

PROGRAMMAZIONE 1 e LABORATORIO (A,B)

a.a. 2010-2011

Prova scritta del 1 luglio 2011

Soluzioni proposte

ESERCIZIO 1 (punti 5)

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

$$\{\alpha \in \Lambda^+ \mid \|\alpha\|_0 = \|\alpha\|_1\}$$

dove la notazione $\|\alpha\|_x$ indica il numero di occorrenze del simbolo x nella stringa α .

Soluzione

```
S ::= 01 | 10 | 1S0 | 0S1 | SS
```

ESERCIZIO 2 (punti 5)

Un albero binario si dice *perfettamente bilanciato* se e solo se, per ogni sottoalbero bt , la profondità del sottoalbero sinistro di bt è uguale alla profondità del sottoalbero destro di bt .

Dato il tipo degli alberi binari visto a lezione

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

si definisca una funzione

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

in modo che `bilanciato bt` restituisca *true* se bt è perfettamente bilanciato, *false* altrimenti. Sono consentiti la definizione e l'uso di funzioni ausiliarie.

Soluzione

Definiamo dapprima una funzione ricorsiva che, dato un albero binario, restituisce una coppia il cui primo elemento è un booleano, che indica se l'albero è o meno perfettamente bilanciato, ed il secondo un intero che indica la profondità dell'albero stesso nel caso in cui sia perfettamente bilanciato (e non è significativo se la prima componente è *false*).

```
let rec bildepth bt = match bt with
  Void -> (true, 0) |
  Node(_,Void, rt) when rt <> Void -> (false, -1) |
  Node(_,lt, Void) when lt <> Void -> (false, -1) |
  Node(_,lt,rt) when lt <> Void & rt <> Void ->
    let (b1,p1) = bildepth lt in
      let (b2,p2) = bildepth rt in
        if b1 & b2 & p1=p2
          then (true, 1+p1)
          else (false, -1);;
```

La funzione richiesta è allora

```
let bilanciato bt = let (b,_) = bildepth bt in b;;
```


ESERCIZIO 4 (punti 5)

Senza utilizzare ricorsione esplicita, ma solo funzioni di ordine superiore, si definisca in CAML una funzione

```
splitposneg : int list -> (int list * int list)
```

in modo che se `splitposneg lis = (l1, l2)`, `l1` contenga tutti e soli gli elementi non negativi di `lis` e `l2` contenga tutti e soli gli elementi negativi di `lis`.

Soluzione 1

```
let splitposneg lis =  
  let f x (l1, l2) =  
    if x < 0 then (l1, x::l2) else (x::l1, l2)  
  in  
    foldr f ([], []) lis;;
```

Soluzione 2

```
let splitposneg lis =  
  let neg x = (x < 0) in  
    let non_neg x = not (neg x)  
  in  
    filter non_neg lis, filter neg lis;;
```

ESERCIZIO 5 (punti 5)

Scrivere una funzione C che, dato un array a di dimensione dim , restituisce 1 se la sequenza $a[0]a[1] \dots a[n-1]$ appartiene al linguaggio dell'esercizio 1, restituisce 0 altrimenti.

Soluzione

```
void check (int a[], int dim)
{
    int i = 0;
    int contazeri_uni = 0;
    int trovato = 0;

    while (i < dim && !trovato)
    {
        if (a[i] == 0)
            contazeri_uni = contazeri_uni+1;
        else
            if (a[i] == 1)
                contazeri_uni = contazeri_uni-1;
            else
                trovato = 1;
        i = i+1;
    }

    return (!trovato && contazeri==0);
}
```

ESERCIZIO 6 (punti 5)

Si suppongano date le seguenti definizioni:

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

Scrivere in C una procedura che, data una lista, cancella l'ultimo elemento se e solo se contiene un valore uguale al penultimo. La funzione deve lasciare invariata la lista se contiene meno di due elementi.

Soluzione

```
void cancella (ListaDiElementi lis)
{
    if (lis != NULL)
        if (lis -> next != NULL)
            {
                ListaDiElementi prec = lis, corr = lis -> next;
                while (corr -> next != NULL)
                    {
                        prec = prec -> next;
                        corr = corr -> next;
                    }
                if (prec -> info == corr -> info)
                    {
                        prec -> next = NULL;
                        free(corr);
                    }
            }
}
```