

PROGRAMMAZIONE 1 e LABORATORIO (A,B)

a.a. 2009-2010
Prova scritta del 11 febbraio 2010

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

ESERCIZIO 1 (punti 5)

Un albero \mathcal{T} si dice *destrorso* se e solo se

- \mathcal{T} è l'albero vuoto, oppure
- la radice di \mathcal{T} è una foglia, oppure
- la radice di \mathcal{T} ha n figli A_1, \dots, A_n , con $n > 0$ dove A_1, \dots, A_{n-1} sono foglie e l'albero con radice A_n è destrorso

Scrivere una grammatica che generi il linguaggio $\mathcal{L} = \{a^n b^m c^2 \mid n > 0, m > 0\}$ (sull'alfabeto $\{a, b, c\}$) in modo che tutti gli alberi di derivazione nella grammatica siano destrorsi.

ESERCIZIO 2 (punti 5)

Dato il tipo degli alberi binari visto a lezione

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

e la definizione di albero destrorso dell'esercizio precedente, si definisca una funzione

```
destrorso : 'a btree -> bool
```

in modo che (`destrorso bt`) restituisca *true* se *bt* è destrorso, *false* altrimenti.

ESERCIZIO 3 (punti 5)

Si definisca in C una funzione

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

che restituisce *true* se vale la seguente proprietà

$$\exists i \in [0, \text{dim} - 1). \text{pari}(a[i]) \wedge (\forall j \in [i + 1, \text{dim}). a[j] > a[i])$$

e restituisce *false* altrimenti (si supponga dato il tipo `typedef enum {false, true} boolean;`).

ESERCIZIO 4 (punti 5)

Senza utilizzare ricorsione esplicita, si definisca in CAML una funzione

```
pam : (('a -> 'b) * 'a) list -> 'b list
```

in modo che, se `pam l1 = l2`, ogni elemento della lista *l2* sia ottenuto dall'elemento (f, x) di *l1* nella medesima posizione, applicando *f* a *x*.

ESERCIZIO 5 (punti 5)

Si definisca in C una procedura che prende come parametri due liste di interi e modifica la prima attaccandole in coda la seconda.

ESERCIZIO 6 (punti 5)

Scrivere in C una procedura che data una lista sposta l'ultimo elemento in prima posizione (la lista deve rimanere inalterata se è vuota o è composta da un solo elemento).

Negli esercizi 5) e 6) si suppongano date le seguenti definizioni:

```
struct el { int info; struct el *next;};
```

```
typedef struct el ElementoLista;
```

```
typedef ElementoLista *ListaDiElementi;
```