

## PROGRAMMAZIONE – II Appello del 13/12/2002

Scrivere **in stampatello** COGNOME, NOME e NUMERO DI MATRICOLA (se conosciuto) su ogni foglio consegnato e sul testo, che va consegnato insieme al compito.

### ESERCIZIO 1 (5 punti)

Si consideri l'alfabeto  $\Lambda = \{0, 1, B\}$ . Disegnare un automa che accetta tutte e sole le stringhe di elementi di  $\Lambda$  che contengono, al loro interno, **esattamente** 3 caratteri "B" (non necessariamente adiacenti). Ad esempio 11B00B01B è una stringa che deve essere accettata, come lo è BBB. Esempi di stringhe che non devono essere accettate sono 00101,  $\epsilon$ , BB1, B1B0B0B111.

### ESERCIZIO 2 (7 punti)

Si consideri l'alfabeto  $\Lambda = \{a, b, c\}$  e la seguente grammatica:

```
<S> ::= a<S>b | a<B>b | <AC>
<B> ::= <B>b | b
<AC> ::= <AC><D> | <D>
<D> ::= a | c | epsilon
```

- Descrivere formalmente il linguaggio generato dalla grammatica
- Disegnare l'albero di derivazione della stringa aabbbb
- Disegnare due alberi di derivazione per la stringa ac

### ESERCIZIO 3 (5 punti)

Si supponga di estendere la sintassi dei comandi con la seguente produzione:

```
Com ::= atleast Exp (Ide = Exp);
```

Il comando `atleast E1 (x = E)`; è un assegnamento controllato. Il valore  $v$  dell'espressione  $E1$  deve essere un intero e rappresenta il valore più grande che può essere assegnato all'identificatore  $x$  nell'assegnamento  $x = E$ . Se il valore di  $x$  dopo l'assegnamento è maggiore di  $v$ , allora  $x$  viene posto a  $v$ . Altrimenti il risultato è quello di un normale assegnamento.

Definire le regole di semantica operativa per trattare il nuovo comando. Si faccia riferimento al modello in cui lo stato è rappresentato solo con una pila di frame.

### ESERCIZIO 4 (6 punti)

Completare la definizione del seguente metodo.

```
public int metodo(int[] a)
/** Cerca, nell'array puntato da a, il primo elemento che e' uguale al
 * successivo (l'ultimo elemento di a non viene considerato per la
 * ricerca). Se trovato, restituisce la somma degli elementi a partire dalla
```

```

* posizione dell'elemento trovato fino alla fine dell'array.
* Altrimenti restituisce 0.
*/

```

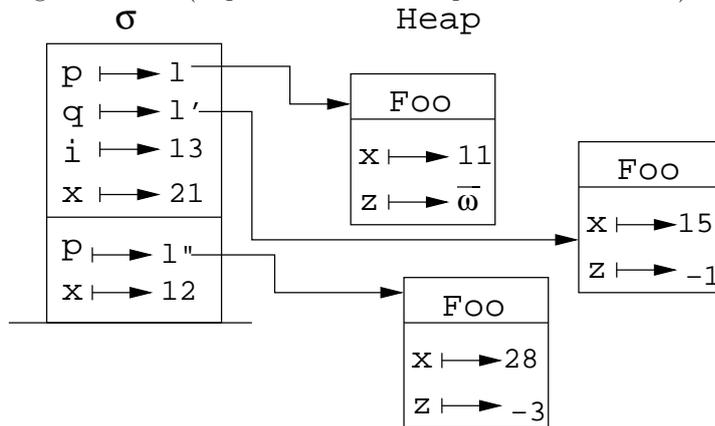
### ESERCIZIO 5 (7 punti)

Si consideri il seguente ambiente delle classi:

Foo		$\{x \mapsto \bar{\omega}, z \mapsto \bar{\omega}\}$	$\rho_m$
-----	--	--	----------

$\rho_m$	m1	y, z	$\{ \text{this.z} = z; \text{if} (\text{this.x} < y) \text{this.x} = y; \}$
----------	----	------	---

e il seguente stato (la pila di frame è composta da due frame):



Si supponga di chiamare il metodo, a partire da questo stato, con il seguente comando:  $p.m1(i, q.x + x);$ . Si disegni lo stato (pila di frame e heap) nei seguenti momenti:

- all'inizio dell'esecuzione del metodo (cioè lo stato in cui viene eseguita la prima istruzione del metodo)
- immediatamente prima dell'uscita (cioè lo stato risultante dall'ultima istruzione del metodo)
- dopo l'esecuzione del metodo (cioè lo stato in cui verrà eseguito il comando seguente a  $p.m1(i, q.x + x);$ ).