

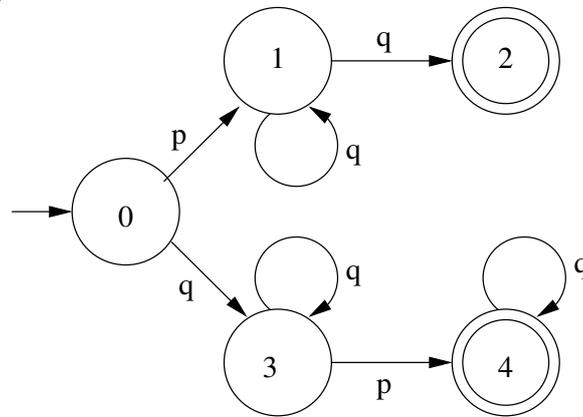
Programmazione A. A. 2004/2005

IV° Appello del 22/06/2005

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 (4 punti)

Descrivere con una espressione regolare o con espressioni su insiemi il linguaggio accettato dal seguente automa:



SOLUZIONE

Il linguaggio accettato è quello denotato da $pq^+ \mid q^+pq^*$.

ESERCIZIO 2 (5 punti)

Scrivere una grammatica che generi il seguente linguaggio:

$$L = \{a^n b^m \mid n, m > 0 \text{ e } m = n + 2\}$$

SOLUZIONE

Il linguaggio si può scrivere come $L = \{a^n bb^n \mid n > 0\}$. In questa forma risulta evidente che basta usare una ricorsione centrale con un caso base opportuno:

$$S \rightarrow aSb \mid abbb$$

ESERCIZIO 3 (7 punti)

Si supponga di estendere la sintassi dei comandi del linguaggio didattico come segue:

Com ::= repeat Com until Ide is zero;

L'esecuzione di un comando repeat C until x is zero; consiste nell'eseguire il comando C e poi testare il valore della variabile intera x. Se tale valore è zero allora il comando termina, altrimenti il ciclo riparte dal nuovo stato.

Definire le regole di semantica operativa per il nuovo comando con riferimento al modello in cui lo stato è composto da una pila di frame.

SOLUZIONE

$$\begin{array}{c} \text{com}_{\text{repeat}=0} \frac{\langle C, \sigma \rangle \rightarrow_{\text{com}} \sigma', \quad \sigma'(x) = \underline{0}}{\langle \text{repeat } C \text{ until } x \text{ is zero};, \sigma \rangle \rightarrow_{\text{com}} \sigma'} \\ \\ \text{com}_{\text{repeat} \neq 0} \frac{\langle C, \sigma \rangle \rightarrow_{\text{com}} \sigma', \quad \sigma'(x) \neq \underline{0}}{\langle \text{repeat } C \text{ until } x \text{ is zero};, \sigma' \rangle \rightarrow_{\text{com}} \sigma''} \\ \langle \text{repeat } C \text{ until } x \text{ is zero};, \sigma \rangle \rightarrow_{\text{com}} \sigma'' \end{array}$$

ESERCIZIO 4 (7 punti)

Si scriva l'implementazione del seguente metodo:

```
\** Restituisce la parita' di una stringa di bit rappresentata
    da un array di boolean.
@param b un array di boolean in cui true rappresenta 1 e false
        rappresenta 0
@return true se nella stringa ci sono un numero pari di 1 (zero
        e' numero pari), false altrimenti oppure se l'array
        non esiste. */
public boolean parity (boolean [] b) {...}
```

SOLUZIONE

```
public boolean parity(boolean[] b){
    if (b == null) return false;
    int num1 = 0;
    for (int i = 0; i < b.length; i++)
        if (b[i]) num1++;
    if (num1 % 2 == 0) return true;
    else return false;
}
```

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

Si considerino le due seguenti pile di frame:

$$\sigma_1 = \{x \mapsto 5, y \mapsto 3\}.\{z \mapsto 5, x \mapsto 7\}.\Omega$$

$$\sigma_2 = \{x \mapsto 4\}.\{y \mapsto 2\}.\{z \mapsto 6, y \mapsto 4, x \mapsto 9\}.\Omega$$

Si dica il valore dell'espressione aritmetica $x + y * z$ in ognuno dei seguenti stati:

1. σ_1
2. σ_2
3. $\sigma_1[2/x]$
4. $\sigma_2[\sigma_1(x)/y, 3/z, 7/w]$
5. $\sigma_3 = \{y \mapsto 4\}.\sigma_4$ dove σ_4 è tale che $\sigma_1 = \varphi.\sigma_4$.

SOLUZIONE

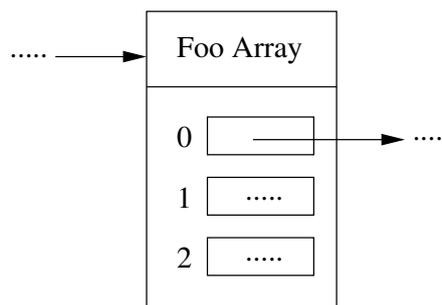
1. $\langle x + y * z, \sigma_1 \rangle \rightarrow_{exp} \underline{5} + [3 * 5] = \underline{20}$
2. $\langle x + y * z, \sigma_2 \rangle \rightarrow_{exp} \underline{4} + [2 * 6] = \underline{16}$
3. $\langle x + y * z, \sigma_1[2/x] \rangle \rightarrow_{exp} \underline{2} + [3 * 5] = \underline{17}$
4. $\langle x + y * z, \sigma_2[\sigma_1(x)/y, {}^3/z, {}^7/w] \rangle \rightarrow_{exp} \underline{4} + [5 * 3] = \underline{19}$
5. Si ha che $\sigma_3 = \{y \mapsto 4\} \cdot \{z \mapsto 5, x \mapsto 7\} \cdot \Omega$, quindi $\langle x + y * z, \sigma_3 \rangle \rightarrow_{exp} \underline{7} + [4 * 5] = \underline{27}$

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

Si consideri il seguente programma dato con la sintassi del linguaggio didattico (array inclusi):

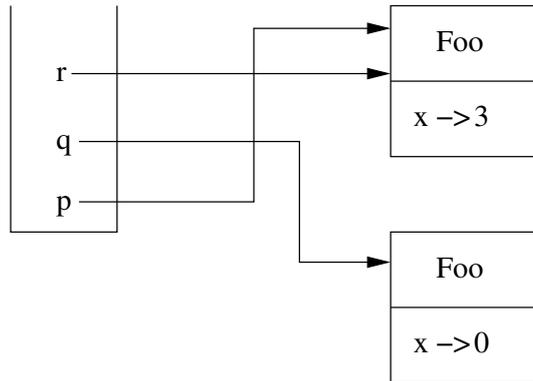
```
progr {  
  class Foo {  
    public int x;  
    public void m(int x) {  
      if (this.x==0) this.x = 100;  
      else this.x = x * this.x;  
    }  
  }  
}  
{  
  Foo p = new Foo; p.x = 3;  
  Foo q = new Foo; q.x = 0;  
  Foo r = p; (1)  
  Foo[] a = new Foo[3];  
  a[0] = p; a[1] = q; a[2] = r; (2)  
  for (int i = 0; i < a.length; i++)  
    a[i].m(2); (3)  
}
```

Si disegni lo stato, composto dalla pila di frame e dallo heap, nei punti del programma (1), (2) e (3). Per disegnare un oggetto array di tre elementi di tipo Foo si suggerisce la seguente rappresentazione grafica:

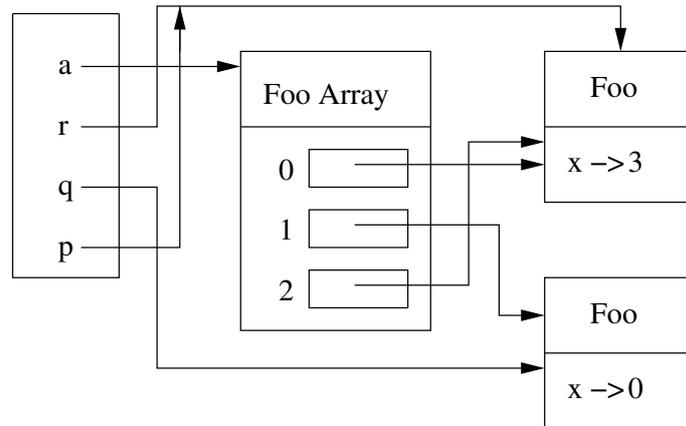


SOLUZIONE

Al punto (1):



Al punto (2):



Al punto (3):

