

PROGRAMMAZIONE I (A,B) - a.a. 2016-17

Quinto Appello – 7 Settembre 2017

Esercizio 1

Si scriva una funzione **C** che, dato un array a di dimensione dim_a e un array b di dimensione dim_b , restituisce il valore di verità della seguente formula

$$\forall i \in [0, dim_a). (\forall j \in [0, dim_b). (a[i] = b[j] \implies \#\{k \mid k \in [0, dim_a) \wedge a[i] = a[k]\} = 1)$$

SOLUZIONE Seguendo pedissequamente la formula logica si può costruire una soluzione che prevede tre cicli (uno per ogni quantificatore e uno per #).

Alternativamente, si noti che la prima parte della formula chiede semplicemente che se un elemento di a appare anche in b , allora deve apparire una sola volta in a . Manipolando la formula, è semplice verificare che le due proprietà possono essere verificate separatamente e poi combinate. Ancora, dato che entrambe le proprietà richiedono il conteggio delle occorrenze di un valore, può bastare un'unica funzione ausiliaria

```
int conta(int c[], int dim_c, int val) {
    int tmp=0;
    for (int j=0; j<dim_c; j++)
        if (val==c[j]) tmp++;
    return tmp;
}
```

A questo punto è facile scrivere una funzione che verifica la proprietà richiesta dall'esercizio

```
int check(int a[], int b[], int dim_a, int dim_b) {
    int ok=1;
    int i=0;
    while (i<dim_a && ok) {
        if ((conta(b, dim_b, a[i])!=0) && (conta(a, dim_a, a[i])!=1) ok=0;
        i++;
    }
    return ok;
}
```

Esercizio 2

Data la seguente grammatica sull'alfabeto $\Lambda = \{a, b\}$

$$S \rightarrow AB \mid CA$$

$$A \rightarrow aAb \mid ab$$

$$B \rightarrow bbB \mid \epsilon$$

$$C \rightarrow aaA \mid \epsilon$$

dove ϵ rappresenta la stringa vuota, si dia il linguaggio generato dalla grammatica e si verifichi se tale linguaggio è regolare o meno (fornendo una opportuna dimostrazione).

SOLUZIONE Controllando la struttura delle grammatica, e le produzioni che partono da S , è facile notare che il linguaggio può essere descritto nel modo seguente

$$\mathcal{L} = \{a^n b^n b^{2m} \mid n > 0 \wedge m \geq 0\} \cup \{a^{2p} a^q b^q \mid p \geq 0 \wedge q > 0\}$$

I due linguaggi separatamente sono chiaramente irregolari, e questo potrebbe far pensare che anche \mathcal{L} lo sia. D'altra parte, il testo diceva esplicitamente che il linguaggio era invece regolare, e sulla base di questo suggerimento, non era difficile accorgersi che vale

$$\mathcal{L} = \{a^n b^m \mid n > 0 \wedge m > 0 \wedge n \% 2 = m \% 2\}$$

al quale è immediato associare un automa a stati finiti.

Esercizio 3

Si suppongano predefiniti i tipi

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

Si scriva in **C** una procedura che, presa una lista l , elimina da essa tutti gli elementi il cui elemento successivo in l contiene un valore maggiore.

SOLUZIONE

```
void (ListaDiElementi *l) {
    ListaDiElementi prec = NULL;
    ListaDiElementi curr = *l;
    if (curr!=NULL) {
        while (curr->next != NULL) {
            if (curr->next->info > curr->info) {
                if (prec==NULL) {
                    *l=curr->next;
                    free(curr);
                    curr=*l;
                }
                else {
                    prec->next=curr->next;
                    free(curr);
                    curr=prec->next;
                }
            }
            else {
                prec=curr;
                curr=curr->next;
            }
        }
    }
}
```

Esercizio 4

Si definisca in CAML, senza usare la ricorsione esplicita, una funzione

```
controllasegni : int list -> bool
```

che, data una lista `lis` di interi, verifica che ogni coppia di elementi consecutivi diversi da zero abbiano lo stesso segno.

SOLUZIONE Al fine di poter confrontare due elementi consecutivi nel corso della scansione della lista operata dalla `foldr`, è utile “portarsi dietro”, oltre a un valore booleano che rappresenta la soluzione che stiamo costruendo, anche il precedente valore incontrato (ossia, l’elemento successivo nella lista, visto che la lista viene scandita partendo dalla coda).

Inizialmente, il valore booleano sarà impostato a `true`. Diverterà `false` se si incontra una coppia di valori discordi. Negli altri casi (valori concordi o uguali a zero) non verrà modificato.

```
let controllasegni lis =
  let f x (y,b) =
    if (!b) then (x,b) else
      if (x>0 && y>0) || (x<0 && y<0) then (x,b)
      else if (x=0) || (y=0) then (x,b)
      else (x,false)
  in
  let (s,t) = foldr f (0,true) lis in t;;
```

La stessa soluzione può essere scritta in modo più compatto come segue.

```
let controllasegni lis =
  let f x (y,b) =
    if (x*y>=0) then (x,b) else (x,false)
  in
  let (s,t) = foldr f (0,true) lis in t;;
```