

## Esercizio 1)

Il linguaggio non è regolare.

Dimostriamolo con il pumping lemma.

Qualunque sia  $n > 0$  prendiamo la stringa del

linguaggio  $w = a^n c c a^{n+1}$

$$|w| = 2n + 3 > n$$

Tutte le possibili divisioni di  $w$  in  $xyz$  tali che

$|xy| \leq n$  e  $y \neq \epsilon$  sono rappresentate da:

$$x = a^s \quad 0 \leq s < n$$

$$y = a^t \quad 0 < t \leq n - s$$

$$z = a^k c c a^{n+1} \quad k = n - t - s$$

con  $i=2$  la stringa  $xy^2z = a^{n+t} c c a^{n+1} \notin L$   
dato che  $n+t \geq n+1$ .

Una grammatica che genera il linguaggio è

$$S \rightarrow ASA \mid ACA \mid SA$$

$$A \rightarrow a \mid b$$

$$C \rightarrow c c A$$

## Esercizio 2)

martedì 12 febbraio 2019 17:03

```
int member (int el, int a[], int dim)
{
    int i;
    int trovato = 0;
    while (i < dim && !trovato)
        if (el == a[i]) trovato = 1;
        else i++;
    return trovato;
}
```

```
int formule (int a[], int dima, int b[], int dimb)
{
    int i = 0;
    int ok = 1;
    while (i < dima && ok)
        if (a[i] > 0)
            if (!member (a[i], b, dimb)) ok = 0;
            else i++;
        else i++;
    return ok;
}
```

# Esercizio 3)

martedì 12 febbraio 2019 17:09

let init l =

let f x y = if x = 1 then []  
          else x :: y

in foldr f [] l ;;

# Esercizio 4)

martedì 12 febbraio 2019 17:10

Soluzione 1

let rec member e l =

match l with

[] → false

| x :: xs → if e = x then true

else member e xs;;

let rec end l = match l with

[] → []

| x :: xs → if member 1 (x :: xs)

then end xs

else x :: xs;;

# Esercizio 4)

## Soluzione 2

let end l =

let rec enda l a =

match l with

[] → a

| x::xs → if x=1 then enda xs []

else enda xs (a@[x])

in enda l [];;