

# PROGRAMMAZIONE 1 e LABORATORIO (A,B)

a.a. 2009-2010  
Prova scritta del 14 gennaio 2010

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

## ESERCIZIO 1 (punti 5)

Data una grammatica  $G = \langle \Lambda, V, S, P \rangle$  che genera il linguaggio  $\mathcal{L}_G$  e supponendo  $\downarrow \notin \Lambda$ , si definisca una nuova grammatica  $G' = \langle \Lambda \cup \{\downarrow\}, V', S', P' \rangle$  che genera il seguente linguaggio

$$\mathcal{L}_{G'} = \{ \alpha_1 \downarrow \alpha_2 \downarrow \dots \downarrow \alpha_n \mid n \text{ pari maggiore di } 0 \text{ e } \alpha_i \in \mathcal{L}_G \text{ con } i \in [1, n] \}$$

(N.B. Indicare con precisione tutte le componenti di  $G'$ )

## ESERCIZIO 2 (punti 5)

Dato il tipo degli alberi binari visto a lezione

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

si definisca una funzione `elim` con tipo

```
elim : 'a btree -> 'a -> 'a btree
```

in modo che `(elim bt x)` sia l'albero ottenuto eliminando da `bt` tutte le foglie etichettate con `x`.

## ESERCIZIO 3 (punti 5)

Si definisca in C una procedura

```
void occur (int a[], int dima, int b[], int dimb)
```

che sostituisce ad ogni elemento del vettore `a` (di dimensione `dima`) il numero delle sue occorrenze nel vettore `b` (di dimensione `dimb`).

## ESERCIZIO 4 (punti 5)

Si definisca in C una funzione

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

che restituisce `true` se e solo se il vettore `vet` (di dimensione `dim`) soddisfa la seguente proprietà:

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

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

## ESERCIZIO 5 (punti 5)

Si completi la seguente definizione CAML

```
let ins x l =  
  let f z y = ...  
in  
  foldr f [x] l;;
```

in modo che, nell'ipotesi che `lis` sia una lista di elementi ordinata in modo non decrescente, il risultato di `(ins x lis)` sia la lista ottenuta inserendo `x` in `lis` e mantenendo l'ordinamento.

## ESERCIZIO 6 (punti 5)

Scrivere in C una procedura che cancella da una lista di interi il primo elemento che contiene un valore  $n$  ed è seguito da un elemento che contiene il valore  $2n$ , e che lascia invariata la lista se un tale elemento non esiste.

Si suppongano date le seguenti definizioni:

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