

## Esercizio 1)

Il linguaggio non è regolare.

Lo proviamo usando il "pumping lemma".

Qualunque sia  $m \in \mathbb{N}$  prendiamo la stringa  $w = a b^m c^m \in L$ .

Vediamo che  $|w| = 2m + 1 > m$ .

Consideriamo tutte le possibili:

suddivisioni  $x, y, z$  di  $w$  tali che  $|xy| \leq m$  e  $y \neq \varepsilon$ .

1)

$$\begin{aligned} x &= \varepsilon \\ y &= a b^t \\ z &= b^{m-t} c^m \end{aligned}$$

per  $i=0$  la stringa  $b^{m-t} c^m \notin L$  dato che manca la  $a$  iniziale

2)

$$x = a b^s$$

$$0 \leq s < m - 1$$

$$y = b^t$$

$$0 < t \leq m - 1 - s$$

$$z = b^k c^m$$

$$k = m - t - s$$

$$|x| + |y| + |z| = 1 + t + k = 1 + t + m - t - s = m - s + 1 < m + 1$$

per  $i=0$  la stringa  $a b^{m-t} c^m \notin L$   
dato che  $t > 0 \Rightarrow m-t < m$

Quindi possiamo concludere che il linguaggio non è regolare.

Una grammatica che lo genera è la seguente

$$S \rightarrow AB$$

$$A \rightarrow a | aA$$

$$B \rightarrow bc | bBc$$

## Esercizio 2)

```
int conte (int el, int a[], int dim)
{
    int c = 0;
    int i;
    for (i = 0; i < dim; i++)
        if (a[i] == el) c++;
    return c;
}
```

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

```
return OK;  
}
```

# Esercizio 3)

L'automato a stati finiti che segue riconosce il linguaggio

