PROGRAMMAZIONE 1 e LABORATORIO (A,B) - a.a. 2013/14

Prova scritta del 3 giugno 2014

Scrivere **IN STAMPATELLO** COGNOME, NOME, MATRICOLA e CORSO su ogni foglio consegnato

ESERCIZIO 1 (punti 5)
Dato l'alfabeto $\Lambda = \{a, b, c\}$, si definisca una grammatica che genera il seguente linguaggio

$$\mathcal{L} = \{a^n \alpha b^n \mid n > 0 \wedge (pari(n) \Rightarrow \alpha \in \{b, c\}^*) \wedge (dispari(n) \Rightarrow \alpha \in \{c\}^+)\}$$

dove *pari* e *dispari* hanno l'ovvio significato.

ESERCIZIO 2 (punti 5)
Definire un automa **deterministico** che riconosce le stringhe sull'alfabeto $\Lambda = \{a, b, c\}$ che contengono almeno due volte la sottostringa *ac*.

ESERCIZIO 3 (punti 5)
Dato il tipo degli alberi binari

```
type 'a btree = Void  | Node of 'a * 'a btree * 'a btree
```

si definisca in `CAML` una funzione `foo` con tipo

```
foo : 'a btree -> int * int
```

in modo che `(foo bt)` sia la coppia costituita dal numero di nodi foglia con etichetta maggiore della radice e dal numero di nodi foglia con etichetta minore o uguale della radice (la funzione non è definita su alberi vuoti).

ESERCIZIO 4 (punti 5)
Si definisca in `C` una funzione

```
int check (int a[], int dim)
```

che restituisce il valore di verità della seguente formula:

$$\exists j \in [1, dim - 1).\ \big((\forall i \in [0, j).\ a[j] < a[i])\ \vee\ (\forall i \in [j + 1, dim).\ a[j] > a[i])\big)$$

ESERCIZIO 5 (punti 5)
Senza utilizzare ricorsione esplicita, definire in CAML una funzione

```
foo : 'a list -> ('a -> bool) -> bool
```

in modo che `(foo lis p)` restituisca `true` se il numero di elementi di `lis` che soddisfano `p` è maggiore del numero di elementi di `lis` che non soddisfano `p`; restituisca `false` altrimenti.

ESERCIZIO 6 (punti 5)
Date le seguenti definizioni:

```
struct el {int info; struct el *next;};
typedef struct el ElementoDiLista;
typedef ElementoDiLista *ListaDiElementi;
```

scrivere in `C` una procedura che, dati in ingresso attraverso opportuni parametri una lista di interi e un intero `x`, elimina dalla lista l'ultimo elemento maggiore di `x`. Se nessun elemento della lista è maggiore di `x`, la procedura elimina, se esiste, l'ultimo elemento della lista.