Linguaggi di Programmazione

Roberta Gori

Logica di Hennessy-Milner II.6

CCS: sintassi

$$
\begin{array}{lcll}
p, q & ::= & \mathbf{nil} & \text{processo inattivo} \\
 & | & x & \text{variabile di processo (per la ricorsione)} \\
 & | & \mu.p & \text{prefisso azione} \\
 & | & p\backslash\alpha & \text{canale ristretto} \\
 & | & p[\phi] & \text{rietichettatura del canale} \\
 & | & p + q & \text{scelta nondeterministica (somma)} \\
 & | & p|q & \text{composizione parallela} \\
 & | & \mathbf{rec}\ x.\ p & \text{ricorsione}
\end{array}
$$

(gli operatori sono elencati in ordine di precedenza)

CCS semantica

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

HML

Logica di Hennessy-Milner

Equivalenza logica

Prendiamo un altro approccio all'equivalenza

definiamo una logica (insieme di formule)

un processo può soddisfare o meno una formula

due processi sono (logicamente) equivalenti

quando soddisfano esattamente le stesse formule

le formule devono descrivere le proprietà comportamentali dei processi

la capacità/incapacità di eseguire transizioni
(logica modale: possibilmente, necessariamente)

allora, possiamo comporre le formule con i soliti operatori

Logica di Hennessy-Milner

Presentiamo gli operatori di base

multi-modale:
gli operatori modali sono parametrizzati da azioni

nessuna negazione:
anche la controparte di una formula può essere scritta come una formula

nessuna ricorsione:
ogni formula esprime proprietà su un numero finito di passi in avanti

HML: sintassi

F, G	$::=$	$\mathbf{tt}$	true
	$\mid$	$\mathbf{ff}$	false
	$\mid$	$\bigwedge_{i \in I} F_i$	conjunction
	$\mid$	$\bigvee_{i \in I} F_i$	disjunction
	$\mid$	$\Diamond_\mu F$	diamond operator $\langle \mu \rangle F$
	$\mid$	$\Box_\mu F$	box operator $[\mu] F$

$\mathcal{L}$ insieme di tutte le formule

HML: semantica

$p \models F$ si legge " *p* soddisfa *F* "

definito induttivamente sulla struttura della formula

$p \models \mathbf{tt}$ tutti i processi soddisfano true
(nessun processo soddisfa false)

$p \models \bigwedge_{i \in I} F_i \quad \text{iff} \quad \forall i \in I.\ p \models F_i$ *p* soddisfa tutte le F_i

$p \models \bigvee_{i \in I} F_i \quad \text{iff} \quad \exists i \in I.\ p \models F_i$ *p* almeno una delle F_i

$p \models \Diamond_\mu F \quad \text{iff} \quad \exists p'.\ p \xrightarrow{\mu} p' \wedge p' \models F$ *p* fa un μ-passo
e poi soddisfa *F*

$p \models \Box_\mu F \quad \text{iff} \quad \forall p'.\ p \xrightarrow{\mu} p' \Rightarrow p' \models F$ *F* e' soddisfatta dopo ogni
μ-passo di *p*

Esempi

$\Diamond_\alpha \mathbf{tt}$ soddisfatta da qualsiasi processo che può fare un α-passo

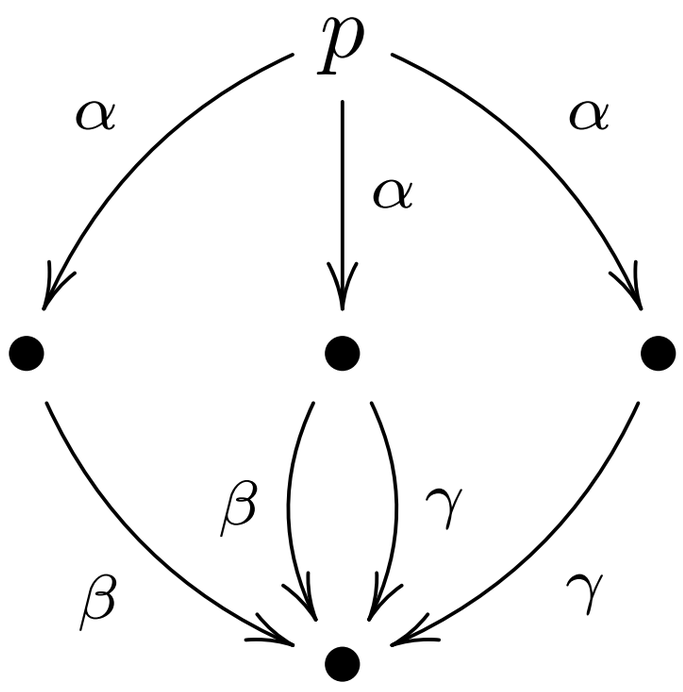
$\Box_\beta \mathbf{ff}$ soddisfatta da qualsiasi processo che non può fare nessun β-passo

$\Diamond_\alpha \mathbf{ff}$ stesso che **ff**
se un processo non può fare α, la modalità non vale (deve esistere una transizione con modalità' α)
se esiste tale transizione dopo non puo' soddisfare **ff**

$\Box_\beta \mathbf{tt}$ stesso che **tt**
se un processo non può fare β la modalità è banalmente valida
se un processo fa β la sua continuazione soddisfa **tt**

$\Diamond_\alpha(\Diamond_\beta \mathbf{tt} \wedge \Box_\gamma \mathbf{ff})$ soddisfatto da qualsiasi processo che può fare α
e raggiungere un processo che può fare β ma non γ

Esempi



$$p \overset{?}{\models} \Diamond_\alpha \mathbf{tt}$$

$$p \overset{?}{\models} \Box_\alpha \Diamond_\beta \mathbf{tt}$$

$$p \overset{?}{\models} \Diamond_\alpha \Box_\beta \mathbf{ff} \ \wedge \ \Diamond_\alpha \Box_\gamma \mathbf{ff}$$

$$p \overset{?}{\models} \Box_\alpha (\Diamond_\beta \mathbf{tt} \vee \Diamond_\gamma \mathbf{tt})$$

$$p \overset{?}{\models} \Box_\alpha (\Diamond_\beta \mathbf{tt} \wedge \Diamond_\gamma \mathbf{tt})$$

$$p \overset{?}{\models} \Diamond_\alpha (\Diamond_\beta \mathbf{tt} \wedge \Diamond_\gamma \mathbf{tt})$$

Negazione

non presente nella sintassi, ma non necessaria

ogni formula F ha una formula inversa F^c tale che

$$\forall p.\ p \models F \quad \text{iff} \quad p \not\models F^c$$

F^c può essere definito per induzione strutturale

$$\mathbf{tt}^c \triangleq \mathbf{ff} \qquad \mathbf{ff}^c \triangleq \mathbf{tt}$$

$$(\bigwedge_{i \in I} F_i)^c \triangleq \bigvee_{i \in I} F_i^c \qquad (\bigvee_{i \in I} F_i)^c \triangleq \bigwedge_{i \in I} F_i^c$$

$$(\Diamond_\mu F)^c \triangleq \Box_\mu F^c \qquad (\Box_\mu F)^c \triangleq \Diamond_\mu F^c$$

esempio

$$(\Diamond_\alpha \mathbf{tt})^c = \Box_\alpha \mathbf{tt}^c = \Box_\alpha \mathbf{ff}$$

(posso fare α)^c =
non posso fare α

Sintassi estesa

$$A = \{\mu_1, ..., \mu_n\}$$

$$\Diamond_A F \triangleq \Diamond_{\mu_1} F \vee \cdots \vee \Diamond_{\mu_n} F = \bigvee_{i \in [1,n]} \Diamond_{\mu_i} F \qquad \Box_A F \triangleq \Box_{\mu_1} F \wedge \cdots \wedge \Box_{\mu_n} F = \bigwedge_{i \in [1,n]} \Box_{\mu_i} F$$

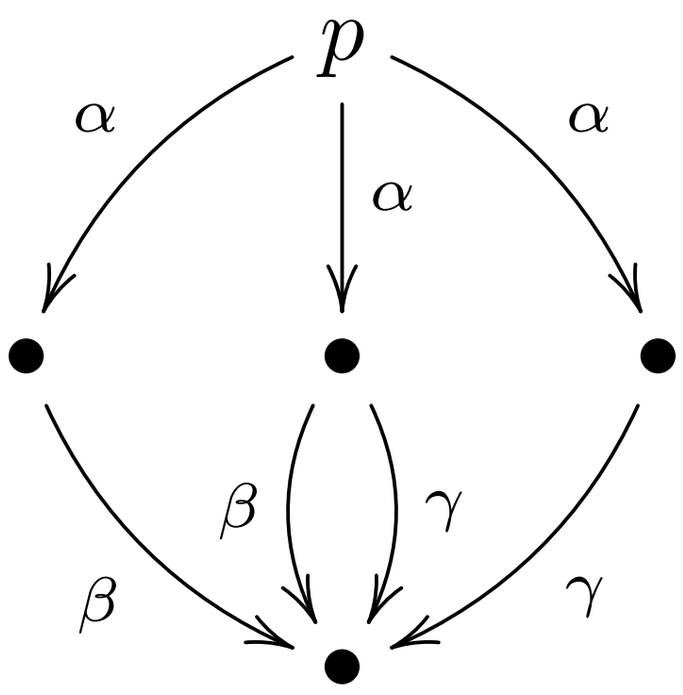
$$\Diamond_\emptyset F \triangleq \mathbf{ff} \qquad \Box_\emptyset F \triangleq \mathbf{tt}$$

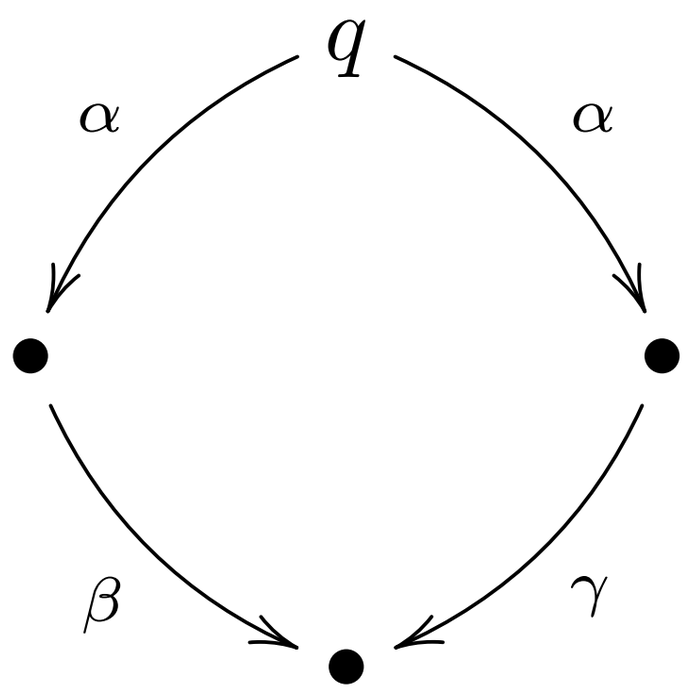
HML: Equivalenza logica

due processi sono equivalenti sse soddisfano le stesse formule

$$p \equiv_{\mathrm{HM}} q \quad \text{sse} \quad \forall F \in \mathcal{L}.\ (p \models F \ \Leftrightarrow \ q \models F)$$

$$p \overset{?}{\equiv}_{\mathrm{HM}} q$$





$p \models F$ $\qquad F \triangleq \Diamond_\alpha(\Diamond_\beta \mathbf{tt} \wedge \Diamond_\gamma \mathbf{tt})$ $\qquad q \not\models F$

$p \not\models F^c$ $\qquad F^c \triangleq \Box_\alpha(\Box_\beta \mathbf{ff} \vee \Box_\gamma \mathbf{ff})$ $\qquad q \models F^c$

Bis. forte come equiv. logica

TH. per qualsiasi processo finitamente ramificato *p,q*

$$p \simeq q \quad \text{sse} \quad p \equiv_{\text{HM}} q$$

(prova omessa)

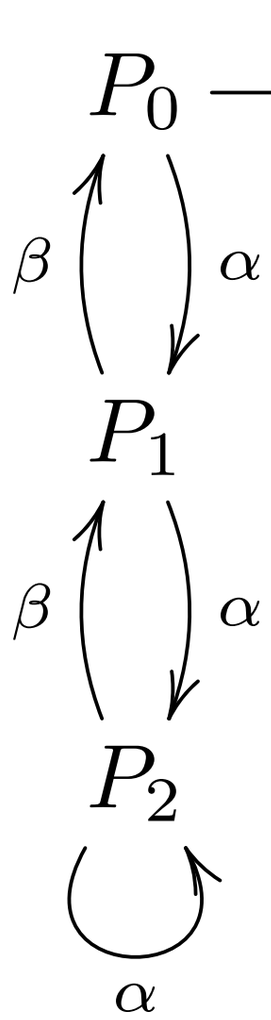
consequenze:

per dimostrare che due processi sono fortemente bisimili: esibire una relazione di bisimulazione forte che li metta in relazione

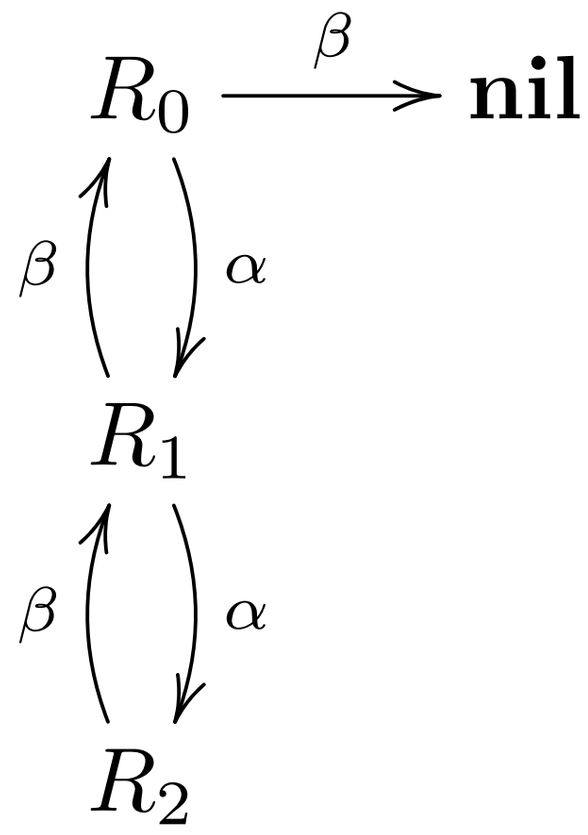
per dimostrare che due processi non sono fortemente bisimili: esibire una formula HML che li distingua

Esercizio

trovare una formula HML che distingua i due processi



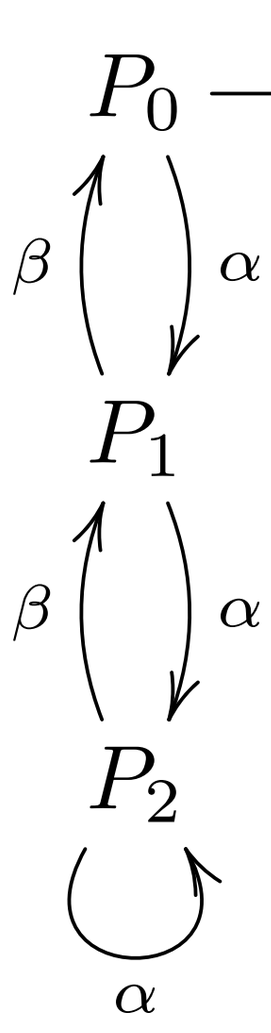
$P_0 \not\approx R_0$



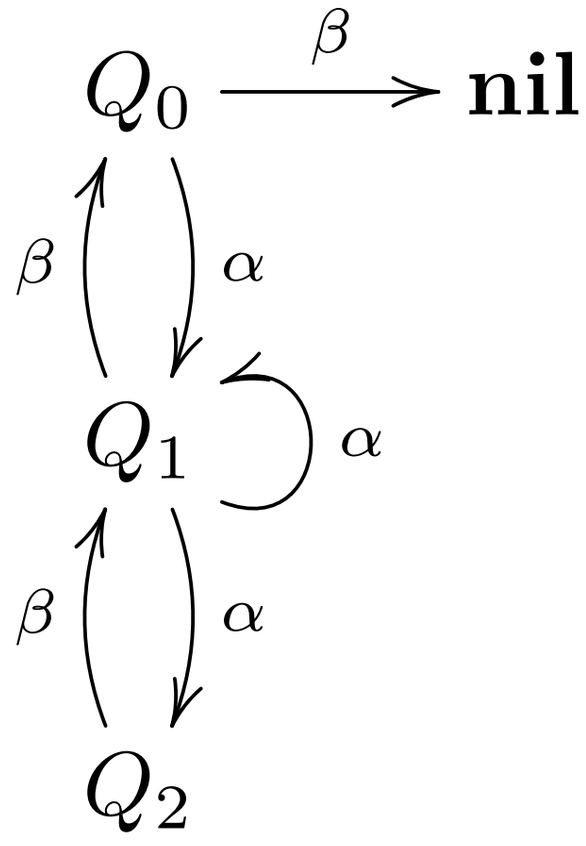
$P_0 \models F$ $\qquad F \triangleq \Diamond_\alpha \Diamond_\alpha \Diamond_\alpha \mathbf{tt}$ $\qquad R_0 \not\models F$

Esercizio

trovare una formula HML che distingua i due processi



$P_0 \not\approx Q_0$



$P_0 \models F$ $\qquad F \triangleq \Diamond_\alpha \Box_\alpha \Diamond_\alpha \mathbf{tt}$ $\qquad Q_0 \not\models F$