Programmazione A. A. 2004/2005

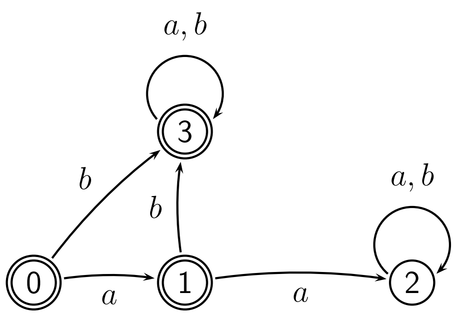
VI° Appello del 16/09/2005

ISTRUZIONI: Scrivere **in stampatello** COGNOME e NOME su ogni foglio. Non occorre consegnare la brutta copia e il testo. Coloro che non vogliono consegnare possono andarsene, consegnando il testo, dopo un'ora dall'inizio del compito ed entro 15 minuti dalla scadenza del tempo.

ESERCIZIO 1 (5 punti)

Scrivere un automa che accetti tutte le stringhe di $\{a, b\}^*$ tranne quelle che iniziano con almeno due a.

SOLUZIONE



ESERCIZIO 2 (5 punti)

Si scriva una grammatica che generi tutte e sole le stringhe ben formate di un linguaggio di markup in cui ci sono tre tag: `a`, `b` e `c`. I tag `a` e `b` possono essere inseriti in qualsiasi punto della stringa e possono essere sia affiancati che annidati uno dentro l'altro a piacere; il modo corretto per inserirli è tramite l'apertura <a>, seguita da una stringa ben formata all'interno, seguita dal relativo tag di chiusura (idem con `b` al posto di `a`). Il tag `c` invece può essere inserito in qualsiasi punto della stringa tramite la sola sequenza <c />, senza la chiusura.

Fra l'apertuta e la chiusura di un tag, o comunque in qualsiasi parte di una stringa ben formata fuori da una sequenza di apertura o chiusura, può apparire il testo cui i tag fanno da markup. Si utilizzi il simbolo terminale `cdata` per rappresentare tale testo (cioè non occorre specificare la sua struttura con delle regole).

SOLUZIONE

Il modo più semplice è usare una ricorsione destra per tutti i tipi di elementi (in modo da simulare la scrittura estesa del testo). Quando si vuole inserire uno dei due tag con chiusura si inserisce un nuovo simbolo non terminale che si occuperà di far rispettare la chiusura e di ottenere all'interno un'altra stringa ben formata:

$$\begin{array}{lcl} S & \rightarrow & \mathtt{cdata}S \mid <\mathtt{c}/>S \mid TS \mid \epsilon \\ T & \rightarrow & <\mathtt{a}>S</\mathtt{a}> \mid <\mathtt{b}>S</\mathtt{b}> \end{array}$$

ESERCIZIO 3 (8 punti)

Si supponga di estendere la sintassi delle dichiarazioni delle variabili intere del linguaggio didattico come segue:

```
Decl ::= int Ide max Num; | int Ide = Exp max Num;
```

Una variabile x dichiarata con una clausola max n non potrà mai avere, durante la sua esistenza, un valore maggiore di n. Qualora venisse assegnata con un valore maggiore la semantica prevede che l'esecuzione si blocchi.

Modificare la struttura dello stato (suggerimento: nei frame associare al nome della variabile anche il valore massimo, oltre al valore corrente; per le variabili dichiarate normalmente porre tale valore a $\bot$ a indicare che non ci sono limiti massimi), le regole di $\rightarrow_{decl}$ per trattare il nuovo caso sintattico (e modificare le altre per inserire $\bot$) e la regola di $\rightarrow_{com}$ per l'assegnamento in modo che il sistema di transizione vada in una configurazione bloccata quando si tenta di assegnare ad una variabile dichiarata nel nuovo modo un valore maggiore del massimo possibile.

SOLUZIONE

Innanzitutto il tipo degli elementi semantici 'frame' cambia in questo modo:

$$\varphi\colon \mathtt{Ide} \longrightarrow \mathtt{Val} \times (\mathbb{N} \cup \{\bot\})$$

Bisogna adattare a questo cambiamento le regole per le dichiarazioni già esistenti:

$$dec_{var-1} \qquad \frac{\sigma = \varphi.\sigma'', \quad \sigma' = \varphi[{}^{(\underline{\omega},\,\bot)}/\mathtt{x}].\sigma''}{\langle \mathtt{T\ x;}, \sigma\rangle \rightarrow_{dec} \sigma'}$$

$$dec_{var-2} \qquad \frac{\langle \mathrm{E}, \sigma\rangle \rightarrow_{exp} \underline{v}, \quad \sigma = \varphi.\sigma'', \quad \sigma' = \varphi[{}^{(\underline{v},\,\bot)}/\mathtt{x}].\sigma''}{\langle \mathtt{T\ x\ =\ E;}, \sigma\rangle \rightarrow_{dec} \sigma'}$$

Poi bisogna aggiungere le due regole per i due nuovi casi sintattici:

$$dec_{var-3} \qquad \frac{\eta(\mathtt{n}) = \underline{n}, \quad \sigma = \varphi.\sigma'', \quad \sigma' = \varphi[{}^{(\underline{\omega},\,\underline{n})}/\mathtt{x}].\sigma''}{\langle \mathtt{int\ x\ max\ n;}, \sigma\rangle \rightarrow_{dec} \sigma'}$$

$$dec_{var-4} \qquad \frac{\eta(\mathtt{n}) = \underline{n}, \quad \langle \mathrm{E}, \sigma\rangle \rightarrow_{exp} \underline{m}, \quad \underline{m} \leq \underline{n}, \quad \sigma = \varphi.\sigma'', \quad \sigma' = \varphi[{}^{(\underline{m},\,\underline{n})}/\mathtt{x}].\sigma''}{\langle \mathtt{int\ x\ =\ E\ max\ n;}, \sigma\rangle \rightarrow_{dec} \sigma'}$$

Infine bisogna inserire il controllo nell'assegnamento. Per far bloccare il sistema basta semplicemente scrivere le regole per i casi giusti e nessuna regola per i casi sbagliati:

$$com_{=-1} \qquad \frac{\sigma(\mathtt{x}) = (\underline{t}, \bot), \quad \langle \mathrm{E}, \sigma\rangle \rightarrow_{exp} \underline{v}}{\langle \mathtt{x\ =\ E;}, \sigma\rangle \rightarrow_{com} \sigma[{}^{(\underline{v},\,\bot)}/\mathtt{x}]}$$

$$com_{=-2} \qquad \frac{\sigma(\mathtt{x}) = (\underline{m}, \underline{n}), \quad \langle \mathrm{E}, \sigma\rangle \rightarrow_{exp} \underline{k}, \quad \underline{k} \leq \underline{n}}{\langle \mathtt{x\ =\ E;}, \sigma\rangle \rightarrow_{com} \sigma[{}^{(\underline{k},\,\underline{n})}/\mathtt{x}]}$$

ESERCIZIO 4 (5 punti)

Si scriva l'implementazione del seguente metodo:

```
\** Restituisce, se c'è, la posizione del primo elemento di un
    array di interi che sia divisibile per due diversi numeri dati
 @param a un array di interi
 @param d1 uno dei due divisori
 @param d2 uno dei due divisori
 @return la posizione del primo elemento di a che è divisibile
     sia per d1 che per d2; -1 se un tale elemento non è presente
 */
public int m(int[] a, int d1, int d2) {...}
```

SOLUZIONE

```
public int m(int[] a, int d1, int d2) {
  // classico schema di ricerca lineare incerta
  boolean trovato = false;
  int i = 0;
  while (i < a.length && !trovato)
    if (a[i] % d1 == 0 && a[i] % d2 == 0)
      trovato = true;
    else i++;
  if (trovato)
    return i;
  else return -1;
}
```

ESERCIZIO 5 - PER GLI STUDENTI DA 5 CFU (7 punti)

Si considerino i linguaggi $L_1 = \{a^n\, b^{n+k} \mid n, k > 0\}$ e $L_2 = \{a^n\, c^k\, b^n \mid n, k > 0\}$. Si definisca un sistema di transizione tra le cui configurazioni ci siano $\{\langle \alpha, \mathsf{START}\rangle \mid \alpha \in L_1\} \cup \{\langle \alpha, \mathsf{STOP}\rangle \mid \alpha \in L_2\}$ e che sia tale che $\langle a^n\, b^{n+k}, \mathsf{START}\rangle \xrightarrow{*} \langle a^n\, c^k\, b^n, \mathsf{STOP}\rangle$. Si scriva dapprima una grammatica per L_1 e si definiscano le regole del sistema guidate da tale sintassi.

SOLUZIONE

Adottiamo un approccio big step con regole guidate dalla sintassi. Innanzitutto definiamo le regole sintattiche per il linguaggio L_1.

$$
\begin{array}{rcl}
S & \rightarrow & aSb \mid aBb \\
B & \rightarrow & bB \mid b
\end{array}
$$

Il sistema è $S = \langle \Gamma, T, \rightarrow\rangle$ dove:

$$\Gamma = \{\langle \alpha, \mathtt{START}\rangle \mid \alpha \in L_1\} \cup \{\langle \beta, \mathtt{STOP}\rangle \mid \beta \in L_2\}$$

$$T = \{\langle \beta, \mathtt{STOP}\rangle \mid \beta \in L_2\}$$

e $\rightarrow$ è definita dalle seguenti regole:

$$(r1)\quad \frac{\alpha \in L(S),\quad \langle \alpha, \mathtt{START}\rangle \rightarrow \langle \beta, \mathtt{STOP}\rangle}{\langle a\alpha b, \mathtt{START}\rangle \rightarrow \langle a\beta b, \mathtt{STOP}\rangle}$$

$$(r2)\quad \frac{\alpha \in L(B),\quad \langle \alpha, \mathtt{START}\rangle \rightarrow \langle \beta, \mathtt{STOP}\rangle}{\langle a\alpha b, \mathtt{START}\rangle \rightarrow \langle a\beta b, \mathtt{STOP}\rangle}$$

$$(r3)\quad \frac{\alpha \in L(B),\quad \langle \alpha, \mathtt{START}\rangle \rightarrow \langle \beta, \mathtt{STOP}\rangle}{\langle b\alpha, \mathtt{START}\rangle \rightarrow \langle c\beta, \mathtt{STOP}\rangle}$$

$$(r4)\quad \frac{}{\langle b, \mathtt{START}\rangle \rightarrow \langle c, \mathtt{STOP}\rangle}$$

ESERCIZIO 5 - PER GLI STUDENTI DA 6 CFU (7 punti)

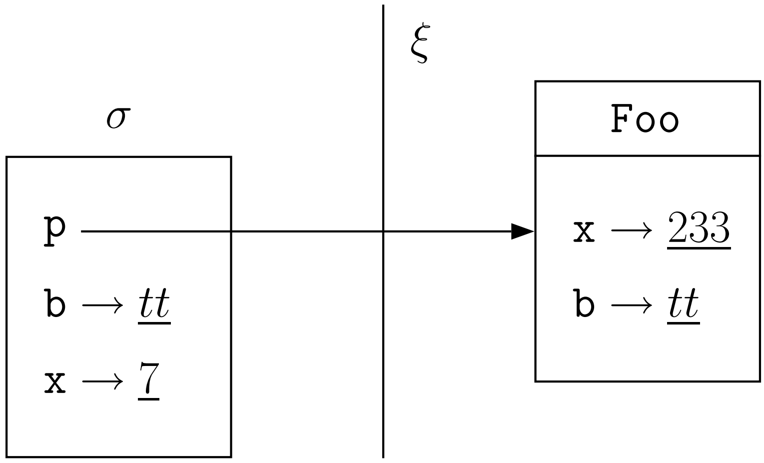
Si consideri il seguente programma nel linguaggio didattico:

```
prog {
  class Foo {
    public int x;
    public boolean b;
    public void m(int x, boolean b) {
      if (b) this.x = this.x * x;
        else if (this.b) this.x = this.x + x;
          else {
            this.x = this.x + x * x;
            this.b = !this.b;
          }
    }
  }
{
int x = 7;
boolean b = true;
Foo p = new Foo;
p.x = 8;
p.b = false;
p.m(x + p.x, b && p.b); (1)
x = p.x * x;
p.m(x, p.b || b); (2)
}}
```

Disegnare lo stato (pila di frame e heap) nei punti (1) e (2) del programma.

SOLUZIONE

Lo stato al punto (1) è il seguente:



Lo stato al punto (2) è il seguente: