Programmazione A. A. 2004/2005

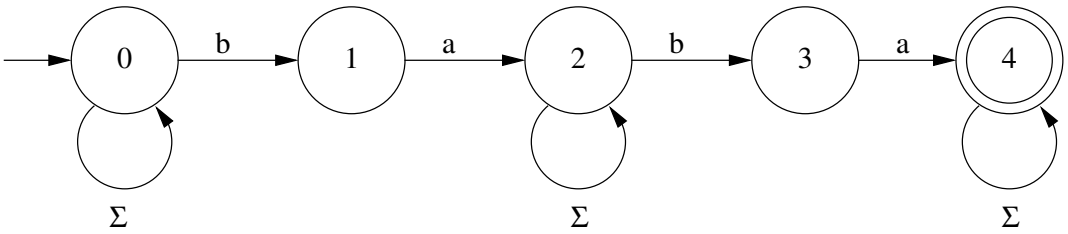
I° Appello del 17/12/2004 - Ascoli Piceno

ISTRUZIONI: Scrivere **in stampatello** COGNOME e NOME su ogni foglio. Vanno consegnati **tutti** i fogli: brutta copia e testo compresi. 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)

Sia $\Sigma = \{a, b, c\}$. Scrivere un automa che accetti tutte le stringhe sull'alfabeto Σ che contengono almeno due volte la sottostringa ba. Ad esempio possono essere accettate $baba$, $ccbabbcbac$, $bbacbaccbaaa$.

SOLUZIONE



ESERCIZIO 2 (5 punti)

Si scriva il linguaggio accettato dalla seguente grammatica mediante espressioni su insiemi:

$$\begin{array}{lcl} S & \rightarrow & aBbAc \\ A & \rightarrow & aaA \mid \epsilon \\ B & \rightarrow & Bb \mid cc \end{array}$$

SOLUZIONE

$$L = \{a\,c\,c\,b^n\,b\,a^{2m}\,c \mid n, m \geq 0\}$$

ESERCIZIO 3 (9 punti)

Si supponga di estendere la categoria sintattica `Com` con la seguente produzione:

```
Com ::= loopaccording (Exp,Exp) Com
```

L'esecuzione di un ciclo `loopaccording (E1,E2) C` consiste nel valutare dapprima, nello stato corrente, le espressioni `E1` ed `E2`, che devono avere un valore nell'insieme $\{\mathbf{tt}, \mathbf{ff}\}$, e poi agire come segue:

1. Se hanno un diverso valore di verità allora il ciclo esce senza fare niente
2. Se hanno lo stesso valore di verità allora viene eseguito il comando `C` e il ciclo viene fatto ripartire.

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

SOLUZIONE

$$com_{loopacc-div} \quad \frac{\langle \texttt{E1}, \sigma\rangle \rightarrow_{exp} \underline{\mathbf{b1}} \quad \langle \texttt{E2}, \sigma\rangle \rightarrow_{exp} \underline{\mathbf{b2}} \quad \underline{\mathbf{b1}}, \underline{\mathbf{b2}} \in \{\underline{\mathbf{tt}}, \underline{\mathbf{ff}}\} \quad \underline{\mathbf{b1}} \neq \underline{\mathbf{b2}}}{\langle \texttt{loopaccording (E1, E2) C}, \sigma\rangle \rightarrow_{com} \sigma}$$

$$com_{loopacc=} \quad \frac{\langle \texttt{E1}, \sigma\rangle \rightarrow_{exp} \underline{\mathbf{b1}} \quad \langle \texttt{E2}, \sigma\rangle \rightarrow_{exp} \underline{\mathbf{b2}} \quad \underline{\mathbf{b1}}, \underline{\mathbf{b2}} \in \{\underline{\mathbf{tt}}, \underline{\mathbf{ff}}\} \quad \underline{\mathbf{b1}} = \underline{\mathbf{b2}} \quad \langle \texttt{C}, \sigma\rangle \rightarrow_{com} \sigma' \quad \langle \texttt{loopaccording (E1, E2) C}, \sigma'\rangle \rightarrow_{com} \sigma''}{\langle \texttt{loopaccording (E1, E2) C}, \sigma\rangle \rightarrow_{com} \sigma''}$$

ESERCIZIO 4 (5 punti)

Si scriva l'implementazione del seguente metodo:

```
/** Conta il numero di elementi di un array che sono divisibili
    per un certo valore
    @param a l'array di interi da controllare
    @param d il valore per cui gli elementi devono essere divisibili
    @return Il numero di elementi di a che sono divisibili per d
  */
public int m(int a[], int d){
...
}
```

SOLUZIONE

```
public int m(int a[], int d) {
  if (d==0) return -1;
  int conta = 0;
  for (int i = 0; i < a.length; i++)
    if (a[i] % d == 0) conta++;
  return conta;
}
```

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

Si consideri la seguente grammatica:

$$\begin{array}{lcl} S & \rightarrow & cS \mid A \\ A & \rightarrow & bA \mid cA \mid a \end{array}$$

Si definisca un sistema di transizioni le cui configurazioni contengano, tra le eventuali altre, le seguenti:

$$\{\langle \alpha, \texttt{START} \rangle \mid \alpha \in L(S)\} \cup \{\langle n, \texttt{STOP} \rangle \mid n \in \mathbb{N}\}$$

e tale che $\langle \alpha, \texttt{START} \rangle \rightarrow^* \langle n, \texttt{STOP} \rangle$ se e solo se n è il numero di simboli c che occorrono in α.

SOLUZIONE

Adottiamo un approccio big step guidato dalla sintassi. Il sistema è $S = \langle \Gamma, T, \rightarrow \rangle$ dove:

$$\Gamma = \{\langle \alpha, \texttt{START} \rangle \mid \alpha \in L(S)\} \cup \{\langle n, \texttt{STOP} \rangle \mid n \in \mathbb{N}\}$$

$$T = \{\langle n, \texttt{STOP} \rangle \mid n \in \mathbb{N}\}$$

e $\rightarrow$ è definita dalle seguenti regole:

$$(r1) \quad \frac{}{\langle a, \texttt{START} \rangle \rightarrow \langle 0, \texttt{STOP} \rangle}$$

$$(r2) \quad \frac{\alpha \in \{b,c\}^* \cdot \{a\} \quad \langle \alpha, \texttt{START} \rangle \rightarrow \langle n, \texttt{STOP} \rangle}{\langle c\,\alpha, \texttt{START} \rangle \rightarrow \langle n+1, \texttt{STOP} \rangle}$$

$$(r3) \quad \frac{\alpha \in \{b,c\}^* \cdot \{a\} \quad \langle \alpha, \texttt{START} \rangle \rightarrow \langle n, \texttt{STOP} \rangle}{\langle b\,\alpha, \texttt{START} \rangle \rightarrow \langle n, \texttt{STOP} \rangle}$$

$$(r4) \quad \frac{\alpha \in \{c\} \cdot L(A) \quad \langle \alpha, \texttt{START} \rangle \rightarrow \langle n, \texttt{STOP} \rangle}{\langle c\,\alpha, \texttt{START} \rangle \rightarrow \langle n+1, \texttt{STOP} \rangle}$$

Nella regola (r4) la premessa $\alpha \in \{c\} \cdot L(A)$ si sovrappone per qualche stringa alla premessa $\alpha \in \{b,c\}^* \cdot \{a\}$ della regola (r2). Quindi in questo modo le regole del sistema non sono mutualmente esclusive. Tuttavia questo non è un problema poiché entrambe le regole fanno la stessa cosa. Una soluzione diversa consiste nell'eliminare del tutto la regola (r2).

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

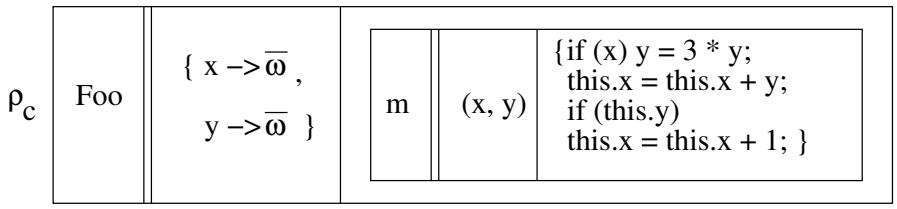
Si consideri il seguente programma nel linguaggio didattico:

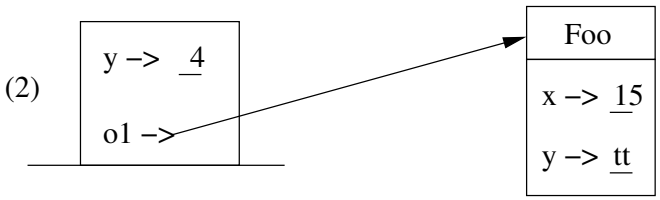
```
prog {
  class Foo {
    public int x;
```

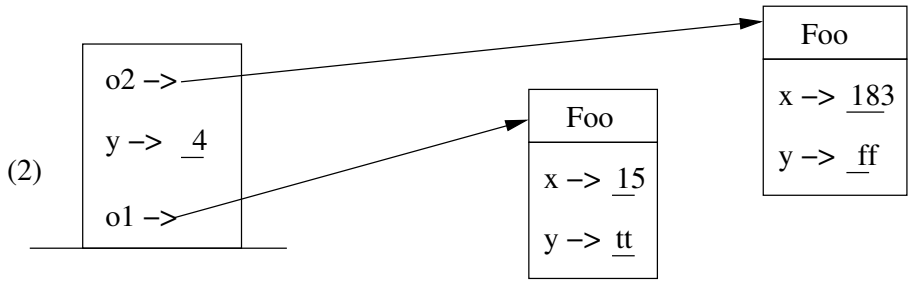
```
    public boolean y;
    public void m(boolean x, int y) {
      if (x) y = 3 * y;
      this.x = this.x + y;
      if (this.y) this.x = this.x + 1;
    }
  }  (1)
 { Foo o1 = new Foo;
   o1.x = 2;
   o1.y = true;
   int y = 4;
   o1.m(o1.y, y);        (2)
   Foo o2 = new Foo;
   o2.x = 3;
   o2.y = false;
   o2.m(true, o1.x * y); (3)
 }
}
```

Si disegni l'ambiente delle classi ρ_c al punto `(1)` e lo stato (pila di frame e heap) ai punti `(2)` e `(3)`.

SOLUZIONE






(2)
o2 –>
y –> 4
o1 –>
Foo
x –> 15
y –> tt
Foo
x –> 183
y –> ff