AltriEsercizi3 (lunedí 4/4/2016) Esercizio1

Esercizio

Si provi che la seguente grammatica G ambigua:
$\mathtt{G} \equiv (\{\mathtt{S}, \mathtt{A}, \mathtt{B}, \mathtt{C}\}, \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}, \mathtt{S}, \{\mathtt{S} \to \mathtt{AS}|\mathtt{BS}|\mathtt{CS}, \mathtt{A} \to \mathtt{aA}|\mathtt{a}, \mathtt{B} \to \mathtt{bB}|\mathtt{b}, \mathtt{C} \to \mathtt{cC}|\mathtt{c}\})$

- $\mathtt{L(G)} = \{\}$, perchè tale è l'insieme $\{\mathtt{w} \in \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}^* \mid \mathtt{S} \to \mathtt{w}\}$
- Il testo conteneva un'omissione

Esercizio

Si provi che la seguente grammatica G ambigua:
$\mathtt{G} \equiv (\{\mathtt{S}, \mathtt{A}, \mathtt{B}, \mathtt{C}\}, \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}, \mathtt{S}, \{\mathtt{S} \to \mathtt{AS}|\mathtt{BS}|\mathtt{CS}|\epsilon, \mathtt{A} \to \mathtt{aA}|\mathtt{a}, \mathtt{B} \to \mathtt{bB}|\mathtt{b}, \mathtt{C} \to \mathtt{cC}|\mathtt{c}\})$

- Proviamo esistenza di $\mathtt{w}$ che è frontiera di più parse tree di G.

AltriEsercizi3 (lunedí 4/4/2016) Esercizio1-bis

Esercizio

Si provi che la seguente grammatica G ambigua:
$\mathtt{G} \equiv (\{\mathtt{S}, \mathtt{A}\}, \{\mathtt{a}\}, \mathtt{S}, \{\mathtt{S} \to \mathtt{AS}|\epsilon, \mathtt{A} \to \mathtt{aA}|\epsilon\})$

- Proviamo che $\mathtt{a}^{\mathtt{n}}$ (incluso n=0) è riconosciuta in modo ambiguo.

- La stessa cosa vale per la grammatica precedente? In caso contrario indicare una stringa ambigua in entrambe le grammatiche di esercizio1 e esrcizio1bis.

AltriEsercizi3 (lunedí 4/4/2016) Esercizio2

Esercizio

(a) Si dia una semantica SOS del Lambda Calcolo.
(b) La si applichi al programma (i.e termine chiuso)

$$(\lambda y.\lambda x.((\lambda y.\lambda x.x + y)(x - 3)y)\ 1\ 5$$

dove i simboli di costante +, -, 1, 2, 5, denotino le ovvie operazioni e interi, rispettivamente.

(a)

Linguaggio. $\Lambda = X \mid \Pi \mid \lambda X.\Lambda \mid \Lambda\Lambda$

Stato. (Ambiente) Sequenza $(x_1, v_1)...(x_k, v_k)$, dove: $k >= 0$, $x_i \in X$, $v_i \in \Pi$

Regole. $x, y \in X$, $v, v_i \in \Pi$, $t, t_i \in \Lambda$

$$\frac{<x,\sigma> \rightarrow v_1 \quad x \neq y}{<x,(y,v_2)\sigma> \rightarrow v_1}\ [\text{var1}] \qquad \frac{x = y}{<x,(y,v)\sigma> \rightarrow v}\ [\text{var2}] \qquad \frac{t \in \Pi}{<t,\sigma> \rightarrow t}\ [\text{const}]$$

$$\frac{<t_2,\sigma> \rightarrow v}{<(\lambda x.t_1)t_2,\sigma> \rightarrow <t_1,(x,v)\sigma>}\ [\text{app}] \qquad \frac{<t_1,\sigma> \rightarrow <t_1',\sigma>}{<t_1t_2,\sigma> \rightarrow <t_1't_2,\sigma>}\ [\text{cong}]$$

Osservazioni.

1. Le regole forniscono un meccanismo per evitare collisione di nome senza uso esplicito di α-conversione.
2. Le regole non forniscono alcun trattamento di η-conversione. Ma per il calcolo non è limitante (η-conversione è coinvolto nella [prova] di proprietà).

3-6. ... altre osservazioni: Le consideriamo dopo soluzione punto (b).

AltriEsercizi3 (lunedí 4/4/2016) Esercizio2 b

Esercizio

(a)....
(b) La si applichi al programma (i.e termine chiuso)

$$(\lambda y.\lambda x.((\lambda y.\lambda x.x + y)(x - 3)y)\ 1\ 5$$

(a) Regole. $x, y \in \mathbb{X},\ v, v_i \in \Pi,\ t, t_i \in \Lambda$

$$\frac{<x,\sigma> \rightarrow v_1 \quad x \neq y}{<x,(y,v_2)\sigma> \rightarrow v_1}\ [\text{var1}] \qquad \frac{x = y}{<x,(y,v)\sigma> \rightarrow v}\ [\text{var2}] \qquad \frac{t \in \Pi}{<t,\sigma> \rightarrow t}\ [\text{const}]$$

$$\frac{<t_2,\sigma> \rightarrow v}{<(\lambda x.t_1)t_2,\sigma> \rightarrow <t_1,(x,v)\sigma>}\ [\text{app}] \qquad \frac{<t_1,\sigma> \rightarrow <t_1',\sigma>}{<t_1 t_2,\sigma> \rightarrow <t_1' t_2,\sigma>}\ [\text{cong}]$$

(b) Sia: $t \equiv (\lambda y.\lambda x.((\lambda y.\lambda x.x + y)(x - 3)y)\ 1\ 5$

$$\begin{aligned}
<t,\perp> &\rightarrow_{const/app}{}^{1} \\
&\rightarrow_{const/app} <\lambda x.((\lambda y.\lambda x.x + y)(x - 3)y)\ 5,(y,1)> \\
&\rightarrow_{const/app} <(\lambda y.\lambda x.x + y)(x - 3)y,(x,5)(y,1)> \\
&\qquad\qquad <(x - 3),(x,5)(y,1)> \rightarrow_{var2/app/-}\ 2 \\
&\rightarrow_{cong/const/app} <(\lambda x.x + y)y,(y,2)(x,5)(y,1)> \\
&\rightarrow_{var2/app} <x + y,(x,2)(y,2)(x,5)(y,1)> \\
&\rightarrow_{var1/var2/app/+}\ 4
\end{aligned}$$

[1] L'ambiente iniziale $\perp$ è assunto indefinito, i.e. senza legami. Il pedice xxx che appare in $\rightarrow_{xxx}$ indica la regola SOS applicata.

AltriEsercizi3 (lunedí 4/4/2016) Esercizio2 - Osservazioni

Esercizio

(a) Si dia una semantica SOS del Lambda Calcolo.
(b) La si applichi al programma (i.e termine chiuso)

$$(\lambda y.\lambda x.((\lambda y.\lambda x.x + y)(x - 5)y)\ 1\ 3$$

dove i simboli di costante +, -, 1, 2, 5, denotino le ovvie operazioni e interi, rispettivamente.

(a)

Linguaggio. $\Lambda = X \mid \Pi \mid \lambda X.\Lambda \mid \Lambda\Lambda$

Stato. (Ambiente) Sequenza $(x_1, v_1)...(x_k, v_k)$, dove: $k >= 0$, $x_i \in X$, $v_i \in \Pi$

Regole[2]. $x, y \in X$, $v, v_i \in \Pi$, $t, t_i \in \Lambda$

$$\frac{<x,\sigma> \rightarrow v_1 \quad x \neq y}{<x,(y,v_2)\sigma> \rightarrow v_1}\ [\text{var1}] \qquad \frac{x = y}{<x,(y,v)\sigma> \rightarrow v}\ [\text{var2}] \qquad \frac{t \in \Pi}{<t,\sigma> \rightarrow t}\ [\text{const}]$$

$$\frac{<t_2,\sigma> \rightarrow v}{<(\lambda x.t_1)t_2,\sigma> \rightarrow <t_1,(x,v)\sigma>}\ [\text{app}] \qquad \frac{<t_1,\sigma> \rightarrow <t_1',\sigma>}{<t_1 t_2,\sigma> \rightarrow <t_1' t_2,\sigma>}\ [\text{cong}]$$

Osservazioni.

3. Non contiene: $(\lambda x.v)t \rightarrow_\Lambda v$, quando $t \equiv (\lambda x.xx)(\lambda x.xx)$.
4. Solo primo ordine: Argomenti valori non funzione (cmq. non definite per ricorsione)
5. Ψ non usabile: $\Psi F \rightarrow_\Lambda F(\Psi F)$ ma l'argomento (ΨF) non può essere ridotto a $v \in \Pi$
6. Solo funzioni totali: Sotto-Calcolo incompleto

[2] Usiamo $\rightarrow_\Lambda$ per indicare la relazione di conversione tra termini definita dagli assiomi α, β, η. Usiamo $\rightarrow$ per la transizione SOS.

AltriEsercizi3 (lunedí 4/4/2016) Esercizio2 - Soluzione

Esercizio

(a) Si dia una semantica SOS del Lambda Calcolo.

(a) Una differente semantica:

Linguaggio. $\Lambda = \mathtt{X} \mid \Pi \mid \lambda\,\mathtt{X}.\Lambda \mid \Lambda\Lambda$

Stato. (Ambiente) Sequenza $(\mathtt{x}_1, \mathtt{Z}_1)...(\mathtt{x}_k, \mathtt{Z}_k)$,

dove: $\mathtt{k} >= 0,\ \mathtt{x}_i \in \mathtt{X},\ \mathtt{Z}_i \in \Lambda \times \mathtt{Ambiente}$

Regole[3]. $\mathtt{x}, \mathtt{y} \in \mathtt{X},\ \mathtt{v}, \mathtt{v}_i \in \Pi,\ \mathtt{t}, \mathtt{t}_i \in \Lambda,\ \mathtt{Z}, \mathtt{Z}_i \in \Lambda \times \mathtt{Ambiente}$

$$\frac{<\mathtt{x}, \sigma> \to \mathtt{Z}_1 \quad \mathtt{x} \neq \mathtt{y}}{<\mathtt{x}, (\mathtt{y}, \mathtt{Z}_2)\sigma> \to \mathtt{Z}_1}\ \text{var1} \qquad \frac{\mathtt{x} = \mathtt{y}}{<\mathtt{x}, (\mathtt{y}, \mathtt{Z})\sigma> \to \mathtt{Z}}\ \text{var2}$$

$$\frac{\mathtt{Z} \equiv [\mathtt{t}, \sigma_\mathtt{t}] \in \Lambda \times \mathtt{Ambiente}}{<\mathtt{Z}, \sigma> \to <\mathtt{t}, \sigma_\mathtt{t}>}\ \text{clos} \qquad \frac{\mathtt{t} \in \Pi}{<\mathtt{t}, \sigma> \to \mathtt{t}}\ \text{const}$$

$$\frac{<\mathtt{t}_2, \sigma> \to \mathtt{v}}{<(\lambda\,\mathtt{x}.\mathtt{t}_1)\mathtt{t}_2, \sigma> \to <\mathtt{t}_1, (\mathtt{x}, \mathtt{v})\sigma>}\ [\text{app}] \qquad \frac{<\mathtt{t}_1, \sigma> \to <\mathtt{t}_1', \sigma>}{<\mathtt{t}_1\mathtt{t}_2, \sigma> \to <\mathtt{t}_1'\mathtt{t}_2, \sigma>}\ [\text{cong}]$$

Osservazioni.

3. Contiene: $(\lambda\,\mathtt{x}.\mathtt{v})\mathtt{t} \to_\Lambda \mathtt{v}$, quando $\mathtt{t} \equiv (\lambda\,\mathtt{x}.\mathtt{xx})(\lambda\,\mathtt{x}.\mathtt{xx})$.
4. Higher Order. Funzioni sono "first class values"
5. Ψ esprimibile. $\Psi\mathtt{F} \to_\Lambda \mathtt{F}(\Psi\mathtt{F})$ e l'argomento $(\Psi\mathtt{F})$ è chiuso nel proprio ambiente.
6. Tutte le calcolabili,i.e. $\mathcal{F}$: Sotto-Calcolo Completo (diverso da Lambda-Calcolo Tipato)

[3] $\to_\Lambda$ indica la conversione tra termini definita dagli assiomi α, β, η. $\to$ indica la transizione SOS.

AltriEsercizi3 (lunedí 4/4/2016) Esercizio3

Esercizio

Esercizio3. Si
a) scriva
b) compili
c) esegua
una versione memoized del fattoriale in C.

- Vedi listing allegato, `MemoizedFact.c`, in cui confrontiamo anche i tempi della versione ordinaria con quella memoized discussa a lezione (vedi lucidi relativi).
- Per il confronto abbiamo utilizzato la libreria `time.h` e le variabili di tipo `clock_t` (tempo espresso in clock/s) e l'operazione `clock()`

AltriEsercizi3 (lunedí 4/4/2016) Esercizio4

Esercizio

Si consideri la versione del fattoriale memoized visto a lezione (slide 22 del 23/3/2016).

(a) Si discuta il ruolo delle dichiarazione static nelle strutture dati introdotte nella definizione di memoizedFact.

(b) Si dica come cambia a funzione calcolata da memoizedFact se il modificatore static rimosso.

- Ci aspettiamo che la memoria relativa sia allocata dinamicamente ad ogni nuova creazione di un Activation Record.
- Di conseguenza l'esecuzione condurrà a... (prendi il listing allegato e fai modifiche e test significativi)

AltriEsercizi2 (lunedì 4/4/2016) Esercizio5

Esercizio

Si consideri il listing `DerivaPack` *allegato contenente una versione strutturata in moduli C del programma Deriva. Si estenda il programma con una versione tail-recursive dell'operazione* `size` *che calcola il numero di alberi di un treeList. In particolare, si:*

a) Scriva ed aggiunga size nell'opportuno modulo

b) Ri-compili e ri-esegua il programma Deriva

(a)

- Ci aspettiamo di modificare il modulo ...
- Scarichiamo il Linsting e procediamo

(b)

- Aggiungiamo, nel main, codice opportuno per un test della nuova operazione

AltriEsercizi2 (lunedì 4/4/2016) Esercizio6

Esercizio

Si scriva un programma che calcola il massimo fattoriale calcolabile per la piattaforma C utilizzata dal programma.

- In un linguaggio con trattamento delle eccezioni potremmo procedere così: Iteriamo il calcolo del fattoriale sugli interi successori a partire da 0 o 1, fino a generare un eccezione per overflow di rappresentazione. Intrappoliamo l'eccezione e restituiamo l'intero precedente.
- Ma il C non ha una gestione delle eccezioni, ha solo una "prassi" per gestire alcune anomalie e tale prassi qui non si applica.
- Dobbiamo e possiamo escogitare una soluzione ad hoc per il fattoriale.
- Ne forniamo una nel listing `MaxFact.c` allegato.

AltriEsercizi2 (lunedì 4/4/2016) Esercizio7

Esercizio

Si discuta un programma per la definizione di tabelle hash a liste di collisione e con funzione hash a divisione.

- L'esercizio6 precedente mostra che l'implementazione del fattoriale con memoization richiede tabelle di memorizzazione di modeste dimensioni e che l'aritmetica delle macchine attuali non operano con fattoriale oltre un piccolo naturale (prossimo a 20)
- Nondimeno, l'impiego di memoization può richiedere l'uso di tabelle hash per rappresentare la mappa finita delle applicazioni di funzione correntemente utilizzate dalla computazione. Limitandoci a funzioni monodiche, l'argomento una chiave, e l'immagine della funzione il valore associato alla chiave.
- **Tabelle a liste di collisione**: Hanno chiavi condivise da più argomenti (distr. unifor.) ed associato ad ogni chiave una lista di coppie $(a_n, v_n)...(a_1, v_1)$, dove a_n è l'n-esimo argomento con tale chiave utilizzato, nella computazione del programma, come argomento della funzione, e v_n è l'immagine calcolata.
- **Funzione hash a divisione**: `chiave(a) = P mod a`, dove P è un intero (primo non prossimo a potenza di 2).
- Implementare in C per il prossimo lunedì 11/4/2016, con operazioni:
  `add, remove, find`.