

Grammatiche e Linguaggi Liberi da Contesto

- Abbiamo visto che molti linguaggi non sono regolari. Consideriamo allora classi più grandi di linguaggi.
- I **Linguaggi Liberi da Contesto** (CFL) sono stati usati nello studio dei linguaggi naturali dal 1950, e nello studio dei compilatori dal 1960.
- Le **grammatiche libere da contesto** (CFG) sono la base della sintassi BNF (Backus-Naur-Form).
- Oggi i CFL sono importanti per XML.
- Studieremo: CFG, i linguaggi che generano e gli alberi sintattici.

Esempio informale di CFG

- Consideriamo $L_{pal} = \{w \in \Sigma^* : w = w^R\}$
- Per esempio: otto $\in L_{pal}$, madamimadam $\in L_{pal}$.
- Sia $\Sigma = \{0, 1\}$ e supponiamo che L_{pal} sia regolare.
- Sia n dato dal pumping lemma. Allora $0^n 1 0^n \in L_{pal}$. Nel leggere 0^n il FA deve passare per un loop. Se omettiamo il loop, contraddizione.
- Definiamo L_{pal} induttivamente:
 - **Base:** ϵ , 0, e 1 sono palindromi.
 - **Induzione:** Se w è palindroma, anche $0w0$ e $1w1$ lo sono.
 - Nessun'altra stringa è palindroma.

Le CFG sono un modo formale per definire linguaggi come L_{pal} .

1. $P \rightarrow \epsilon$
2. $P \rightarrow 0$
3. $P \rightarrow 1$
4. $P \rightarrow 0P0$
5. $P \rightarrow 1P1$

- 0 e 1 sono *terminali*
- P è una *variabile* (o *nonterminale*, o *categoria sintattica*)
- P è in questa gramatica anche il *simbolo iniziale*.
- 1–5 sono *produzioni* (o *regole*)

Definizione formale di CFG

Una *grammatica libera da contesto* è una quadrupla

$$G = (V, T, P, S)$$

dove

- V è un insieme finito di *variabili* o [*simboli*] *non terminali* o *categorie sintattiche*. (Rappresentano insiemi di stringhe)
- T è un insieme finito di [*simboli*] *terminali*.
- P è un insieme finito di *produzioni* (regole ricorsive che definiscono le variabili) della forma $A \rightarrow \alpha$, dove A è una variabile, la *testa della produzione*, e $\alpha \in (V \cup T)^*$ è il *corpo della produzione*. Possono esserci regole alternative per la stessa variabile.
- S è una variabile distinta chiamata il *simbolo iniziale*.

Esempi

- $G_{pal} = (\{P\}, \{0, 1\}, A, P)$, dove $A = \{P \rightarrow \epsilon, P \rightarrow 0, P \rightarrow 1, P \rightarrow 0P0, P \rightarrow 1P1\}$.
- A volte raggruppiamo le produzioni con la stessa testa:
 $A = \{P \rightarrow \epsilon | 0 | 1 | 0P0 | 1P1\}$.
- Le espressioni regolari su $\{0, 1\}$ possono essere definite dalla grammatica

$$G_{regex} = (\{E\}, \{0, 1\}, A, E)$$

dove A corrisponde a

$$\{E \rightarrow \mathbf{0}, E \rightarrow \mathbf{1}, E \rightarrow E.E, E \rightarrow E + E, E \rightarrow E^*, E \rightarrow (E)\}$$

Esempio

Espressioni (semplici) in un tipico linguaggio di programmazione.

Gli operatori sono $+$ e $*$, e gli operandi sono identificatori, cioè stringhe in $L((\mathbf{a} + \mathbf{b})(\mathbf{a} + \mathbf{b} + \mathbf{0} + \mathbf{1})^*)$

Usiamo la grammatica $G = (\{E, I\}, T, P, E)$ dove

$T = \{+, *, (,), a, b, 0, 1\}$ e P è il seguente insieme di produzioni:

1. $E \rightarrow I$
2. $E \rightarrow E + E$
3. $E \rightarrow E * E$
4. $E \rightarrow (E)$
5. $I \rightarrow a$
6. $I \rightarrow b$
7. $I \rightarrow Ia$
8. $I \rightarrow Ib$
9. $I \rightarrow I0$

Derivazioni usando le grammatiche

- *Inferenza ricorsiva*, usando le produzioni dal corpo alla testa
- *Derivazioni*, usando le produzioni dalla testa al corpo

Esempio di inferenza ricorsiva:

	Stringa	Ling.	Prod.	Stringhe usate
(i)	a	I	$5.I \rightarrow a$	-
(ii)	b	I	$6.I \rightarrow b$	-
(iii)	$b0$	I	$9.I \rightarrow I0$	(ii)
(iv)	$b00$	I	$9.I \rightarrow I0$	(iii)
(v)	a	E	$1.E \rightarrow I$	(i)
(vi)	$b00$	E	$1.E \rightarrow I$	(iv)
(vii)	$a + b00$	E	$2.E \rightarrow E + E$	(v), (vi)
(viii)	$(a + b00)$	E	$4.E \rightarrow (E)$	(vii)
(ix)	$a * (a + b00)$	E	$3.E \rightarrow E * E$	(v), (viii)

Derivazioni

- Sia $G = (V, T, P, S)$ una CFG, $A \in V$, $\{\alpha, \beta\} \subset (V \cup T)^*$, e $A \rightarrow \gamma \in P$.

- Allora scriviamo

$$\alpha A \beta \xRightarrow{G} \alpha \gamma \beta$$

o, se è ovvia la G ,

$$\alpha A \beta \Rightarrow \alpha \gamma \beta$$

e diciamo che da $\alpha A \beta$ **si deriva** $\alpha \gamma \beta$.

- Definiamo $\xRightarrow{*}$ la chiusura riflessiva e transitiva di \Rightarrow , cioè':
 - **Base:** Sia $\alpha \in (V \cup T)^*$. Allora $\alpha \xRightarrow{*} \alpha$.
 - **Induzione:** Se $\alpha \xRightarrow{*} \beta$, e $\beta \Rightarrow \gamma$, allora $\alpha \xRightarrow{*} \gamma$.

Esempio

Derivazione di $a * (a + b00)$ da E nella grammatica delle espressioni:

$$\begin{aligned}
 E &\Rightarrow E * E \Rightarrow I * E \Rightarrow a * E \Rightarrow a * (E) \Rightarrow \\
 &a * (E + E) \Rightarrow a * (I + E) \Rightarrow a * (a + E) \Rightarrow a * (a + I) \Rightarrow \\
 &a * (a + I0) \Rightarrow a * (a + I00) \Rightarrow a * (a + b00)
 \end{aligned}$$

- Ad ogni passo potremmo avere varie regole tra cui scegliere, ad esempio
 $I * E \Rightarrow a * E \Rightarrow a * (E)$, oppure
 $I * E \Rightarrow I * (E) \Rightarrow a * (E)$.
- Non tutte le scelte portano a derivazioni di una particolare stringa, per esempio

$$E \Rightarrow E + E$$

non ci fa derivare $a * (a + b00)$.

Derivazioni a sinistra e a destra

- **Derivazione a sinistra** \Rightarrow_{lm} : rimpiazza sempre la variabile più e a sinistra con il corpo di una delle sue regole.
- **Derivazione a destra** \Rightarrow_{rm} : rimpiazza sempre la variabile più a destra con il corpo di una delle sue regole.
- Der. a sinistra: quella del lucido precedente.
- A destra:

$$\begin{aligned}
 & E \xRightarrow{rm} E * E \xRightarrow{rm} \\
 E * (E) & \xRightarrow{rm} E * (E + E) \xRightarrow{rm} E * (E + I) \xRightarrow{rm} E * (E + I0) \\
 & \xRightarrow{rm} E * (E + I00) \xRightarrow{rm} E * (E + b00) \xRightarrow{rm} E * (I + b00) \\
 & \xRightarrow{rm} E * (a + b00) \xRightarrow{rm} I * (a + b00) \xRightarrow{rm} a * (a + b00)
 \end{aligned}$$

Possiamo concludere che $E \xRightarrow{rm}^* a * (a + b00)$

Il linguaggio di una grammatica

- Se $G(V, T, P, S)$ è una CFG, allora il **linguaggio di G** è

$$L(G) = \{w \in T^* : S \xRightarrow[G]{*} w\}$$

cioè l'insieme delle stringhe su T^* derivabili dal simbolo iniziale.

- Se G è una CFG, chiameremo $L(G)$ un **linguaggio libero da contesto**.
- Esempio: $L(G_{pal})$ è un linguaggio libero da contesto.
- Il linguaggio è visto come l'insieme delle stringhe generate dalla grammatica (approccio *generativo-sintetico*), mentre finora come l'abbiamo visto come l'insieme delle stringhe riconosciute o accettate dagli automi (approccio *riconoscitivo-analitico*).

Teorema 5.7:

$$L(G_{pal}) = \{w \in \{0, 1\}^* : w = w^R\}$$

Dimostrazione: (direzione \supseteq) Supponiamo $w = w^R$. Mostriamo per induzione su $|w|$ che $w \in L(G_{pal})$.

- **Base:** $|w| = 0$, o $|w| = 1$. Allora w è ϵ , 0 , o 1 . Dato che $P \rightarrow \epsilon$, $P \rightarrow 0$, e $P \rightarrow 1$ sono produzioni, concludiamo che $P \xRightarrow[G]{*} w$ in tutti i casi base.

- **Induzione:** Supponiamo $|w| \geq 2$. Dato che $w = w^R$, abbiamo $w = 0x0$, o $w = 1x1$, e $x = x^R$.
 Se $w = 0x0$ sappiamo che per l'ipotesi induttiva $P \xRightarrow{*} x$.
 Allora

$$P \Rightarrow 0P0 \xRightarrow{*} 0x0 = w$$

Quindi $w \in L(G_{pal})$.

Il caso di $w = 1x1$ è simile.

(direzione \subseteq) Supponiamo che $w \in L(G_{pal})$ e dobbiamo mostrare che $w = w^R$.

Dato che $w \in L(G_{pal})$, abbiamo $P \xRightarrow{*} w$.

Faremo un'induzione sulla lunghezza di $\xRightarrow{*}$.

- **Base:** La derivazione $P \xRightarrow{*} w$ ha 1 passo.
Allora w deve essere ϵ , 0, o 1, tutte palindromi.
- **Induzione:** Sia $n \geq 1$, e supponiamo che la derivazione abbia $n + 1$ passi e che l'enunciato sia vero per tutte le derivazioni di n passi (se $P \xRightarrow{*} x$ in n passi, allora $x = x^R$). Allora una derivazione di $n + 1$ passi deve essere del tipo

$$w = 0x0 \xleftarrow{*} 0P0 \rightarrow P \text{ oppure } w = 1x1 \xleftarrow{*} 1P1 \rightarrow P$$

dove la seconda derivazione di x ha n passi. Infatti $n + 1 > 1$ e le produzioni $P \leftarrow 0P0$ e $P \leftarrow 1P1$ sono le uniche che permettono passi aggiuntivi.

Per l'ipotesi induttiva, x è palindroma. Lo sarà quindi anche la stringa w .

Forme sentenziali

- Sia $G = (V, T, P, S)$ una CFG, e $\alpha \in (V \cup T)^*$.
- Se $S \xRightarrow{*} \alpha$ diciamo che α è una **forma sentenziale** (*sentential form*).
- Se $S \xRightarrow{lm} \alpha$ diciamo che α è una **forma sentenziale sinistra**,
- Se $S \xRightarrow{rm} \alpha$ diciamo che α è una **forma sentenziale destra**
- Nota: $L(G)$ contiene le forme sentenziali (o *sentence*) che sono in T^* .

Esempi

- Prendiamo la G delle espressioni. Allora $E * (I + E)$ è una forma sentenziale perché

$$E \Rightarrow E * E \Rightarrow E * (E) \Rightarrow E * (E + E) \Rightarrow E * (I + E)$$

Questa derivazione non è né a sinistra né a destra

- $a * E$ è una forma sentenziale sinistra, perché

$$E \underset{lm}{\Rightarrow} E * E \underset{lm}{\Rightarrow} I * E \underset{lm}{\Rightarrow} a * E$$

- $E * (E + E)$ è una forma sentenziale destra, perché

$$E \underset{rm}{\Rightarrow} E * E \underset{rm}{\Rightarrow} E * (E) \underset{rm}{\Rightarrow} E * (E + E)$$