



Linguaggi di Programmazione

Roberta Gori

Bisimilarita' come punto fisso

CCS: sintassi

p, q	$::=$	nil	processo inattivo
		x	variabile di processo (per la ricorsione)
		$\mu.p$	prefisso azione
		$p \setminus \alpha$	canale ristretto
		$p[\phi]$	rietichettatura del canale
		$p + q$	scelta nondeterministica (somma)
		$p q$	composizione parallela
		rec $x. p$	ricorsione

(gli operatori sono elencati in ordine di precedenza)

CCS semantica operativa

$$\begin{array}{l} \text{Act)} \frac{}{\mu.p \xrightarrow{\mu} p} \\ \text{Res)} \frac{p \xrightarrow{\mu} q \quad \mu \notin \{\alpha, \bar{\alpha}\}}{p \setminus \alpha \xrightarrow{\mu} q \setminus \alpha} \\ \text{Rel)} \frac{p \xrightarrow{\mu} q}{p[\phi] \xrightarrow{\phi(\mu)} q[\phi]} \end{array}$$

$$\begin{array}{l} \text{SumL)} \frac{p_1 \xrightarrow{\mu} q}{p_1 + p_2 \xrightarrow{\mu} q} \\ \text{SumR)} \frac{p_2 \xrightarrow{\mu} q}{p_1 + p_2 \xrightarrow{\mu} q} \end{array}$$

$$\begin{array}{l} \text{ParL)} \frac{p_1 \xrightarrow{\mu} q_1}{p_1 | p_2 \xrightarrow{\mu} q_1 | p_2} \\ \text{Com)} \frac{p_1 \xrightarrow{\lambda} q_1 \quad p_2 \xrightarrow{\bar{\lambda}} q_2}{p_1 | p_2 \xrightarrow{\tau} q_1 | q_2} \\ \text{ParR)} \frac{p_2 \xrightarrow{\mu} q_2}{p_1 | p_2 \xrightarrow{\mu} p_1 | q_2} \end{array}$$

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

bisimilarita' forte

\mathcal{P} insiemi di processi $\mathbf{R} \subseteq \mathcal{P} \times \mathcal{P}$ una relazione binaria

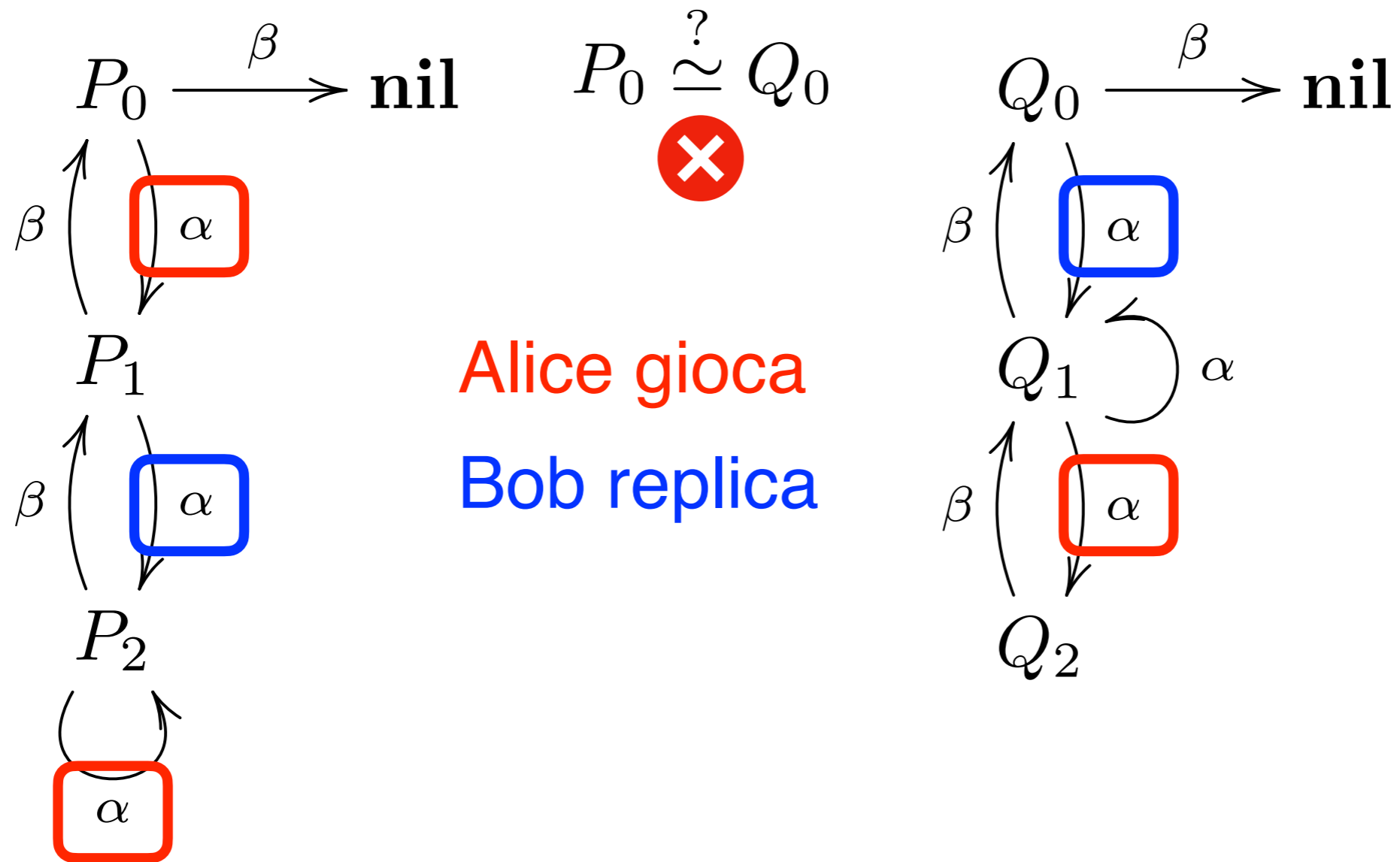
\mathbf{R} e' una bisimulazione forte se

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

bisimilarita' forte $\simeq \triangleq \bigcup_{\mathbf{R} \text{ s.b.}} \mathbf{R}$ e' un'equivalenza e' una bisimulazione forte

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

il gioco della bisimulazione



CCS

Bisimilarita' come punto fisso

Bisimulazione come pre punto fisso

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

$\Phi : \wp(\mathcal{P} \times \mathcal{P}) \rightarrow \wp(\mathcal{P} \times \mathcal{P})$ mappa relazioni in relazioni

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

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

una bisimulazione forte

Bisimilarita' come punto fisso

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

$\Phi : \wp(\mathcal{P} \times \mathcal{P}) \rightarrow \wp(\mathcal{P} \times \mathcal{P})$ mappa relazioni in relazioni

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

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

la bisimilarita' forte e' un punto fisso

Punto fisso: quale OPC?

Possiamo usare il teorema di Kleene?

Dominio $\wp(\mathcal{P} \times \mathcal{P})$

vogliamo trovare **la piu' grande** relazione,
non **la piu' piccola**

Idea: invertiamo l'ordine (inclusione)!

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

una relazione con più coppie è
più piccola di una con meno coppie

$$\mathbf{R} \sqsubseteq \mathbf{R}' \iff \mathbf{R}' \subseteq \mathbf{R}$$

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

Minimo punto fisso...invertito

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

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

$$\top = \emptyset$$

$$\text{pre-punti fissi } \Phi(\mathbf{R}) \sqsubseteq \mathbf{R}$$

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

↓ piu' grande

bisimulazione forte

↑ piu' piccola

minimo pre-punto fisso
bisimilarita' forte

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

⋮

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

$$\mathbf{R}_1 = \Phi(\mathbf{R}_0)$$

$$\mathbf{R}_0 \perp = \mathcal{P} \times \mathcal{P}$$

Calcolare il punto fisso

possiamo riutilizzare il teorema del punto fisso di Kleene per calcolare \simeq ?

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

intersezione

partiamo dalla relazione universale (tutte le coppie, una partizione unica)

tutti i processi sono equivalenti

applichiamo Φ per distinguere sempre più processi

\mathbf{R}_1 distinguibili in un passo

\mathbf{R}_2 distinguibili in due passi

⋮

il numero di partizioni aumenta ad ogni passo

TH. Φ e' monotona

prova.

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

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

prendiamo $(p, q) \in \Phi(\mathbf{R}_2)$ dobbiamo provare $(p, q) \in \Phi(\mathbf{R}_1)$

dal momento che $(p, q) \in \Phi(\mathbf{R}_2)$ abbiamo $\forall p \xrightarrow{\mu} p'$ con
 $\exists q \xrightarrow{\mu} q'$

e $(p', q') \in \mathbf{R}_2 \subseteq \mathbf{R}_1$ quindi, per def di Φ , abbiamo che
 $(p, q) \in \Phi(\mathbf{R}_1)$

Per $q \xrightarrow{\mu} q'$ e $p \xrightarrow{\mu} p'$ analogo al caso precedente

TH. Φ e' continua (su processi finitamente ramificati)

prova.

prendiamo una catena

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

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

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

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

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

segue per la monotonia

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

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

prendiamo $(p, q) \in \bigcap_{n \in \mathbb{N}} \Phi(\mathbf{R}_n)$ vogliamo provare

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

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

TH. Φ e' continua (su processi finitamente ramificati)

prova.

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

prendiamo $p \xrightarrow{\mu} p'$ vogliamo trovare $q \xrightarrow{\mu} q'$ con $(p', q') \in \bigcap_{n \in \mathbb{N}} \mathbf{R}_n$

dal momento che

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

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

q e' finitamente ramificato: $\{q' \mid q \xrightarrow{\mu} q'\}$ e' finito

quindi $\exists m \in \mathbb{N}$ tali che $\{n \mid q_n = q_m\}$ e' infinito

quindi $\forall n. (p', q_m) \in \mathbf{R}_n$ e prendiamo $q' = q_m$

$$(p', q') \in \bigcap_{n \in \mathbb{N}} \mathbf{R}_n$$

inverso, analogo al caso precedente

Bisimilarita' come punto fisso

\mathcal{P}_f processi finitamente ramificati

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

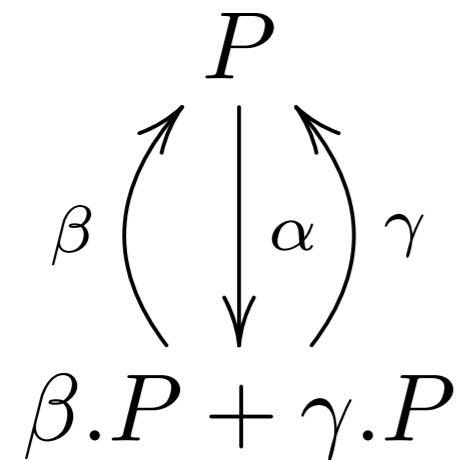
come facciamo a sapere che un processo è finitamente ramificato?

possiamo restringere la sintassi: processi guardati

Esempio

Guardato!

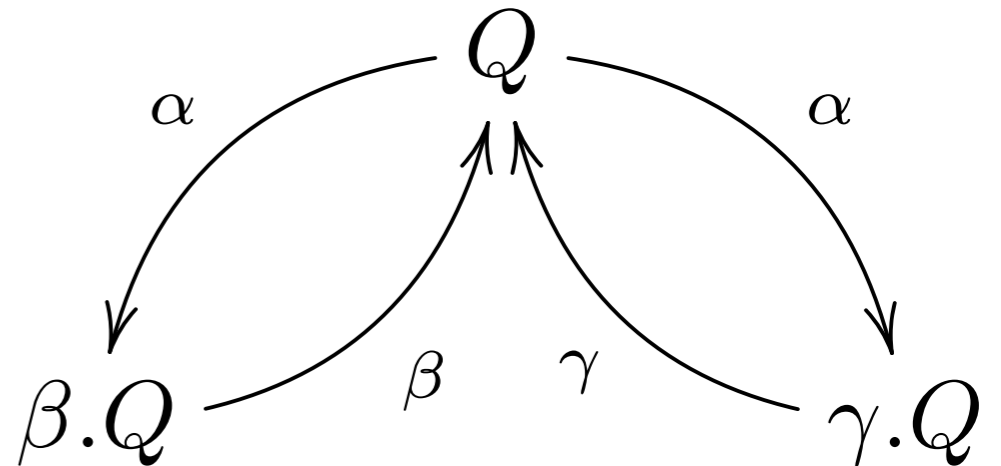
$$P \triangleq \alpha.(\beta.P + \gamma.P)$$



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

Guardato!

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

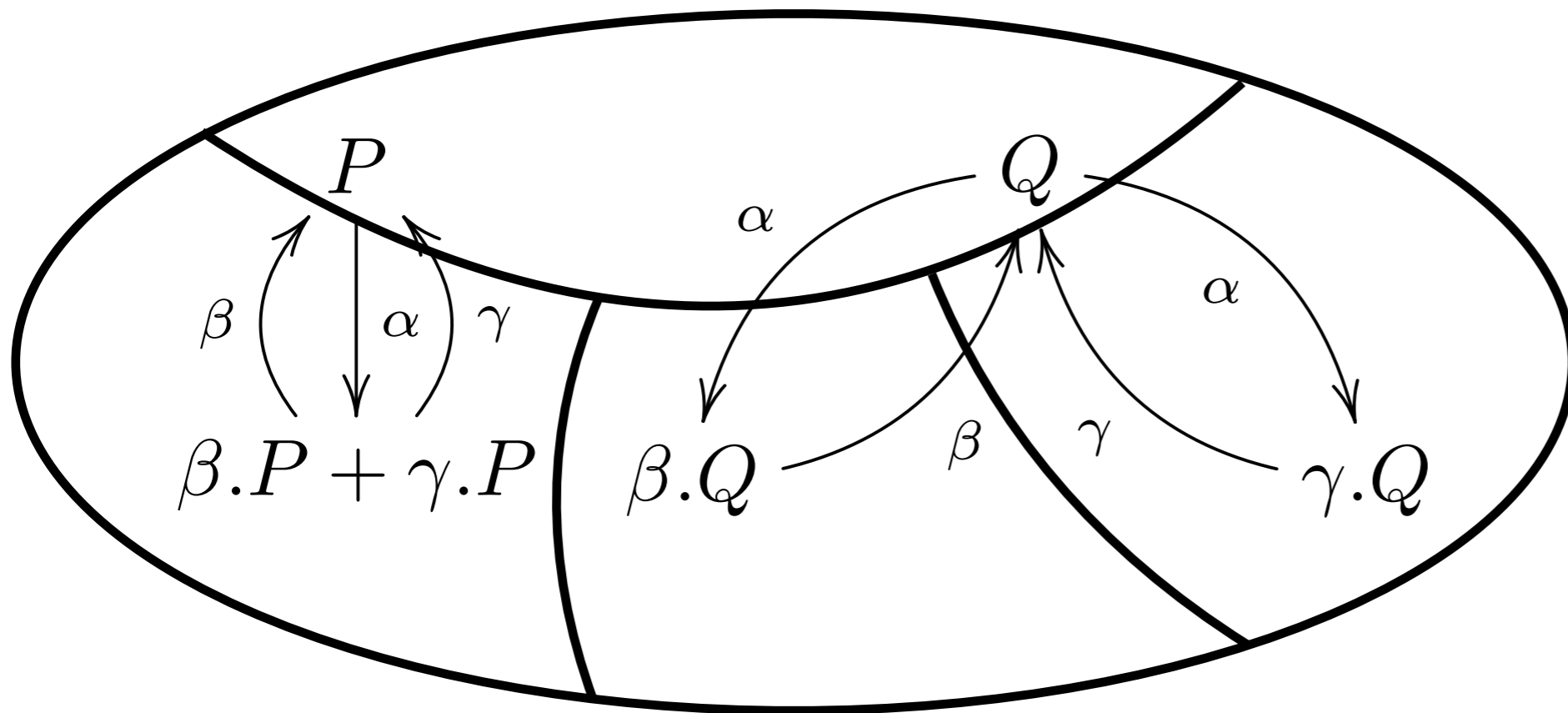


$$\mathbf{R}_0 = \{ \{P, Q, \beta.P + \gamma.P, \beta.Q, \gamma.Q\} \}$$

Esempio

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

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



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

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

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

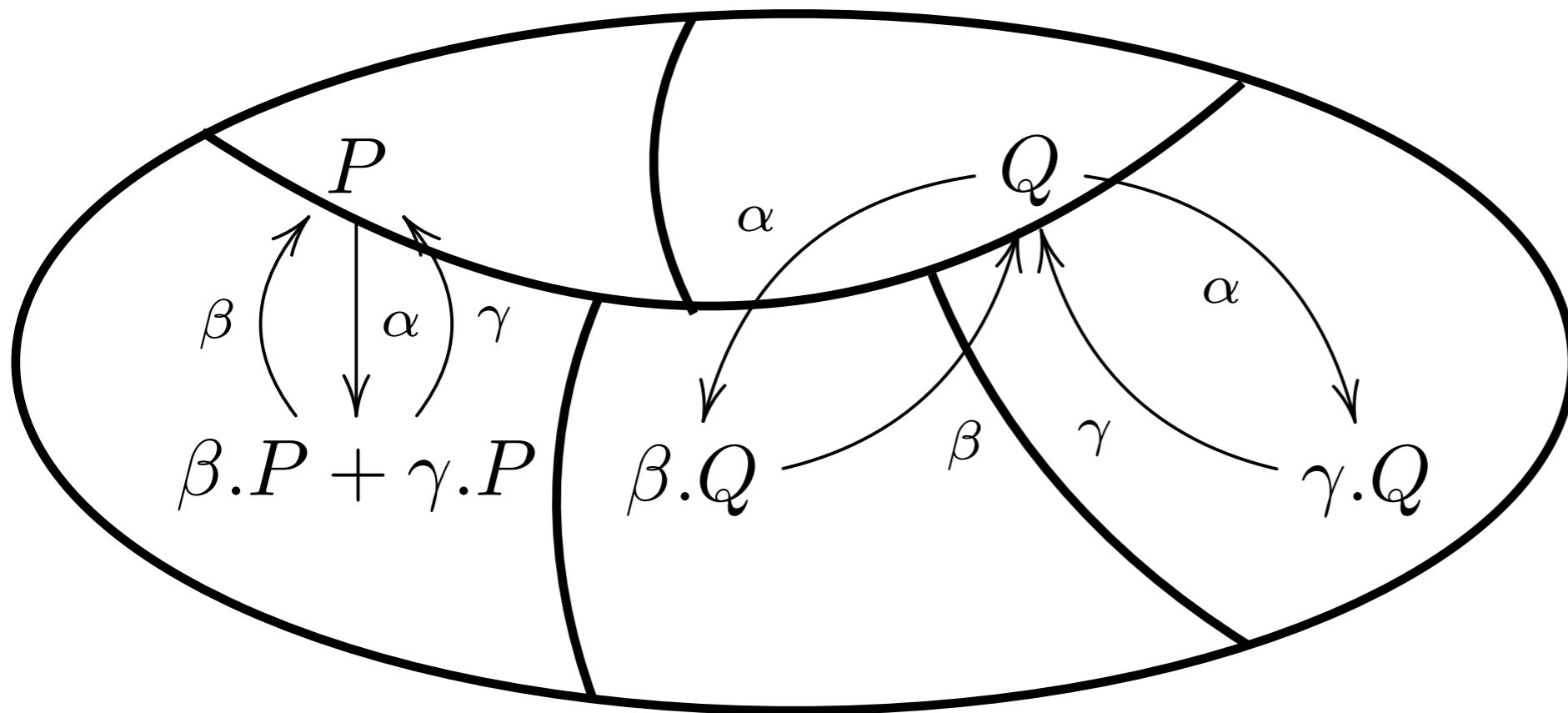
$$\gamma.Q \xrightarrow{\gamma}$$

Si devono distinguere i processi con capacità diverse

Esempio

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

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



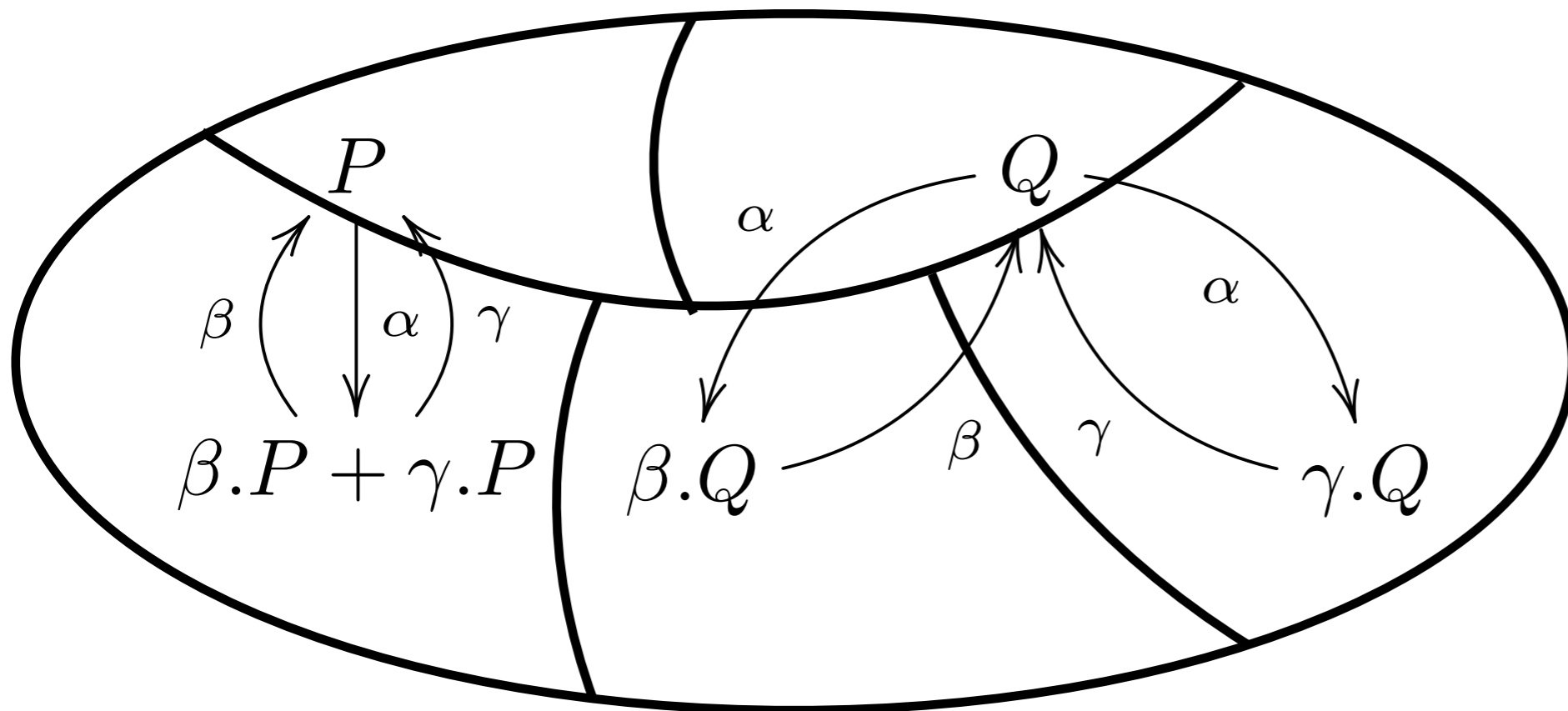
$$P \xrightarrow{\alpha} [\beta.P + \gamma.Q]$$
$$Q \xrightarrow{\alpha} [\beta.Q], [\gamma.Q]$$

Le transizioni α di P e Q finiscono in partizioni diverse

Esempio

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

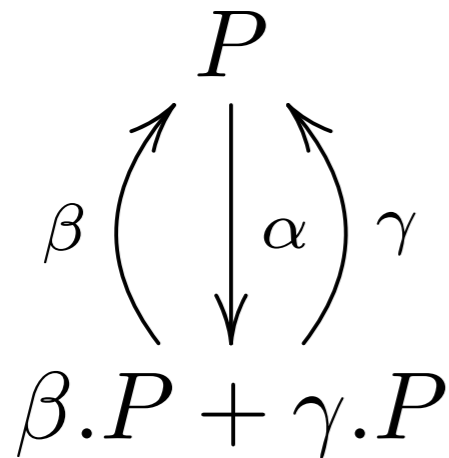
$$\mathbf{R}_2 = \{ \{P\}, \{Q\}, \{\beta.P + \gamma.P\}, \{\beta.Q\}, \{\gamma.Q\} \}$$



Esempio

Guardato!

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

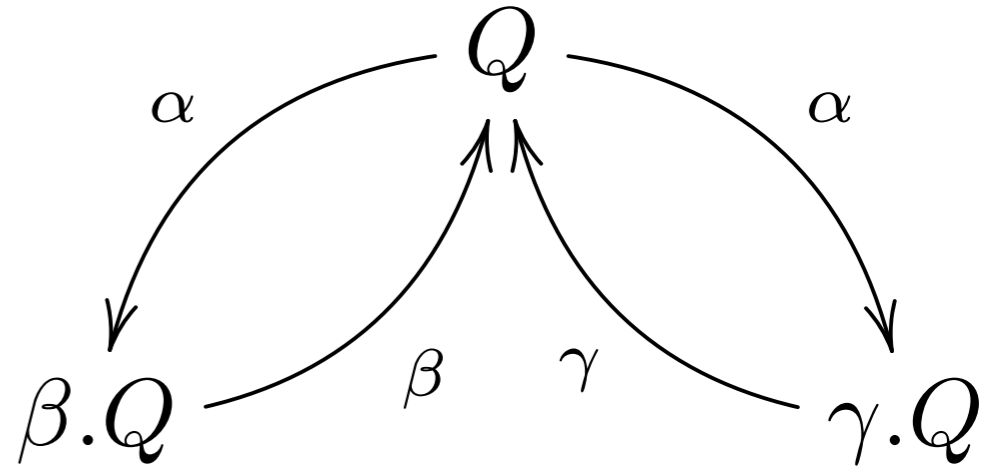


$$P \stackrel{?}{\simeq} Q$$



Guardato!

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



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

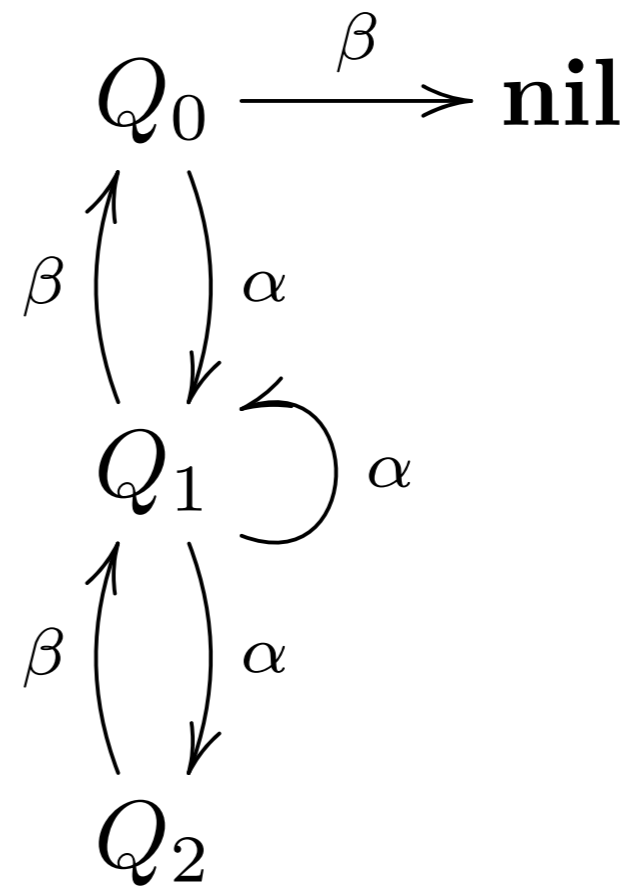
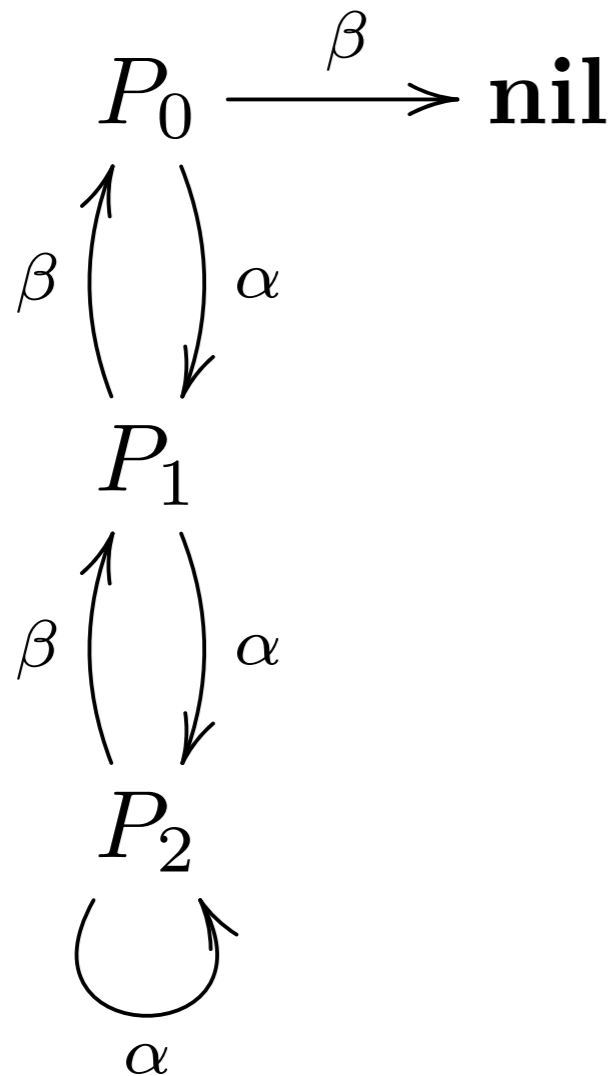
$$\mathbf{R}_1 = \{ \{P, Q\}, \{\beta.P + \gamma.P\}, \{\beta.Q\}, \{\gamma.Q\} \}$$

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

Solo partizioni con un elemento, ci possiamo fermare $P \neq Q$

Esercizio

finitamente ramificato! $P_0 \stackrel{?}{\simeq} Q_0$ finitamente ramificato!



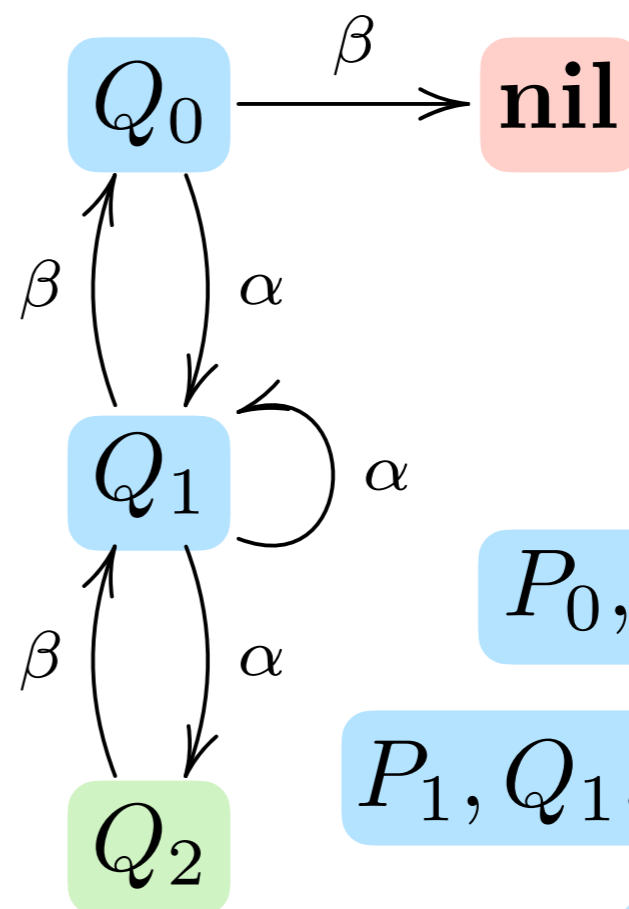
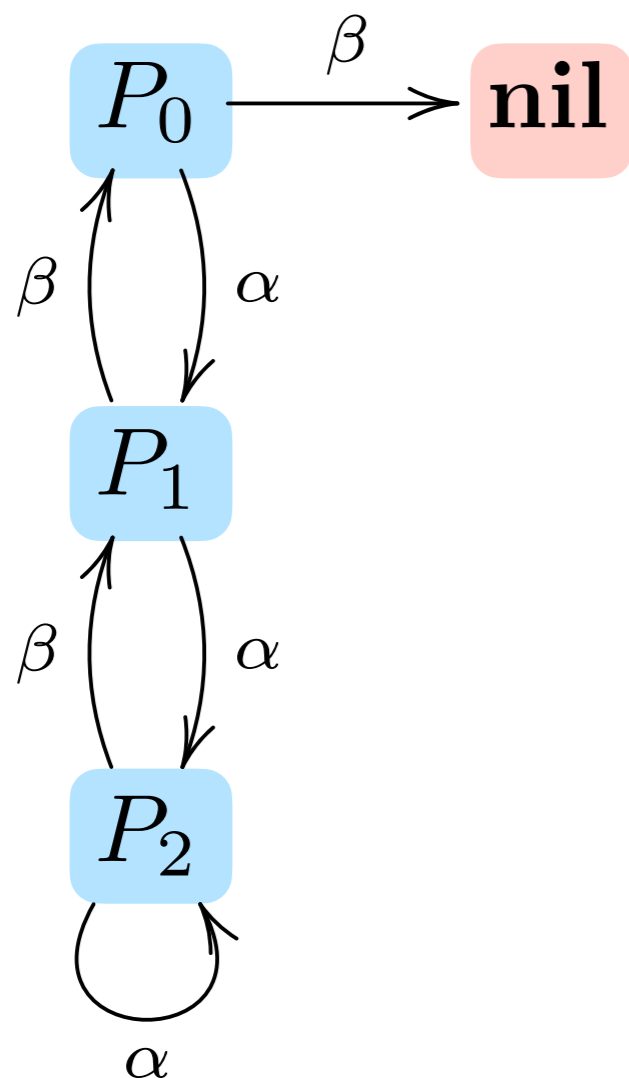
$\mathbf{nil} \not\rightarrow$
 $Q_2 \xrightarrow{\beta}$

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

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

Esercizio

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



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

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

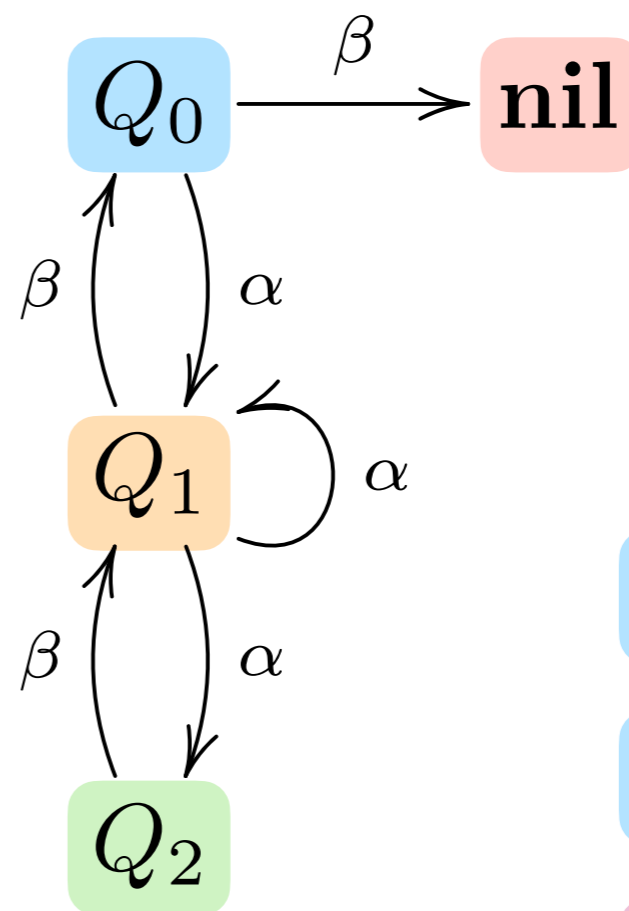
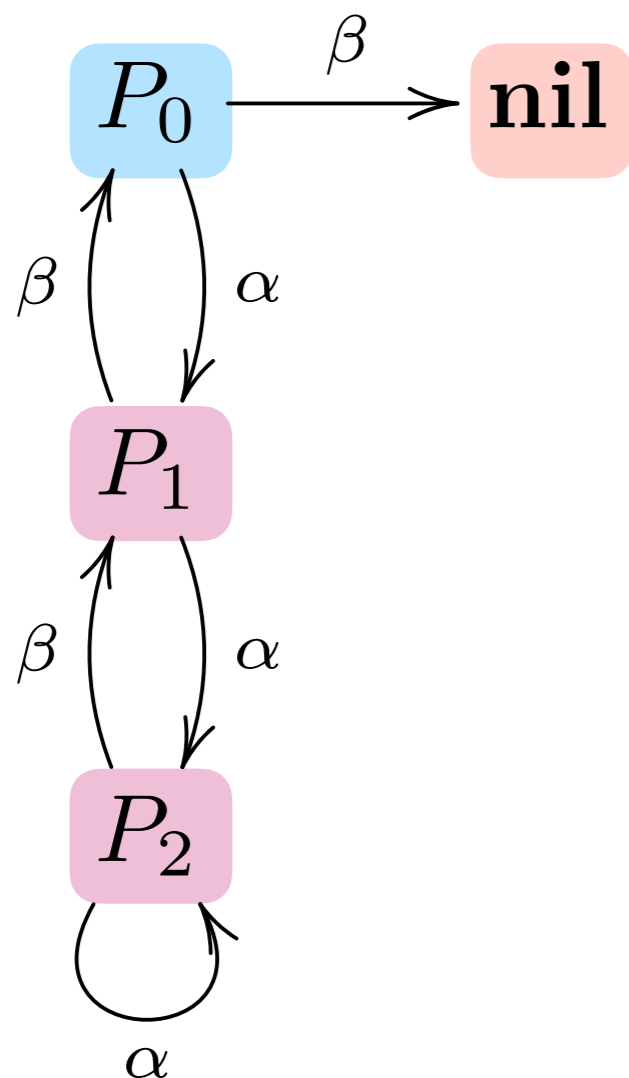
$$Q_1 \xrightarrow{\alpha} [Q_2]$$

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

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

Esercizio

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



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

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

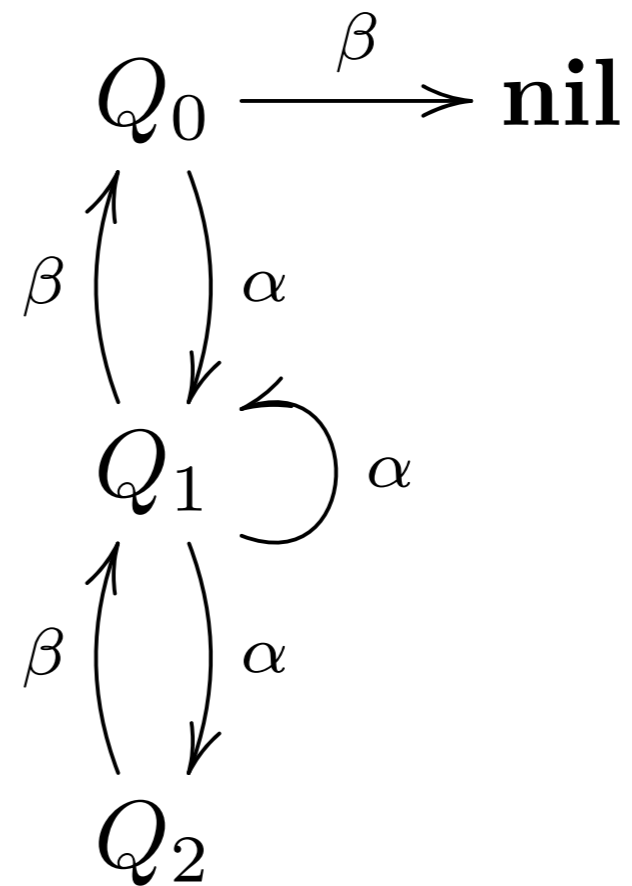
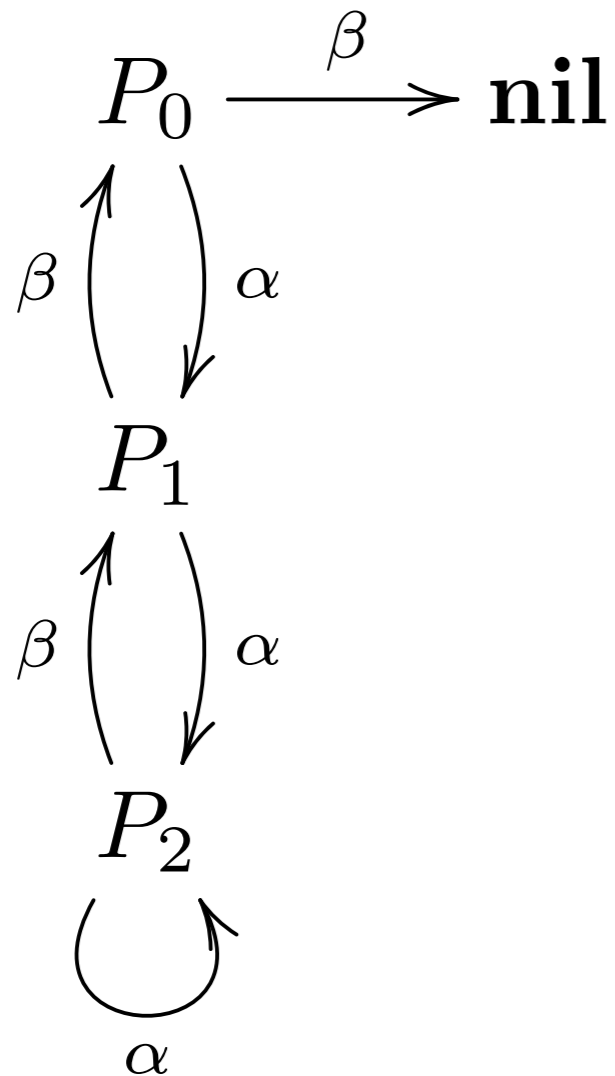
$$P_1 \xrightarrow{\beta} [P_0]$$

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

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

Esercizio

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

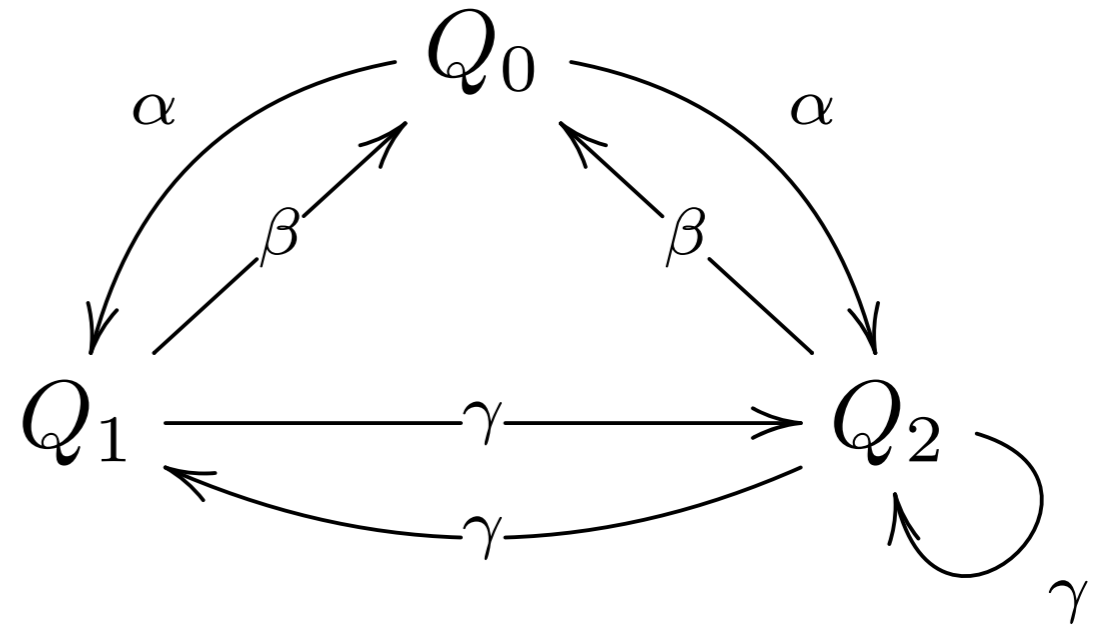
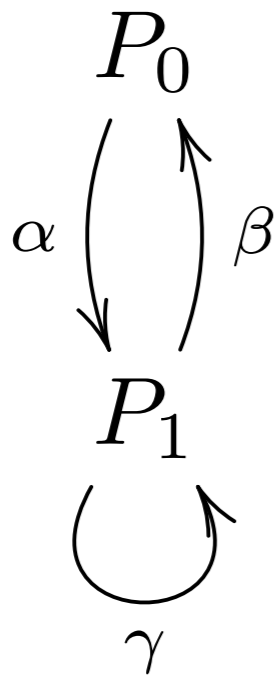


$$P_0 \not\approx Q_0$$

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

Esercizio

finitamente ramificato! $P_0 \stackrel{?}{\simeq} Q_0$ finitamente ramificato!



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

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

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

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

Non ci sono più ragioni per discriminare!

processi non guardati?

E il caso generale? (processi non guardati)

ogni power set ordinato per inclusione definisce un reticolo completo

Reticolo completo: (D, \sqsubseteq) OP tale che

ogni $X \subseteq D$ ha un least upper bound $\bigsqcup X$

ogni $X \subseteq D$ ha un greatest lower bound $\bigsqcap X$

ha il bottom e top $\perp = \bigsqcap D$ $\top = \bigsqcup D$

TH. [Knaster-Tarski] (D, \sqsubseteq) reticolo completo

$f : D \rightarrow D$ monotona

ha un least e un greatest fixpoint

$d_{\min} \triangleq \bigsqcap \{d \in D \mid f(d) \sqsubseteq d\}$ e' un least fixpoint
glb pre punti fissi

$d_{\max} \triangleq \bigsqcup \{d \in D \mid d \sqsubseteq f(d)\}$ e' un greatest fixpoint
lub post punti fissi

il minimo e il massimo punto fisso esistono... ma come calcolarli?