

PROGRAMMAZIONE – VIII Appello del 23/09/2003 — Camerino

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 (6 punti)

Siano $L_1 = \{a^n d c^m \mid n > 0, m \geq 0\}$ e $L_2 = \{b^n d c^m \mid n > 0, m \geq 0\}$ due linguaggi su $\{a, b, c, d\}^*$. Disegnare un automa che accetti il seguente linguaggio:

$$L = \{c s_1 c s_2 \cdots c s_k \mid k > 0, s_i \in L_1 \cup L_2 \text{ per } i = 1, 2, \dots, k\}$$

ESERCIZIO 2 (10 punti)

Si supponga di voler estendere il linguaggio delle espressioni aritmetiche visto a lezione con le *espressioni condizionali multiple*. Una espressione condizionale multipla è una lista di una o più espressioni con guardia. Una espressione con guardia è una espressione preceduta da una espressione booleana e da due punti. Le espressioni con guardia sono separate da virgole, cominciano con la parola `cond` e terminano con la parola `dnoc`. Ad esempio:

`cond x > 1 : y + 7 * 9, p && q : pippo + 4, z == 0 : z+1 dnoc`
è una espressione condizionale multipla di 3 espressioni con guardia. La prima espressione con guardia è `x > 1 : y + 7 * 9` dove `x > 1` è la guardia e `y + 7 * 9` è l'espressione.

(1) Estendere il linguaggio dalla categoria sintattica `Exp` facendo in modo che generi anche le espressioni condizionali multiple.

(2) Definire un sottosistema di transizioni del sistema \rightarrow_{exp} per valutare le espressioni condizionali multiple. Il valore di una espressione condizionale multipla deve essere il valore dell'espressione della prima espressione con guardia la cui guardia è vera. Le espressioni con guardia devono essere considerate dalla prima a partire da sinistra. Se nessuna guardia è vera, allora il valore deve essere quello dell'espressione dell'ultima espressione con guardia.

ESERCIZIO 3 (7 punti)

Si scriva il codice del metodo `public boolean pred(int [] a, int [] b)` che riceve in ingresso come parametri due puntatori ad array di interi `a` e `b` e restituisce `true` o `false`. Il metodo deve controllare se tutti gli elementi dell'array `a` sono minori di tutti gli elementi dell'array `b` e in tal caso restituire `true`. Nel caso contrario la risposta deve essere `false`.

ESERCIZIO 4 (7 punti)

Si consideri il seguente programma:

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

```

    public int inc() {
        this.x++;
    }
}

class El {
    public Val p;
    public int y;

    public void calc(int q) {
        if (p != null) {
            p.inc();
            this.y = p.x * q;
        } else this.y = 0;
    }
}

{
    Val a = new Val;
    a.x = 3;
    El b = new El;
    b.p = a;
    b.calc(4); (1)
    a = null;
    El c = b;
    b.p.inc();
    c.calc(2); (2)
}
}

```

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