Automi

Mattia Patetta

22 ottobre 2025

Indice

1 Linguaggi Regolari

1.1 DFA

1.2 Linguaggio Accettato

1.3 Linguaggi Regolari

1.4 Operazioni sui linguaggi

1.5 Non Determinismo

1.6 NFA

1.7 Teorema: REG = $\mathcal{L}$(NFA)

1.8 Espressioni Regolari

1.9 Pumping Lemma

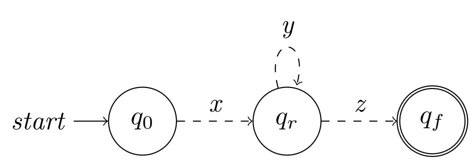
Per provare la non regolarità di un linguaggio, usiamo un teorema chiamato **pumping lemma**, che mostra una proprietà che solo i linguaggi regolari possiedono.
Questa proprietà afferma che, per ogni parola sufficientemente lunga del linguaggio, è possibile ripetere una certa parte della parola in modo che la nuova stringa ottenuta appartenga ancora al linguaggio.

Teorema 1.1. *Se L è regolare, allora esiste un valore $p \in \mathbb{N}$ (lunghezza del pumping) t.c., presa una stringa $w \in L$ con $|w| \geq p$, allora w può essere scomposta in $w = xyz$ (tre sottostringhe), in modo tale che:*

i) $\forall\ i \geq 0,\ xy^i z \in L$

ii) $|y| > 0$

iii) $|xy| \leq p$



Dimostrazione. Sia M (Q, Σ, δ, q_1, F) t.c. $L(M) = L$ e sia $p = |Q|$. Consideriamo $w = w_1, w_2, \ldots, w_n$ t.c. $n \geq p$.
Sia $r_1, r_2, \ldots, r_{n+1}$ la sequenza di stati attraversati da M su input w (in cui $r_1 = q_1$ e $r_{n+1} \in F$).
In altre parole:

$$\delta(r_i, w_i) = r_{i+1}, \text{ con } i = 1, 2, \ldots, n$$

Pigeonhole Principle

Poiché $|w| \geq p$, l'automa visita $|w| + 1$ stati. Avendo solo p stati disponibili, per il principio dei cassetti almeno uno stato deve essere visitato due volte. Questo ci assicura l'esistenza di un ciclo nell'automa.

Per il principio appena enunciato, nella sequenza considerata c'è almeno uno stato che si ripete.
Scompongo la stringa $w = xyz$ e pongo:

1. $x = w_1, \ldots, w_{i-1}$
2. $y = w_i, \ldots, w_{l-1}$
3. $z = w_l, \ldots, w_n$

Siccome x porta M da $r_1 = q_1$ ad r_i, y porta M da $r_i = r_l$ a $r_{n+1} \in F$, allora:

$$\forall i \geq 0, xy^i z \in L(M)$$

Siccome $i \neq l$, allora $|y| > 0$.
Infine $l \leq p + 1$, allora $|xy| = l - 1 \leq p$

$\square$

2 Linguaggi Non Regolari

Le grammatiche context-free sono un metodo più potente per descrivere i linguaggi e sono caratterizzati dalla possibilità di descrivere alcuni aspetti tramite una struttura ricorsiva.
I linguaggi associati a grammatiche context-free si chiamano linguaggi context-free. La classe dei linguaggi context-free include tutti i linguaggi regolari e molti altri.

2.1 Grammatiche Acontestuali

Prendiamo per esempio una grammatica e chiamiamola G_1

$$\begin{cases} A \to 0A1 \quad (R_1) \\ A \to B \quad (R_2) \\ B \to \# \quad (R_3) \end{cases}$$

Una grammatica consiste di un insieme di regole di sostituzione, dette **produzioni**. Ogni regola è formata da un simbolo e una stringa separati da una freccia:

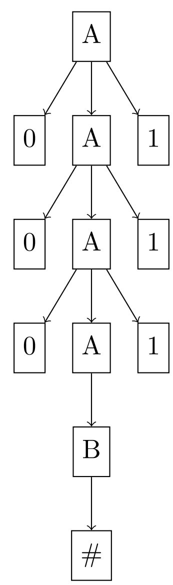
1. I simboli sono chiamati **variabili**
2. La stringa è formata da variabili e altri simboli chiamati **terminali**

Una variabile particolare è chiamata **variabile iniziale** e generalmente si trova sul lato sinistro in alto.
Una grammatica può essere usata per descrivere un linguaggio generando ogni stringa del linguaggio così:

1. Scrivo la variabile iniziale
2. Sostituisco le variabili con altre espressioni seguendo le regole della grammatica
3. Ripeto fino a che non ci sono più variabili

esempio:

$$A \Rightarrow 0A1 \Rightarrow 00A11 \Rightarrow 000A111 \Rightarrow 000B111 \Rightarrow 000\#111$$



Rappresentazione tramite albero sintattico

Tutte le stringhe generate in questo modo generano il linguaggio della grammatica che chiameremo $L(G_1)$. Ogni linguaggio che può essere generato da una grammatica context-free è chiamato linguaggio context-free (CFL). Una sequenza di sostituzioni per ottenere una stringa è chiamata una **derivazione**.

2.2 Definizione Formale di Grammatica Context-Free

CFG (Context-Free Grammar)

Una grammatica context-free è una quadrupla (V, Σ, R, S), dove:

1. V è un insieme finito i cui elementi sono chiamati **variabili**
2. Σ è un insieme finito, disgiunto da V, i cui elementi sono chiamati **terminali**
3. R è un insieme finito di **regole**, dove ciascuna regola è una variabile e una stringa di variabili e terminali
4. S è la variabile iniziale

Prendiamo due variabili u e v e diciamo che u deriva v, denotandolo con $u \overset{*}{\Rightarrow} v$ se $u = v$ o se esiste una sequenza $u_1, u_2, \ldots, u_k$ con $k \geq 0$ e

$$u \Rightarrow u_1 \Rightarrow \ldots \Rightarrow u_k = v.$$

2.3 Unione di Grammatiche

Supponiamo di avere $G_i = (V_i, \Sigma_i, R_i, S_i)$ e che siano tutte CFG $\forall i \in \mathbb{N}$ e vogliamo delle CFG $G = (V, \Sigma, R, S)$ t.c. $L(G) = \bigcup_{i=1}^{n} L(G_i)$.

Idea Naturale

- $V = \bigcup_{i}^{n} V_i \cup \{S\}$ senza perdita di generalità. Possiamo assumere $V_i \cap V_j = \emptyset, \forall i, j \in \mathbb{N} : i \neq j$
- S nuova variabile iniziale
- $\Sigma = \bigcup_i \Sigma_i$
- $R = \bigcup_i R_i \cup \{S \to S_1 | \ldots | S_k\}$

Proprietà 1. *Correttezza:* $\bigcup_i L(G_i) = L(G)$

Dimostrazione. Dimostro prima $\bigcup_i L(G_i) \subseteq L(G)$. Sia $w \in \bigcup_i L(G_i) : \exists j \in \mathbb{N}$ t.c. $w \in L(G_i)$, ovvero:

$$S_i \overset{*}{\Rightarrow}_{G_i} w$$

Ma allora per definizione:

$$S \Rightarrow S_i \overset{*}{\Rightarrow}_{G_i} w$$

Ovvero, $w \in L(G)$
Adesso dimostriamo l'altra direzione: $L(G) \subseteq \bigcup_i L(G_i)$. Sia $w \in L(G)$, ovvero: $S \overset{*}{\Rightarrow}_G w$ e per definizione $\exists i \in \mathbb{N}$ t.c.

$$S \Rightarrow S_i \overset{*}{\Rightarrow}_G w$$

Siccome V_i è disgiunto da tutte le altre variabili:

$$S \Rightarrow S_i \overset{*}{\Rightarrow}_{G_i} w$$

Quindi:

$$w \in L(G_i) \subseteq \bigcup L(G_i)$$

$\square$

Proprietà 2. *Da DFA a CFG*

Proprietà 3. *Ricorsione*

Definizione 2.1. *Forma Normale di Chomsky*

Teorema 2.1. *Ogni CFG ammette una CFG equivalente in forma normale*