

Automati e Linguaggi Formali

Proprieta' dei linguaggi liberi da contesto

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



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Proprieta' dei linguaggi liberi da contesto

- **Semplificazione**
 - CFG possono essere semplificate
 - Se L e' CFL allora la sua grammatica ha una forma speciale
- **Pumping Lemma per i CFL**
 - Simile al lemma per i LR
 - Se L non e' CFL il pumping lemma mostrera' una contraddizione
- **Proprieta' di chiusura**
 - Alcune proprieta' di chiusura di LR valgono anche per CFL
- **Proprieta' di decisione**
 - Controllare alcune proprieta': e.g., $w \in L$ e $L = \emptyset$
 - Equivalenza *non e' decidibile*



Automati e Linguaggi Formali – A.A. 2014-2015
Docente: Enrico Mezzetti

2 of 20

Semplificazione e forme normali

- Ogni CFL (senza ε) e' generato da una CFG dove tutte le produzioni sono della forma

$$A \rightarrow BC \text{ oppure } A \rightarrow a$$

dove A, B, e C sono variabili, e a e un simbolo terminale

- **Forma normale di Chomsky (CNF)**
- Semplificazioni preliminari per ottenerla
 - Eliminare i *simboli inutili*, che non appaiono in nessuna derivazione $S \xRightarrow{*} w$
 - Eliminare le *ε -produzioni*, della forma $A \rightarrow \varepsilon$
 - Eliminare le *produzioni unitarie*, della forma $A \rightarrow B$, dove A e B sono variabili
 - Eliminare le produzioni con piu di due non-terminali come corpo

Esempi di semplificazione preliminare

- **Simboli inutili**
$$\begin{array}{l} S \rightarrow AB|a \\ A \rightarrow b \end{array} \Rightarrow \begin{array}{l} S \rightarrow \cancel{AB}|a \\ A \rightarrow \cancel{b} \end{array} \Rightarrow S \rightarrow a$$
- **ε -produzioni** $[L(G) \rightarrow L(G) - \{\varepsilon\}]$
$$\begin{array}{l} S \rightarrow AB \\ A \rightarrow aAA|\varepsilon \\ B \rightarrow bBB|\varepsilon \end{array} \Rightarrow \begin{array}{l} S \rightarrow AB \\ A \rightarrow aAA|\cancel{\varepsilon} \\ B \rightarrow bBB|\cancel{\varepsilon} \end{array} \Rightarrow \begin{array}{l} S \rightarrow AB|A|B \\ A \rightarrow aAA|aA|a \\ B \rightarrow bBB|bB|b \end{array}$$
- **Produzioni unitarie**
$$\begin{array}{l} I \rightarrow a|b|a|b|I|I \\ F \rightarrow I|(E) \\ T \rightarrow F|T \times F \\ E \rightarrow T|E + T \end{array} \Rightarrow \begin{array}{l} I \rightarrow a|b|a|b|I|I \\ F \rightarrow I|(E)|a|b|a|b|I|I \\ T \rightarrow \cancel{F}|T \times F|(E)|a|b|a|b|I|I \\ E \rightarrow \cancel{T}|E + T|\cancel{T} \times F|(E)|a|b|a|b|I|I \end{array}$$



Automati e Linguaggi Formali – A.A. 2014-2015
Docente: Enrico Mezzetti

3 of 20



Automati e Linguaggi Formali – A.A. 2014-2015
Docente: Enrico Mezzetti

4 of 20

Esempio di CNF

- Produzioni della forma $A \rightarrow BC$ oppure $A \rightarrow a$
- Consideriamo la CFG delle espressioni "semplificate"

$$E \rightarrow E + T \mid T \times F \mid (E) \mid a \mid b \mid \mid \mid 0 \mid 1$$

$$T \rightarrow T \times F \mid (E) \mid a \mid b \mid \mid \mid 0 \mid 1$$

$$F \rightarrow (E) \mid a \mid b \mid \mid \mid 0 \mid 1$$

$$I \rightarrow a \mid b \mid \mid \mid 0 \mid 1$$
- Aggiungiamo le produzioni sui terminali

$$A \rightarrow a, B \rightarrow b, Z \rightarrow 0, U \rightarrow 1$$

$$P \rightarrow +, M \rightarrow \times, S \rightarrow (, D \rightarrow)$$
- Otteniamo la grammatica

$$E \rightarrow EPT \mid TMF \mid SED \mid A \mid B \mid IA \mid IB \mid IZ \mid IU$$

$$T \rightarrow TMF \mid SED \mid A \mid B \mid IA \mid IB \mid IZ \mid IU$$

$$F \rightarrow SED \mid A \mid B \mid IA \mid IB \mid IZ \mid IU$$

$$I \rightarrow A \mid B \mid IA \mid IB \mid IZ \mid IU$$

$$A \rightarrow a, B \rightarrow b, Z \rightarrow 0, U \rightarrow 1$$

$$P \rightarrow +, M \rightarrow \times, S \rightarrow (, D \rightarrow)$$



Esempio di CNF - 2

- Produzioni della forma $A \rightarrow BC$ oppure $A \rightarrow a$
- Eliminazione corpi con piu' di due non-terminali
 - $E \rightarrow EPT \Rightarrow E \rightarrow EC_1, C_1 \rightarrow PT$
 - $E \vee T \rightarrow TMF \Rightarrow E \vee T \rightarrow TC_2, C_2 \rightarrow MF$
 - $E \vee T \vee F \rightarrow SED \Rightarrow E \vee T \vee F \rightarrow LC_3, C_3 \rightarrow ER$
- Otteniamo la grammatica

$$E \rightarrow EC_1 \mid TC_2 \mid SC_3 \mid A \mid B \mid IA \mid IB \mid IZ \mid IU$$

$$T \rightarrow TC_2 \mid SC_3 \mid A \mid B \mid IA \mid IB \mid IZ \mid IU$$

$$F \rightarrow SC_3 \mid A \mid B \mid IA \mid IB \mid IZ \mid IU$$

$$I \rightarrow A \mid B \mid IA \mid IB \mid IZ \mid IU$$

$$A \rightarrow a, b \rightarrow b, Z \rightarrow 0, U \rightarrow 1$$

$$P \rightarrow +, M \rightarrow \times, S \rightarrow (, D \rightarrow)$$

$$C_1 \rightarrow PT, C_2 \rightarrow MF, C_3 \rightarrow ED$$



Proprieta' CNF

- Grammatiche CNF producono solo *alberi binari*
- Esempio

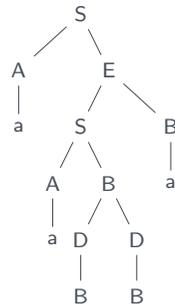
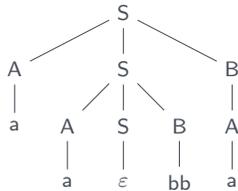
$$S \rightarrow ASB \mid \varepsilon \quad S \rightarrow AE \mid AB$$

$$A \rightarrow aAS \mid a \quad A \rightarrow CF \mid CA \mid a$$

$$B \rightarrow SbS \mid A \mid bb \quad B \rightarrow SG \mid DS \mid SD \mid b \mid CF \mid CA \mid a \mid DD$$

$$C \rightarrow a \quad D \rightarrow b$$

$$E \rightarrow SB \quad F \rightarrow AS \quad G \rightarrow DS$$



Proprieta' CNF - 2

Th.7.17: Dato un albero sintattico con prodotto w conforme ad una grammatica CFG G in CNF, se l'altezza dell'albero e' n , allora $|w| \leq 2^{n-1}$.

Prova: Per induzione sull'altezza n dell'albero.

$n=1$ Albero costituito da un'unica produzione del tipo $A \rightarrow a$

- $|w| = 1 \leq 2^0 = 1$

$n>1$ $n > 1 \Rightarrow$ la radice usa un produzione del tipo $A \rightarrow BC$

- Nessun cammino nei sottoalberi B, C puo' avere altezza maggiore di $n-1$
- Per ipotesi induttiva i prodotti dei sotto-alberi w_B e w_C hanno lunghezza $\leq 2^{(n-1)-1} = 2^{n-2}$
- $|w| = |w_B + w_C| \leq 2^{n-2} + 2^{n-2} = 2^{n-1}$



Pumping lemma per CFL

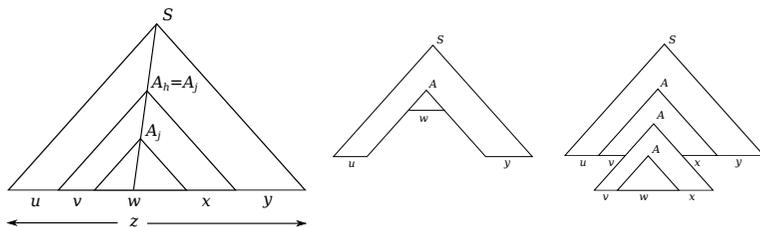
- **Informalmente** In ogni stringa sufficientemente lunga di un CFL si possono trovare due sottostringhe vicine che e possibile eliminare o ripetere (insieme), ottenendo sempre stringhe del linguaggio
- **Formalmente** Sia L un CFL. Esiste una costante n tale che, se $z \in L$ e $|z| \geq n$, possiamo scrivere $z = uvwxy$ con le seguenti condizioni
 - 1 $|vwx| \leq n$
 - 2 $vx \neq \varepsilon$
 - 3 $\forall i \geq 0, uv^iwx^iy \in L$

Informalmente

- Se la stringa w e' sufficientemente lunga, l'albero sintattico che produce $w = uvwxy$ ha un simbolo non terminale che si ripete in un cammino dalla radice ad una foglia. Supponiamo $A_i = A_j$
- Individuiamo il sottoalbero con radice A_j e chiamiamo w il suo prodotto
- Individuiamo il sottoalbero con radice A_i e chiamiamo vwx il suo prodotto
- Poiche' $A_i = A_j$ possiamo rimpiazzare il sottoalbero di A_i con quello di A_j , ottenendo quindi $uvwxy$, che deve ancora appartenere a L .
- Oppure possiamo rimpiazzare il sottoalbero di A_j con quello di A_i , ottenendo $uvvwxxy$, ancora generata da L .

Prova sketch

- Consideriamo una grammatica in CNF G per L
 - Supponiamo che G abbia m variabili e scegliamo $n = 2m$
 - Per proprieta' CNF: albero con cammino lungo m (una occorrenza per variabile) ha prodotto di lunghezza 2^{m-1}
 - Ma $|z| = 2^m \Rightarrow$ cammino di lunghezza $\geq m + 1$
 - Esistono almeno due derivazioni di una stessa variabile (sottoalbero)



Esempio

- Consideriamo $L = \{0^n 1^n 2^n | n \geq 1\}$
- Dato un n generico, scegliamo $z = 0^n 1^n 2^n$
- Per qualunque scomposizione di z in $uvwxy$, con $|vwx| \leq n$ e $vx \neq \varepsilon$, vwx non puo contenere sia 0 che 2
 - Ultimo 0 e primo 2 sono lontani $n + 1$ posti
- Per casi:
 - vwx non contiene 2 $\Rightarrow vx$ ha solo 0 e 1
 $uvwxy \notin L$ dato che ha n 2, ma meno di n 0 o 1
 - vwx non contiene 0 \Rightarrow Analogo

Altri esempi

- I CFL non sanno abbinare coppie con lo stesso numero di simboli, se le coppie sono intrecciate
 - E.g., $L = \{0^i 1^j 2^i 3^j \mid i, j \geq 1\}$
 - Dato n , scegliamo $z = 0^n 1^n 2^n 3^n$. Quindi vwx contiene un solo simbolo o due simboli. In ogni caso, le stringhe generate non sono in L .
- I CFL non sanno abbinare due stringhe di lunghezza arbitraria, se sono su un alfabeto di più di un simbolo
 - E.g., $L = \{ww \mid w \in \{0, 1\}^*\}$
 - Dato n , scegliamo $z = 0^n 1^n 0^n 1^n$.
Comunque la scomponiamo, non otteniamo stringhe in L .

✎ Altri esempi...



Proprietà di chiusura dei CFL

Th.7.24: I CFL sono chiusi sotto *unione*, *concatenazione*, *chiusura di Kleene* e *chiusura positiva* (+)

Prova: E' sufficiente operare sulle grammatiche:

- per l'unione: $S \rightarrow A|B$
- per la concatenazione: $S \rightarrow AB$
- per la chiusura di Kleene: $S \rightarrow SA|\epsilon$
- per la chiusura positiva: $S \rightarrow SA|A$



Chiusura rispetto all'inversione

Th.: Se L è CF, allora L^R è anche CF

Prova: Poiché L è CF, allora sarà generata da $G = (V, T, P, S)$.

- Costruiamo $G^R = (V, T, P^R, S)$ dove

$$P^R = \{A \rightarrow \alpha^R : A \rightarrow \alpha \in P\}$$

- Si può dimostrare per induzione sulla lunghezza delle derivazioni in G e in G^R che $(L(G))^R = L(G^R)$



Chiusura rispetto all'intersezione

- I CFL non sono chiusi sotto l'intersezione
- Sia $L_1 = \{0^n 1^n 2^i : n \geq 1, i \geq 1\}$ CFL con grammatica

$$\begin{aligned} S &\rightarrow AB \\ A &\rightarrow 0A1|01 \\ B &\rightarrow 2B|2 \end{aligned}$$

- Sia $L_2 = \{0^i 1^n 2^n : n \geq 1, i \geq 1\}$ CFL con grammatica

$$\begin{aligned} S &\rightarrow AB \\ A &\rightarrow 0A|0 \\ B &\rightarrow 12B|12 \end{aligned}$$

- $L_1 \cap L_2 = \{0^n 1^n 2^n : n \geq 1\}$ non è CF.



Intersezione tra CFL e linguaggi regolari

Th.7.27: Se L e' CF, e R e' regolare, allora $L \cap R$ e' CF

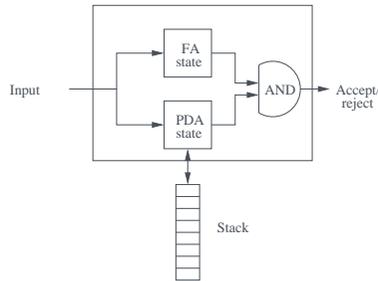
Prova: Sia L accettato dal PDA

$$P = (Q_P, \Sigma, \Gamma, \delta_P, q_P, Z_0, F_P)$$

per stato finale e R sia accettato dal FA

$$A = (Q_A, \Sigma, \Gamma, \delta_A, q_A, F_A)$$

- Costruiamo un PDA per $L \cap R$



Intersezione tra CFL e linguaggi regolari - 2

- Formalmente

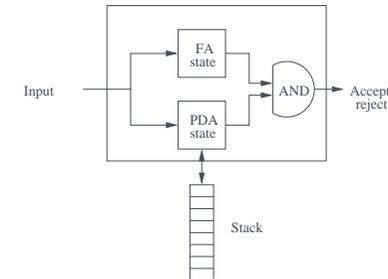
$$P' = (Q_P \times Q_A, \Sigma, \Gamma, \delta_P, (q_P, q_A), Z_0, (F_P \times F_A))$$

dove

$$\delta((q, p), a, X) = \{((r, \hat{\delta}_A(p, a)), \gamma) : (r, \gamma) \in \delta_P(q, a, X)\}$$

- Per induzione su \vdash^* si prova che

$$\text{In } P \ (q_P, w, Z_0) \vdash^* (q, \varepsilon, \gamma) \iff ((q_P, q_A), w, Z_0) \vdash^* ((q, \hat{\delta}(p_A, w)), \varepsilon, \gamma) \text{ In } P'$$



Differenza e complemento tra CFL e LR

Th.7.29: Siano L, L_1, L_2 CFL e R regolare

- 1 $L \setminus R$ e' CF
- 2 \bar{L} non e' necessariamente CF
- 3 $L_1 \setminus L_2$ non e' necessariamente CF

Prova:

- (1) $L \setminus R = L \cap \bar{R}$ e sappiamo che \bar{R} e' regolare e che $L \cap \bar{R}$ e' regolare per Th.7.27
- (2) Se L fosse CF allora
 - $\overline{L_1 \cup L_2}$ sarebbe CF
 - ma $\overline{L_1 \cup L_2} = L_1 \cap L_2$ che non e' CF (vedi esempio)
- (3) Σ^* e' CFL per ogni alfabeto Σ e quindi anche $\Sigma^* \setminus L = \bar{L}$ sarebbe CFL, in contraddizione con punto 2



Problemi indecidibili per CFL

- Non esiste un algoritmo in grado di decidere i seguenti problemi
 - Verificare che una CFG e' ambigua
 - Verificare che un CFL e' inerentemente ambiguo
 - Verificare che l'intersezione di due CFL e' vuota
 - Verificare che due CFL sono uguali
 - Verificare che un CFL e' uguale a Σ^*

