Esercizi sul CCS

Server parallelo e server sequenziale. Si considerino i seguenti server

$$\begin{array}{ll} S = a.S + b.S & \text{server di } a \text{ e di } b \\ S_a = a.S_a & \text{server di } a \\ S_b = b.S_b & \text{server di } b \end{array}$$

Dire se S e $S_a|S_b$ sono equivalenti per qualche nozione di equivalenza (equiv. forte, equiv. debole, congruenza debole).

Server sequenziale e server con gemmazione. Si considerino i seguenti server di a

$$\begin{array}{ll} S_1 = a.S_1 & \text{server sequenziale di } a \\ S_2 = a.(S_1|S_2) & \text{server di } a \text{ con gemmazione} \\ S_1' = a.S_1' + \tau.0 & \text{server sequenziale di } a \text{ che puo' terminare} \end{array}$$

Quali server sono equivalenti e quali no per qualche nozione di equivalenza (equiv. forte, equiv./congruenza debole)?

Equivalenza tra consumatori Si considerino i seguenti tipi di consumatori di risorse a e b

$$\begin{array}{ll} C_1 = a.b.\overline{ok} & \text{consumatore 1} \\ C_2 = b.a.\overline{ok} & \text{consumatore 2} \\ C_3 = C_1 + C_2 & \text{consumatore 3} \end{array}$$

Dire se per qualche $R_a\colon\{\overline{a}\}$ e $R_b\colon\{\overline{b}\}$ c'e' modo di distinguere (rispetto all'equivalenza/congruenza debole) tra i consumatori quando sono utilizzati in un contesto della forma $E = (R_a|R_b|X)\backslash\{a,b\}$. Si osservi che se C e' un consumatore allora $E\{X:=C\}$ ha sort $\{\overline{ok},\tau\}$.
NOTA: sebbene ci siano un'infinita di scelte per R_a ed R_b basta considerare un numero finito di possibilita'. Dire quali, giustificando informalmente perche'.

Catene di montaggio Si consideri i seguenti tipi di operaio:

$$\begin{array}{ll} W_1 = a.\overline{b}.W_1 & \text{operaio forte} \\ W_2 = a.g.\overline{r}.\overline{b}.W_2 & \text{operaio debole, richiede una risorsa} \\ W_3 = a.g.\overline{b}.\overline{r}.W_3 & \text{altro operaio debole} \\ R = \overline{g}.r.R & \text{risorsa usata da operaio debole} \end{array}$$

Dire quali delle seguenti catene di montaggio sono equivalenti (nel senso della equivalenza/congruenza debole):

$$J_1 = W_1|W_1 \quad J_2 = (W_1|W_2|R)\backslash\{g,r\} \quad J_3 = (W_2|W_2|R)\backslash\{g,r\}$$
$$J_4 = (W_1|W_3|R)\backslash\{g,r\} \quad J_5 = (W_2|W_3|R)\backslash\{g,r\} \quad J_6 = (W_3|W_3|R)\backslash\{g,r\}$$

NOTA: Dire quanti stati ha ciascuna catena di montaggio, trascurando gli stati non raggiungibili da quello iniziale. Sfruttando le proprieta' equazionali della equivalenza forte cercare di ridurre l'insieme di stati da prendere in considerazione, identificando quelli che sono fortemente equivalenti.

Macchinetta del caffe' Sia K il seguente insieme di azioni osservabili:

$$\begin{array}{ll} m & \text{accetta moneta} \\ \overline{m} & \text{restituisci moneta} \\ c & \text{accetta richiesta di caffe'} \\ \overline{c} & \text{fornisci il caffe'} \end{array}$$

Formalizzare mediante agenti di sort $K \cup \{\tau\}$ i seguenti tipi di macchinetta del caffe'

- puo' non essere in grado di fornire il caffe', ma non ruba
- puo' non essere in grado di fornire il caffe', e puo' rubare
- puo' non essere in grado di fornire il caffe', ed in tal caso non restituisce la moneta

Vi sono vari modi di interpretare le specifiche dei requisiti:

- "puo' non essere in grado di fornire caffe', ma non ruba" si puo' interpretare nei seguenti modi

  1) la macchinetta accetta in ogni caso moneta e richiesta di caffe' (in qualche ordine prefissato oppure no), quindi verifica la disponibilita' del caffe' e si porta (con un τ) in due possibili stati: quello che fornisce il caffe', quello che restituisce i soldi.

  2) la macchinetta si puo' trovare in due stati iniziali, quello in cui e' disponibile il caffe' e quello in cui non e' disponibile, in quest'ultimo caso potrebbe decidere di non accettare moneta (ma questo sembra poco realistico) e puo' portarsi nello stato in cui e' disponibile il caffe' effettuando una azione τ (corrisponde ad una operazione di ricarica).

- "puo' non essere in grado di fornire il caffe', e puo' rubare" la possibilita' di rubare (cioe' non restituire la moneta) non va' interpretata come certezza (che e' invece quanto richiesto nella terza specifica).

- un aspetto importante da modellare e' chi effettua la scelta se dare il caffe' o restituire la moneta. Questa scelta dovrebbe essere della macchinetta, non dell'ambiente. Quindi il seguente agente non va' bene

  $$MC = m.c.(\overline{m}.MC + \overline{c}.MC)$$

  poiche' corrisponde ad una macchinetta che e' sempre in grado di fornire il caffe' (ipotesi poco realistica), e lascia all'utente la possibilita' di scegliere se farsi restituire i soldi o prendere il caffe'.

Compitino sul CCS

1 Slot machine. Sia $K = \{in_m | m \in \mathsf{Nat}\} \cup \{\overline{out}_m | m \in \mathsf{Nat}\}$, dove le azioni hanno la seguente interpretazione:

in_m	accetta giocata di m lire
$\overline{out}_m$	dai vincita di m lire

Formalizzare mediante agenti (in CCS puro non value-passing) di sort $K \cup \{\tau\}$ i seguenti tipi di slot machine:

- accetta qualunque giocata $m > 0$ (e incrementa il jackpot), puo' far vincere qualunque cifra tra 0 ed il jackpot.
- accetta qualunque giocata $m > 0$ (incrementando il jackpot), puo' far vincere tutto il jackpot o niente.
- accetta qualunque giocata $m > 0$, si trattiene 1 lira di tassa, puo' far vincere tutto il jackpot (al netto delle tasse) o niente.

Come test di la correttezza della formalizzazione verificare che $(SM_0|T_m)\backslash K$ non sia debolmente equivalente ad $\overline{ok}$, dove SM_0 e' la slot machine nello stato iniziale (cioe' jackpot=0), e $T_m = \overline{in}_m.out_m.\overline{ok}$ con $m > 0$.
SUGGERIMENTO: si disegni il LTS corrispondente a $(SM_0|T_m)\backslash K$ per ciascuna delle tre slot machine.

Soluzione. Ogni slot machine viene descritta da un insieme di equazioni $SM_n = P_n$, dove SM_j e' lo stato della slot machine con jack-pot pari ad j:

- $SM_j = \sum_{m>0} in_m.SM'_{j+m} \qquad SM'_j = \sum_{n\leq j} \tau.\overline{out}_n.SM_{j-n}$

  $(SM_0|T_m)\backslash K$ e' fortemente equivalente a $\tau.(\tau + \tau.\overline{ok})$
- $SM_j = \sum_{m>0} in_m.SM'_{j+m} \qquad SM'_j = (\tau.\overline{out}_0.SM_j) + (\tau.\overline{out}_j.SM_0)$

  $(SM_0|T_m)\backslash K$ e' fortemente equivalente a $\tau.(\tau + \tau.\overline{ok})$
- $SM_j = \sum_{m>0} in_m.SM'_{j+m-1} \qquad SM'_j = (\tau.\overline{out}_0.SM_j) + (\tau.\overline{out}_j.SM_0)$

  $(SM_0|T_m)\backslash K$ e' fortemente equivalente a $\tau.\tau$

2 Server parallelo e server non-deterministico. Si considerino i seguenti server

$S = \tau.a.S + \tau.b.S$	server non-deterministico di a e di b
$S_a = \tau.a.S_a$	server di a
$S_b = \tau.b.S_b$	server di b

Dire se S e $S_a|S_b$ sono equivalenti per qualche nozione di equivalenza (equiv. forte, equiv. debole, congruenza debole), giustificando la risposta. In particolare, si disegni il LTS per i due server.

Soluzione. S e $S' \stackrel{\Delta}{\equiv} S_a|S_b$ non sono equivalenti nemmeno nel senso debole. I LTS per tali agenti sono

$$\begin{tikzcd}
S_0 \arrow[r, shift left, "a"] & S \arrow[l, shift left, "\tau"] \arrow[r, shift right, "\tau"'] & S_1 \arrow[l, shift right, "b"']
\end{tikzcd}
\qquad
\begin{tikzcd}
S' \arrow[r, shift right, "\tau"'] \arrow[d, shift left, "\tau"] & S'_1 \arrow[l, shift right, "b"'] \arrow[d, shift right, "\tau"'] \\
S'_0 \arrow[r, shift left, "\tau"] \arrow[u, shift left, "a"] & S'_2 \arrow[l, shift left, "b"] \arrow[u, shift right, "a"']
\end{tikzcd}$$

S' facendo due τ si porta in S'_2. Da S facendo solo τ si puo' arrivare in qualunque stato, ma nessuno di questi e' debolmente equivalente ad S'_2: infatti S_0 non puo' fare b, S_1 non puo' fare a, mentre S puo' fare τ e portarsi in uno stato non equivalente ad S'_2.

3 Server sequenziale e server con gemmazione. Si considerino i seguenti server di a

$S_1 = a.S_1$	server sequenziale di a
$S_2 = a.S_2 + \tau.0$	server sequenziale di a che puo' terminare
$S_3 = a.(S_2 \vert S_3)$	altro server di a con gemmazione

Dire se S_1 e S_3 sono equivalenti per qualche nozione di equivalenza (equiv. forte, equiv./congruenza debole), giustificando la risposta.

Soluzione. S_1 e S_3 sono debolmente equivalenti, e poiche' non hanno azioni τ iniziali sono anche debolmente congruenti. Dopo un certo numero di azioni (a o τ) l'agente S_3 si trovera' in uno stato della forma $S * S_3$ dove $S \in \{0, S_2\}^*$ e' una lista di agenti e $S * S_3$ e' una abbreviazione definita per induzione sulla lunghezza di S

$$nil * S_3 \stackrel{\Delta}{\equiv} S_3 \qquad (P{:}{:}S) * S_3 \stackrel{\Delta}{\equiv} (P|S * S_3)$$

Per far vedere che S_1 e' debolmente equivalente a S_3 basta dimostrare che la relazione

$$R \stackrel{\Delta}{\equiv} \{(S_1, S * S_3) | s \in \{0, S_2\}^*\}$$

e' una bisimulazione debole.

- l'unica mossa $S_1 \xrightarrow{a} S_1$ si puo' simulare con la mossa $S * S_3 \xrightarrow{a} S * (S_2|S_3)$
- $S * S_3$ puo' compiere azioni τ causate da un S_2 in S che fa' $S_2 \xrightarrow{\tau} 0$ (in questo caso S_1 non si muove), oppure azioni a causate da un S_2 in S che fa' $S_2 \xrightarrow{a} S_2$, oppure un azione a causata da $S_3 \xrightarrow{a} (S_2|S_3)$. Gli ultimi due casi sono simulati da S_1 facendo l'azione $S_1 \xrightarrow{a} S_1$.