

Espressioni Regolari

Dato un alfabeto Σ , si definisce espressione regolare di Σ una stringa R rappresentante un linguaggio $L(R) \subseteq \Sigma^*$. Quindi ogni espressione regolare R è in verità il linguaggio $L(R)$ ad essa associata.

Le **espressioni regolari (regex)** non sono solo delle stringhe con simboli speciali, ma in **teoria della computazione** sono una notazione compatta per descrivere un **linguaggio formale**, cioè un insieme di stringhe costruite su un alfabeto Σ .

Ogni espressione regolare R corrisponde ad un linguaggio $L(R)$.

Costruzione delle espressioni regolari

L'insieme delle espressioni regolari su un alfabeto Σ , indicato con $re(\Sigma)$, si definisce induttivamente:

Casi base

- $\emptyset \in re(\Sigma)$
 - Linguaggio vuoto: $L(\emptyset) = \{\}$
- $\varepsilon \in re(\Sigma)$
 - Linguaggio contenente solo la stringa vuota: $L(\varepsilon) = \varepsilon$
- $a \in re(\Sigma)$ con $a \in \Sigma$
 - Linguaggio con la stringa " a ": $L(a) = a$

Costruzioni induttive (operazioni sui linguaggi)

Se $R1, R2 \in re(\Sigma)$, allora:

- **Unione:** $R1 \cup R2 \in re(\Sigma)$
 - Linguaggio: $L(R1 \cup R2) = L(R1) \cup L(R2)$ e significa "**o R1 o R2**"
- **Concatenazione:** $R1 \circ R2 \in re(\Sigma)$
 - Linguaggio: $L(R1 \circ R2) = xy \mid x \in L(R1), y \in L(R2)$ e significa "**prima R1, poi R2**"
Se $R \in re(\Sigma)$, allora:
- **Star (Kleene star):** $R^* \in re(\Sigma)$
 - Linguaggio: $L(R^*) = \varepsilon, w, ww, www, \dots \mid w \in L(R)$ e significa "**zero o più ripetizioni di R**"
- **Plus (Kleene plus):** $R^+ \in re(\Sigma)$
 - Linguaggio: $L(R^+) = L(R) \circ L(R^*)$ e significa "**una o più ripetizioni di R**"

Classi dei linguaggi descritti da esp. reg.

Dato un alfabeto Σ , definiamo come classe dei linguaggi di Σ descritti da un'espressione regolare il seguente insieme:

$$L(re) = L \subseteq \Sigma^* \mid \exists R \in re(\Sigma) \text{ t. c. } L = L(R)$$