# Modelli di Sistemi Sequenziali e Concorrenti

**esercizi**

Stefano Martina
stefano.martina@stud.unifi.it
Università degli Studi di Firenze
Scuola di Scienze Matematiche, Fisiche e Naturali
Corso magistrale di Informatica

21 febbraio 2016

# Indice

## Esercizio 3.15

### Testo:

Dimostrare che, per ogni espressione aritmetica $E$ descritta nella Sezione 3.3,

$$E \rightarrow E_1,\ E \rightarrow E_2 \quad \text{implica} \quad E_1 \twoheadrightarrow n,\ E_2 \twoheadrightarrow n$$

### Soluzione:

**Premessa:** Si noti che se vale la tesi, allora vale anche che $E \twoheadrightarrow n$, infatti per l'equivalenza tra semantica di computazione e semantica di valutazione:

$$\begin{aligned}
&E \rightarrow E_1 \twoheadrightarrow n\\
&E \rightarrow E_1 \xrightarrow{*} n\\
&E \xrightarrow{*} n\\
&E \twoheadrightarrow n
\end{aligned}$$

Oppure in modo analogo passando da $E_2$.

La dimostrazione procede per induzione sulla lunghezza dell'espressione $E$, o in modo equivalente sul numero di operatori in $E$.

**Caso base:** Per il caso base si considera un'espressione $E$ con un operatore (non possono esserci zero operatori altrimenti non si potrebbe applicare l'ipotesi), l'unica regola applicabile è la $(op)$ della semantica di computazione, e quindi se $E \rightarrow E_1$ e $E \rightarrow E_2$ allora $E_1 = E_2 = n$ e vale la tesi.

**Passo induttivo:** Per una generica espressione $E$ tale che $E \rightarrow E_1$ e $E \rightarrow E_2$, si può osservare che se è stata usata $(op)$ nel passo $E \rightarrow E_1$ significa che $E = m_1\ op\ m_2$ e quindi ha un solo operatore e siamo nel caso base.

Se è stata usata $(redl)$ nel passo $E \rightarrow E_1$ significa che $E$ non è in una forma in cui è possibile applicare $(op)$ quindi nel passo $E \rightarrow E_2$ è possibile solo applicare $(redl)$ o $(redr)$. Nel Primo caso significa che $E_1 = E_2$ e non è rilevante per la dimostrazione, quindi assumiamo che sia stata usata $(redr)$. Chiamando $E \equiv E_a\ op\ E_b$, $E_1 \equiv E'_a\ op\ E_b$, $E_2 \equiv E_a\ op\ E'_b$ si hanno:

$$\frac{E_a \rightarrow E'_a}{E_a\ op\ E_b \rightarrow E'_a\ op\ E_b}\ (redl)$$

$$\frac{E_b \rightarrow E'_b}{E_a\ op\ E_b \rightarrow E_a\ op\ E'_b}\ (redr)$$

Si noti che è stato possibile escludere che $E_1 = k_1$ o $E_2 = k_2$ in quanto in questi casi si esclude che sia possibile applicare $(redl)$ e $(redr)$ contemporaneamente e si riduce ad un caso banale.

$E_a$ e $E_b$ sono più corte di $E$ quindi si può applicare l'ipotesi induttiva, in modo che per qualunque coppia di cammini intrapresi $E_a \rightarrow E_{a1}$, $E_a \rightarrow E_{a2}$, $E_b \rightarrow E_{b1}$, $E_b \rightarrow E_{b2}$, si ha che $E_{a1} \twoheadrightarrow m_a$, $E_{a2} \twoheadrightarrow m_a$, $E_{b1} \twoheadrightarrow m_b$,

$E_{b2} \twoheadrightarrow m_b$ e, per quanto detto nella premessa, anche $E_a \twoheadrightarrow m_a$ e $E_b \twoheadrightarrow m_b$. Di conseguenza valgono:

$$\begin{aligned}
E_a &\twoheadrightarrow m_a \\
E_b &\twoheadrightarrow m_b \\
E'_a &\twoheadrightarrow m_a \\
E'_b &\twoheadrightarrow m_b
\end{aligned}$$

e si può applicare:

$$\frac{E'_a \twoheadrightarrow m_a \qquad E_b \twoheadrightarrow m_b}{E'_a \ op\ E_b \twoheadrightarrow n} \ (m_a \ op\ m_b = n)$$

$$\frac{E_a \twoheadrightarrow m_a \qquad E'_b \twoheadrightarrow m_b}{E_a \ op\ E'_b \twoheadrightarrow n} \ (m_a \ op\ m_b = n)$$

Dimostrando che $E_1 \twoheadrightarrow n$ e $E_2 \twoheadrightarrow n$, e quindi la tesi.

# Esercizio 4.11

## Testo:

Relativamente all'Esempio 4.10, si dimostri che se $s$ è una traccia di $F$ (ossia $F \Rightarrow 1$) allora $s$ è una traccia anche di $E$.

## Soluzione:

È necessario dimostrare che se $F \overset{s}{\Rightarrow} 1$ allora $E \overset{s}{\Rightarrow} 1$. La dimostrazione procede per induzione sulla lunghezza della traccia $s$.

**Caso base** $|s| = 0$ se la traccia è di lunghezza zero, allora essa è necessariamente $s = \varepsilon$, e quindi la tesi è dimostrata perché vale $E \overset{s}{\Rightarrow} 1$ per l'assioma $(Star_1)$.

**Passo induttivo** $|s| > 1$ si ha che la traccia può essere $s = as'$ oppure $s = bs'$, e senza perdere di generalità si assume il primo caso. Inoltre vale l'ipotesi induttiva che se $F \overset{s'}{\Rightarrow} 1$ allora $E \overset{s'}{\Rightarrow} 1$.

Vale che

$$F \overset{a}{\rightarrow} 1; a^*; F \overset{\varepsilon}{\rightarrow} a; F \overset{\varepsilon}{\rightarrow} F \overset{\varepsilon}{\rightarrow} 1; F \overset{\varepsilon}{\rightarrow} F \overset{\varepsilon}{\rightarrow} 1$$

o in altri termini valgono:

$$\begin{array}{ccccc}
F & \overset{a}{\Rightarrow} & 1; a^*; F & \overset{\varepsilon}{\Rightarrow} & F \\
F & \overset{a}{\Rightarrow} & a; F & \overset{\varepsilon}{\Rightarrow} & F \\
F & \overset{a}{\Rightarrow} & F & \overset{\varepsilon}{\Rightarrow} & F \\
F & \overset{a}{\Rightarrow} & 1; F & \overset{\varepsilon}{\Rightarrow} & F \\
F & \overset{a}{\Rightarrow} & 1 & &
\end{array}$$

Che sono anche le uniche transizioni che $F$ può eseguire tramite $a$, giustificate dalle:

$$\cfrac{\cfrac{\cfrac{\cfrac{}{a \xrightarrow{a} 1}\ (Atom)}{a^* \xrightarrow{a} 1; a^*}\ (Star_2)}{a^* + b^* \xrightarrow{a} 1; a^*}\ (Sum_1)}{(a^* + b^*)^* \xrightarrow{a} 1; a^*; (a^* + b^*)^*}\ (Star_2)$$

$$\cfrac{\cfrac{}{1 \xrightarrow{\varepsilon} 1}\ (Tic)}{1; a^*; (a^* + b^*)^* \xrightarrow{\varepsilon} a^*; (a^* + b^*)^*}\ (Seq_2)$$

$$\cfrac{\cfrac{}{a* \xrightarrow{\varepsilon} 1}\ (Star_1)}{a^*; (a^* + b^*)^* \xrightarrow{\varepsilon} (a^* + b^*)^*}\ (Seq_2)$$

$$\cfrac{\cfrac{\cfrac{}{a^* \xrightarrow{\varepsilon} 1}\ (Star_1)}{a^* + b^* \xrightarrow{\varepsilon} 1}\ (Sum_1)\ \text{(o similmente } (Sum_2))}{(a^* + b^*)^* \xrightarrow{\varepsilon} 1; (a^* + b^*)^*}\ (Star_2)$$

$$\cfrac{\cfrac{}{1 \xrightarrow{\varepsilon} 1}\ (Tic)}{1; (a^* + b^*)^* \xrightarrow{\varepsilon} (a^* + b^*)^*}\ (Seq_2)$$

$$\cfrac{}{(a^* + b^*)^* \xrightarrow{\varepsilon} 1}\ (Star_1)$$

Per ipotesi: $F \xRightarrow{as'} 1$, e per quanto detto prima, dopo aver fatto una $\xRightarrow{a}$ si ha che vale $X \xRightarrow{s'} 1$ con $X$ che può essere uno dei:

$1; a^*; F$
$a; F$
$F$
$1; F$
$1$

Nell'ultimo caso si ha che necessariamente $s' = \varepsilon$ e $s = a$, la tesi vale perché esiste la computazione $E \xrightarrow{a} 1; (a + b)^* \xrightarrow{\varepsilon} E \xrightarrow{\varepsilon} 1$ e quindi $E \xRightarrow{a} 1$. Tale computazione è giustificata dalle regole:

$$\dfrac{\dfrac{\dfrac{}{a \stackrel{a}{\to} 1}\ (Atom)}{a+b \stackrel{a}{\to} 1}\ (Sum_1)}{(a+b)^* \stackrel{a}{\to} 1;(a+b)^*}\ (Star_2)$$

$$\dfrac{\dfrac{}{1 \stackrel{\varepsilon}{\to} 1}\ (Tic)}{1;(a+b)^* \stackrel{\varepsilon}{\to} (a+b)^*}\ (Seq_2)$$

$$\dfrac{}{(a+b)^* \stackrel{\varepsilon}{\to} 1}\ (Star_1)$$

Per gli altri casi si ha che $X \stackrel{\varepsilon}{\Rightarrow} F$ è l'unica computazione possibile, quindi $F \stackrel{a}{\Rightarrow} X \stackrel{\varepsilon}{\Rightarrow} F \stackrel{s'}{\Rightarrow} 1$ e usando l'ipotesi induttiva vale che $E \stackrel{s'}{\Rightarrow} 1$. La tesi si ottiene giustificando che esiste $E \stackrel{a}{\Rightarrow} E$ e quindi $E \xRightarrow{as'=s} 1$. Per fare ciò si noti che esistono i passi $E \stackrel{a}{\to} 1; E \stackrel{\varepsilon}{\to} E$ già giustificati con regole viste prima.

# Esercizio 5.5

## Testo:

Dati:

$$\begin{array}{rcl} \mathbf{S} & \equiv & \lambda xyz.xz(yz) \\ \mathbf{K} & \equiv & \lambda xy.x \\ \mathbf{I} & \equiv & \lambda x.x \end{array}$$

mostrare che $\mathbf{SK} = \mathbf{KI}$.

## Soluzione:

Valgono i passaggi:

$$\begin{array}{rcl} \mathbf{SK} & = & (\lambda xyz.xz(yz))(\lambda xy.x) \\ & \longrightarrow_\beta & \lambda yz.(\lambda xy.x)z(yz) \\ & \longrightarrow_\beta & \lambda yz.(\lambda y.z)(yz) \\ & \longrightarrow_\beta & \lambda yz.z \\ & \equiv & \lambda xy.y \end{array}$$

e anche:

$$\begin{array}{rcl} \mathbf{KI} & = & (\lambda xy.x)(\lambda x.x) \\ & \longrightarrow_\beta & \lambda y.(\lambda x.x) \\ & \equiv & \lambda yx.x \\ & \equiv & \lambda xy.y \end{array}$$

quindi la tesi è dimostrata perchè i due termini riducono alla stessa forma.

# Esercizio 6.10

## Testo:

Sia $D$ un cpo. Una funzione $r : D \to D$ si dice idempotente se $r(r(x)) = r(x)$, per ogni $x \in D$. Dimostrare che l'insieme di tutte le funzioni continue idempotenti da $D$ in $D$ è un cpo.

## Soluzione:

Poiché $D$ è un cpo, lo spazio delle funzioni continue da $D$ a $D$ denotato da $[D \to D] = (D \to D, \sqsubseteq)$, così come in definizione 6.33, è un cpo. L'insieme delle funzioni continue e idempotenti $G$ è un sottoinsieme dell'insieme delle funzioni continue $D \to D$, quindi considerando $\sqsubseteq_G$ la restrizione di $\sqsubseteq$ a $G$, si ha che $\mathbb{G} = (G, \sqsubseteq_G)$ è un poset. Per dimostrare che $\mathbb{G}$ è anche un cpo è necessario mostrare che con la restrizione da $[D \to D]$ a $\mathbb{G}$ vengono mantenuti il minimo (punto 1) e i sup di ogni catena in $\mathbb{G}$ (punto 2).

1. Il minimo di $[D \to D]$ è $\Omega \equiv \lambda x.\perp_D$ con $\perp_D$ il minimo di $D$. Tale funzione è anche idempotente perché $\Omega(x) = \Omega(\Omega(x)) = \perp_D$ per qualunque $x$, quindi appartiene anche a $G$.

2. Data una generica catena di funzioni continue e idempotenti $\{f_i | i \in I\}$, è necessario dimostrare che il sup di tale catena, che sappiamo dalla dimostrazione del teorema 6.34 essere:

   $$g \equiv \lambda x. \sup\{f_i x | i \in I\}$$

   sia $\in G$ e quindi idempotente, e quindi che $g(gx) = g(x)$ per ogni $x$.

   La dimostrazione procede nel seguente modo:

   $$\begin{array}{rcl} g(gx) & = & (\lambda x'. \sup\{f_i x' | i \in I\})((\lambda x''. \sup\{f_i x'' | i \in I\})x) \\ & = & (\lambda x'. \sup\{f_i x' | i \in I\}) \sup\{f_i x | i \in I\} \\ & = & \sup\{f_i(\sup\{f_j x | j \in I\}) | i \in I\} \end{array}$$

   e per la continuità delle $f_i$

   $$= \quad \sup\{\sup\{f_i(f_j x) | j \in I\} | i \in I\}$$

a questo punto disponendo in forma matriciale ordinata gli $f_i$ e gli $f_j$ è possibile applicare la proposizione 6.27(2) e si ottiene

$$= \quad \sup\{f_i(f_i x) | i \in I\}$$

e poiché tutte le $f_i$ sono idempotenti

$$\begin{aligned} &= \sup\{f_i x | i \in I\} \\ &= gx \end{aligned}$$

$\square$

# Esercizio 7.6

## Testo:

Fornire semantica operazionale e denotazionale del programma

$$\mathbf{letrec} f(x) \Leftarrow f(x) \mathbf{in} f(5)$$

## Soluzione:

Cominciando con la semantica operazionale con chiamata per nome si ha:

$$f(5) \xrightarrow{(FUN)}_D f(5) \xrightarrow{(FUN)}_D \ldots$$

e in modo simile per quella con chiamata per valore usando $(FUN')$. Quindi divergono.

Per la semantica denotazionale, usando un approccio bottom up, si ha:

$$\begin{aligned}
\mathcal{T}[[5]] &= \lambda f.\lambda x.5 \\
\mathcal{T}[[x]] &= \lambda f.\lambda x.x \\
\mathcal{T}[[f(5)]] &= \lambda f.\lambda x.f(\mathcal{T}[[5]]fx) \\
&= \lambda f.\lambda x.f(5) \\
\mathcal{T}[[f(x)]] &= \lambda f.\lambda x.f(\mathcal{T}[[x]]fx) \\
&= \lambda f.\lambda x.f((\lambda f.\lambda x'.x')fx) \\
&= \lambda f.\lambda x.f(x) \\
\mathcal{D}[[f(x) \Leftarrow f(x)]] &= fix(\lambda f.\mathcal{T}[[f(x)]]f) \\
&= fix(\lambda f.(\lambda f'.\lambda x.f'(x))f) \\
&= fix(\lambda f.\lambda x.f(x)) \\
&= \sup\{(\lambda f.\lambda x.f(x))^i \lambda x.\perp | i \in \mathbb{N}\} \\
&= \lambda x.\perp \\
&= \Omega \\
\mathcal{P}[[\mathbf{letrec} f(x) \Leftarrow f(x) \mathbf{in} f(5)]] &= \mathcal{T}[[f(5)]]\mathcal{D}[[f(x) \Leftarrow f(x)]]0 \\
&= (\lambda f.\lambda x.f(5))\Omega 0 \\
&= \Omega 5 \\
&= \perp
\end{aligned}$$

Dalla valutazione della semantica operazionale si nota che il programma diverge, e infatti la valutazione della semantica denotazionale lo conferma restituendo, come prevedibile, un risultato non definito.

# Esercizio 8.11

## Testo:

Si aggiunga al linguaggio TINY un comando **stop** con la semantica informale di far terminare il programma. Se ne dia la semantica e si dimostri che $c1; \mathbf{stop}$ e $c1; \mathbf{stop}; c2$ sono semanticamente equivalenti.

## Soluzione:

È possibile estendere la semantica operazionale in modo da gestire lo **stop** modificando leggermente ($\text{Seq}_2$) in:

$$\frac{< c_1, \sigma > \longrightarrow < c_1', \sigma' >}{< c_1; c_2, \sigma > \longrightarrow < c_1'; c_2, \sigma' >} (c_1' \neq \mathbf{stop}) \qquad (\text{Seq}_2')$$

e aggiungendo le due regole

$$\begin{aligned} < \mathbf{stop}, \sigma > &\longrightarrow\ < \mathbf{noaction}, \sigma > && (\text{Stop}_1) \\ < \mathbf{stop}; c, \sigma > &\longrightarrow\ < \mathbf{stop}, \sigma > && (\text{Stop}_2) \end{aligned}$$

Per quanto riguarda la semantica denotazionale è più difficile ottenere lo stesso risultato. È necessario cambiare anche il dominio delle funzioni di interpretazione semantica dei comandi in:

$$\mathcal{C} : Com \longrightarrow \mathbb{STATO} \longrightarrow (\mathbb{STATO} + (\mathbb{STATO} \times \{stop\}) + \{error\})$$

per aggiungere la possibilità che la computazione ritorni una coppia $< \sigma, stop >$ che va letta come lo stato $\sigma$ con l'aggiunta dell'informazione che la computazione è terminata a causa di un comando **stop**. Va quindi modificata la denotazione corrispondente alla sequenzializzazione di comandi in:

$$\begin{aligned} \mathcal{C}[\![c_1; c_2]\!] = \lambda\sigma.cases(\mathcal{C}[\![c_1]\!]\sigma)of& \\ &\sigma' : \mathcal{C}[\![c_2]\!]\sigma'; \\ &< \sigma', stop >:< \sigma', stop >; \\ &error : error \\ endcases& \end{aligned}$$

e aggiunta la denotazione per lo **stop**:

$$\mathcal{C}[\![\mathbf{stop}]\!] = \lambda\sigma. < \sigma, \mathbf{stop} >$$

Con queste aggiunte andrebbero anche riviste e modificate la dimostrazione di equivalenza tra le due semantiche e la formalizzazione in punto fisso della denotazione del **while** che fa uso della sequenzializzazione. Queste ultime modifiche vengono tralasciate perché non inerenti all'esercizio.

Per dimostrare che $c_1; \textbf{stop}$ e $c_1; \textbf{stop}; c_2$ sono semanticamente equivalenti, si può usare la semantica operazionale per mostrare che le computazioni dei due programmi portano a configurazioni finali con stati identici, dato un qualsiasi stato iniziale $\sigma$. Per $c_1; \textbf{stop}$ vale che:

$$
\begin{aligned}
&< c_1; \mathit{stop}, \sigma > \\
&\downarrow_{(\mathrm{Seq}_2')} \\
&< \mathit{stop}, \sigma' > \\
&\downarrow_{(\mathrm{Stop}_1)} \\
&< \mathit{noaction}, \sigma' >
\end{aligned}
$$

dove è stata usata l'istanza di $(\mathrm{Seq}_2')$:

$$\frac{< c_1, \sigma > \longrightarrow \; < c_1', \sigma' >}{< c_1; \textbf{stop}, \sigma > \longrightarrow \; < \textbf{stop}, \sigma' >}$$

Per $c_1; \textbf{stop}; c_2$ vale che:

$$
\begin{aligned}
&< c_1; \mathit{stop}; c_2, \sigma > \\
&\downarrow_{(\mathrm{Seq}_2')} \\
&< \mathit{stop}; c_2, \sigma' > \\
&\downarrow_{(\mathrm{Stop}_2)} \\
&< \mathit{stop}, \sigma' > \\
&\downarrow_{(\mathrm{Stop}_1)} \\
&< \mathit{noaction}, \sigma' >
\end{aligned}
$$

dove è stata usata l'istanza di $(\mathrm{Seq}_2')$:

$$\frac{< c_1, \sigma > \longrightarrow \; < c_1', \sigma' >}{< c_1; \textbf{stop}; c_2, \sigma > \longrightarrow \; < \textbf{stop}; c_2, \sigma' >}$$

con $c_1'$ e $\sigma'$ gli stessi di prima in quanto la premessa è la stessa. Quindi poiché entrambe le computazioni hanno lo stesso stato finale $< \textbf{noaction}, \sigma' >$, la tesi è vera.

È possibile anche dimostrare l'equivalenza usando la semantica denotazionale, osservando che sia nella denotazione di $\mathcal{C}[\![c_1; \textbf{stop}]\!]$ che in quella di $\mathcal{C}[\![c_1; \textbf{stop}; c_2]\!]$ viene prima valutato $\mathcal{C}[\![c_1]\!]$ e poi applicato $\sigma'$ a $\mathcal{C}[\![\textbf{stop}]\!]$ in un caso e $\mathcal{C}[\![\textbf{stop}; c_2]\!]$ nell'altro. In seguito l'unica differenza è che nel primo caso vale subito che $\mathcal{C}[\![\textbf{stop}]\!]\sigma'$ è $< \sigma', \mathit{stop} >$, nel secondo caso c'è un ulteriore passaggio in cui viene usato il secondo *case* della denotazione della sequenzializzazione e che però non cambia il risultato.

# Esercizio 9.7

## Testo:

Estendere il linguaggio SMALL introducendo i comandi **restart** ed **exit**, il cui significato è di saltare rispettivamente all'inizio ed alla fine del blocco più interno.

## Soluzione:

Per gestire i due comandi richiesti è necessario modificare il dominio:

$$\begin{aligned}\mathcal{C} : Com \longrightarrow \mathbb{AMB} \longrightarrow \mathbb{MEM} \longrightarrow \\ ((\mathbb{MEM} \times \{restart, stop, normal\}) + \{error\})\end{aligned}$$

che aggiunge la possibilità per i comandi di indicare il tipo di ritorno in abbinamento al valore di ritorno della memoria. Informalmente *normal* indica che la valutazione del comando che li ha ritornati non ha portato a nessun comando di **stop** o **restart**; *stop* e *restart* indicano che la valutazione del comando ha portato rispettivamente ad uno **stop** e ad un **restart**.

Per gestire il cambio di flusso del programma vengono modificate le regole di interpretazione semantica di programmi, di blocchi, e della sequenzializzazione nel seguente modo:

$$\begin{aligned}\mathcal{P}[\![\textbf{program } c]\!]in \ = \ & fix(\lambda\Theta.cases(\mathcal{C}[\![c]\!]\rho_0(\lambda x.unused)[in/lin][nil/lout])of \\ & \quad < \sigma, normal >: \sigma(lout); \\ & \quad < \sigma, stop >: \sigma(lout); \\ & \quad < \sigma, restart >: \Theta \\ & endcases)\end{aligned}$$

$$\begin{aligned}\mathcal{C}[\![\textbf{begin } d;\ c \textbf{ end}]\!]\rho \ = \ & fix(\lambda\Theta.\mathcal{D}[\![d]\!]\rho \ \star \ \lambda\rho'.\mathcal{C}[\![c]\!]\rho[\rho'] \ \star \ \lambda\sigma s.\ cases\ s\ in: \\ & \quad normal :< \sigma, normal >; \\ & \quad stop :< \sigma, normal >; \\ & \quad restart : \Theta \\ & endcases)\end{aligned}$$

$$\begin{aligned}\mathcal{C}[\![c_1; c_2]\!]\rho \ = \ & \mathcal{C}[\![c_1]\!]\rho \ \star \ \lambda\sigma s.\ cases\ s\ in: \\ & \quad normal : \mathcal{C}[\![c_2]\!]\rho; \\ & \quad stop :< \sigma,\ stop >; \\ & \quad restart :< \sigma,\ restart >; \\ & endcases\end{aligned}$$

Oltre a queste regole è necessario modificare leggermente le altre regole di interpretazione dei comandi per dare in output la coppia $< \sigma,\ normal >$ invece della sola memoria:

$$\begin{aligned}\mathcal{C}[\![e := e']\!]\rho \ = \ & \mathcal{E}[\![e]\!]\rho \ \star \ checkLOC \ \star \ \lambda l.\mathcal{R}[\![e']\!]\rho \ \star \\ & \lambda v\sigma. < \sigma[v/l],\ normal >\end{aligned}$$

$$\mathcal{C}[\![\textbf{while } e \textbf{ do } c]\!]\rho \;=\; fix(\lambda\Theta.\mathcal{R}[\![e]\!]\rho \;\star\; checkBOOL$$
$$\star\;\; \lambda b.b \to \mathcal{C}[\![c]\!]\rho \;\star\; \Theta,\; \lambda\sigma. <\sigma,\; normal>)$$

$$\mathcal{C}[\![\textbf{output } e]\!]\rho \;=\; \mathcal{R}[\![e]\!]\rho \;\star\; \lambda b\sigma. <\sigma[b :: \sigma(lout)/lout],\; normal>$$

Non è invece necessario modificare le regole per l'**if** e per l'applicazione di procedure, in quanto la coppia viene generata a livello superiore.

Infine è necessario specificare le regole di interpretazione semantica per i due nuovi comandi **stop** e **restart**:

$$\mathcal{C}[\![\textbf{stop}]\!]\rho \;=\; \lambda\sigma. <\sigma,\; stop>$$

$$\mathcal{C}[\![\textbf{restart}]\!]\rho \;=\; \lambda\sigma. <\sigma,\; restart>$$

# Esercizio 11.10

## Testo:

Si dimostri la Proposizione 11.34 che fornisce una definizione alternativa di bisimulazione.

(Sia $<Q, A, \to>$ un LTS. Una relazione $R \subseteq Q \times Q$ è una bisimulazione se e solo se $R$ e $R^{-1}$ sono simulazioni.)

## Soluzione:

Si dimostrano i due lati della doppia implicazione. Nel testo di questo esercizio "$\implies$" è da intendersi come "*implica*", e non come sinonimo di "$\stackrel{\varepsilon}{\Rightarrow}$".

($\implies$) Si dimostra che se $R$ è una bisimulazione, allora $R$ e $R^{-1}$ sono simulazioni.

Per il punto 1 della definizione 11.8 si ha che

$$\begin{aligned}
&\forall <p, q> \in R,\ \text{vale:}\\
&\quad \forall a \in A,\ \forall p' \in Q :\\
&\qquad (p \xrightarrow{a} p') \implies \exists q' \in Q : (q \xrightarrow{a} q' \wedge\ <p', q'> \in R),
\end{aligned}$$

ma se vale questo vale anche che $R$ è una simulazione per la definizione 11.31.

Per il punto 2 della definizione 11.8 si ha che

$$\begin{aligned}
&\forall <p, q> \in R,\ \text{vale:}\\
&\quad \forall a \in A,\ \forall q' \in Q :\\
&\qquad (q \xrightarrow{a} q') \implies \exists p' \in Q : (p \xrightarrow{a} p' \wedge\ <p', q'> \in R),
\end{aligned}$$

per la definizione di relazione inversa si ha $<q, p> \in R^{-1} \iff <p, q> \in R$, quindi

$$\begin{aligned}
&\forall <q, p> \in R^{-1},\ \text{vale:}\\
&\quad \forall a \in A,\ \forall q' \in Q :\\
&\qquad (q \xrightarrow{a} q') \implies \exists p' \in Q : (p \xrightarrow{a} p' \wedge\ <q', p'> \in R^{-1}).
\end{aligned}$$

Ridenominando $p$ in $q$, $p'$ in $q'$, $q$ in $p$ e $q'$ in $p'$, tutte quantificate, si ottiene che $R^{-1}$ è una simulazione dalla definizione 11.31.

$(\Longleftarrow)$ Si dimostra che se $R$ e $R^{-1}$ sono simulazioni, allora $R$ è una bisimulazione.

Applicando la definizione 11.31 ad $R$ ed $R^{-1}$, ridenominando nella seconda $p$ in $q$, $p'$ in $q'$, $q$ in $p$ e $q'$ in $p'$ si ha

$$\begin{aligned}
&\forall < p, q > \in R, \text{ vale:}\\
&\quad \forall a \in A,\ \forall p' \in Q :\\
&\qquad (p \xrightarrow{a} p') \implies \exists q' \in Q : (q \xrightarrow{a} q' \ \wedge\ < p', q' > \in R),\\
&\forall < q, p > \in R^{-1}, \text{ vale:}\\
&\quad \forall a \in A,\ \forall q' \in Q :\\
&\qquad (q \xrightarrow{a} q') \implies \exists p' \in Q : (p \xrightarrow{a} p' \ \wedge\ < q', p' > \in R^{-1}),
\end{aligned}$$

per la definizione di relazione inversa si ha $< q, p > \in R^{-1} \iff < p, q > \in R$, quindi

$$\begin{aligned}
&\forall < p, q > \in R, \text{ vale:}\\
&\quad \forall a \in A,\ \forall p' \in Q :\\
&\qquad (p \xrightarrow{a} p') \implies \exists q' \in Q : (q \xrightarrow{a} q' \ \wedge\ < p', q' > \in R),\\
&\forall < p, q > \in R, \text{ vale:}\\
&\quad \forall a \in A,\ \forall q' \in Q :\\
&\qquad (q \xrightarrow{a} q') \implies \exists p' \in Q : (p \xrightarrow{a} p' \ \wedge\ < p', q' > \in R),
\end{aligned}$$

o in modo alternativo

$$\begin{aligned}
&\forall < p, q > \in R, \text{ valgono:}\\
&\quad \forall a \in A,\ \forall p' \in Q :\\
&\qquad (p \xrightarrow{a} p') \implies \exists q' \in Q : (q \xrightarrow{a} q' \ \wedge\ < p', q' > \in R),\\
&\quad \forall a \in A,\ \forall q' \in Q :\\
&\qquad (q \xrightarrow{a} q') \implies \exists p' \in Q : (p \xrightarrow{a} p' \ \wedge\ < p', q' > \in R),
\end{aligned}$$

e, per la definizione 11.8, $R$ è una bisimulazione.

# Esercizio 12.1

## Testo:

Dimostrare la proposizione seguente che introduce alcune leggi per gli operatori statici derivabili a partire da quelle nelle Tabelle 12.6, 12.7 e 12.8.

**Proposizione 12.64 (Leggi derivabili).** Le seguenti leggi

1. $p|q \ = \ q|p$
2. $p|(q|r) \ = \ (p|q)|r$
3. $p|nil \ = \ p$
4. $p\backslash L \ = \ p$ se $\mathcal{L}(p) \cap (L \cup \overline{L}) = \emptyset$
5. $p\backslash K\backslash L \ = \ p\backslash(K \cup L)$
6. $p[f]\backslash L \ = \ p\backslash f^{-1}(L)[f]$
7. $(p|q)\backslash L \ = \ p\backslash L|q\backslash L$ se $\mathcal{L}(p) \cap \overline{\mathcal{L}(q)} \cap (L \cup L) = \emptyset$
8. $p[Id] \ = \ p$
9. $p[f] \ = \ p[f']$ se $f \restriction_{\mathcal{L}(p)} = f' \restriction_{\mathcal{L}(p)}$
10. $p[f][f'] \ = \ p[f \circ f']$
11. $(p|q)[f] \ = \ p[f]|q[f]$ se $f \restriction_{(L \cup \overline{L})}$ è biunivoca, dove $L = \mathcal{L}(p) \cup \mathcal{L}(q)$

sono corrette rispetto a $\sim$.

## Soluzione:

Per ognuna delle regole $t_1 \ = \ t_2$ è necessario verificare che esiste una bisimulazione forte $R$ tale che $< t_1, t_2 > \in R$. La struttura della dimostrazione è simile per tutti i punti: si definisce

$$R \triangleq \{< t_1, t_2 >: t_1, t_2 \in \mathcal{P}_{CCS}\} \cup Id$$

e si procede quindi a dimostrare che $R$ è chiusa rispetto alle transizioni. Alternativamente è possibile dimostrare che $\forall \mu \in \mathcal{A}_{CCS} : \ t_1 \xrightarrow{\mu} t \iff t_2 \xrightarrow{\mu} t$.

1. $\boxed{p|q = q|p}$: Se $p|q \xrightarrow{\mu} t$ significa che è stata usata una tra $(PAR1)$, $(PAR2)$ e $(PAR3)$, quindi si ha che

   - $p \xrightarrow{\mu} p'$ e quindi $t = p'|q$ oppure
   - $q \xrightarrow{\mu} q'$ e quindi $t = p|q'$ oppure
   - se $\mu = \tau$: $p \xrightarrow{\alpha} p''$ e $q \xrightarrow{\overline{\alpha}} q''$ per qualche $\alpha$ e quindi $t = p''|q''$.

   Quindi per le stesse regole vale che $q|p \xrightarrow{\mu} t'$ e

   - $p \xrightarrow{\mu} p'$ e quindi $t' = q|p'$ oppure
   - $q \xrightarrow{\mu} q'$ e quindi $t' = q'|p$ oppure
   - se $\mu = \tau$: $p \xrightarrow{\alpha} p''$ e $q \xrightarrow{\overline{\alpha}} q''$ per qualche $\alpha$ e quindi $t' = q''|p''$.

   La tesi viene dal fatto che qualunque percorso viene scelto $t$ e $t'$ rimangono in forma $p|q$ e $q|p$ e quindi $< t, \ t' > \in R$.

2. $\boxed{p|(q|r) = (p|q)|r}$: La dimostrazione avviene mostrando che $R$ è chiusa rispetto alle transizioni.

   In maniera schematica è possibile vedere che se per un generico $\mu$ si ha $p|(q|r) \xrightarrow{\mu} t$ allora a seconda di che regole vengono usate, $t$ può essere in una delle forme riassunte nello schema seguente (dove $\implies$ è l'implicazione):

   - $(PAR1): \quad p \xrightarrow{\mu} p' \quad \implies \quad t = p'(q|r)$ $(P_1)$
   - $(PAR2): \quad (q|r) \xrightarrow{\mu} t' \quad \implies$ uno tra:
     - $(PAR1): \quad q \xrightarrow{\mu} q' \quad \implies \quad t' = q'|r \quad \implies \quad t = p|(q'|r)$ $(Q_1)$
     - $(PAR2): \quad r \xrightarrow{\mu} r' \quad \implies \quad t' = q|r' \quad \implies \quad t = p|(q|r')$ $(R_1)$
     - $(SINC): \quad \mu = \tau,\ q \xrightarrow{\alpha} q_1'',\ r \xrightarrow{\overline{\alpha}} r_1'' \quad \implies$
       $t' = q_1''|r_1'' \quad \implies \quad t = p|(q_1''|r_1'')$ $(QR_1)$
   - $(SINC): \quad \mu = \tau,\ p \xrightarrow{\beta} p_1'',\ (q|r) \xrightarrow{\overline{\beta}} t_1'' \quad \implies$ uno tra:
     - $(PAR1): \quad q \xrightarrow{\overline{\beta}} q_1''' \implies t_1'' = q_1'''|r \quad \implies \quad t = p_1''|(q_1'''|r)$ $(PQ_1)$
     - $(PAR2): \quad r \xrightarrow{\overline{\beta}} r_1''' \implies t_1'' = q|r_1''' \quad \implies \quad t = p_1''|(q|r_1''')$ $(PR_1)$
     - $(SINC)$ : non possibile perché $\overline{\beta} \neq \tau$

   In modo analogo è possibile vedere che se per lo stesso $\mu$ si ha $(p|q)|r \xrightarrow{\mu} t$ allora a seconda di che regole vengono usate, $t$ può essere in una delle forme riassunte nello schema seguente:

   - $(PAR1): \quad (p|q) \xrightarrow{\mu} t' \quad \implies$ uno tra:
     - $(PAR1): \quad p \xrightarrow{\mu} p' \quad \implies \quad t' = p'|q \quad \implies \quad t = (p'|q)|r$ $(P_2)$
     - $(PAR2): \quad q \xrightarrow{\mu} q' \implies t' = p|q' \quad \implies \quad t = (p|q')|r$ $(Q_2)$
     - $(SINC): \quad \mu = \tau,\ p \xrightarrow{\gamma} p_2'',\ q \xrightarrow{\overline{\gamma}} q_2'' \quad \implies$
       $t' = p_2''|q_2'' \quad \implies \quad t = (p_2''|q_2'')|r$ $(PQ_2)$
   - $(PAR2): \quad r \xrightarrow{\mu} r' \quad \implies \quad t = (p|q)|r'$ $(R_2)$
   - $(SINC): \quad \mu = \tau,\ (p|q) \xrightarrow{\delta} t_2'',\ r \xrightarrow{\overline{\delta}} r_2'' \quad \implies$ uno tra:
     - $(PAR1): \quad p \xrightarrow{\delta} p_2''' \implies t_2'' = p_2'''|q \quad \implies \quad t = (p_2'''|q)|r_2''$ $(PR_2)$
     - $(PAR2): \quad q \xrightarrow{\delta} q_2''' \implies t_2'' = p|q_2''' \quad \implies \quad t = (p|q_2''')|r_2''$ $(QR_2)$
     - $(SINC)$ : non possibile perché $\delta \neq \tau$.

   Non è difficile vedere che ad esempio per il caso $(P_1)$ vale che $p \xrightarrow{\mu} p'$ e $p|(q|r) \xrightarrow{\mu} p'(q|r)$ e per il caso $(P_2)$ $p \xrightarrow{\mu} p'$ con stesso $\mu$ e quindi anche stesso $p'$ e $(p|q)|r \xrightarrow{\mu} (p'|q)|r$. $< p'|(q|r),\ (p'|q)|r > \in R$ perché $R$ contiene tutte le coppie nella forma $< p|(q|r),\ (p|q)|r >$, quindi per questo caso $R$ è chiusa per le transizioni. In modo analogo si può dimostrare i casi $(Q_1)$-$(Q_2)$ e quelli $(R_1)$-$(R_2)$. Per gli altri casi è necessario notare che ad esempio per il caso $(QR_1)$ si ha una sincronizzazione tra $q$ ed $r$, così come per il caso $(QR_2)$, questo significa che $\alpha$ e $\gamma$ coincidono, e quindi $q_1'' = q_2'''$ e $r_1'' = r_2''$. In modo analogo si dimostra che: $r_1'' = r_2'' = r_1'''$, $p_1'' = p_2'' = p_2'''$, $q_1''' = q_2''$. Quindi anche per i casi $(QR_1)$-$(QR_2)$, $(PQ_1)$-$(PQ_2)$ e $(PR_1)$-$(PR_2)$ vale che $R$ è chiusa per le transizioni.

3. $\boxed{p|nil = p}$: Si dimostra con il fatto che $R$ è chiusa per transizioni. L'unica regola della semantica applicabile a $p|nil$ è $(PAR1)$ quindi $p|nil \xrightarrow{\mu} p'|nil$ e $p \xrightarrow{\mu} p'$ e $< p'|nil,\ p' >\in R$.

4. $\boxed{p\backslash L = p \quad \text{se } \mathcal{L}(p) \cap (L \cup \overline{L}) = \emptyset}$: Se $p \xrightarrow{\mu} p'$ allora per la regola $(RES)$ della semantica $p\backslash L \xrightarrow{\mu} p'\backslash L$ solo se $\mu, \overline{\mu} \notin L$ equivalentemente se $\mu \notin (L \cup \overline{L})$. Applicando la definizione 12.18 di sorta si evince che se $\mathcal{L}(p) \cap (L \cup \overline{L}) = \emptyset$ significa che il processo $p$ non potrà mai fare un'azione di $L$ o una sua negata, quindi vale sempre che $p\backslash L \xrightarrow{\mu} p'\backslash L$, e la tesi vale perché $< p'\backslash L,\ p' >\in R$ e dalla definizione di sorta se $\mathcal{L}(p) \cap (L \cup \overline{L}) = \emptyset$ allora $\mathcal{L}(p') \cap (L \cup \overline{L}) = \emptyset$.

5. $\boxed{p\backslash K\backslash L = p\backslash(K \cup L)}$: Se $p\backslash K\backslash L \xrightarrow{\mu} p'$ l'unica regola della semantica applicabile è $(RES)$ nel seguente modo:

$$\dfrac{\dfrac{p \xrightarrow{\mu} p'}{p\backslash K \xrightarrow{\mu} p'\backslash K}\ (\mu, \overline{\mu} \notin K)}{p\backslash K\backslash L \xrightarrow{\mu} p'\backslash K\backslash L}\ (\mu, \overline{\mu} \notin L)$$

Quindi $\mu, \overline{\mu} \notin L$ e $\mu, \overline{\mu} \notin K$. Equivalentemente $\mu, \overline{\mu} \notin (K \cup L)$.

Allo stesso modo per i termini $p\backslash(K \cup L)$ si può applicare

$$\dfrac{p \xrightarrow{\mu} p'}{p\backslash(K \cup L) \xrightarrow{\mu} p'\backslash(K \cup L)}\ (\mu, \overline{\mu} \notin (K \cup L))$$

e i $\mu$ per cui valgono sono gli stessi, inoltre $R$ è chiuso per le transizioni perché $< p'\backslash K\backslash L,\ p'\backslash(K \cup L) >\in R$.

6. $\boxed{p[f]\backslash L = p\backslash f^{-1}(L)[f]}$: Sul membro sinistro è possibile applicare solo:

$$\dfrac{\dfrac{p \xrightarrow{\mu} p'}{p[f] \xrightarrow{\widehat{f}(\mu)} p'[f]}}{p[f]\backslash L \xrightarrow{\widehat{f}(\mu)} p'[f]\backslash L}\ (\widehat{f}(\mu),\ \overline{\widehat{f}(\mu)}\ \notin L)$$

su quello destro solo

$$\dfrac{\dfrac{p \xrightarrow{\mu} p'}{p\backslash f^{-1}(L) \xrightarrow{\mu} p'\backslash f^{-1}(L)}\ (\mu,\ \overline{\mu}\ \notin f^{-1}(L))}{p\backslash f^{-1}(L)[f] \xrightarrow{\widehat{f}(\mu)} p'\backslash f^{-1}(L)[f]}$$

Si nota che poiché $L \subseteq \Lambda$ e $f : \Lambda \to \Lambda$ e per la definizione di $\widehat{f}$ vale che

$$\begin{aligned}(\mu,\ \overline{\mu}\ \notin f^{-1}(L)) &\implies (\widehat{f}(\mu),\ \widehat{f}(\overline{\mu})\ \notin L)\\ &\implies (\widehat{f}(\mu),\ \overline{\widehat{f}(\mu)}\ \notin L)\end{aligned}$$

e quindi le condizioni dei due membri si equivalgono e $p[f]\backslash L \xrightarrow{\widehat{f}(\mu)} p'[f]\backslash L$ se e solo se $p\backslash f^{-1}(L)[f] \xrightarrow{\widehat{f}(\mu)} p'\backslash f^{-1}(L)[f]$. La tesi viene dal fatto che $R$ è chiusa per transizioni perché $< p'[f]\backslash L,\ p'\backslash f^{-1}(L)[f] > \in R$.

7. $\boxed{(p|q)\backslash L = p\backslash L|q\backslash L \quad \text{se } \mathcal{L}(p) \cap \overline{\mathcal{L}(q)} \cap (L \cup \overline{L}) = \emptyset}$: Per il membro sinistro si può applicare

$$\frac{p|q \xrightarrow{\mu} t'}{(p|q)\backslash L \xrightarrow{\mu} t}\ (\mu, \overline{\mu} \notin L)$$

e quindi si può applicare una delle regole del parallelismo per ottenere i possibili valori di $t'$ e $t$

- $(PAR1):\quad p \xrightarrow{\mu} p' \quad\implies\quad t' = p'|q \quad\implies\quad t = (p'|q)\backslash L$
- $(PAR2):\quad q \xrightarrow{\mu} q' \quad\implies\quad t' = p|q' \quad\implies\quad t = (p|q')\backslash L$
- $(SINC):\quad p \xrightarrow{\alpha} p'',\ q \xrightarrow{\overline{\alpha}} q'' \implies t' = p''|q'' \implies t = (p''|q'')\backslash L$

Per il membro destro si applica subito una delle regole del parallelismo e se $p\backslash L|q\backslash L \xrightarrow{\mu} t$ allora

- $(PAR1):\quad \frac{p \xrightarrow{\mu} p'}{p\backslash L \xrightarrow{\mu} p'\backslash L}(\mu, \overline{\mu} \notin L) \quad\implies\quad t = p'\backslash L|q\backslash L$
- $(PAR2):\quad \frac{q \xrightarrow{\mu} q'}{q\backslash L \xrightarrow{\mu} q'\backslash L}(\mu, \overline{\mu} \notin L) \quad\implies\quad t = p\backslash L|q'\backslash L$
- $(SINC):\quad \frac{p \xrightarrow{\alpha} p''}{p\backslash L \xrightarrow{\alpha} p''\backslash L}(\alpha, \overline{\alpha} \notin L),\ \frac{q \xrightarrow{\overline{\alpha}} q''}{q\backslash L \xrightarrow{\overline{\alpha}} q''\backslash L}(\overline{\alpha}, \alpha \notin L) \quad\implies$
$t = p''\backslash L|q''\backslash L$

La regola è quasi verificata perché $R$ sarebbe chiusa per tutte le transizioni se non fosse per le condizioni aggiuntive imposte sul membro destro nella sincronizzazione. Tali condizioni impediscono a $p$ e $q$ di sincronizzarsi su $\alpha$ quando questi o il suo complementare appartengono ad $L$, ma questa condizione è coperta da quella più forte imposta nella regola: $\mathcal{L}(p) \cap \overline{\mathcal{L}(q)} \cap (L \cup \overline{L}) = \emptyset$. Infatti per la definizione di sorta, questa impone che nei processi $p$ e $q$ non compaiano mai coppie di processi complementari contenuti in $L$.

8. $\boxed{p[Id] = p}$: È possibile applicare

$$\frac{p \xrightarrow{\mu} p'}{p[Id] \xrightarrow{\widehat{Id}(\mu)} p'[Id]}$$

e per la definizione di $\widehat{f}$ e $Id$, si ha che $\widehat{Id}(\mu) = Id(\mu) = \mu$ quindi $p[Id] \xrightarrow{\mu} p'[Id]$ se e solo se $p \xrightarrow{\mu} p'$. $R$ è chiusa per transizioni perché $< p'[Id],\ p' > \in R$.

9. $\boxed{p[f] = p[f'] \quad \text{se } f\restriction_{\mathcal{L}(p)} = f'\restriction_{\mathcal{L}(p)}}$: Si può applicare $(REL)$ al membro sinistro e a quello destro:

$$\frac{p \xrightarrow{\mu} p'}{p[f] \xrightarrow{\widehat{f}(\mu)} p'[f]}$$

$$\frac{p \xrightarrow{\mu} p'}{p[f'] \xrightarrow{\widehat{f'}(\mu)} p'[f']}$$

Si nota facilmente che se $\widehat{f} = \widehat{f'}$ allora $p[f]$ e $p[f']$ hanno le stesse $\mu$-derivate. La condizione imposta nella regola indica che la restrizione di $f$ su tutte le possibili azioni di $p$ deve essere uguale alla restrizione di $f'$ su tutte le possibili azioni di $p$, e quindi per ogni possibile $\mu$ vale che $f(\mu) = f'(\mu)$ e quindi $\widehat{f}(\mu) = \widehat{f'}(\mu)$.

10. $\boxed{p[f][f'] = p[f' \circ f]}$: Per il membro sinistro si può applicare $(REL)$ due volte:

$$\frac{\dfrac{p \xrightarrow{\mu} p'}{p[f] \xrightarrow{\widehat{f}(\mu)} p'[f]}}{p[f][f'] \xrightarrow{\widehat{f'}(\widehat{f}(\mu))} p'[f][f']}$$

Per il membro destro si può applicare $(REL)$ una volta

$$\frac{p \xrightarrow{\mu} p'}{p[f' \circ f] \xrightarrow{(\widehat{f' \circ f})(\mu)} p'[f' \circ f]}$$

Dalla definizione di $\widehat{f}$ e di funzione composta si verifica facilmente che $(\widehat{f' \circ f})(\mu) = \widehat{f'}(\widehat{f}(\mu))$. Quindi $R$ è chiusa per transizioni.

11. $\boxed{(p|q)[f] = p[f]|q[f] \quad \text{se } f\restriction_{(L \cup \overline{L})} \text{ è biunivoca, dove } L = \mathcal{L}(p) \cup \mathcal{L}(q)}$: Per il membro sinistro si può applicare $(REL)$:

$$\frac{p|q \xrightarrow{\mu} t'}{(p|q)[f] \xrightarrow{\widehat{f}(\mu)} t}$$

e per valutare i valori di $t'$ e $t$ è necessario vedere quale regole del parallelismo sono usate:

- $(PAR1): \quad p \xrightarrow{\mu} p' \quad \Longrightarrow \quad t' = p'|q \quad \Longrightarrow \quad t = (p'|q)[f]$
- $(PAR2): \quad q \xrightarrow{\mu} q' \quad \Longrightarrow \quad t' = p|q' \quad \Longrightarrow \quad t = (p|q')[f]$
- $(PAR1): \quad \mu = \tau,\ p \xrightarrow{\alpha} p'',\ q \xrightarrow{\overline{\alpha}} q'' \quad \Longrightarrow$
$t' = p''|q'' \quad \Longrightarrow \quad t = (p''|q'')[f]$

Per il membro destro si possono applicare subito le regole per il parallelo:

$$(PAR1): \quad \cfrac{\cfrac{p \xrightarrow{\mu} p'}{p[f] \xrightarrow{\widehat{f}(\mu)} p'[f]}}{p[f]|q[f] \xrightarrow{\widehat{f}(\mu)} p'[f]|q[f]}$$

$$(PAR2): \quad \cfrac{\cfrac{q \xrightarrow{\mu} q'}{q[f] \xrightarrow{\widehat{f}(\mu)} q'[f]}}{p[f]|q[f] \xrightarrow{\widehat{f}(\mu)} p[f]|q'[f]}$$

$$(SINC): \quad \cfrac{\cfrac{p \xrightarrow{\beta} p'''}{p[f] \xrightarrow{\widehat{f}(\beta)} p'''[f]} \qquad \cfrac{q \xrightarrow{\gamma} q'''}{q[f] \xrightarrow{\widehat{f}(\gamma)} q'''[f]}}{p[f]|q[f] \xrightarrow{\tau=\widehat{f}(\tau)} p'''[f]|q'''[f]} \; (\widehat{f}(\beta) = \overline{\widehat{f}(\gamma)})$$

la condizione di *biunivocità* (limitatamente a tutte le possibili azioni che possono fare $p$ e $q$) imposta ad $f$ garantisce che se $\widehat{f}(\beta) = \overline{\widehat{f}(\gamma)} = \widehat{f}(\overline{\gamma})$ allora $\beta = \overline{\gamma}$ e quindi $p$ e $q$ si possono sincronizzare solo sugli stessi $\alpha$ e $\overline{\alpha}$ visti per il membro sinistro, e quindi $p'' = p'''$ e $q'' = q'''$.

Detto questo allora $R$ è chiusa per le transizioni e quindi la legge vale.

# Esercizio 12.2

## Testo:

Mostrare che la legge seguente

$$(\tau 4) \qquad p + \tau.(p+q) = \tau.(p+q)$$

è derivabile dall'insieme $E_4$ definito in Tabella 12.9.

## Soluzione:

Si ha che:

$$\begin{aligned}
p + \tau.(p+q) &= p + ((p+q) + \tau.(p+q)) && \text{da } (\tau 2)\\
&= (p + (p+q)) + \tau.(p+q) && \text{da } (A2)\\
&= ((p+p)+q) + \tau.(p+q) && \text{da } (A2)\\
&= (p+q) + \tau.(p+q) && \text{da } (A4)\\
&= \tau.(p+q) && \text{da } (\tau 2)
\end{aligned}$$

quindi vale che $E_4 \vdash p + \tau.(p+q) = \tau.(p+q)$.