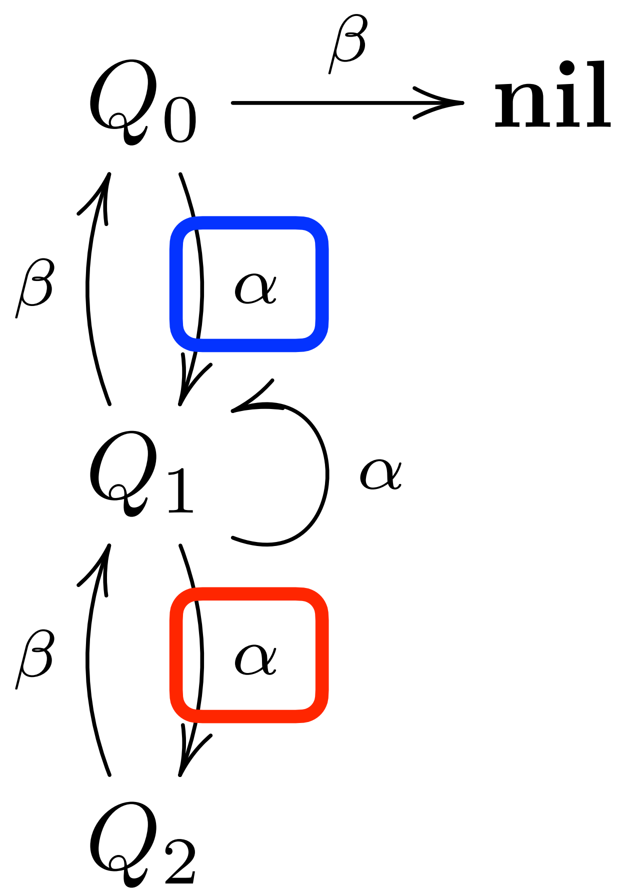
Bisimulation game



$P_0 \overset{?}{\simeq} Q_0$

Alice plays

Bob replies



CCS
Bisimilarity as a fixpoint

Strong bis as fix

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

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

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

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

a strong bisimulation

Strong bis as fix

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

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

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

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

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)!

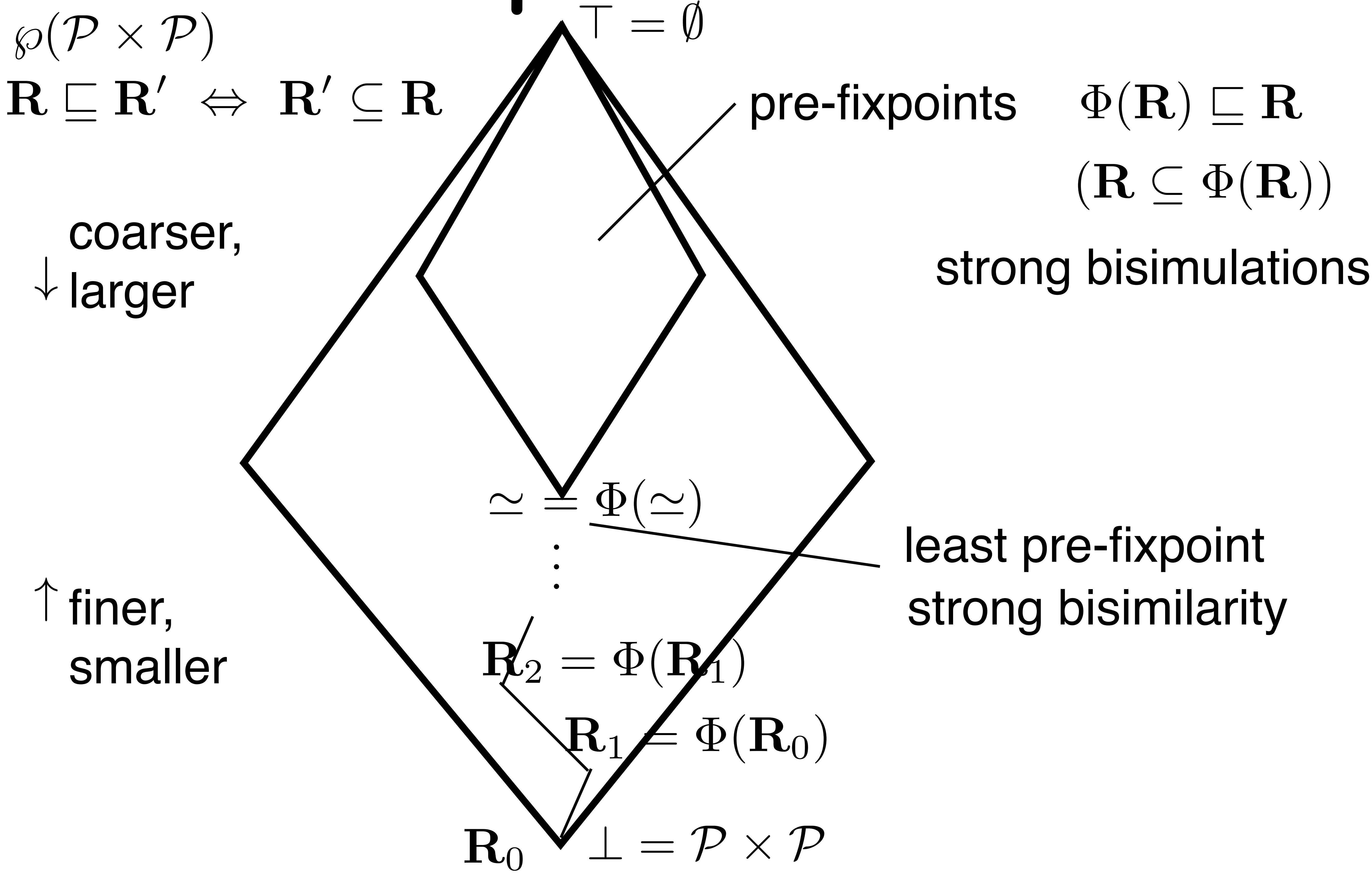
$(\wp(\mathcal{P} \times \mathcal{P}), \sqsubseteq)$

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

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

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

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 $\Phi(\mathbf{R}_1) \sqsubseteq \Phi(\mathbf{R}_2)$

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

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

$$\mathbf{R}_0 \sqsubseteq \mathbf{R}_1 \sqsubseteq \cdots \sqsubseteq \mathbf{R}_n \sqsubseteq \ldots$$

$$\mathbf{R}_0 \supseteq \mathbf{R}_1 \supseteq \cdots \supseteq \mathbf{R}_n \supseteq \ldots$$

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

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

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)

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

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 $(p',q_n) \in \mathbf{R}_n$

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

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

Example

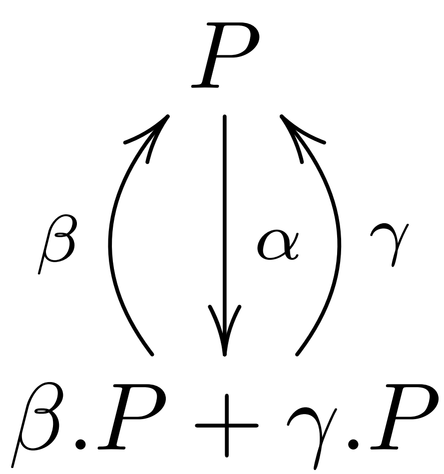
Guarded!

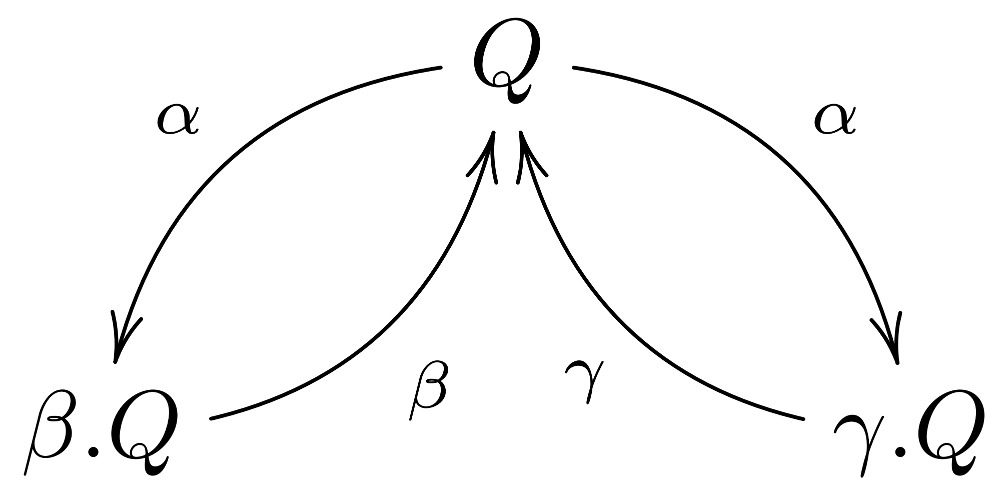
$$P \triangleq \alpha.(\beta.P + \gamma.P)$$

$$P \overset{?}{\simeq} Q$$

Guarded!

$$Q \triangleq \alpha.\beta.Q + \alpha.\gamma.Q$$



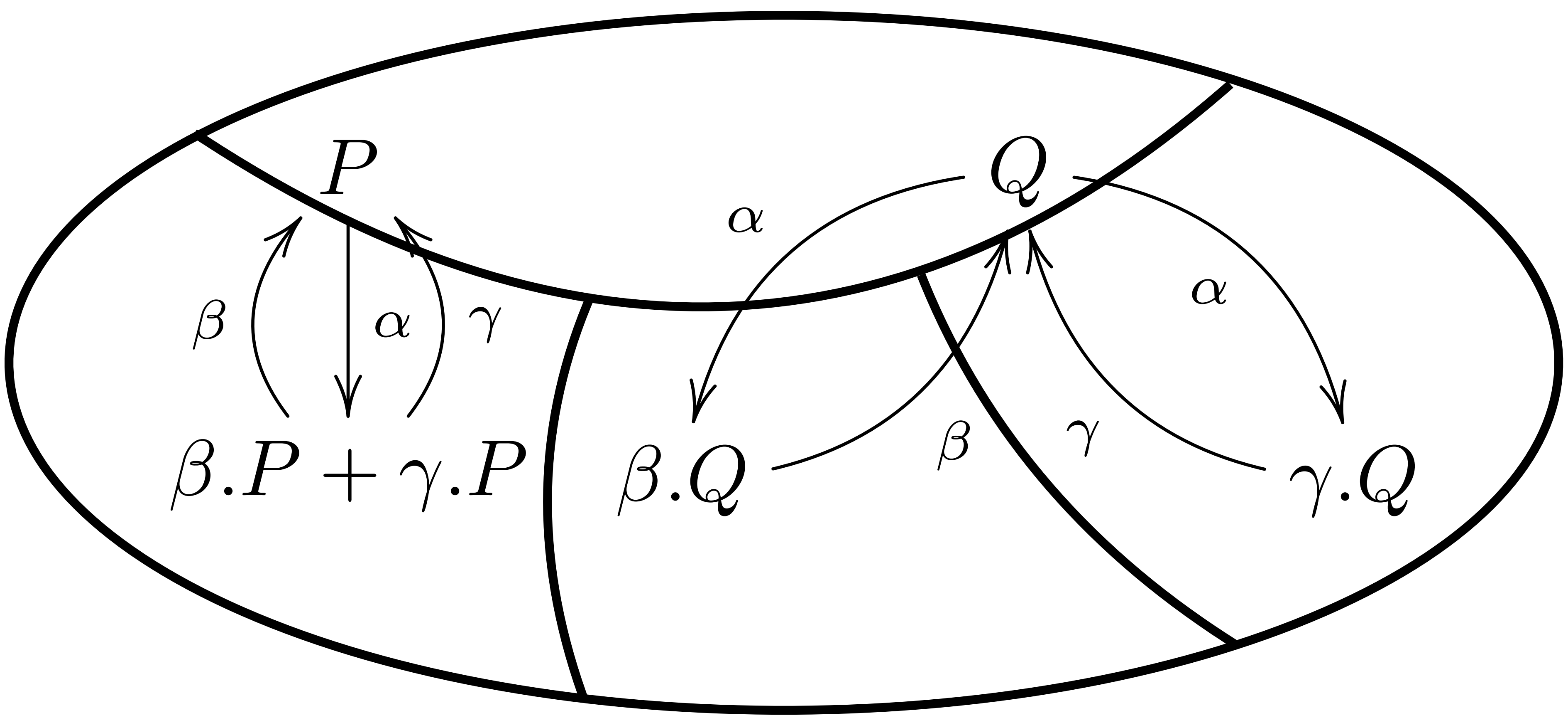


$$\mathbf{R}_0 = \{ \ \{P \ , \ Q \ , \ \beta.P + \gamma.P \ , \ \beta.Q \ , \ \gamma.Q\} \ \}$$

Example

$$P \triangleq \alpha.(\beta.P + \gamma.P) \qquad P \overset{?}{\simeq} Q \qquad Q \triangleq \alpha.\beta.Q + \alpha.\gamma.Q$$

$$\mathbf{R}_0 = \{\ \{P\ ,\ Q\ ,\ \beta.P + \gamma.P\ ,\ \beta.Q\ ,\ \gamma.Q\}\ \}$$



$$P, Q \xrightarrow{\alpha}$$

$$\beta.P + \gamma.P \xrightarrow{\beta,\gamma}$$

$$\beta.Q \xrightarrow{\beta}$$

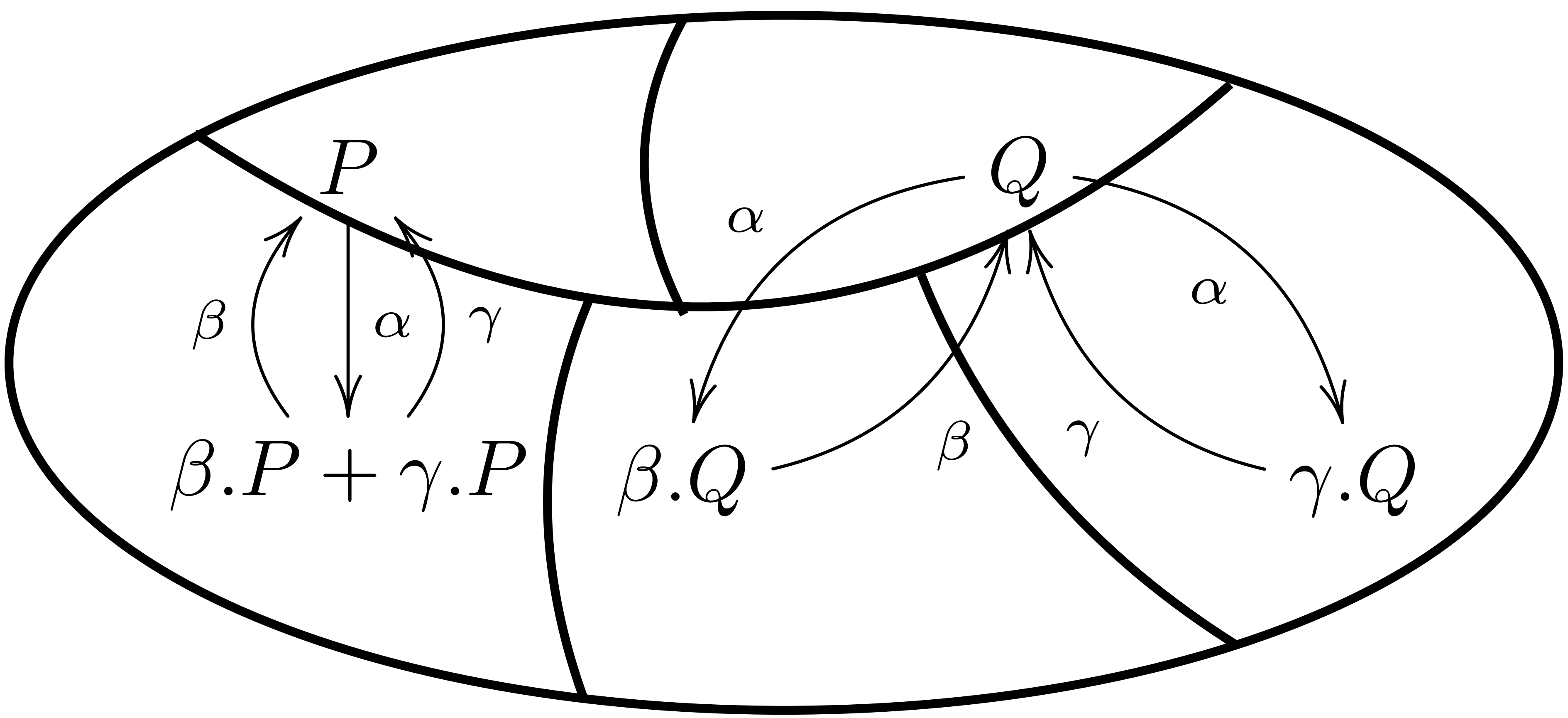
$$\gamma.Q \xrightarrow{\gamma}$$

Processes with different capabilities must be distinguished

Example

$P \triangleq \alpha.(\beta.P + \gamma.P) \qquad P \overset{?}{\simeq} Q \qquad Q \triangleq \alpha.\beta.Q + \alpha.\gamma.Q$

$$\mathbf{R}_1 = \{\ \{P\ ,\ Q\}\ ,\ \{\beta.P + \gamma.P\}\ ,\ \{\beta.Q\}\ ,\ \{\gamma.Q\}\ \}$$



$P \xrightarrow{\alpha} [\beta.P + \gamma.Q]$

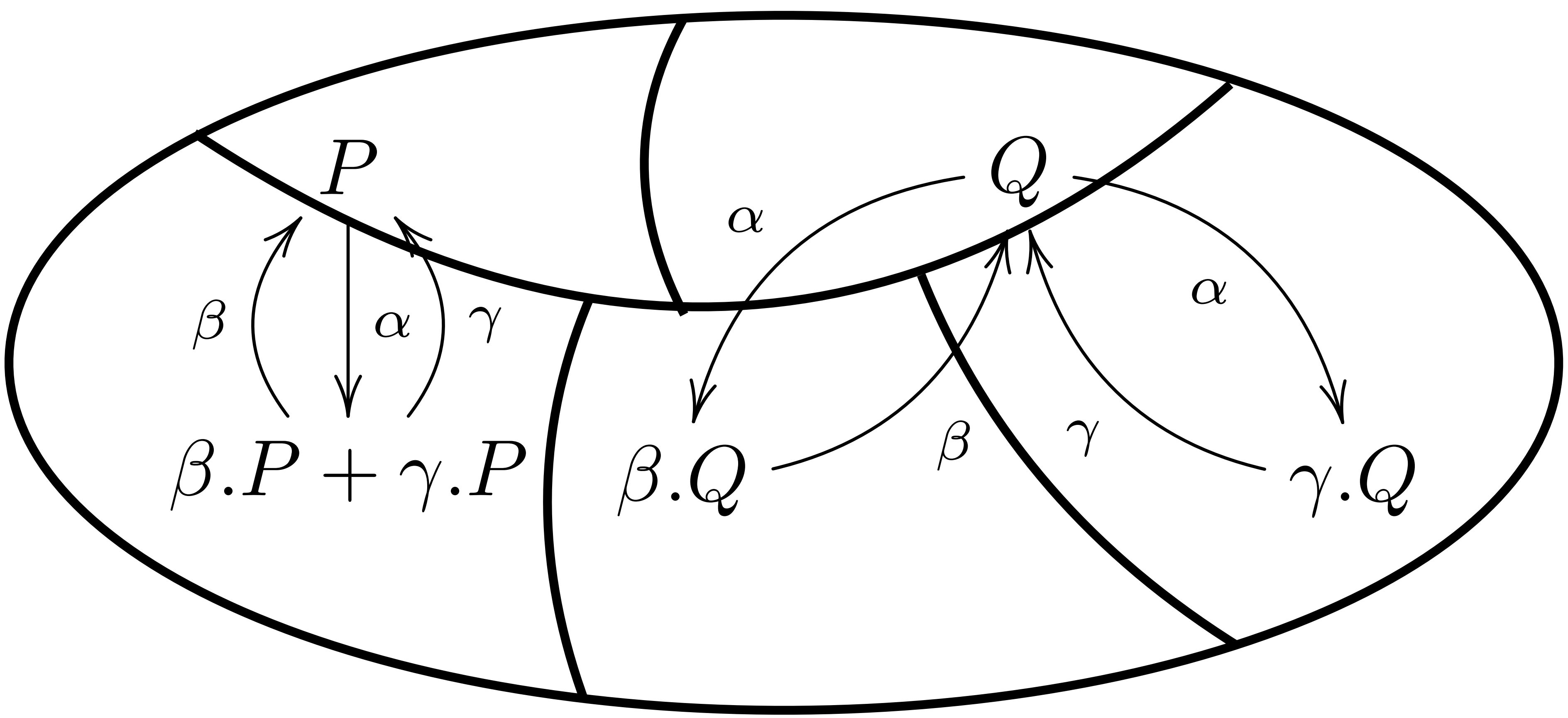
$Q \xrightarrow{\alpha} [\beta.Q], [\gamma.Q]$

The α transitions of P and Q ends in different partitions

Example

$P \triangleq \alpha.(\beta.P + \gamma.P) \qquad P \overset{?}{\simeq} Q \qquad Q \triangleq \alpha.\beta.Q + \alpha.\gamma.Q$

$$\mathbf{R}_2 = \{\ \{P\}\ ,\ \{Q\}\ ,\ \{\beta.P + \gamma.P\}\ ,\ \{\beta.Q\}\ ,\ \{\gamma.Q\}\ \}$$



Example

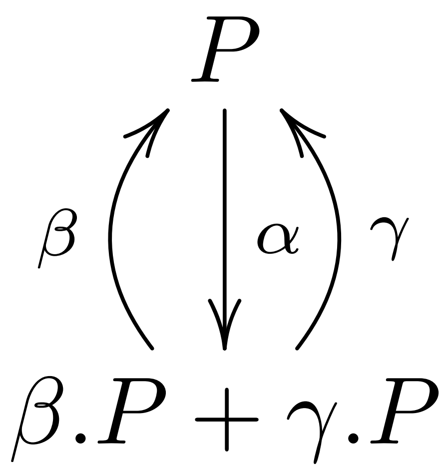
Guarded!

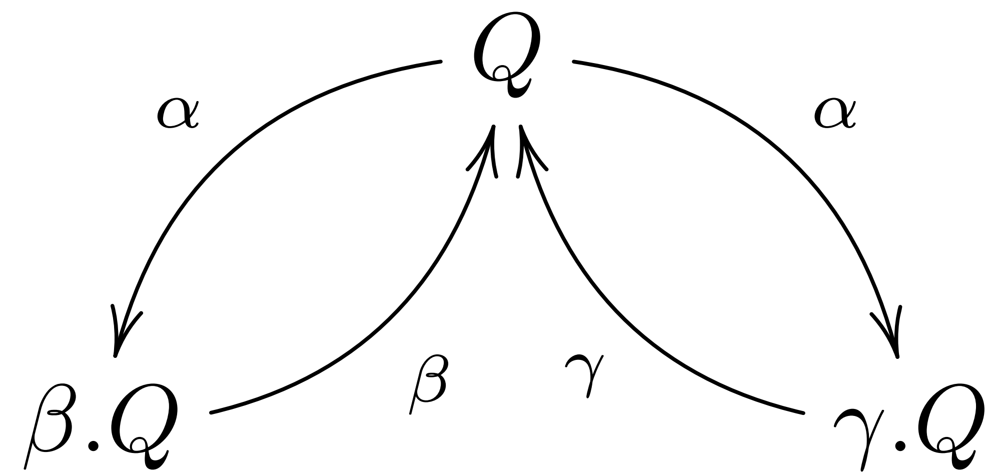
$$P \triangleq \alpha.(\beta.P + \gamma.P)$$

$$P \overset{?}{\simeq} Q$$

Guarded!

$$Q \triangleq \alpha.\beta.Q + \alpha.\gamma.Q$$





$$\mathbf{R}_0 = \{\ \{P\ ,\ Q\ ,\ \beta.P + \gamma.P\ ,\ \beta.Q\ ,\ \gamma.Q\}\ \}$$

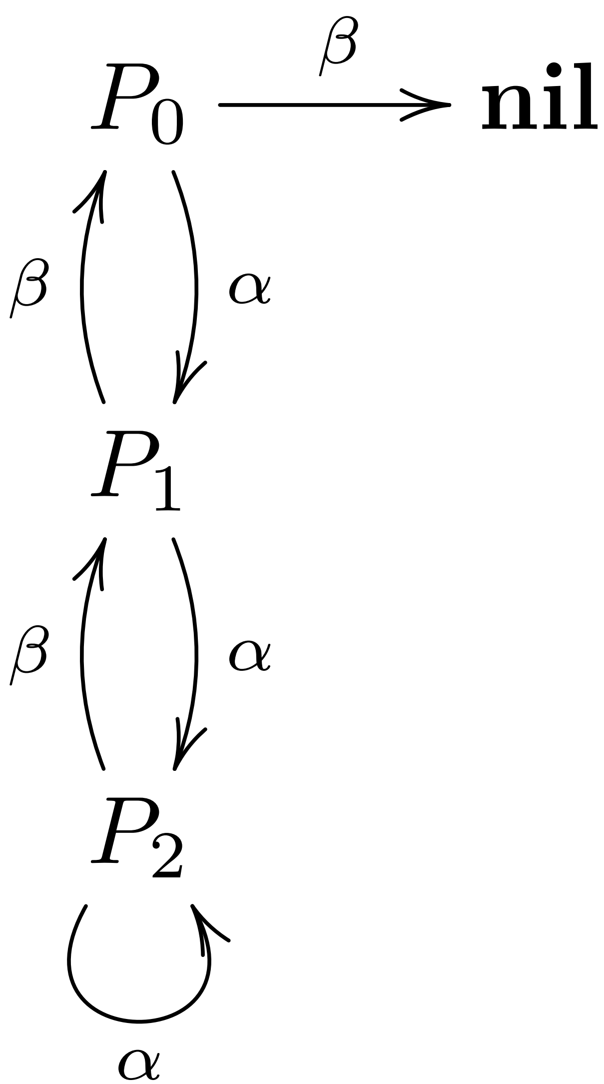
$$\mathbf{R}_1 = \{\ \{P\ ,\ Q\}\ ,\ \{\beta.P + \gamma.P\}\ ,\ \{\beta.Q\}\ ,\ \{\gamma.Q\}\ \}$$

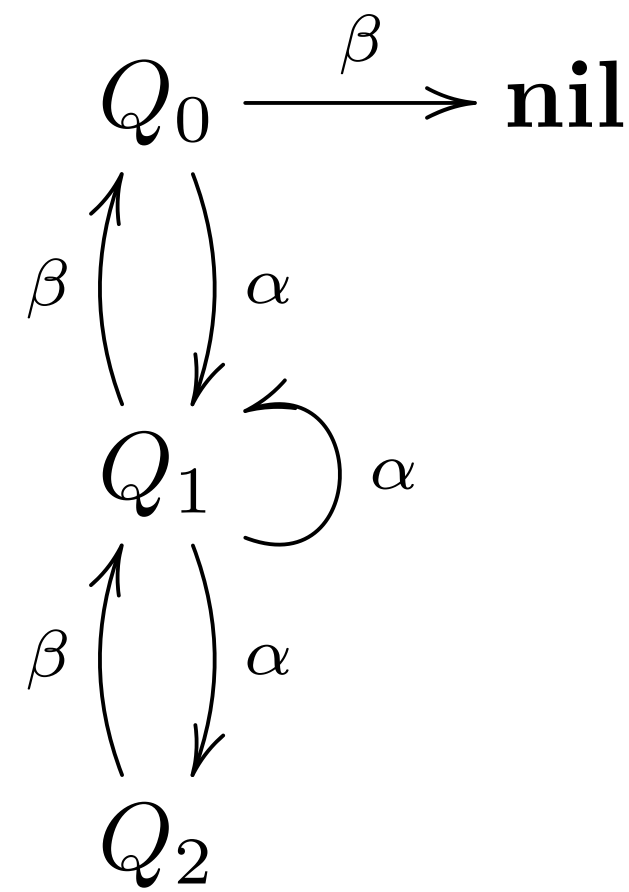
$$\mathbf{R}_2 = \{\ \{P\}\ ,\ \{Q\}\ ,\ \{\beta.P + \gamma.P\}\ ,\ \{\beta.Q\}\ ,\ \{\gamma.Q\}\ \}$$

Only singletons partitions, we can stop $P \not\simeq Q$

Exercise

finitely branching! $P_0 \overset{?}{\simeq} Q_0$ finitely branching!





$\mathbf{nil} \not\rightarrow$

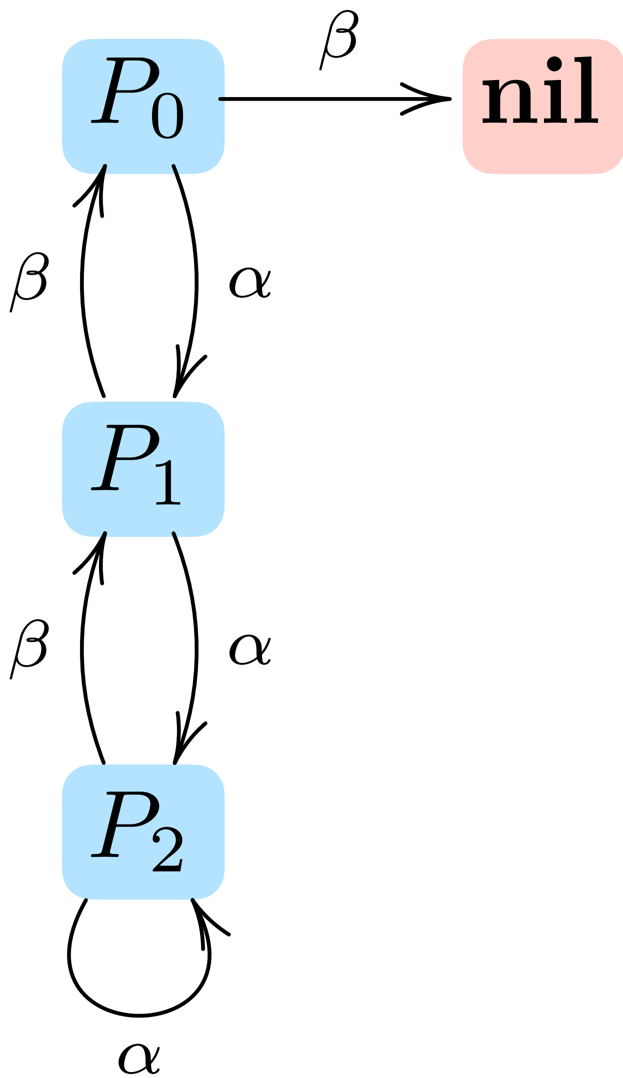
$Q_2 \xrightarrow{\beta}$

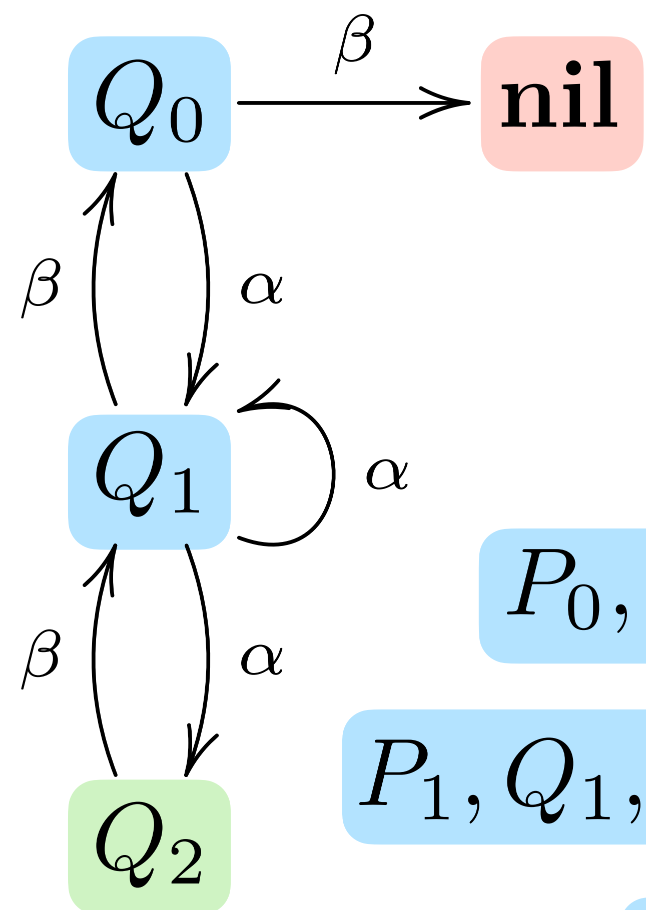
$P_0, Q_0, P_1, Q_1, P_2 \xrightarrow{\alpha,\beta}$

$$\mathbf{R}_0 = \{\ \{P_0, Q_0, P_1, Q_1, P_2, Q_2, \mathbf{nil}\}\ \}$$

Exercise

$$P_0 \overset{?}{\simeq} Q_0$$





$$P_0, Q_0 \xrightarrow{\beta} [\mathbf{nil}]$$

$$P_1, Q_1, P_2 \not\xrightarrow{\beta} [\mathbf{nil}]$$

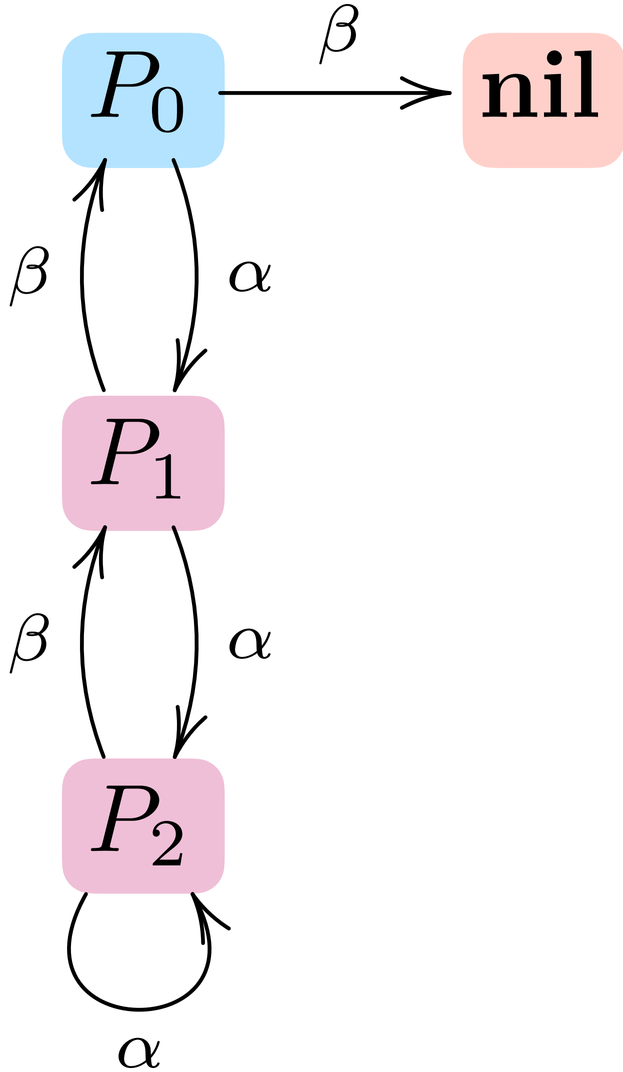
$$Q_1 \xrightarrow{\alpha} [Q_2]$$

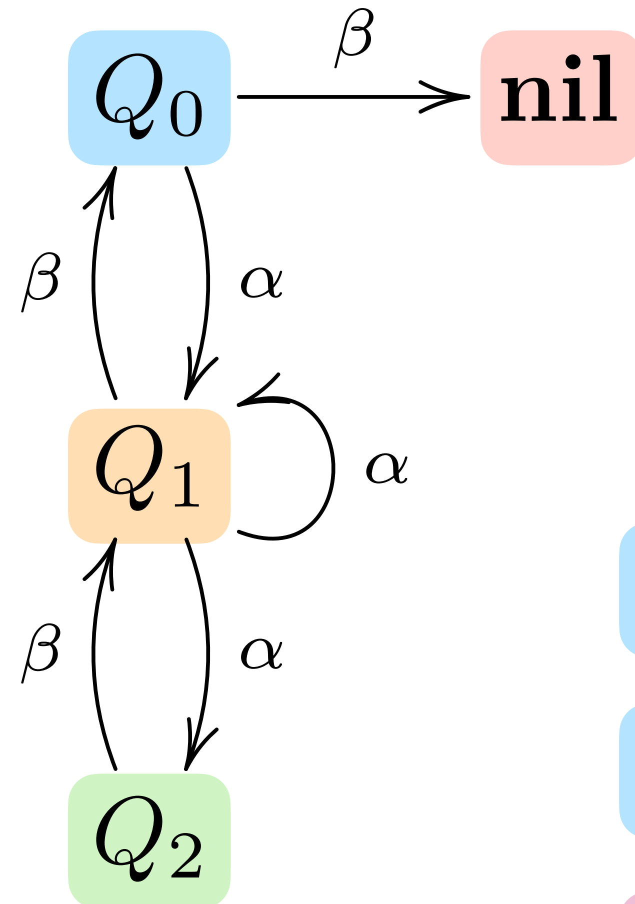
$$P_1, P_2 \not\xrightarrow{\alpha} [Q_2]$$

$$\mathbf{R}_1 = \{ \{P_0, Q_0, P_1, Q_1, P_2\}, \{Q_2\}, \{\mathbf{nil}\} \}$$

Exercise

$$P_0 \overset{?}{\simeq} Q_0$$





$P_0 \xrightarrow{\alpha} [P_1]$

$Q_0 \not\xrightarrow{\alpha} [P_1]$

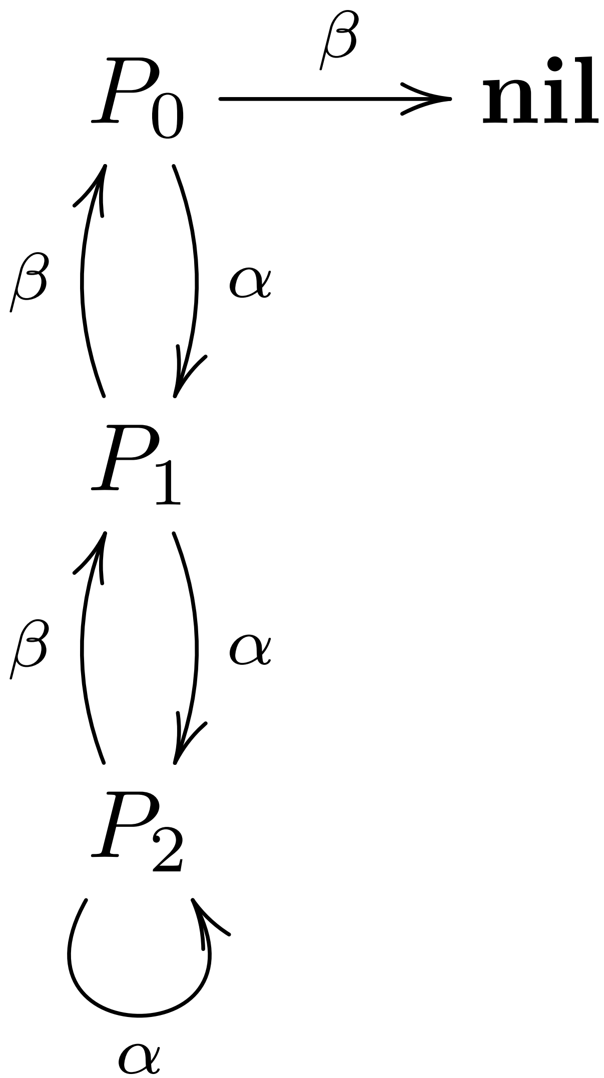
$P_1 \xrightarrow{\beta} [P_0]$

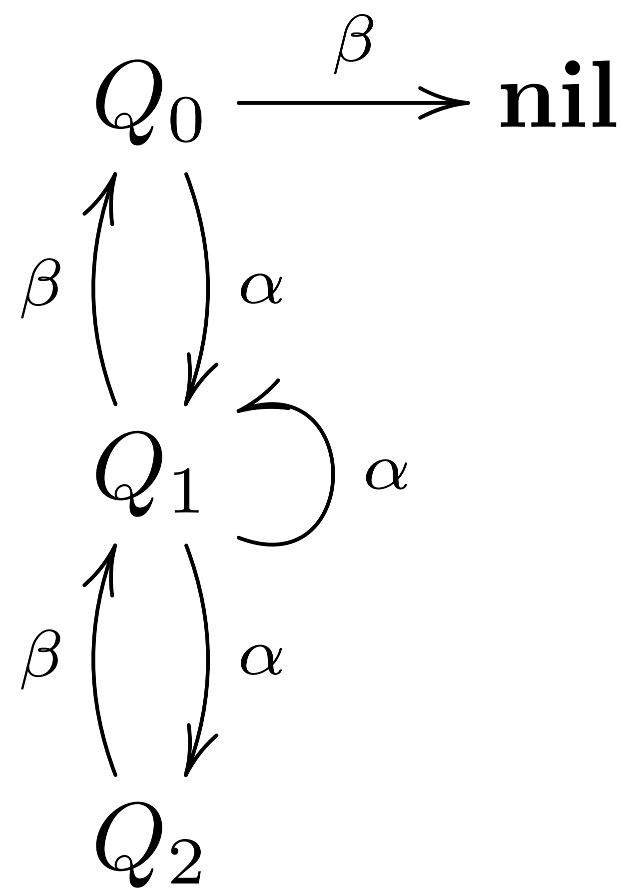
$P_2 \not\xrightarrow{\beta} [P_0]$

$$\mathbf{R}_2 = \{ \{P_0, Q_0\}, \{P_1, P_2\}, \{Q_1\}, \{Q_2\}, \{\mathbf{nil}\} \}$$

Exercise

$$P_0 \overset{?}{\simeq} Q_0$$



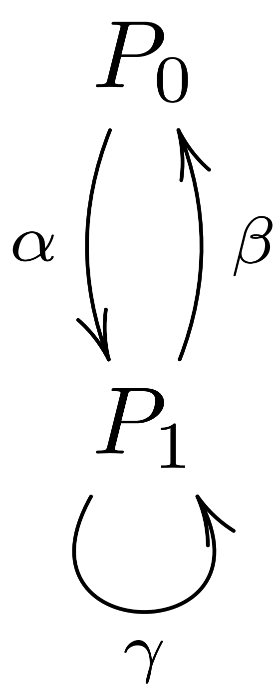


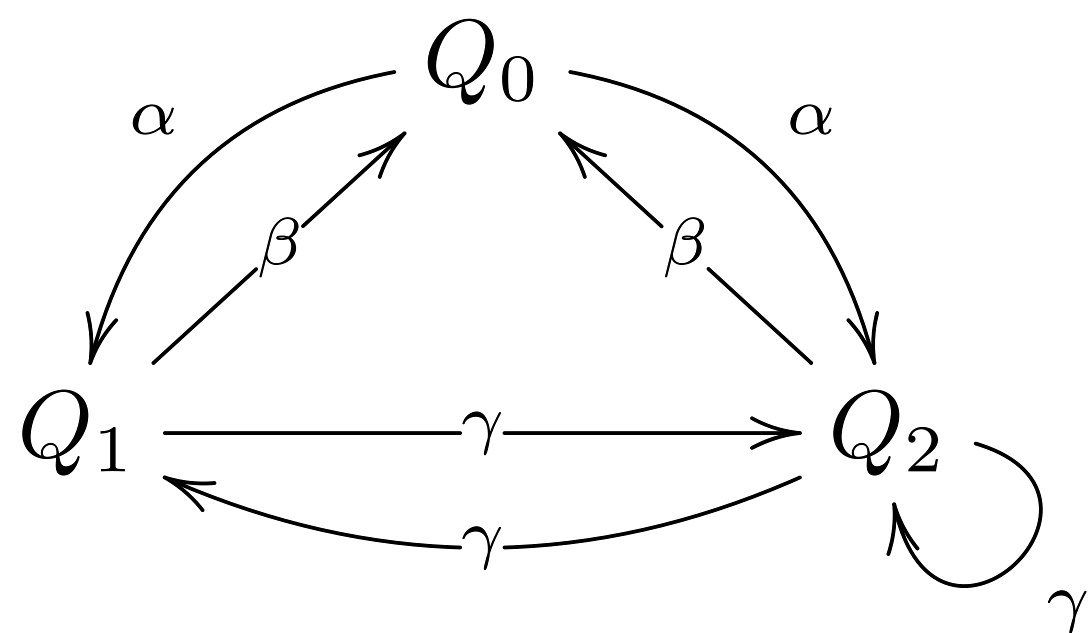
$$P_0 \not\simeq Q_0$$

$$\mathbf{R}_3 = \{ \ \{P_0\} \ , \ \{Q_0\} \ , \ \{P_1\} \ , \ \{P_2\} \ , \ \{Q_1\} \ , \ \{Q_2\} \ , \ \{\mathbf{nil}\} \ \}$$

Exercise

finitely branching! $P_0 \overset{?}{\simeq} Q_0$ finitely branching!





$$\mathbf{R}_0 = \{ \ \{P_0, Q_0, P_1, Q_1, Q_2\} \ \}$$

$$P_0, Q_0 \xrightarrow{\alpha}$$

$$P_1, Q_1, Q_2 \xrightarrow{\beta,\gamma}$$

$$\mathbf{R}_1 = \{ \ \{P_0, Q_0\} \ , \ \{P_1, Q_1, Q_2\} \ \}$$

No more reasons to discriminate!

Unguarded processes?

What about the general case? (unguarded processes)

any powerset ordered by inclusion defines a complete lattice

Complete lattice: $(D, \sqsubseteq)$ PO such that

any $X \subseteq D$ has a least upper bound $\bigsqcup X$

any $X \subseteq D$ has a greatest lower bound $\bigsqcap X$

it has bottom and top elements $\bot = \bigsqcap D$ $\top = \bigsqcup D$

TH. [Knaster-Tarski] $(D, \sqsubseteq)$ complete lattice

$f : D \to D$ monotone

has least and greatest fixpoint

$d_{\mathsf{min}} \triangleq \bigsqcap \{d \in D \mid f(d) \sqsubseteq d\}$ is the least fixpoint

glb — pre-fixpoints

$d_{\mathsf{max}} \triangleq \bigsqcup \{d \in D \mid d \sqsubseteq f(d)\}$ is the greatest fixpoint

lub — post-fixpoints

least and greatest fixpoint exist… but how to compute them?