CORSO DI MODELLI E SISTEMI CONCORRENTI
ESAME FINALE 2024

Svolgere i due esercizi obbligatori e 2 esercizi opzionali a scelta. Consegna almeno 3 giorni prima dell'orale.

1. **(Esercizio obbligatorio 1)** Si consideri l'algebra di processi *MINI*, definita dalla seguente sintassi:

$$p \quad ::= \quad \mathbf{0} \quad | \quad a.p \quad | \quad \underline{a}.p \quad | \quad p+p$$

dove a è l'unica azione disponibile. Assumendo che σ denoti una generica sequenza non vuota di a ($\sigma \in a^+$), le regole di semantica operazionale sono definite come segue:

$$\text{(Pref)} \quad \frac{}{a.p \xrightarrow{a} p} \qquad \text{(S-Pref)} \quad \frac{p \xrightarrow{\sigma} p'}{\underline{a}.p \xrightarrow{a\sigma} p'}$$

$$\text{(Sum}_1\text{)} \quad \frac{p \xrightarrow{\sigma} p'}{p+q \xrightarrow{\sigma} p'} \qquad \text{(Sum}_2\text{)} \quad \frac{q \xrightarrow{\sigma} q'}{p+q \xrightarrow{\sigma} q'}$$

1. Disegnare l'LTS per il processo $a.\underline{a}.(a.\mathbf{0}+\underline{a}.a.\mathbf{0})+\underline{a}.\mathbf{0}$, fornendo la dimostrazione di ogni singola transizione; quindi, indicare l'insieme delle sue tracce complete $CTr(p)$.

2. Dimostrare che ogni processo p di *MINI* è terminante, cioè tutte le sue computazioni terminano.

3. Verificare quali delle seguenti sono proprietà valide per l'equivalenza di bisimulazione forte:

(i) $a.p \sim \underline{a}.p$
(ii) $\underline{a}.\mathbf{0} \sim \mathbf{0}$
(iii) $\underline{a}.(p+q) \sim \underline{a}.p+\underline{a}.q$
(iv) $a.(p+q) \sim a.p+a.q$
(v) $\underline{a}.a.p \sim a.\underline{a}.p$

4. Dimostrare che $\sim$ è una congruenza per gli operatori di *MINI*.

5. L'insieme di assiomi $\{A_1, A_2, A_3, A_4, S_1, S_2\}$, dove

S_1 $\underline{a}.\mathbf{0} = \mathbf{0}$
S_2 $\underline{a}.(p+q) = \underline{a}.p+\underline{a}.q$

è una assiomatizzazione sound e completa di $\sim$ per *MINI*? Giustificare la risposta.

6. Individuare un processo p tale che soddisfi tutte le seguenti formule HML:

(i) $\langle a\rangle\langle aa\rangle T$
(ii) $[a]\langle a\rangle T$
(iii) $\langle a\rangle[aa]F$
(iv) $[aa]F$
(v) $\langle a\rangle\langle a\rangle\langle a\rangle T$

7. Sia *MICRO* il sottocalcolo di *MINI* che non usa l'operatore di prefisso forte (prefisso sottolineato). Si studi se sia possibile compilare *MINI* su *MICRO* rispettando la semantica per tracce complete; in altre parole, si chiede se sia possibile definire un encoding $[\![-]\!]$ da *MINI* su *MICRO*, tale che il processo p di *MINI* e il suo compilato $[\![p]\!]$ di *MICRO* abbiano le stesse tracce complete.

2. **(Esercizio obbligatorio 2)** Dato un generico LTS $(Q, A, \rightarrow)$, si definiscano le funzioni D, L, S e O, tutte da 2^Q in 2^Q, tali che il massimo punto fisso di D sia l'insieme degli stati divergenti (che possono eseguire una computazione infinita di τ), il massimo punto fisso di L sia l'insieme degli stati di livelock (stati che possono solo eseguire computazioni infinite di τ), il minimo punto fisso di S sia l'insieme degli stati che possono terminare il calcolo in modo silente (cioè facendo una computazione che va in deadlock con solo transizioni etichettate τ) e il massimo punto fisso di O sia l'insieme degli stati che possono eseguire una computazione infinita solo di azioni osservabili. Si consideri la figura di LTS in Figura 1. Si mostri il calcolo, per approssimazioni successive, degli stati divergenti, di livelock, convergenti in modo silente e divergenti in modo osservabile. Infine, definire formule di HML con ricorsione che caratterizzino le proprietà D, L, S e O.

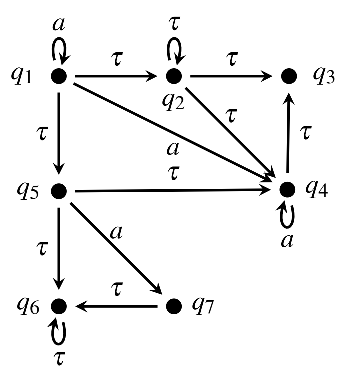

Figure 1: LTS per l'esercizio 2

3. **(Esercizio opzionale facile)** Dimostrare che $L = \{a^{2n+1}b^{2m} \mid n, m \geq 0\}$ è un linguaggio descrivibile in finite-state CCS, cioè esiste un processo p di finite-state CCS tale che $WCTr(p) = L$.

4. **(Esercizio opzionale facile)** Dimostrare che se $p \stackrel{\tau}{\Longrightarrow} p'$, con $p' \approx q$, e $q \stackrel{\tau}{\Longrightarrow} q'$, con $p \approx q'$, allora $p \approx^c q$.

5. **(Esercizio opzionale facile)** Dimostrare che se $p \approx^c q$, allora $(\nu a)p \approx^c (\nu a)q$ per ogni a.

6. **(Esercizio opzionale medio)** Dato un cammino silente

$q_1 \xrightarrow{\tau} q_2 \xrightarrow{\tau} q_3 \dots q_n \xrightarrow{\tau} q_{n+1}$, (dove i vari q_i possono fare anche altre transizioni) dimostra che se $q_1 \approx q_{n+1}$, allora $q_i \approx q_j$ per tutti gli $i, j = 1, 2, \ldots n+1$, dove $\approx$ denota l'equivalenza per bisimulazione debole. (*Suggerimento:* Prova prima con $n = 2$.)

7. **(Esercizio opzionale medio)** Modellare una vending machine che (i) consente di scegliere tra tea $(ask-t)$ e/o coffee $(ask-c)$, entrambe al costo di una moneta $(coin)$, (ii) produce tea $(\bar{t})$ e/o coffee $(\bar{c})$, (ii) alterna la bevanda selezionabile, nel senso che la prima volta permette solo di selezionare tea, la seconda solo coffee, la terza solo tea, e così via; infine (ii) può tener credito fino a due monete.

   Verificare se la traccia $coin\ ask-t\ \bar{t}\ coin$ e la traccia $coin\ coin\ ask-t\ \bar{t}$, eseguite a partire dallo stato iniziale, portano sullo stesso stato.

   Verificare quali stati soddisfano le seguenti formule:

   (i) $[coin]\langle coin\rangle\langle ask-t\rangle T$
   (ii) $\langle\bar{t}\rangle\langle coin\rangle\langle ask-c\rangle T$
   (iii) $[coin][coin]F$
   (iv) $\langle coin\rangle\langle coin\rangle[ask-t]F$

8. **(Esercizio opzionale medio)** Considera la specifica dell'*unbounded bag buffer*:

$$
\begin{array}{lcl}
UB_0 & \stackrel{def}{=} & in.UB_1 \\
UB_i & \stackrel{def}{=} & in.UB_{i+1} + \overline{out}.UB_{i-1} \quad \text{per } 0 < i
\end{array}
$$

   Disegna (il frammento inziale del) l'associato LTS. Considera il processo BPP

$$UB \stackrel{def}{=} in.(UB \,|\, \overline{out}.\mathbf{0})$$

   Dimostra che $UB_0 \sim UB$, costruendo una opportuna strong bisimilation up to.

9. **(Esercizio opzionale medio)** Dimostrare che la team bisimilarity su reti di Petri BPP è una relazione di equivalenza. (*Suggerimento:* Bisogna dimostrare la Proposizione 13 sul lucido 9 della lezione 33, dando per assodata la Proposizione 12 del lucido 5)

10. **(Esercizio opzionale medio)** Dimostra che $\approx^c$ è una relazione di equivalenza e che se $C \stackrel{def}{=} \tau.C + p$ allora $C \approx^c \tau.p$. (Suggerimento: per provare che $\approx^c$ è transitiva, serve risolvere prima l'esercizio 2.53 (lezione 7, pagina 16), adattato alla weak bisimulation)

11. **(Esercizio opzionale medio-difficile)** Sia $A^\omega \stackrel{def}{=} a.A^\omega$. Sia $a^0 = \mathbf{0}$ e $a^{i+1} = a.a^i$. Dimostra che, per ogni formula F di HML, $A^\omega \models F$ se e soltanto se $a^i \models F$, dove i è la *modal depth* di F.

12. **(Esercizio opzionale medio-difficile)** Dimostrare che se R è una weak bisimulation up to $\approx$, allora $R \subseteq \approx$. (*Suggerimento:* Dimostrare che se R è una weak bisimulation up to $\approx$, allora $\approx R \approx$ è una weak bisimulation.)

13. **(Esercizio opzionale difficile)** Considera il produttore

    $Pr = produce.(\overline{send}.\mathbf{0} \,|\, Pr)$, il consumatore $C_1 = send.consume.C_1$ and il processo di finite-net CCS

    $UPC \stackrel{def}{=} (\nu send)(Pr \,|\, C_1)$.

    Considera l'unbounded BPP buffer UB dell'esercizio 8, e

    $P_2 \stackrel{def}{=} produce.\overline{in}.P_2 \quad C_2 \stackrel{def}{=} out.consume.C_2$.

    Dimostra che $PUBC \stackrel{def}{=} (\nu in, out)((P_2 \,|\, UB) \,|\, C_2)$ è weak bisimile a UPC.

14. **(Esercizio opzionale difficile)** Mostrare che $L = \{ww^R \mid w \in (a+b)^*\}$ è un linguaggio di finitary CCS, ovvero esiste un processo p di finitary CCS tale che $WCTr(p) = L$.

15. **(Esercizio opzionale difficile)** Costruire una specifica Multi-CCS (e possibilmente anche la corrispondente rete di Petri) per il problema dei filosofi a cena (caso con 3 filosofi) che soddisfi il requisito di essere fully-distributed, deadlock-free, divergent-free e che garantisca weak non-starvation. Definire in HML con ricorsione la propretà di weak non-starvation.

16. **(Esercizio opzionale molto difficile)** Continuando l'esercizio precedente, provare a definire una variante del problema dei filosofi a cena (sempre con 3 filosofi) che garantisca strong non-starvation. Definire in HML con ricorsione la proprietà di strong non-starvation.

17. **(Esercizio opzionale estremamente difficile)** Uno stack di simboli in A, che segue l'usuale disciplina LIFO (*last-in-first-out*), può essere descritto in full value-passing CCS dalla famiglia di costanti $Stack(\sigma)$, ciascuna parametrizzata su una sequenza $\sigma \in A^*$, come segue:

    $$\begin{array}{rcl} Stack(\varepsilon) & \stackrel{def}{=} & empty.Stack(\varepsilon) + push(x).Stack(x) \\ Stack(x\sigma) & \stackrel{def}{=} & \overline{pop}(x).Stack(\sigma) + push(y).Stack(yx\sigma) \end{array}$$

    dove x e y sono variabili e l'elemento top di $Stack(\sigma)$ è il simbolo più a sinistra.

    Uno stack di simboli può essere descritto in finitary value-passing CCS come segue:

    $$\begin{array}{rcl} S & \stackrel{def}{=} & empty.S + push(x).((\nu a)(S_1(x) \,|\, a.S)) \\ S_1(x) & \stackrel{def}{=} & \overline{pop}(x).\bar{a}.\mathbf{0} + push(y).((\nu b)(S_2(y) \,|\, b.S_1(x))) \\ S_2(y) & \stackrel{def}{=} & \overline{pop}(y).\bar{b}.\mathbf{0} + push(x).((\nu a)(S_1(x) \,|\, a.S_2(y))) \end{array}$$

    Nota che, se $|A| = m$, allora il processo CCS puro utilizza $2 \times m + 1$ costanti and $2 \times m + 1$ azioni, con, in aggiunta, le due azioni ausiliarie a e b.

    Dimostra che $Stack(\varepsilon)$ e S sono weakly bisimili. (Suggerimento: adatta la prova vista a lezione per il contatore.)