Linguaggi di Programmazione

Roberta Gori

CCS semantica astratta-11.4

CCS
isomorfismo tra grafi

CCS sintassi

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

(gli operatori sono elencati in ordine di precedenza)

CCS semantica operazionale

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

Isomorfismo tra LTS

processi sintatticamente diversi
possono mostrare esattamente lo stesso comportamento

$\mathbf{nil} \qquad \mathbf{nil}\backslash\alpha \qquad \mathbf{nil}[\phi]$

$p \qquad p + \mathbf{nil} \qquad p + p \qquad p|\mathbf{nil}$

$p + q \qquad q + p$

$p|q \qquad q|p$

i loro LTS sono diversi (stati sintatticamente diversi)

l'isomorfismo tra grafi astrae dagli stati

Isomorfismo tra grafi

$$G = (V, E) \qquad G' = (V', E')$$

$$v \xrightarrow{\mu} w \quad \Leftrightarrow \quad f(v) \xrightarrow{\mu} f(w)$$

$f : V \to V'$ bigettiva

$$\begin{tikzcd}
\alpha \arrow[d, "\alpha"'] \arrow[r, dash, dashed, bend left] & \alpha + \mathbf{nil} \arrow[d, "\alpha"'] \arrow[r, dash, dashed, bend left] & \alpha + \alpha \arrow[d, "\alpha"'] \arrow[r, dash, dashed, bend left] & \alpha | \mathbf{nil} \arrow[d, "\alpha"'] \\
\mathbf{nil} \arrow[r, dash, dashed, bend right] & \mathbf{nil} \arrow[r, dash, dashed, bend right] & \mathbf{nil} \arrow[r, dash, dashed, bend right] & \mathbf{nil} | \mathbf{nil}
\end{tikzcd}$$

Isomorfismo tra grafi

$$G = (V, E) \qquad\qquad G' = (V', E')$$

$f : V \to V'$ bigettiva

$$v \xrightarrow{\mu} w \quad \Leftrightarrow \quad f(v) \xrightarrow{\mu} f(w)$$

$$\begin{tikzcd}
\alpha + \beta.\gamma \arrow[d, "\beta"] \arrow[dd, bend right, "\alpha"'] \arrow[rr, dash, dotted, bend left] & & \beta.\gamma + \alpha \arrow[d, "\beta"'] \arrow[dd, bend left, "\alpha"] \\
\gamma \arrow[d, "\gamma"] \arrow[rr, dash, dotted, bend right] & & \gamma \arrow[d, "\gamma"'] \\
\mathbf{nil} \arrow[rr, dash, dotted, bend right] & & \mathbf{nil}
\end{tikzcd}$$

Isomorfismo tra grafi

$$G = (V, E) \qquad\qquad G' = (V', E')$$

$$f : V \to V' \quad \text{bigettiva}$$

$$v \xrightarrow{\mu} w \quad \Leftrightarrow \quad f(v) \xrightarrow{\mu} f(w)$$

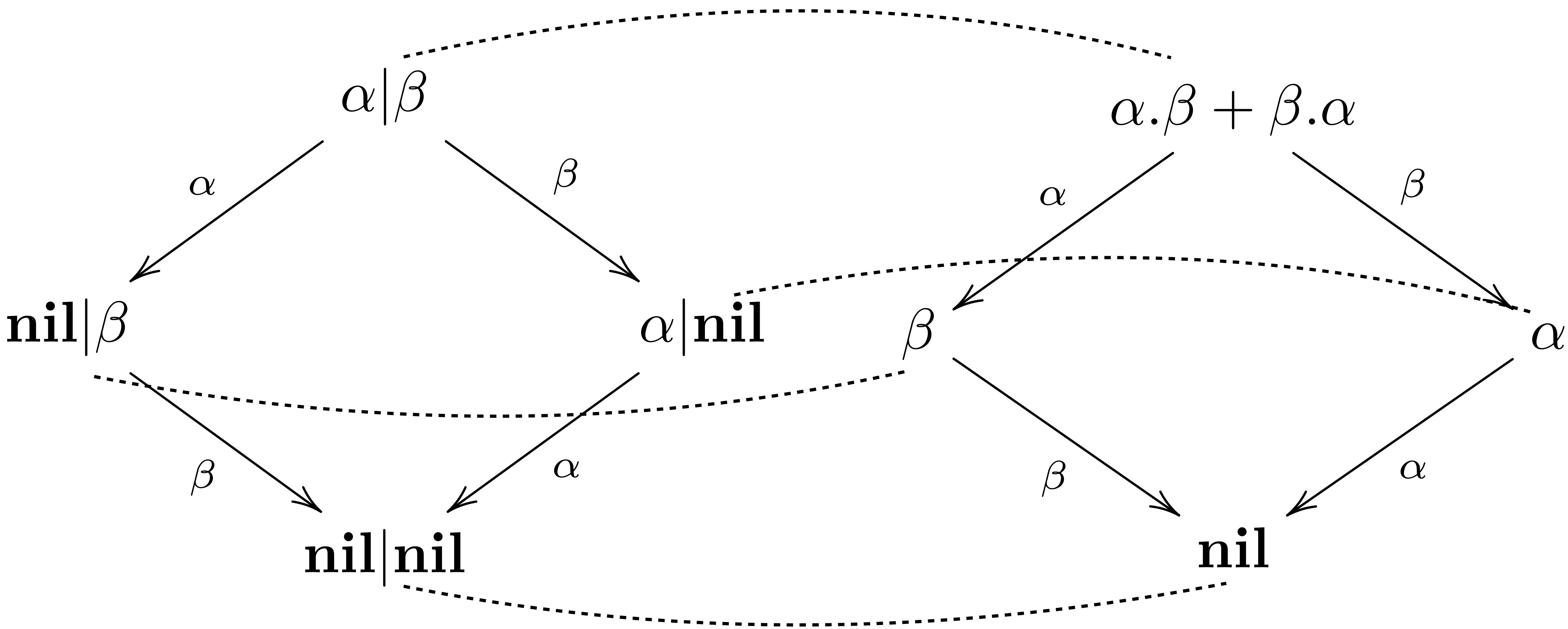
$$\begin{tikzcd}
 & \alpha|\overline{\alpha} \arrow[dl, "\alpha"'] \arrow[dd, "\tau"'] \arrow[dr, "\overline{\alpha}"] \arrow[rrr, dash, dashed, bend left] & & & \overline{\alpha}|\alpha \arrow[dl, "\overline{\alpha}"'] \arrow[dd, "\tau"'] \arrow[dr, "\alpha"] & \\
\mathbf{nil}|\overline{\alpha} \arrow[dr, "\overline{\alpha}"'] \arrow[rrrrr, dash, dashed, bend right] & & \alpha|\mathbf{nil} \arrow[dl, "\alpha"] \arrow[r, dash, dashed, bend left] & \mathbf{nil}|\alpha \arrow[dr, "\alpha"'] & & \overline{\alpha}|\mathbf{nil} \arrow[dl, "\overline{\alpha}"] \\
 & \mathbf{nil}|\mathbf{nil} \arrow[rrr, dash, dashed, bend right] & & & \mathbf{nil}|\mathbf{nil} &
\end{tikzcd}$$

Isomorfismo tra grafi

$$G = (V, E) \qquad G' = (V', E')$$

$$f : V \to V' \quad \text{bigettiva}$$

$$v \xrightarrow{\mu} w \quad \Leftrightarrow \quad f(v) \xrightarrow{\mu} f(w)$$



Equivalenti o no?

$$
\begin{tikzcd}
B^2_0 \arrow[d, bend left, "in"] \\
B^2_1 \arrow[u, bend left, "\overline{out}"] \arrow[d, bend left, "in"] \\
B^2_2 \arrow[u, bend left, "\overline{out}"]
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
& B^1_0|B^1_0 \arrow[dl, bend left, "in"] \arrow[dr, bend left, "in"] & \\
B^1_1|B^1_0 \arrow[ur, bend left, "\overline{out}"] \arrow[dr, bend left, "in"] & & B^1_0|B^1_1 \arrow[ul, bend left, "\overline{out}"] \arrow[dl, bend left, "in"] \\
& B^1_1|B^1_1 \arrow[ul, bend left, "\overline{out}"] \arrow[ur, bend left, "\overline{out}"] &
\end{tikzcd}
$$

intuitivamente equivalenti, ma non isomorfi!

Iso e' troppo concreta

se due processi hanno LTS isomorfi
allora devono essere considerati equivalenti

L'isomorfismo tra grafi cattura alcune equivalenze interessanti

ma e' abbastanza?

$P \triangleq \alpha.P$

$$\begin{tikzcd} \mathbf{rec}\ x.\ \alpha.x \arrow[loop below, "\alpha"] \end{tikzcd}$$

$Q \triangleq \alpha.\alpha.Q$

$$\begin{tikzcd} \mathbf{rec}\ x.\ \alpha.\alpha.x \arrow[d, bend right, "\alpha"'] \\ \alpha.\mathbf{rec}\ x.\ \alpha.\alpha.x \arrow[u, bend right, "\alpha"'] \end{tikzcd}$$

$\alpha.P$

$$\begin{tikzcd} \alpha.\mathbf{rec}\ x.\ \alpha.x \arrow[d, "\alpha"'] \\ \mathbf{rec}\ x.\ \alpha.x \arrow[loop below, "\alpha"] \end{tikzcd}$$

intuitivamente equivalenti, ma non isomorfi!

CCS
equivalenza tra le tracce

Equivalenza per traccia

nella teoria degli automi: equivalenza tra linguaggi

nozione di tracce (finite) $p \xrightarrow{\mu_1 \mu_2 \cdots \mu_k} q$

$$p = p_0 \xrightarrow{\mu_1} p_1 \xrightarrow{\mu_2} \cdots \xrightarrow{\mu_k} p_k = q$$

semantica delle tracce (finite) di un processo

$$\mathcal{T}(p) = \{\mu_1 \mu_2 \cdots \mu_k \mid \exists q.\ p \xrightarrow{\mu_1 \mu_2 \cdots \mu_k} q\}$$

due processi sono equivalenti per traccia se hanno la stessa semantica delle tracce

$$p \equiv_{\text{tr}} q \text{ iff } \mathcal{T}(p) = \mathcal{T}(q)$$

l'equivalenza per traccia è una buona nozione per sistemi concorrenti?

Equivalenza per traccia

isomorfismo tra grafi implica l'equivalenza per traccia

conserviamo tutte le equivalenze utili viste prima

$$p \equiv_{\mathsf{tr}} p + \mathbf{nil} \equiv_{\mathsf{tr}} p + p \equiv_{\mathsf{tr}} p|\mathbf{nil}$$

$$p + q \equiv_{\mathsf{tr}} q + p$$

$$p|q \equiv_{\mathsf{tr}} q|p$$

la semantica delle tracce è chiusa per prefissi
(se una traccia è presente, sono presenti anche tutti i suoi prefissi)

Equivalenza per traccia

$$\begin{tikzcd}
\mathbf{rec}\ x.\ \alpha.x \arrow[loop below, "\alpha"]
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathbf{rec}\ x.\ \alpha.\alpha.x \arrow[d, bend right, "\alpha"'] \\
\alpha.\mathbf{rec}\ x.\ \alpha.\alpha.x \arrow[u, bend right, "\alpha"']
\end{tikzcd}
\qquad
\begin{tikzcd}
\alpha.\mathbf{rec}\ x.\ \alpha.x \arrow[d, "\alpha"'] \\
\mathbf{rec}\ x.\ \alpha.x \arrow[loop below, "\alpha"]
\end{tikzcd}$$

$$\mathcal{T}(\mathbf{rec}\ x.\ \alpha.x) = \{\alpha^n \ \mid \ n \in \mathbb{N}\}$$

$$\mathcal{T}(\mathbf{rec}\ x.\ \alpha.\alpha.x) = \{\alpha^n \ \mid \ n \in \mathbb{N}\}$$

$$\mathcal{T}(\alpha.\mathbf{rec}\ x.\ \alpha.x) = \{\alpha^n \ \mid \ n \in \mathbb{N}\}$$

Equivalenza per traccia

$$\begin{tikzcd}
B_0^2 \arrow[d, bend left, "in"] \\
B_1^2 \arrow[u, bend left, "\overline{out}"] \arrow[d, bend left, "in"] \\
B_2^2 \arrow[u, bend left, "\overline{out}"]
\end{tikzcd}$$

$$\begin{tikzcd}
 & B_0^1 \mid B_0^1 \arrow[dl, bend left, "in"] \arrow[dr, bend right, "in"'] & \\
B_1^1 \mid B_0^1 \arrow[ur, bend left, "\overline{out}"] \arrow[dr, bend left, "in"] & & B_0^1 \mid B_1^1 \arrow[ul, bend right, "\overline{out}"'] \arrow[dl, bend right, "in"'] \\
 & B_1^1 \mid B_1^1 \arrow[ul, bend left, "\overline{out}"] \arrow[ur, bend right, "\overline{out}"'] &
\end{tikzcd}$$

$$\mathcal{T}(B_0^2) = \mathcal{T}(B_0^1 | B_0^1)$$

tentativo di descrizione:
0 ≤ #*in* - #*out* ≤ 2 (per ogni prefisso)

2 distributori automatici

$$coin.(\overline{coffee} + \overline{tea}) \qquad\qquad coin.\overline{coffee} + coin.\overline{tea}$$

$$\begin{tikzcd}
coin.(\overline{coffee} + \overline{tea}) \arrow[d, "coin"'] \\
\overline{coffee} + \overline{tea} \arrow[d, bend right, "\overline{coffee}"'] \arrow[d, bend left, "\overline{tea}"] \\
\mathbf{nil}
\end{tikzcd}
\qquad
\begin{tikzcd}
& coin.\overline{coffee} + coin.\overline{tea} \arrow[dl, "coin"'] \arrow[dr, "coin"] & \\
\overline{coffee} \arrow[dr, "\overline{coffee}"'] & & \overline{tea} \arrow[dl, "\overline{tea}"] \\
& \mathbf{nil} &
\end{tikzcd}$$

non isomorfi, ma equivalenti per traccia

La visione del cliente

quale distributore automatico preferiresti?

$$\begin{tikzcd}
coin.(\overline{coffee} + \overline{tea}) \arrow[d, "coin"'] \\
\overline{coffee} + \overline{tea} \arrow[d, bend right, "\overline{coffee}"'] \arrow[d, bend left, "\overline{tea}"] \\
\mathbf{nil}
\end{tikzcd}$$

inserisci una moneta, poi scegli la bevanda

La visione del cliente

quale distributore automatico preferiresti?

$$
\begin{tikzcd}
 & coin.\overline{coffee} + coin.\overline{tea} \arrow[dl, "coin"'] \arrow[dr, "coin"] & \\
\overline{coffee} \arrow[dr, "\overline{coffee}"'] & & \overline{tea} \arrow[dl, "\overline{tea}"] \\
 & \mathbf{nil} &
\end{tikzcd}
$$

inserire una moneta, poi ottenere la bevanda scelta dalla macchina

Distributori ricorsivi

$$M_2 \triangleq coin.\overbrace{\overline{coffee}.M_2}^{M_2'} + coin.\overbrace{\overline{tea}.M_2}^{M_2''}$$

$$M_1 \triangleq coin.\overbrace{(\overline{coffee}.M_1 + \overline{tea}.M_1)}^{M_1'}$$

$$M_1 \equiv_{\mathsf{tr}} M_2$$

$$\begin{tikzcd}
M_1 \arrow[d, "coin"'] \\
M_1' \arrow[u, bend left=60, "\overline{coffee}"] \arrow[u, bend right=60, "\overline{tea}"']
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
M_2 \arrow[r, "coin"'] \arrow[d, "coin"'] & M_2'' \arrow[l, bend right=60, "\overline{tea}"'] \\
M_2' \arrow[u, bend left=60, "\overline{coffee}"] &
\end{tikzcd}$$

la visione del sistema

$$M_1 \triangleq coin.\overbrace{(\overline{coffee}.M_1 + \overline{tea}.M_1)}^{M_1'}$$

$$M_1 \equiv_{\mathsf{tr}} M_2 \qquad M_2 \triangleq coin.\overbrace{\overline{coffee}.M_2}^{M_2'} + coin.\overbrace{\overline{tea}.M_2}^{M_2''}$$

$$C \triangleq \overline{coin}.\overbrace{coffee.C}^{C'} \qquad S \triangleq \{coin, coffee, tea\}$$

$$P \triangleq (C|M_1)\backslash S \qquad P \equiv_{\mathsf{tr}} Q \qquad Q \triangleq (C|M_2)\backslash S$$

$$\begin{tikzcd}
P \arrow[d, bend right, "\tau"'] \\
(C'|M_1')\backslash S \arrow[u, bend right, "\tau"']
\end{tikzcd}
\qquad
\begin{tikzcd}
& Q \arrow[dl, bend right, "\tau"'] \arrow[dr, bend left, "\tau"] & \\
(C'|M_2')\backslash S \arrow[ur, bend right, "\tau"'] & & \boxed{(C'|M_2'')\backslash S}
\end{tikzcd}$$

deadlock!

prossimamente: bisimilarita'

l'isomorfismo tra grafi distingue troppi processi

l'equivalenza delle tracce identifica troppi processi

abbiamo bisogno di qualche nozione di equivalenza tra i due

introduciamo la nozione di *strong bisimilarity*

come un gioco

come un punto fisso

come un equivalenza logica

da tenere a mente: due processi sono equivalenti a meno che abbiamo delle buone ragioni per distinguerli