

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

## Prova scritta del 25 giugno 2014

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

### ESERCIZIO 1 (punti 5)

Dato l'alfabeto  $\Lambda = \{0, 1, 2\}$ , si definisca una grammatica che genera il seguente linguaggio

$$\mathcal{L} = \{\alpha \mid \alpha \in \Lambda^+ \wedge \Sigma\alpha \bmod 2 = 0 \wedge \alpha \bmod 10 = 0\}$$

dove  $\Sigma\alpha$  è la somma di tutte le cifre di  $\alpha$ .

Ad esempio,  $121 \notin \mathcal{L}$ ,  $10 \notin \mathcal{L}$  e  $2110 \in \mathcal{L}$ .

### ESERCIZIO 2 (punti 5)

Definire un automa a stati finiti **deterministico** che riconosce le stringhe del linguaggio

$$\mathcal{L} = \{\alpha a a b \mid \alpha \in \Lambda^*\}$$

sull'alfabeto  $\Lambda = \{a, b, c\}$ .

### ESERCIZIO 3 (punti 5)

Data la seguente definizione di tipo

```
type 'a bilist = Tail of 'a * 'a | Cons of 'a * 'a * 'a bilist
```

si definisca in CAML una funzione `check` con tipo

```
check : 'a bilist -> bool
```

che restituisce `true` se tutti gli elementi di tipo `'a` dell'argomento sono distinti tra loro e restituisce `false` altrimenti.

**Suggerimento:** definire una funzione ausiliaria analoga alla funzione `member` sulle liste.

### 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:

$$\exists j \in (0, dim - 1). \#\{k \mid k \in [0, j) \wedge a[k] = a[j]\} = 1 \wedge \#\{k \mid k \in (j, dim) \wedge a[k] = a[j]\} = 1$$

Si ricorda che  $\#A$  indica la cardinalità (ovvero il numero di elementi) dell'insieme  $A$ .

### ESERCIZIO 5 (punti 5)

Senza utilizzare ricorsione esplicita, definire in CAML una funzione

```
subst : ('a * 'a) list -> bool list
```

che, data una lista di coppie, restituisce la lista di booleani nella quale il valore `true` (risp. `false`) compare in posizione  $i$  se e solo se nella stessa posizione della lista originaria compare una coppia di elementi uguali (risp. diversi).

### 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  $x$ , sposta in prima posizione l'ultima occorrenza dell'elemento  $x$ .

**N.B.:** non è consentito utilizzare assegnamenti sui campi `info` degli elementi della lista.