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.

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 $0^n10^n \in L_{pal}$. Nel leggere 0^n il FA deve passare per un loop. Se omettiamo il loop, contraddizione.
- Definiamo L_{pal} induttivamente:

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 $0^n10^n \in L_{pal}$. Nel leggere 0^n il FA deve passare per un loop. Se omettiamo il loop, contraddizione.
- Definiamo L_{pal} induttivamente:
  - **Base:** $\epsilon, 0$, 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. $P \to \epsilon$
2. $P \to 0$
3. $P \to 1$
4. $P \to 0P0$
5. $P \to 1P1$

- 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 definiscone le variabili) della forma $A \to \alpha$, 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 $A = \{P \to \epsilon, P \to 0, P \to 1, P \to 0P0, P \to 1P1\}$.
- A volte raggruppiamo le produzioni con la stessa testa: $A = \{P \to \epsilon | 0 | 1 | 0P0 | 1P1\}$.
- Le espressioni regolari su $\{0,1\}$ possono essere definite dalla grammatica

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

dove A corrisponde a

$$\{E \to \mathbf{0}, E \to \mathbf{1}, E \to E.E, E \to E + E, E \to E^{\star}, E \to (E)\}$$

Esempio

Espressioni (semplici) in un tipico linguaggio di programmazione.
Gli operatori sono + e *, e gli operandi sono identificatori, cioè stringhe in $L((\boldsymbol{a}+\boldsymbol{b})(\boldsymbol{a}+\boldsymbol{b}+\boldsymbol{0}+\boldsymbol{1})^*)$
Usiamo la grammatica $G = (\{E, I\}, T, P, E)$ dove
$T = \{+, *, (,), a, b, 0, 1\}$ e P è il seguente insieme di produzioni:

1. $E \to I$
2. $E \to E + E$
3. $E \to E * E$
4. $E \to (E)$
5. $I \to a$
6. $I \to b$
7. $I \to Ia$
8. $I \to Ib$
9. $I \to I0$
10. $I \to I1$

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

Derivazioni

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

$$\alpha A \beta \underset{G}{\Rightarrow} \alpha \gamma \beta$$

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

Esempio

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

$$E \Rightarrow E*E \Rightarrow I*E \Rightarrow a*E \Rightarrow a*(E) \Rightarrow$$
$$a*(E+E) \Rightarrow a*(I+E) \Rightarrow a*(a+E) \Rightarrow a*(a+I) \Rightarrow$$
$$a*(a+I0) \Rightarrow a*(a+I00) \Rightarrow a*(a+b00)$$

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

$$E \underset{rm}{\Rightarrow} E * E \underset{rm}{\Rightarrow}$$

$$E * (E) \underset{rm}{\Rightarrow} E * (E + E) \underset{rm}{\Rightarrow} E * (E + I) \underset{rm}{\Rightarrow} E * (E + I0)$$

$$\underset{rm}{\Rightarrow} E * (E + I00) \underset{rm}{\Rightarrow} E * (E + b00) \underset{rm}{\Rightarrow} E * (I + b00)$$

$$\underset{rm}{\Rightarrow} E * (a + b00) \underset{rm}{\Rightarrow} I * (a + b00) \underset{rm}{\Rightarrow} a * (a + b00)$$

Possiamo concludere che $E \overset{*}{\underset{rm}{\Rightarrow}} a * (a + b00)$

Il linguaggio di una grammatica

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

$$L(G) = \{w \in T^* : S \overset{*}{\underset{G}{\Rightarrow}} w\}$$

  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*).