

GRAMMATICHE A CONTESTUALI

Context-free grammars (CFG). Un modo diverso di definire linguaggi. Vediamo:

- Più espressivo, consente di rappresentare linguaggi non regolari ($\{0^n 1^n : n \geq 0\}$).
- Le CFG sono equivalenti ed autonome e stanno proprio "evoluti" del DFA/NFA. Si chiama PDA (push-down automaton) e avrà una PILA.
- Non tutti i linguaggi sono contestuali; ci sono un PUMPING LEMMA anche per le CFG.

Prima di dare le definizioni, vediamo un esempio di grammatica.

$$A \rightarrow 0A1$$

$$A \rightarrow B$$

$$B \rightarrow \#$$

variabile
non terminale.

Ingredienti: Variabili (A, B), regole ($A \rightarrow B$), simboli terminali ($0, 1, \#$).

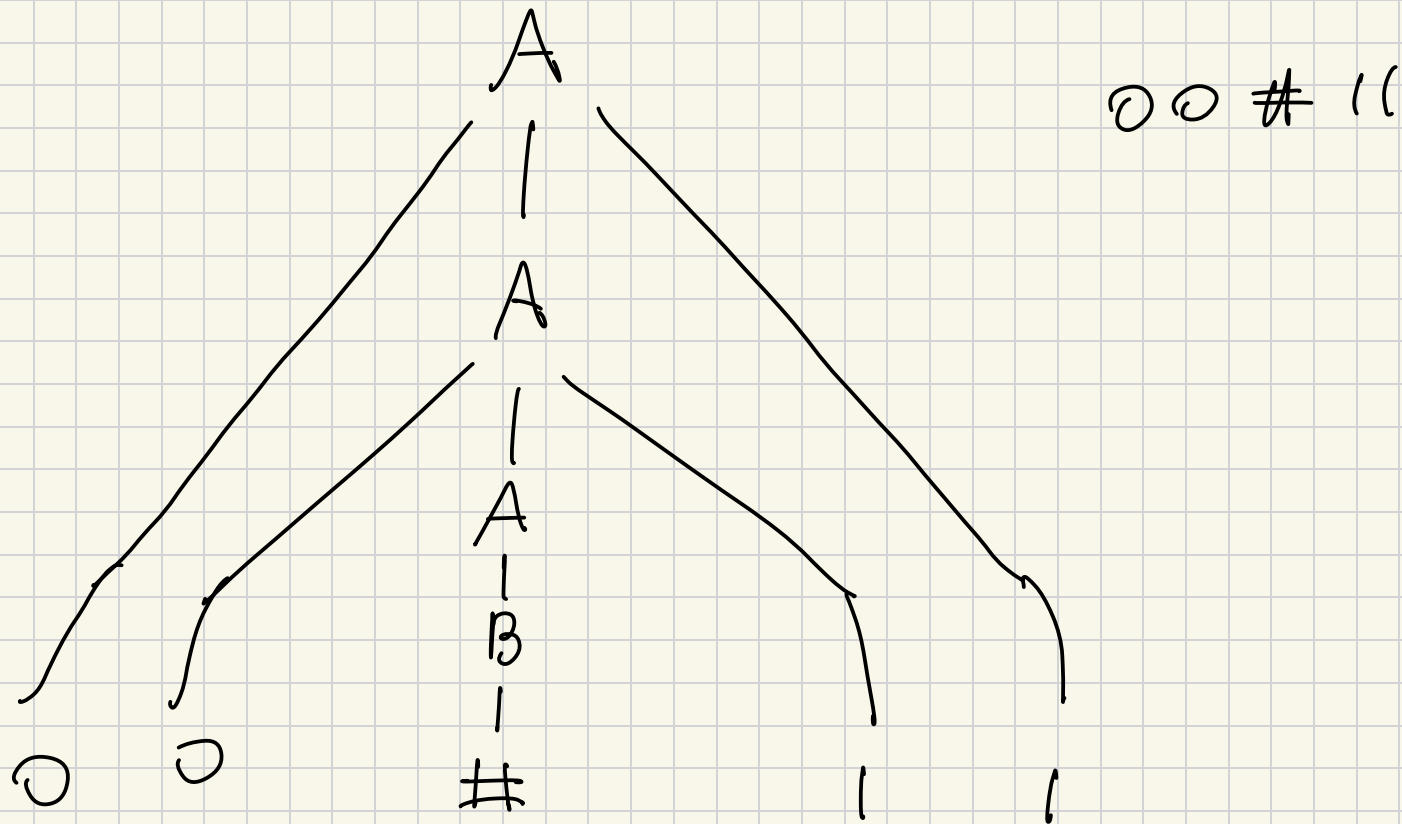
Per generare stringhe: Parto da A e applico le regole in un ordine qualsiasi, fino a che non ottengo solo terminali.

Ad esempio: la CFG di sopra genera $000\#111$.

$A \Rightarrow 0 A 1 \Rightarrow 00 A 11 \Rightarrow 000 A 111$

$\Rightarrow 000 B 111 \Rightarrow 000 \# 111$

Rappresentazione visuale: Albero di derivazione.



Applicazione principale: parser computer del
linguaggio di programmazione.

Esempio:

$$E \rightarrow E + E$$

$$E \rightarrow E \times E$$

$$E \rightarrow (E)$$

$$E \rightarrow 0 \mid 1 \mid 2 \mid \dots \mid 9$$

Questa CFG genera $(3+4) \times 4$, ma non genera

$$3 + 4 \times$$

$$\begin{aligned}
 E &\Rightarrow E \times E \Rightarrow (E) \times E \Rightarrow (E + E) \times E \\
 &\Rightarrow (E + E) \times 4 \Rightarrow (3 + E) \times 4 \\
 &\Rightarrow (3 + 4) \times 4
 \end{aligned}$$

$$\Sigma = \{ 0, \dots, 9, x, +, (,) \}$$

DEF Une grammaire $\bar{G}$ une tuple (V, Σ, R, S)
 où :

- V ensemble FINI TO de VARIABLES
- Σ ensemble II de TERMINAUX
 $(V \cap \Sigma = \emptyset)$
- S VARIABLE INITIALE

- R insieme FINITO di regole

$$A \rightarrow w; A \in V$$

w insieme di
vowel/wh/
Terminal

Definiamo il linguaggio associato a CFG.

Se $u, v, w \in \Sigma^* V$ e R contiene

$$A \rightarrow w$$

allora diremo che $u A v \Rightarrow u w v$
() produce

Per μ in generale $\mu \xRightarrow{*} \nu$ se:
 $\hookrightarrow$ oppure

— $\mu = \nu$

— oppure esistono $\mu_1, \dots, \mu_k$ $k > 0$

$$\mu \Rightarrow \mu_1, \mu_1 \Rightarrow \mu_2, \dots, \mu_k \Rightarrow \nu$$

Questo w porta a dire: Il linguaggio μ è
sotto e una CFG $= (V, \Sigma, R, S)$

$$L = \{ w \in \Sigma^* : S \xRightarrow{*} w \}$$

Vogliamo dimostrare che riguardano le CFG.

1) Si possono unire le CFG. Anche nel
caso di grammatiche complesse.

$$G_i = (V_i, \Sigma_i, R_i, S_i) \quad i = 1 \dots K$$

vogliamo G f.c. $L(G) = \bigcup_i L(G_i)$

$$G = (\bigcup_i V_i \cup \{S\}, \bigcup_i \Sigma_i, R, S)$$

$$R = \bigcup_i R_i \cup \{ \pi : S \rightarrow S_1 | S_2 | \dots | S_k \}$$

Non è difficile dimostrare: $L(G) = \bigcup_i L(G_i)$

Imponiamo: Se $w \in \bigcup_{j \in I} L(G_j)$ allora
 $w \in L(G_i)$ per $i \in I$. $\curvearrowright w \in L(G)$

Overo: $S_j \stackrel{*}{\Rightarrow} w$. Ma allora
 $S \Rightarrow S_j \stackrel{*}{\Rightarrow} w$ e quindi $S \stackrel{*}{\Rightarrow} w$.

D'altra parte, supponiamo $w \in L(G)$:

$$S \stackrel{*}{\Rightarrow} w$$

Per definizione di R , $\exists i$ t.c.

$$S \Rightarrow S_i \stackrel{*}{\Rightarrow} w \quad \text{overo}$$

$$S \Rightarrow S_i \Rightarrow \mu_1 \Rightarrow \mu_2 \Rightarrow \dots \Rightarrow \mu_m \Rightarrow w$$

L'unico teorema che in generale non è garantito che tutte le regole associate a queste produzioni siano regole di della CFG G ;

Esempio: $L = \{0^m 1^m : m \geq 0\} \cup \{1^m 0^m : m \geq 0\}$

$$S_1 \rightarrow 0 S_1 1 \mid \varepsilon$$

$$L(G_1) = L_1$$

$$S_2 \rightarrow 1 S_2 0 \mid \varepsilon$$

$$L(G_2) = L_2$$

$$S \rightarrow S_1 \mid S_2$$

$$L(G) = L_1 \cup L_2$$

$$S_1 \rightarrow 0 S_1 1 \mid \varepsilon$$

$$S_2 \rightarrow 1 S_2 0 \mid \varepsilon$$

$$S \rightarrow T$$

Non andare bene.

$$T \rightarrow 0 T 1 \mid \epsilon$$

$$T \rightarrow 1 T 0 \mid \epsilon$$

2) Ovviamente, le CFL possono fare tutto quello che fanno i DFA. Ovvero che un DFA non è sufficiente definire CFL equivalenti:

- Metto V_i variabile per ogni stato, con $\delta = V_0$ che corrisponde allo stato iniziale q_{start} .

- Se $q_i \in F$, allora aggiungo $V_i \rightarrow \epsilon$.

- Se vale $\delta(q_i, a) = q_j$, aggiungo la
regola $V_i \rightarrow a V_j$

Si può dimostrare formalmente che queste $\bar{\Gamma}$
equivalenti al dato linguaggio regolare.