

# Fondamenti dell'Informatica

27 luglio 2016

## Esercizio 1

Sia  $\Sigma$  un alfabeto e sia  $\min: \Sigma^* \rightarrow \Sigma^*$  definita, per ogni  $L \subseteq \Sigma^*$ , da

$$\min(L) = \{ z \in L \mid \forall x \in \Sigma^* : (\exists y \in \Sigma^+ . z = xy) \implies x \notin L \}.$$

Si dimostri formalmente che i linguaggi regolari sono chiusi rispetto all'operazione  $\min$ .

## Esercizio 2

Sia  $L$  il linguaggio generato dalla grammatica minima le cui produzioni sono  $S \rightarrow aAA$ ,  $A \rightarrow aS \mid bS \mid a$ . Si definisca un APND con al più tre stati che accetti  $L$  per stato finale.

### **Esercizio 3**

Si enuncino tre caratterizzazioni degli insiemi ricorsivi e se ne dimostri l'equivalenza.

## Esercizio 4

Si dica, motivando adeguatamente, se il seguente problema è decidibile:

Data una macchina di Turing  $M$  e un input per la stessa, dire se la macchina, nel corso della computazione, sposterà mai la testina a sinistra.

## Esercizio 5

Sia  $\Sigma = \{0, 1\}$  e, per  $b \in \Sigma$ , sia  $n_b(\varepsilon) = 0$  e, per ogni  $a \in \Sigma$  ed ogni  $w \in \Sigma^*$ ,  $n_b(aw) = 1 - |a - b| + n_b(w)$ . Se  $p$  ed  $x$  sono due stringhe, diciamo che  $p$  è un *prefisso proprio* di  $x$  se  $p$  è un prefisso di  $x$  con  $p \neq \varepsilon$  e  $p \neq x$ . Per  $w \in \Sigma^*$  diciamo che  $w$  è *bilanciata* se  $n_0(w) = n_1(w)$ . Si consideri il linguaggio  $L$  di tutte le stringhe bilanciate su  $\Sigma$ :

$$L = \{ w \in \Sigma^* \mid n_0(w) = n_1(w) \}.$$

Sia  $y \in L$  tale che  $y = y_1 y_2 \cdots y_k$  e  $y_1 = y_k$ . Si dimostri che  $y$  ha un prefisso proprio bilanciato.

## Esercizio 6

Si considerino le dichiarazioni

```
z : integer;  
procedure p (x:integer) <corpo>;
```

Si scriva il corpo di p ('<corpo>') in modo che il codice

```
z := 1; p(z); write(z)
```

si comporti in modi tutti diversi nel caso di passaggio dei parametri per valore, per riferimento, per valore-risultato.

## Esercizio 7

Si consideri la seguente dichiarazione in C++:

```
int fact(int x) {  
    if (x == 0)  
        return 1;  
    else return fact(--x) * x;  
}
```

Si risponda alle seguenti domande, motivando adeguatamente la risposta:

1. Quale o quali possono essere i risultati ritornati dalla chiamata `fact(3)`?
2. Cosa cambia se sostituiamo l'espressione `fact(--x) * x` con l'espressione `fact(x--) * x`?
3. Cosa cambia se sostituiamo l'espressione `fact(--x) * x` con l'espressione `x * fact(--x)`?
4. Cosa cambia se sostituiamo il comando `return fact(--x) * x;` con `{int y = x; return fact(--x) * y;}`?