Espressioni Regolari

Possiamo vederle come delle espressioni algebriche, ma definiscono dei linguaggi su un certo alfabeto.

Esempio

$$(0 \cup 1) \bigcirc 0^* \quad \text{che equivale a} \quad \{0, 1\} \bigcirc 0^*$$

Definizione

Sia Σ un alfabeto possiamo definire un'espressione regolare ssu Σ (denotata con $\mathrm{re}(\Sigma)$) in modo ricorsivo:

$$\text{caso base} \begin{cases} \emptyset \in \mathrm{re}(\Sigma) \\ \varepsilon \in \mathrm{re}(\Sigma) \\ a \in \mathrm{re}(\Sigma) \quad \text{con} \quad a \in \Sigma \end{cases}$$

$$\text{induzione} \begin{cases} R_1 \cup R_2 \quad \text{se} \quad R_1, R_2 \in \mathrm{re}(\Sigma) \\ R_1 \bigcirc R_2 \quad \text{se} \quad R_1, R_2 \in \mathrm{re}(\Sigma) \\ (R_1)^* \quad \text{se} \quad R_1 \in \mathrm{re}(\Sigma) \end{cases}$$

Ogni espressione regolare ha un solo linguaggio associato:

$$L(r) \quad \text{t.c.} \quad r \in \mathrm{re}(\Sigma)$$

Vediamolo ricorsivamente:

$$\text{caso base} \begin{cases} L(r) = \emptyset \quad \text{se} \quad r = \emptyset \\ L(r) = \varepsilon \quad \text{se} \quad r = \varepsilon \\ L(r) = \{a\} \quad \text{se} \quad r = a \end{cases}$$

$$\text{induzione} \begin{cases} L(r) = L(R_1) \cup L(R_2) \quad \text{se} \quad r = R_1 \cup R_2 \\ L(r) = L(R_1) \bigcirc L(R_2) \quad \text{se} \quad r = R_1 \bigcirc R_2 \\ L(r) = (L(R_1))^* \quad \text{se} \quad r = R_1^* \end{cases}$$

Qualche esempio:

- $0^*10^* = \{w : w \ \text{ contiene esattamente un } 1\}$
- $\Sigma^*1\Sigma^* = \{w : w \ \text{ contiene almeno un } 1\}$
- $\Sigma^*001\Sigma^* = \{w : w \ \text{ contiene } 001 \text{ come sottostringa}\}$
- $(0 \cup 1000)^* = \{w : \text{ogni occorrenza di } 1 \text{ é seguita da } 000\}$

Teorema

Un linguaggio regolare è regolare se e solo se esiste un'espressione regolare che lo descrive:

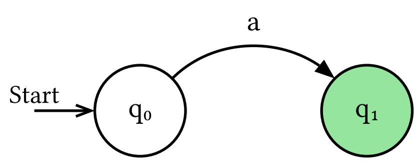
$$\text{REG} \equiv L(\text{re})$$

Per dimostrarlo dobbiamo dimostrare la doppia implicazione $L(\text{re}) \subseteq \text{REG}$ e $\text{REG} \subseteq L(\text{re})$.

Iniziamo dimostrando $L(\text{re}) \subseteq \text{REG}$, quindi data un'espressione regolare r costruiamo un NFA N_r tale che $L(N_r) = L(r)$.

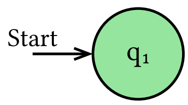
I casi base sono 3, quando l'espressione regolare è un solo carattere, quando l'espression regolare è la stringa vuota oppure quando è l'insieme vuoto.

Data l'espressione regolare $r = a$ con $a \in \Sigma$ costruiamo l'NFA che la riconosce:



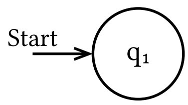
Dove:
- $N_r = (\{q_0, q_1\}, \Sigma, \delta, q_0, \{q_1\})$
- $\delta(q_0, a) = q_1$
- $\delta(q, b) = \emptyset$ con $q \neq q_1 \wedge b \neq a$

Se invece abbiamo $r = \varepsilon$, possiamo costrutire:



Definiamo:
- $N_r = (\{q_1\}, \Sigma, \delta, q_1, \{q_1\})$
- $\delta(q_1, b) = \emptyset, \forall b \in \Sigma$

Se invece $r = \emptyset$ costruiamo:



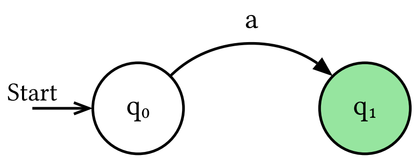
Definiamo:
- $N_r = (\{q_1\}, \Sigma, \delta, q_1, \emptyset)$
- $\delta(q_1, b) = \emptyset, \forall b \in \Sigma$

Adesso per il caso induttivo dobbiamo considerare un'espressione regolare $r = R_1 \cup R_2$ e per ipotesi induttiva sappiamo che $\exists M_1, M_2$ t.c. $L(M_1) = L(R_1) \wedge L(M_2) = L(R_2)$ e per chiusura di REG sappiamo che questo implica che $\exists M \quad \text{t.c.} \quad L(M) = L(R_1) \cup L(R_2)$. Questo è analogo per la concatenazione e l'operatore star.

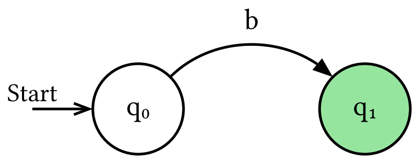
Esempio

Vogliamo costruire un NFA che riconosce il linguaggio $(ab \cup a)^*$

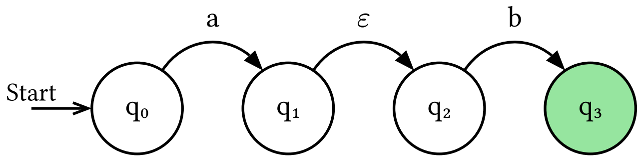
- Possiamo costruire prima l'automa che riconosce a:



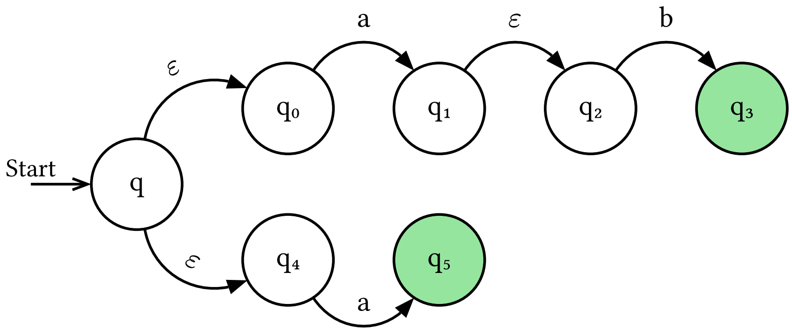
- Poi costruiamo l'automa che riconosce b:



- Adesso possiamo ricavare quello che riconosce ab:



- Costruiamo l'automa che riconosce $(ab \cup a)$:



- Infine costruiamo l'NFA che riconosce il linguaggio dell'espressione regolare: