

PROGRAMMAZIONE I (A,B) - a.a. 2018-19

Primo appello - 17/1/2019

Esercizio 1

Sia dato il seguente linguaggio sull'alfabeto $\Lambda = \{a, b\}$

$$L = \{\alpha\beta \mid \alpha, \beta \in \Lambda^+ \wedge \beta = \alpha^R\}$$

dove α^R è definito nel modo seguente:

$$\begin{aligned} x^R &= x && \text{se } x \in \Lambda \\ (x\alpha)^R &= \alpha^R x && \text{se } x \in \Lambda, \alpha \in \Lambda^+ \end{aligned}$$

Si verifichi se tale linguaggio è regolare o meno e si dia una grammatica che lo genera.

Esercizio 2

Si scriva una funzione **C** che, dato un array a di dimensione dim , restituisca il valore della seguente formula:

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

dove $\#\{\dots\}$ rappresenta il numero degli elementi (cardinalità) dell'insieme.

Esercizio 3

Si definisca in CAML, USANDO FOLDR, una funzione

```
prec : 'a list -> 'a list
```

che, data una lista, restituisce la lista degli elementi che sono preceduti da un elemento minore.

Si precisa che il primo elemento di una lista NON è preceduto da un elemento minore.

Esempio: `prec [3;4;5;2;-1;7;3;8] = [4;5;7;8]`

Esercizio 4

Si definisca in CAML, USANDO la ricorsione esplicita, la funzione

```
prec : 'a list -> 'a list
```

dell'esercizio precedente.