1 MFI 2001/2002 - Foglio esercizi n. 1

1.1 Semantica Statica

1. Vogliamo arricchire il linguaggio didattico con il seguente comando di "assegnamento in parallelo" $x_1 := e_1 \parallel x_2 := e_2$ dove x_1, x_2 sono nomi di variabili distinte ed e_1, e_2 sono espressioni rispettivamente dello stesso tipo di x_1, x_2. Descriverne formalmente la semantica statica

2. In quali ambienti statici sono corretti i seguenti termini di LW?

   - **while** $(x > 0)\{$**if** $(x == y)\{x = 7;\}$ **else** $\{p(y)\}\}$
   - **if** $(f(x, y, z))\{x = y + z;\}$ **else** $\{p(y)\}\}$
   - **int** x;
bool $y = x == 4$;
   - **int** x;
void $f($**bool** $x)\{$**if** $(x)\{y = p(y)\}$ **else** $\{x = z\}\}$;

3. Si arricchisca LF con la produzione `Exp ::= Type function(PDecs){Exp}` (che gioca il ruolo della λ-astrazione), ottenendo il linguaggio LF+

4. In quali ambienti statici sono corretti i seguenti termini di LF+?

   - **int** $x = 1$;
int $f($**int** $\to$ **bool** $x)\{$**if** $(x(y))\{p(x)\}$ **else** $\{p(z)\}\}$;
   - **int** $\to$ **bool** $p($**int** $x)$
$\{$**if** $(x > 0)\{$**bool function** $($**int** $z)\{z + x\}\}$
else $\{$**bool function** $($**int** $z)\{z - x\}\}$;

5. Vogliamo arricchire il linguaggio didattico con la dichiarazione di n (con n un qualsiasi intero maggiore di 1) variabili *distinte* corrispondenti ad una stessa locazione (*aliasing*). Descriverne formalmente la sintassi e la semantica statica (i vincoli del linguaggio per la dichiarazione di variabile restano validi anche per questo nuovo tipo)

1.2 Semantica Dinamica

1. Descrivere formalmente la semantica dinamica dell'assegnamento parallelo introdotto in Sezione 1.1, esercizio 1. Intuitivamente, l'effetto deve essere di assegnare *simultaneamente* il valore di e_1 a x_1 e il valore di e_2 a x_2.

2. Valutare formalmente la semantica dinamica dei termini introdotti nell'esercizio 2 dlla Sezione 1.1 a partire da un ambiente r_0 che ne garantisca la correttezza statica ed uno stato iniziale m_0 il cui dominio include l'immagine di r_0.

3. Definire la semantica dinamica di LF+, come introdotto nell'esercizio 3 dlla Sezione 1.1.

4. Valutare formalmente la semantica dinamica dei termini introdotti nell'esercizio 4 della Sezione 1.1 a partire da un ambiente r_0 che ne garantisca la correttezza statica.

5. Valutare formalmente la semantica dinamica dei seguenti termini a partire da un ambiente r_0 che ne garantisca la correttezza statica ed uno stato iniziale m_0 il cui dominio include l'immagine di r_0.

   - **int** x;
int $y = 4$;
int $p($**int** $x)\{x = x * x;$ **result** $x * x\}$;
void $f($**bool** $x)\{$**if** $(x)\{y = p(y)\}$ **else** $\{x = z\}\}$;
main $()f($**true**$)$
   - **int** $x = 2$;
int $p($**int** $z)\{z = z * x;$ **result** $z * z\}$;
void $f($**int** $x)\{$**if** $(x > 2)\{x = p(x)\}$ **else** $\{x = z\}\}$;
main $()f(6)$
   - **int** $x = 2$;
int $p($**int** $z)\{$**if** $(z == x)\{z = z * x;\}$ **else** $\{z = x * p(z - 1);\}$**result** $z\}$;

   (si usi una prova induttiva)

- **while** $(x > 9)\{z = z + x; x = x - 1\}$;
  (si usi una prova induttiva)
- Descrivere formalmente la semantica

6. Descrivere formalmente la semantica dinamica della dichiarazione con aliasing introdotto in Sezione 1.1, esercizio 5. Intuitivamente, l'effetto deve essere di assegnare a tutte le variabili la stessa locazione.

7. Sotto quali assunzioni su m_0 le seguenti coppie di termini sono semanticamente equivalenti?

   - **int** $p($**int** $x)\{x = x * x;$ **result** $x * x\}$;
     e
     int $p($**int** $z)\{z = z * z;$ **result** $z * z\}$;
   - **int** $p($**int** $x)\{x = z * x;$ **result** $z * x\}$;
     main $()p(6)$
     e
     int $p($**int** $z)\{z = x * z;$ **result** $x * z\}$;
     main $()p(6)$
   - **int** $p($**int** $x)\{x = z * x;$ **result** $z * x\}$;
     main $()p(z)$
     e
     int $p($**int** $x)\{x = x * x;$ **result** $x * x\}$;
     main $()p(z)$
   - **int** $p($**int** $x)\{x = z * x;$ **result** $z * x\}$;
     main $()p(z)$
     e
     int $p($**int** $x)\{x = x * x;$ **result** $x * x\}$;
     main $()p(t)$

1.3 λ-calcolo

1. Si considerino le seguenti espressioni del λ-calcolo puro, dove $\mathcal{E}$ indica una generica espressione.

   - $\mathcal{E}_1 = \lambda\, Y\, .\, (\, \lambda\, X\, .\, (\, X\ \mathcal{E}\,)\ (\, Y\ \lambda\, X\, .\, (\, \mathcal{E}\ X\,)\,)\,)$
   - $\mathcal{E}_2 = (\, \mathcal{E}_1\ \lambda\, X\, .\, \lambda\, Y\, .\, (\, X\ Y\,)\,)$
   - $\mathcal{E}_3 = (\, \mathcal{E}_1\ \lambda\, X\, .\, \lambda\, Y\, .\, (\, Y\ X\,)\,)$

   Esibire una $\mathcal{E}$ tale che $\mathcal{E}_2$ ed $\mathcal{E}_3$ siano α-β equivalenti ed esibire una $\mathcal{E}$ tale che $\mathcal{E}_2$ ed $\mathcal{E}_3$ *non* siano α-β equivalenti.

2. $e_1 = (\, (\, \lambda\, X,Y\, .\, (\, X\ (\, Y\ X\,)\,)\ \lambda\, X\, .\, X\,)\ \lambda\, Z,Z\, .\, Z\,)$,
   $e_2 = (\, (\, (\, \lambda\, X,Y,Z\, .\, (\, Z\ (\, Z\ Z\,)\,)\ \lambda\, T\, .\, T\,)\ \lambda\, W\, .\, W\,)\ \lambda\, Q\, .\, Q\,)$;
   $e_3 = (\, (\, \lambda\, X,Y\, .\, (\, X\ (\, Y\ X\,)\,)\ \lambda\, X\, .\, X\,)\ \lambda\, Z,Z\, .\, Z\,)$;
   $e_4 = (\, (\, (\, \lambda\, X,Y,X,Y\, .\, (\, X\ (\, Y\ (\, X\ Y\,)\,)\,)\ \lambda\, Z,T\, .\, Z\,)\ \lambda\, Z,T\, .\, Z\,)\ \lambda\, T\, .\, Y\,)$;
   per $i = 1, 4$, esibire la forma normale di e_i ed esibire un'altra espressione α-β-equivalente ad e_i in cui appare solo la variabile "P".

3. Dire se le seguenti coppie di espressioni del λ-calcolo sono α-β-equivalenti:

   - $(\, \lambda\, Y\, .\, (\, Y\ \lambda\, Z\, .\, (\, Y\ (\, X\ Z\,)\,)\,)\ \lambda\, Z\, .\, (\, Z\ Z\,)\,)$
     $(\, \lambda\, T\, .\, (\, (\, Y\ T\,)\ (\, Y\ T\,)\,)\ \lambda\, T\, .\, (\, (\, Y\ T\,)\ (\, Y\ T\,)\,)\,)$;
   - $(\, \lambda\, X\, .\, (\, (\, I\ X\,)\ X\,)\ \lambda\, X\, .\, (\, (\, I\ X\,)\ X\,)\,)$
     $(\, \lambda\, W\, .\, (\, I\ (\, W\ W\,)\,)\ \lambda\, Z\, .\, (\, I\ (\, Z\ Z\,)\,)\,)$
     dove $I = \lambda\, X\, .\, X$.
   - $(\, \lambda\, X,X\, .\, (\, X\ X\,)\ (\, \lambda\, Y\, .\, (\, Y\ Y\,)\ \lambda\, Z\, .\, (\, Z\ Z\,)\,)\,)$
     $(\, \lambda\, Z,T\, .\, (\, Z\ T\,)\ (\, \lambda\, X,X\, .\, (\, X\ X\,)\ (\, \lambda\, Z\, .\, X\ X\,)\,)\,)$
   - $(\, (\, \lambda\, X,Y\, .\, (\, X\ T\,)\ \lambda\, Y,X\, .\, (\, X\ Y\,)\,)\ Z\,)$
     $(\, (\, \lambda\, Y,X\, .\, (\, Y\ Z\,)\ \lambda\, X,Y\, .\, (\, Y\ X\,)\,)\ T\,)$
   - $(\, (\, \lambda\, X,Y\, .\, (\, \lambda\, Y,Z\, .\, (\, (\, X\ Z\,)\ X\,)\ Y\,)\ \lambda\, X\, .\, X\,)\ (\, Z\ Z\,)\,)$
     $(\, \lambda\, Z,Y\, .\, (\, Y\ Z\,)\ \lambda\, Y\, .\, Y\,)$.

4. Esibire e_1, e_2 espressioni del λ-calcolo puro tale che le seguenti coppie di espressioni siano α-β-equivalenti:
   - $(\lambda X . (\lambda Y . (((Y\ X) (Y\ Z)) \lambda Y . (Y\ Z)) \lambda Y, T . Y) e_1)$,
     O;
   - $(\lambda X . (\lambda X . (e_1\ X) e_1) X)$,
     $(\lambda X . (\lambda X . (X\ e_2) X) e_2)$.

1.4 PARWHILE

Si assuma che nei seguenti programmi PARWHILE, le variabile siano tutte di tipo intero, implicitamente inizializzate ad 0, i file di input, output facciano parte dello stato ed i risultati finali delle esecuzioni terminanti siano dati solamente dal contenuto del file di output, non considerando quindi la memoria.

1. Per ogni gruppo di programmi PARWHILE dati di seguito dire quali sono semanticamente equivalenti.

   - $\{< X = 1; Y = 2 + X; Y = 3 + X >; !(X); !(Y); ||$
     $< Y = 1; X = 2 + Y; X = 3 + Y >; !(X); !(Y); ||$
     $\mathbf{nil}\}$

     $\{X = 1; Y = 2 + X; Y = 3 + X; !(X); !(Y); ||$
     $< Y = 1; X = 2 + Y; X = 3 + Y >; !(X); !(Y); ||$
     $\mathbf{nil}\}$

     $\{< X = 1; Y = 2 + X; Y = 3 + X >; !(X); !(Y); ||$
     $Y = 1; X = 2 + Y; X = 3 + Y; !(X); !(Y); ||$
     $\mathbf{nil}\}$

   - $\{\mathbf{while}\ (X <= 10)\{\mathbf{await}(Y > X); X = X + 1; \}!(X); ||$
     $\mathbf{while}\ (Y <= 10)\{\mathbf{await}(X => Y); Y = Y + 1; \}!(Y); ||$
     $\mathbf{nil}\}$

     $\{\mathbf{while}\ (X <= 10)\{\mathbf{await}(Y > X); X = X + 1; \}!(X); ||$
     $\mathbf{while}\ (Y <= 10)\{Y = Y + 1; \}!(Y); ||$
     $\mathbf{nil}\}$

     $\{\mathbf{while}\ (X <= 10)\{\mathbf{await}(Y > X); X = X + 1; \}!(X); ||$
     $\mathbf{while}\ (Y <= 10)\{Y = Y + 1; \}!(Y); ||$
     $\mathbf{nil}\}$

   - $\{\mathbf{await}(Y > X); X = X + 1; !(X)||$
     $\mathbf{await}(Z > Y); Y = Y + 2; !(Y)||$
     $\mathbf{await}(X > Z); Z = Z + 3; !(Z)||$
     $\mathbf{nil}\}$

     $\{\mathbf{await}(Y => X); X = X + 1; !(X); ||$
     $\mathbf{await}(Z > Y); Y = Y + 2; !(Y); ||$
     $\mathbf{await}(X > Z); Z = Z + 3; !(Z); ||$
     $\mathbf{nil}\}$

     $\{\mathbf{await}(Y => X); X = X + 1; !(X); ||$
     $\mathbf{await}(Z > Y); Y = Y + 2; !(Y); ||$
     $\mathbf{await}(X > Z); Z = Z + 3; !(Z); ||$
     $\mathbf{nil}\}$

   - $\{Y = 1; \mathbf{while}\ (X > 0)\{Y = X + 1; X = Y; !(X)\}||$
     $X = 1; \mathbf{while}\ (Y > 0)\{X = Y + 1; Y = X; !(Y)\}||$
     $\mathbf{nil}\}$

     $\{Y = 1; \mathbf{while}\ (X > 0)\{< Y = X + 1; X = Y; !(X) >\}||$
     $X = 1; \mathbf{while}\ (Y > 0)\{< X = Y + 1; Y = X; !(Y) >\}||$
     $\mathbf{nil}\}$

   - $\{\mathbf{while}\ (K < 4)\{< X = X + 1; K = K + 1 >; Y = X\}||$
     $\mathbf{while}\ (K < 4)\{< Z = Z - 1; K = K + 1 >; Y = Z\}||$
     $\mathbf{while}\ (K < 4)\{!(Y)\}||$
     $\mathbf{nil}\}$

     $\{\mathbf{while}\ (K < 4)\{< X = X + 1; K = K + 1 >; !(X)\}||$
     $\mathbf{while}\ (K < 4)\{< Z = Z - 1; K = K + 1 >; !(Z)\}||$
     $\mathbf{nil}\}$

2. Dire i possibili risultati dei seguenti programmi.

   - $\{X = 1; Y = 2; ||$
     $\textbf{await}(Y > 0)!(0); X = 0; ||$
     $\textbf{await}(X > 0)!(1); ||$
     $\textbf{nil}\}$

3. Si consideri il seguente programma PARWHILE

   $\{Z = 0; Y = 1; X = 1; ||$
   $\textbf{await}(Z > 0)!(0); ||$
   $\textbf{await}(Y > 0)!(1); ||$
   $\textbf{await}(X > 0)!(2); ||$
   $\ldots ||\textbf{nil}\}$

   Calcolarne i risultati quando i puntini vengono sostituiti con **skip**.

   Sostituire i puntini in modo da ottenere un programma il cui unico risultato sia non terminazione.

   Sostituire i puntini in modo da ottenere un programma il cui tutti i risultati siano terminazioni corrette, e dire quali sono.

1.5 CCS

1. Dire se i seguenti gruppi di behaviour CCS sono equivalenti rispetto alla semantica di traccia e/o a quella di bisimulazione forte.

   Esibirne inoltre gli alberi di transizione e di sincronizzazione.

   $\overline{\alpha} \,.\, \beta \,.\, \textbf{nil} \;\; || \;\; \overline{\beta} \,.\, \textbf{nil}$,

   $\overline{\alpha} \,.\, \beta \,.\, \overline{\beta} \,.\, \textbf{nil} \; + \; \overline{\alpha} \,.\, \overline{\beta} \,.\, \beta \,.\, \textbf{nil} \; + \; \beta \,.\, \overline{\alpha} \,.\, \overline{\beta} \,.\, \textbf{nil} \; + \; \beta \,.\, \overline{\beta} \,.\, \overline{\alpha} \,.\, \textbf{nil} \; + \; \overline{\beta} \,.\, \beta \,.\, \overline{\alpha} \,.\, \textbf{nil} \; + \; \overline{\beta} \,.\, \overline{\alpha} \,.\, \alpha \,.\, \textbf{nil}$;

2. Siano α e β due azioni *prefissate*, si consideri quindi il seguente calcolo CCS$^{\%}$ ottenuto aggiungendo al CCS il combinatore ! la cui sintassi è data da

   $E ::= E_1 \; ! \; E_2$

   e la cui semantica operazionale è definita dalle seguenti regole:

   $$\frac{x \xrightarrow{\alpha} x' \qquad y \xrightarrow{\overline{\alpha}} y'}{x \; ! \; y \xrightarrow{\tau} (x' \; ! \; y') \;\; || \;\; y} \qquad\qquad \frac{x \xrightarrow{\overline{\beta}} x' \qquad y \xrightarrow{\beta} y'}{x \; ! \; y \xrightarrow{\tau} x \;\; || \;\; (x' \; ! \; y')}$$

   $$\frac{x \xrightarrow{l} x'}{x \; ! \; y \xrightarrow{l} x' \; ! \; y} \; l \neq \alpha, \overline{\alpha}, \beta, \overline{\beta} \qquad\qquad \frac{y \xrightarrow{l} y'}{x \; ! \; y \xrightarrow{l} x \; ! \; y'} \; l \neq \alpha, \overline{\alpha}, \beta, \overline{\beta}$$

   - Dare gli alberi di sincronizzazione dei seguenti behaviour di CCS$^{\%}$
     $(\alpha \,.\, \gamma \,.\, \textbf{nil}) \; ! \; (\overline{\gamma} \,.\, \overline{\alpha} \,.\, \textbf{nil})$
     $(\gamma \,.\, \alpha \,.\, \textbf{nil}) \; ! \; (\overline{\gamma} \,.\, \overline{\alpha} \,.\, \textbf{nil})$.
   - Dire se ! è associativo e /o commutativo rispetto alla semantica di bisimulazione forte.
   - Dire, motivando, se è vero o falso che
     "$p_1 \; ! \; p_2$ è in bisimulazione forte con $p_1 \;\; || \;\; p_2$
     sse
     p_1 e p_2 non contengono nè α nè β".

3. - Si pensi di definire una variante CCS^2 del CCS in cui la regola di sincronizzazione è sostituita da regole che modellano una sincronizzazione doppia: perchè ci sia una sincronizzazione con risultato τ ci deve essere un raddoppio delle capacità complementari e sullo stesso canale.
     Si diano le regole appropriate.
   - Si discuta l'associatività e commutatività del parallelo nel nuovo calcolo.

4. Dare gli alberi di transizione e di sincronizzazione dell'espressione P:

   $$P \equiv \alpha \,.\, \beta \,.\, \textbf{nil} \;\; || \;\; \overline{\beta} \,.\, \overline{\alpha} \,.\, \gamma \,.\, \textbf{nil} \;\; || \;\; \beta \,.\, \overline{\gamma} \,.\, \textbf{nil}$$

   Trovare, se esiste, Q in bisimulazione forte con P ma senza l'operatore $||$.

1.5.1 Costrutti ancora da vedere

1. Si considerino i seguenti behaviours CCS:

   $H = (\alpha \,.\, H) \;+\; (\alpha \,.\, \mathbf{nil})$

   $X = X \;+\; (\alpha \,.\, \mathbf{nil})$

   $Y = Y \;+\; X$

   $Z = (\alpha \,.\, H) \;+\; Z$

   Dire quali di questi processi sono o no equivalenti per bisimulazione forte.