Automi e Linguaggi Formali

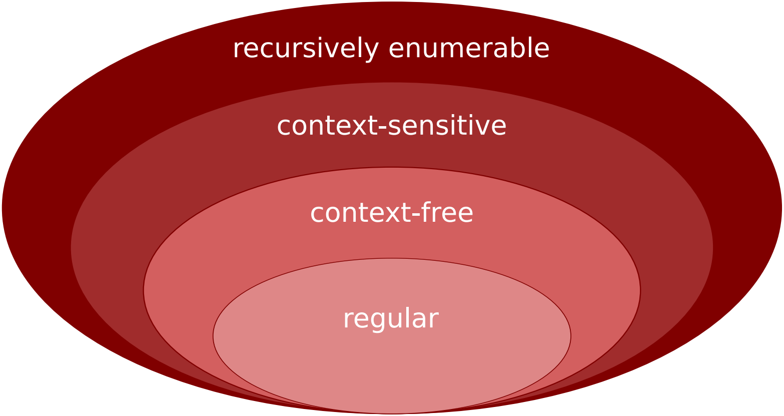
Grammatiche libere da contesto

A.A. 2014-2015
Enrico Mezzetti
emezzett@math.unipd.it

Università degli Studi di Padova

Grammatiche e Linguaggi Liberi da Contesto

- *Classe RL* non comprende molti linguaggi *interessanti*
  - E.g., ww^R, *palindromi*



Grammatiche e linguaggi liberi da contesto

- Consideriamo allora classi piu' espressive di linguaggi.
- **Linguaggi Liberi da Contesto** (Context-free languages)
  - Studio dei linguaggi naturali 1950→
  - Studio dei compilatori (*parser*) 1960→
  - In `DTD` per doc `XML`
- **Grammatiche libere da contesto** (Context-free grammars)
  - Base della sintassi BNF (*Backus-Naur-Form*)
    - *Regole di derivazione* → linguaggi di programmazione
- **Focus**
  - CFG e linguaggi che esse generano/descrivono
  - *Alberi sintattici*
  - *Automi a pila*
  - *Proprieta' dei CFL* (come per RL)

Esempio informale di CFG

- Linguaggio dei *palindormi*

$$L_{pal} = \{w \in \Sigma^* : w = w^R\}$$

  - E.g., $\mathtt{otto} \in L_{pal}$, $\mathtt{radar} \in L_{pal}$.

- $L_{pal} \notin \mathrm{RL}$ (gia' visto)
  - Sia $\Sigma = \{0, 1\}$ e supponiamo che L_{pal} sia regolare
  - Sia n dato dal pumping lemma, alora $0^n10^n \in L_{pal}$
    - Abbiamo visto per $k = 0$, $xy^kz \notin L_{pal}$

- Come decidere se una stringa $w \in L_{pal}$?
  - Definiamo L_{pal} induttivamente:

Base: ϵ, 0, e 1 sono palindromi

Induzione: Se w e' una palindrome, anche $0w0$ e $1w1$ lo sono

  - Nessun altra stringa e' palindroma

Esempio informale di CFG - 2

- CFG sono un modo formale per definire linguaggi come L_{pal}
  - Approccio formale *generativo*
- **Produzioni**

$$
\begin{array}{ll}
1. & P \rightarrow \epsilon \\
2. & P \rightarrow 0 \\
3. & P \rightarrow 1 \\
4. & P \rightarrow 0P0 \\
5. & P \rightarrow 1P1
\end{array}
$$

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

Definizione formale di CFG

- **CFG** e' definita da una *quadrupla*

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

dove

  - V e' un insieme finito di *variabili*.
  - T e' un insieme finito di *simboli* o *terminali*.
  - P e' un insieme finito di *produzioni* o *regole* della forma

$$\underset{\textit{testa}\ \uparrow}{A} \rightarrow \underset{\uparrow\ \textit{corpo}}{\alpha}$$

  dove

$$A \text{ e' una variabile e } \alpha \in (V \cup T)^*$$

  - S e' una variabile distinta che costituisce il *simbolo iniziale*.

Esempio

- $L_{pal} = \{w \in \Sigma^* : w = w^R\}$

$$G_{pal} = (\{P\}, \{0,1\}, A, P)$$

  dove le regole A sono:

$$\begin{aligned} P &\rightarrow \epsilon \\ P &\rightarrow 0 \\ P &\rightarrow 1 \\ P &\rightarrow 0P0 \\ P &\rightarrow 1P1 \end{aligned}$$

- Possiamo ”raggruppare” le produzioni con la stessa testa

$$P \rightarrow \epsilon|0|1|0P0|1P1$$

.

Esempio - 2

- Espressioni regolari su $\Sigma^{\{0,1\}}$ $\quad L_{regexp} = \{w : w = \{0,1\}^*\}$

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

dove le regole A sono:

$$
\begin{aligned}
E &\rightarrow \epsilon \\
E &\rightarrow 0 \\
E &\rightarrow 1 \\
E &\rightarrow E.E \\
E &\rightarrow E + E \\
E &\rightarrow E^* \\
E &\rightarrow (E)
\end{aligned}
$$

Esempio - 3

- Espressioni sempificate di un ling. di programmazione *PL*
  - Ammettiamo solo 2 operatori: $+$ e $\times$
  - Identificatori composti da lettere (a o b) o cifre (0 o 1)
    - Inizia con a o b e continua con una stringa in $\{a, b, 0, 1\}^*$

$$G_{PL} = (\{E, I\}, T, P, E)$$

- Variabili
  - E per le espressioni in PL che vogliamo definire
  - I per gli identificatori, e' un *linguaggio regolare*

$$L(I) = L((\mathbf{a} + \mathbf{b})(\mathbf{a} + \mathbf{b} + \mathbf{0} + \mathbf{1})^*)$$

- Terminali $T = \{+, \times, (,), a, b, 0, 1\}$
- Produzioni P

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

Appartenenza ad un linguaggio usando CFG

- *Inferenza ricorsiva*
  - Usando le produzioni dal corpo alla testa
  - Es: $a0101 + b01abb \to I + I \to E + E \to E \in L(PL)$
- *Derivazione*
  - Usando le produzioni dalla testa al corpo
  - Es: $E \to E{+}E \to I{+}I \to aI{+}bI \to \cdots \to a0101{+}b01abb \in L(PL)$
- Esempio di inferenza ricorsiva:

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

Derivazioni

- Consideriamo CFG $G = (V, T, P, S)$
  - Siano $A \in V$
  - $\{\alpha, \beta\} \subset (V \cup T)^*$ (stringhe in $(V \cup T)^*$)
  - $A \rightarrow \gamma \in P$
- Allora denotiamo una derivazione come

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

o se non e' ambiguo

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

- Diciamo che da $\alpha A\beta$ **si deriva** $\alpha\gamma\beta$
- Definiamo $\overset{*}{\Rightarrow}$ la chiusura riflessiva e transitiva di $\Rightarrow$
  - **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 $a \times (a + b00)$ da E in G_{PL}

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

- Diverse produzioni sono applicabili ad ogni passo

  $I \times E \Rightarrow a \times E \Rightarrow a \times (E)$, ma anche
  $I \times E \Rightarrow I \times (E) \Rightarrow a \times (E)$

- Non tutte le scelte sono funzionali a derivazioni

  $E \Rightarrow E + E$ non ci fa derivare $a \times (a + b00)$.

- *Inferenza* e *derivazione* sono **equivalenti**

Derivazioni a sinistra e a destra

- Ridurre lo spazio delle derivazioni possibili
  - Imposizione di un *ordine*
- **Derivazione a sinistra** (leftmost) $\underset{lm}{\Rightarrow}$
  - Rimpiazza sempre la variabile piu' a sinistra con il corpo di una delle sue regole
- **Derivazione a destra** (rightmost) $\underset{rm}{\Rightarrow}$
  - Rimpiazza sempre la variabile piu' a destra con il corpo di una delle sue regole.

Derivazioni a sinistra e a destra - 2

- Es. derivazione di $a \times (a + b00)$ da E in G_{PL}
- Der. a sinistra (appena vista)

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

- A destra:

$$E \underset{rm}{\Rightarrow} E \times E \underset{rm}{\Rightarrow} E \times (E) \underset{rm}{\Rightarrow} E \times (E + E) \underset{rm}{\Rightarrow}$$
$$E \times (E + I) \underset{rm}{\Rightarrow} E \times (E + I0) \underset{rm}{\Rightarrow} E \times (E + I00) \underset{rm}{\Rightarrow}$$
$$E \times (E + b00) \underset{rm}{\Rightarrow} E \times (I + b00) \underset{rm}{\Rightarrow} E \times (a + b00) \underset{rm}{\Rightarrow}$$
$$I \times (a + b00) \underset{rm}{\Rightarrow} a \times (a + b00)$$

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

Il linguaggio di una grammatica

- Se $G(V, T, P, S)$ e' una CFG, allora il **linguaggio di** G e'
$$L(G) = \{w \in T^* : S \underset{G}{\overset{*}{\Rightarrow}} w\}$$
l'insieme delle stringhe su T^* derivabili a partire da S

- Se L e' linguaggio di una CGF allora L e' un **linguaggio libero da contesto** (CFL)

- Esempio: $L(G_{pal})$ e' un linguaggio libero da contesto

Il linguaggio di una grammatica - 2

Th. 5.7 $L(G_{pal})$ dove $G_{pal} = (\{P\}, \{0,1\}, P \rightarrow \epsilon|0|1|0P0|1P1, P)$ e' l'insieme delle aplindrome in $\{0,1\}$

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

Se Supponiamo $w = w^R$
Dimostriamo per induzione su $|w|$ che $w \in L(G_{pal})$.

- **Base:** $|w| = 0$ o $|w| = 1 \Rightarrow$ w e' $\epsilon, 0,$ o 1
  Dato che $P \rightarrow \epsilon, P \rightarrow 0,$ and $P \rightarrow 1$ sono produzioni, concludiamo che $P \overset{*}{\underset{G}{\Rightarrow}} w$ in tutti i casi base.
- **Induzione:** $|w| \geq 2$.
  Dato che $w = w^R \rightarrow w = 0x0$, o $w = 1x1$, e $x = x^R$.
  Se $w = 0x0$ sappiamo che per l'ipotesi induttiva $P \overset{*}{\Rightarrow} x$.
  Allora $P \Rightarrow 0P0 \overset{*}{\Rightarrow} 0x0 = w \in L(G_{pal})$.
  Il caso di $w = 1x1$ e' simile.

Solo se Supponiamo che $w \in L(G_{pal})$
Dobbiamo mostrare che $w = w^R$.
Dato che $w \in L(G_{pal})$, abbiamo $P \overset{*}{\Rightarrow} w$.
Faremo un'induzione sulla lunghezza di $\overset{*}{\Rightarrow}$.

- **Base:(1 passo)** La derivazione $P \overset{*}{\Rightarrow} w$ ha 1 passo
  $\Rightarrow w = \epsilon, 0,$ o 1 che sono tutte palindrome
- **Induzione:(n+1 passi)** Per ipotesi induttiva l'enunciato e' vero per derivazioni di n passi.
  Una drivazione di $n+1$ passi sara' nella forma

$$w = 0x0 \overset{*}{\Leftarrow} 0P0 \Leftarrow P$$

o

$$w = 1x1 \overset{*}{\Leftarrow} 1P1 \Leftarrow P$$

La seconda derivazione ha n passi
$\Rightarrow$ per ipotesi induttiva x e' una palindrome
Poiche' anche $0x0$ e $1x1$ lo sono
$\Rightarrow$ w e' un palindromo

Forme sentenziali

- Sia $G = (V, T, P, S)$ una CFG, e $\alpha \in (V \cup T)^*$.
  - Se $S \overset{*}{\Rightarrow} \alpha$ diciamo che α e' una **forma sentenziale**.
  - Se $S \underset{lm}{\Rightarrow} \alpha$ diciamo che α e' una **forma sentenziale sinistra**,
  - Se $S \underset{rm}{\Rightarrow} \alpha$ diciamo che α e' una **forma sentenziale destra**
- Quindi $L(G)$ contiene le forme sentenziali che sono in T^*
  - Cio che consistono unicamente di terminali

Esempio

- Prendiamo la G delle espressioni. Allora $E \times (I + E)$ e' una forma sentenziale perche'

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

  Questa derivazione non e' ne' a sinistra ne' a destra
- $a \times E$ e' una forma sentenziale sinistra, perche'

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

- $E \times (E + E)$ e' una forma sentenziale destra, perche'

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