UNIVERSITÀ DEGLI STUDI DI TRENTO

Dipartimento di Ingegneria e Scienza dell'Informazione

Corso di Laurea in
Informatica

Appunti di

Linguaggi Formali e Compilatori

Prof.ssa Paola Quaglia

Autore
Emanuele Nardi

Revisore
Filippo Frezza

Anno accademico 2017/2018

Introduzione

Il materiale didattico trattato in questo documento si trova nella cartella Google Drive del corso di Informatica.

Hai trovato un errore? Inviami un'e-mail o contattami direttamante su Telegram.

Ulteriori contatti dell'ateneo si trovano sulla pagina del DISI.

Indice

Elenco delle figure

1 Panoramica

L'obiettivo dell'analisi predittiva è quello di produrre modelli accurati.

[...]

2 Grammatiche

Alcune definizioni:

Alfabeto insieme finito e non vuoto di simboli;

Stringa/parola sequenza finita *o nulla* i simboli dell'alfabeto, ottenuta per giustapposizione di simboli;

Parola vuota viene denotata dal simbolo $\varepsilon \notin Q$;

Lunghezza di una parola è data dal no. di simboli dell'alfabeto che lo compongono ed è $\varnothing$ se la parola è vuota.

2.1 Grammatiche generative

Una grammatica si definisce come un insieme:

$$G = \{V, T, S, P\} \tag{1}$$

Dove:

- V: *vocabolario*, insieme di simboli, finito e non vuoto. I *simboli* si divisono a loro volta in simboli terminali e simboli non terminali;
- T: insieme dei *simboli terminali*, tale che $T \subset V$;
- S: simbolo iniziale, $S \in V \setminus T$, dove:
  - $V \setminus T$ sono l'insieme dei simboli non-terminali;
- P: insieme di produzioni, in generale hanno una forma $\alpha \to \beta$, dove:
  - α è una stringa non vuota su V contenente almeno un elemento non terminale;
  - β è una stringa su V, oppure è ε.

2.1.1 Grammatiche e convenzioni

Un esempio di grammatica:

$$G_1 = (\underbrace{\{S, a, b\}}_{V}, \underbrace{\{a, b\}}_{T}, S, \underbrace{\{S \longrightarrow aSb, S \longrightarrow \varepsilon\}}_{P})$$

D'ora in avanti utilizzeremo le seguenti **convenzioni**:

- simboli in $V \setminus T$ (non-terminali) denotati da lettere maiuscole, nella quale si cerca di non utilizzare X ed Y;
- simboli in T (terminali) denotati da lettere minuscole;
- X, Y denotate da un generico simbolo in V;
- $\alpha, \beta, \gamma \ldots$ denotate da parole su V^*[1].

[1] Il simbolo * si chiama *Kleene star* e denota la ripetizione di 0 o più volte di simboli all'interno dell'insieme di cui fanno parte.

2.2 Grammatica libera (da contesto)

Prendiamo un insieme di produzioni definito nel seguente modo:

$$\{S \longrightarrow aSb,\, S \longrightarrow A,\, S \longrightarrow \varepsilon\}$$

Lo riscriviamo così:

$$\begin{aligned} S &\longrightarrow aSb \\ S &\longrightarrow A \\ S &\longrightarrow \varepsilon \end{aligned}$$

O in forma semplificata utilizzando il simbolo di *pipe*:

$$S \longrightarrow aSb \mid A \mid \varepsilon$$

Banalmente un grammatica è *libera da contesto* se alla sinistra del simbolo di produzione $\longrightarrow$ non sono presenti simboli non-terminali.

Ad esempio una produzione $aS \longrightarrow b$ denota una grammatica non *libera da contesto.*

Definizione 1 (grammatica libera da contesto)**.** *Una grammatica generica è libera da contesto (o libero o context-free) se ogni produzione ha la forma $A \longrightarrow B$ (cioè che contiene un ed un solo simbolo terminale)*

Derivazioni destre (sinistre) Nel caso di grammatiche libere si definiscono le derivazioni destre e sinistre (rigthmost derivation/leftmost derivation).
Nel caso rightmost (leftmost) si richiede che ad ogni passo di derivazione $\mu \implies \gamma$ venga rimpiazzato il non-terinale più a destra (sinistra) in μ.

Definizione 2 (grammatica ambigua)**.** *G* è ambigua *se esiste $w \in L(G)$ tale che esistono* ***due*** *derivazioni canoniche distinte, entrambe destre oppure entrambe sinistre.*

2.3 Parole generate dalla grammatica

Prendiamo come esempio le produzioni di una grammatica:

$$S \longrightarrow aSb \mid \varepsilon \tag{2}$$

Come posso generare parole dell'alfabeto?

$$S \implies \varepsilon \qquad \varepsilon \text{ è generata dalla grammatica } G$$

ε è quindi una delle parole generate da queste grammatica, ma non è l'unica parola che possono generare.

$$\begin{aligned} &S \implies aSb \implies ab \qquad ab \text{ è generata dalla grammatica } G \\ &S \implies aSb \implies aaSbb \implies aabb \end{aligned}$$

Continuare è banale. Qual è quindi la linguaggio generato dalla grammatica che ha come produzioni quelle mostrate nell'equazione 2?.

La risposta viene può essere data dall'osservazione del numero crescente di a e di b. La grammatica denotata dalle produzioni in equazione 2 producono quindi il linguaggio:

$$\{a^n b^n \mid n \geqslant 0\}$$

2.4 Derivazione

Vediamo ora un paio di definizioni formali

Definizione 3 (derivazione in un passo). *$\mu \Longrightarrow \gamma$ (un passo generico) (γ deriva in un passo da μ, data la grammatica G) se (è vero che) $\mu = \mu_1 \propto \mu_2 \wedge \alpha \longrightarrow \beta$ è una produzione di $G \wedge \gamma = \mu_1 \beta \mu_2$ (copio il contesto sinistro e poi quello destro)*

Definizione 4 (derivazione in più passi). *$\mu \underset{\Longrightarrow}{+} \gamma$ (γ deriva in uno o più passi da μ, data la grammatica G) se (è vero che) (esiste una sequenza del tipo) $\mu \Longrightarrow \delta_0 \Longrightarrow \delta_1 \Longrightarrow \ldots \Longrightarrow \gamma$.*

Quest'ultima è una definizione generica: non sto dicendo che γ è composta solo da simboli terminali.

$$L(G) = \{w \mid w \in T^* \wedge S \underset{\Longrightarrow}{+} w\}$$

2.5 Esercizi

Esercizio

$$\begin{aligned} S &\longrightarrow aAb \\ aA &\longrightarrow aaAb \\ A &\longrightarrow \varepsilon \end{aligned}$$

Innanzitutto notiamo che questa grammatica non è *libera da contesto*.
Proviamo a scrivere dei linguaggi e poi troviamo dei controesempi per provare che non sia il linguaggio prodotto da questa grammatica.

$$\begin{aligned} &L_1 : \{a^n b^m \mid n \geqslant 0 \wedge (m = 0 \vee m = 1)\} \\ &L_2 : \{a^n b^m \mid n > 0 \wedge m = n - 1\} \\ &L_3 : \{a^n b^n \mid n > 0\} \\ &L_4 : \{a^{2n+1} b^{n+1} \mid n \geqslant 0\} \\ &L_5 : \{a^n b^n \mid n \geqslant 1\} \end{aligned}$$

L_2 non può essere perché $ab \in L(G) \wedge \notin L_2$, $aabb \notin L_4$.
Prendiamo in considerazione la grammatica G_2 che il seguente insieme di produzioni

$$S \longrightarrow aSb \mid ab$$

G_1 e G_2 sono completamente diverse, poiché hanno produzioni diverse, ma il linguaggio generato è lo stesso.

Ricorda. *Dato il linguaggio L possono esistere più grammatiche diverse fra loro che generano L.*

<u>É indecidibile</u> il linguaggio generato da una grammatica G per G arbitrario.

$$\Big\downarrow \text{o meglio}$$

$\underline{\nexists}$ algoritmo che dato G e dato L, dicesse $L = L(G)$ oppure no.

$$\Big\downarrow \text{o meglio}$$

<u>Non può esistere</u> . . .

Esercizio

$G_3 = \{\{S, A, B, a, b\}, \{a, b\}, S, \{\dots\}\}$

$$\begin{aligned} S &\longrightarrow AB \\ A &\longrightarrow aA \\ A &\longrightarrow a \\ B &\longrightarrow Bb \\ B &\longrightarrow b \end{aligned} \qquad\qquad \begin{aligned} S &\longrightarrow AB \\ A &\longrightarrow aA \mid a \\ \\ B &\longrightarrow Bb \mid b \end{aligned}$$

Prima di tutto notiamo che $A \longrightarrow aA$ produce a^n con $n \geqslant 1$, e $B \longrightarrow Bb$ produce b^m con $m \geqslant 1$.

Dopodiché notiamo che la grammatica G_3 non è altro che la concatenazione di stringhe.

$$L(G_3) = \{w_4 w_5 \mid w_4 \in L(G_4) \wedge w_5 \in L(G_5)\}$$

dove G_4 e G_5 sono definiti dalle produzioni

$$\begin{aligned} G_4 &: A \longrightarrow aA \mid a \\ G_5 &: B \longrightarrow Bb \mid b \end{aligned}$$

Esercizio

$$\begin{aligned} S &\longrightarrow aSBc \mid abc \\ cB &\longrightarrow Bc \\ bB &\longrightarrow bb \end{aligned}$$

La grammatica *non è libera*.

Esercizio

$G_6 : S \longrightarrow aB$ è una grammatica?

É possibile ricostruire la grammatica dalle sue produzioni:

$$(\{S, B, a\}, \{a\}, S, \{S \longrightarrow aB\})$$

Quale linguaggio genera? Nessuno ✗, vuoto ✓

$$L(G_6) = \varnothing$$

Esercizio

$G_7 : S \longrightarrow \varepsilon$ è una grammatica?

$$(\{S\}, \varnothing, S, \{S \longrightarrow \varepsilon\})$$

Quale linguaggio genera?

$$L(G_7) = \{\varepsilon\}$$

Esercizio

$$S \longrightarrow \varnothing B \mid 1A$$
$$cB \longrightarrow \varnothing \mid \varnothing S \mid 1AA$$
$$bB \longrightarrow \mid 1S \mid \varnothing BB$$

Il linguaggio che ha come insieme di produzioni quello elencato sopra produce il seguente linguaggio

$$L(G_8) = \{w \mid count(0,w) = count(1,w)\}$$

Questa grammatica produce l'insieme delle parole con le combinazioni di zeri e uni, con lo stesso no. di zeri e di uni. In questo caso l'ordine degli zeri e degli uni non conta.

Esercizio

Definisci G_9 tale che $L(G_2) = \{a^k b^n\}$.

Possono esserci diversi approcci per risolvere il problema, il primo e più semplice è quello di approcciare il problema tramite dividi-et-impera: si scompone quindi un problema complesso in tanti problemi più semplici.

La soluzione è quindi la seguente:

$$S \longrightarrow AB$$
$$A \longrightarrow aA \mid a$$
$$B \longrightarrow bB \mid b$$

Notiamo che il primo passaggio consiste nella trasformazione del simbolo iniziale in simboli non-terminali che a loro volta vengono scomposti in simboli terminali, almeno parzialmente.

Un altro approccio consiste nel produrre una sola stringa a in posizione iniziale.

$$S \longrightarrow aS \mid aB$$
$$A \longrightarrow bB \mid b$$
$$B \longrightarrow B \mid b$$

In modo tale che i coefficienti k ed n siano realmente diversi fra loro, a differenza dell'approccio dividi-et-impera nella quale i coeffienti risultavano in un caso particolare essendo uguali.

Un terzo approccio consiste nell'usa la definizione formale di derivazione.

$$S \longrightarrow ab \mid aS \mid Sb$$

Producendo così $S \underset{\Longrightarrow}{+} a^i S b^i$.

Esercizio

Definisci G_9 tale che $L(G_2) = \{a^k b^n c^{2n}\}$.

In questo caso bisogna far sì che il no. di occorrenze della stringa b sia esattamente il doppio del no. di occorrenze della stringa c.

$$S \longrightarrow AB \mid aB$$
$$A \longrightarrow aA \mid a$$
$$B \longrightarrow bBcc \mid bcc$$

La parole più piccola che è possibile produrre con questa grammatica è $abcc$ dopo il no. di occorrenze di c è il doppio di quello di b.

Esercizio

Definisci G_{10} tale che $L(G_2) = \{a^k b^n d^k\}$.

$$\begin{aligned} S &\longrightarrow AB \qquad\qquad\qquad \text{2 parametri: } k \text{ ed } n \\ A &\longrightarrow abb \\ B &\longrightarrow db \end{aligned}$$

Nota che la stringa a e la stringa d hanno lo stesso no. di occorrenze.

Ricorda. *Sintetico è bello.*

Una soluzione alternativa alla precedente potrebbe essere.

$$\begin{aligned} S &\longrightarrow aSdd \longrightarrow aBdd \\ B &\longrightarrow bB \mid b \end{aligned}$$

2.6 Albero di derivazione

$$\begin{aligned} S &\longrightarrow aSb \\ S &\longrightarrow aSb \implies aabb \end{aligned}$$

$aabb$ è la parola generata.

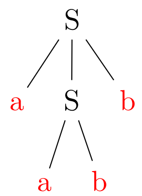


Figura 1: Albero di derivazione

Consideriamo ad esempio la produzione $E \longrightarrow E + E \mid E * E \mid 4$ e la parola $4 + 4 + 4$.
L'albero di derivazione è il seguente:

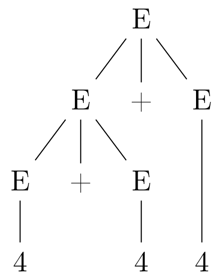


Figura 2: Esempio di *left-most derivation*: è associativa a sinistra come la somma

Mentre se consideriamo la parola $4 + 4 * 4$, l'albero di derivazione è il seguente:

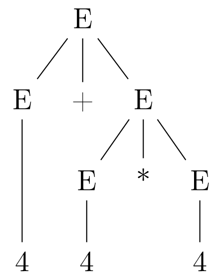


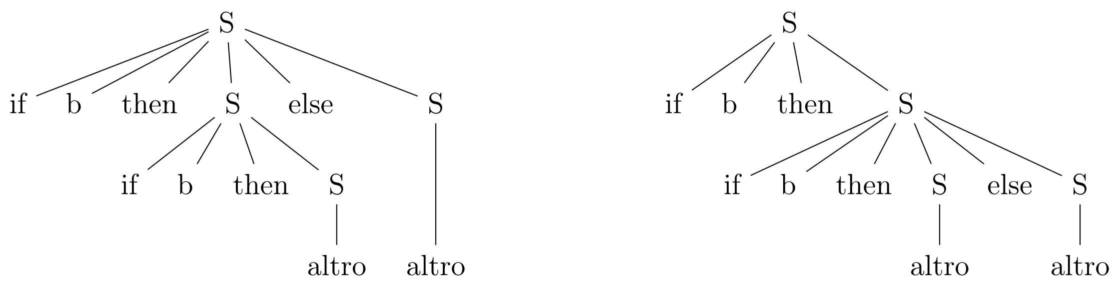
Figura 3: Esempio di *right-most derivation*: è associativa a destra come la moltiplicazione

$$S \longrightarrow \text{ if } b \text{ then } S \mid \text{if } b \text{ then } S \text{ else } S \mid \textit{altro}$$

che produce la seguente parola appartenente al linguaggio

$$w: \text{if } b \text{ then if } b \text{ then } \underline{\textit{altro}} \text{ else } \underline{\textit{altro}}$$

rappresentiamolo come albero di derivazione:



(a) left-most derivation

(b) right-most derivation

Figura 4: Esempio di derivazione di un *if-then-else statement*

w risulta essere *ambigua*: sono presenti due derivazioni (una *left-most* ed una *right-most*) che producono la stessa parola.

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

è una grammatica libera se tutte le produzioni sono del tipo $A \longrightarrow B$, dove $B \in V^*$ e A è un simbolo non-terminale.

Un linguaggio L è libero da contesto se esiste una grammatica libera da contesto G tale che $L = L(G)$.

Dato L possono esistere più grammatiche distinte che generano L.

In generale, dato un linguaggio L e una grammatica G, **non esiste** un lagoritmo per dimostrare che $L = L(G)$.

Inoltre **non esiste** un algoritmo per dimostrare che G è ambigua.

2.7 Unione di linguaggi

Lemma 1. ***La classe dei linguaggi liberi è chiusa rispetto all'unione****. Cioè dati generici linguaggi liberi L_1 e L_2, il linguaggio che contiene tutte e sole le parole $w \in L_1 \cup L_2$ appartiene essa stessa alla classe dei linguaggi liberi.*

Se L_1 ed L_2 sono definiti come segue

$$L_1 \text{ libero} \Longrightarrow \exists\, G_1 = (V_1, T_1, S_1, P_1) \text{ t.c. } L_1 = L(G_1)$$
$$L_2 \text{ libero} \Longrightarrow \exists\, G_2 = (V_2, T_2, S_2, P_2) \text{ t.c. } L_2 = L(G_2)$$

allora la loro unione definisce L_3

$$L_3 \longrightarrow G_3 = (V_1 \cup V_2 \cup \{S\}, T_1 \cup T_2, \{S\}, P_1 \cup P_2 \cup \{S \longrightarrow S_1 \mid S_2)$$

dove S è nuovo rispetto a V_1 e a V_2.

Nota. *Tutto questo è vero solo se abbiamo* ***ridenominato i simboli non-terminali*** *di G_1 e G_2 in modo da* ***non avere omonimie****.*

$$L_1 \qquad S_1 \longrightarrow a \qquad (\{S_1, a\}, \{a\}, S_1, \underbrace{S_1 \longrightarrow a}_{P_1})$$

$$L_2 \qquad S_2 \longrightarrow b \qquad (\{S_2, b\}, \{b\}, S_2, \underbrace{S_2 \longrightarrow b}_{P_2})$$

$$L_3 \qquad S \longrightarrow S1 \mid S_2 \qquad (\{S_1, a, S_2, b\}, \{a, b\}, \{S\}, \{S \longrightarrow S_1 \mid S_2, \underbrace{S_1 \longrightarrow a}_{P_1}, \underbrace{S_2 \longrightarrow b}_{P_2}\})$$

2.8 Concatenazione di linguaggi

Lemma 2. *concatenazione di linguaggi* ***La classe dei linguaggi liberi è chiusa rispetto alla concatenazione****. Cioè se L_1 ed L_2 sono liberi, allora L_3 definito come $\{w_1 w_2 \mid w_1 \in L_1 \wedge w_2 \in L_2\}$ è un linguaggio libero.*

Nota. *L'unione delle parole appartenenti alle produzioni di due linguaggi liberi produce un linguaggio libero.*

$$G_1 \quad \{aa\} \left\{ \begin{array}{l} S_1 \longrightarrow aA_1 \\ A_1 \longrightarrow a \end{array} \right\}$$

$$G_2 \quad \{bb\} \left\{ \begin{array}{l} S_2 \longrightarrow bA_2 \\ A_2 \longrightarrow b \end{array} \right\}$$

Ricorda. *Se ho due modi diversi per derivare aa vuol dire che la grammatica è* ***ambigua****, non vuol dire che non sia libera.*

Ricorda. *Per lo stesso linguaggio potremmo trovare una grammatica libera ambigua, ed una grammatica non libera e non ambigua. Come nel caso precedente devo preoccuparmi di ridenominare i simboli non-terminali in modo tale da evitare che ci siano omonimie.*

Sia G_2' la grammatica (V_2', T_2, S_2', P_2'), dove V_2', S_2', P_2' sono possibili ridenominazioni dei non terminali per evitare clash con i simboli non terminali di G_1.
Sia S nuovo tale che $S \notin V_1 \cup V_2'$.

Allora $G = (V_1 \cup V_2 \cup \{S\}, T_1 \cup T_2, S, P_1 \cup P_2' \cup \{S \longrightarrow S_1\, S_2\})$
Visto come albero di derivazione

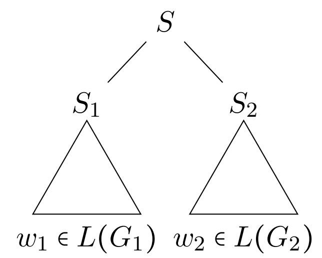


Figura 5: Albero di derivazione

Nota. *I simboli non-terminali hanno solo una funzione ausiliaria, poiché guidano le trascrizioni, ma di fatto — alla fine — scompaiono.*

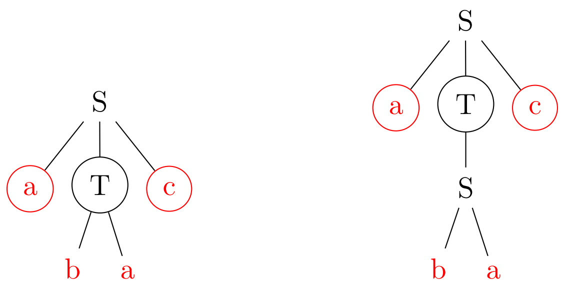
Esercizio

Determina se la grammatica è ambigua

$$
\begin{aligned}
G_i \qquad S &\longrightarrow aSc \mid aTc \mid T \\
T &\longrightarrow bTa \mid ba
\end{aligned}
$$

Ricorda. *La prima cosa da fare quando avete una grammatica è capire che linguaggio genera.*

$$L_i = \{a^n b^m a^m c^n \mid n \geqslant 0, m > 0\}$$



(a) . . . con 2 passi di derivazione

(b) . . . con 3 passi di derivazione

Figura 6: Esempio di grammatica ambigua

Nota. *Dopo un passaggio di derivazione siamo arrivati alla stessa situazione nei due alberi di derivazione.*

La grammatica G è *ambigua*: esistono cioè due derivazioni differrenti (entrmbi *left-most* o *right-most*) che portano alla stessa parola w.

Esercizio con grammatica non libera

$$
\begin{aligned}
S &\longrightarrow CD \\
C &\longrightarrow aCA \mid bCB \\
AD &\longrightarrow aD \\
BD &\longrightarrow bD \\
Aa &\longrightarrow aA \\
Ab &\longrightarrow bA \\
Ba &\longrightarrow aB \\
Bb &\longrightarrow bB \\
C &\longrightarrow \varepsilon \\
D &\longrightarrow \varepsilon
\end{aligned}
$$

$$S \xRightarrow{1} CD \xRightarrow{10} \xRightarrow{11} \varepsilon$$
$$S \xRightarrow{1} CD \xRightarrow{2} aCaD \xRightarrow{10} \xRightarrow{11} aa$$
$$S \xRightarrow{1} CD \xRightarrow{3} bCBD \xRightarrow{5} bCbD \xRightarrow{10} \xRightarrow{11} bb$$
$$S \xRightarrow{1} CD \xRightarrow{2} aCAD \xRightarrow{3} abCBAD \xRightarrow{4} abCBaD \xRightarrow{8} abCaBD \xRightarrow{5} abCabD \xRightarrow{10} \xRightarrow{11} abab$$

$$L = \{ww \mid w \text{ è una stringa sull'alfabeto } \{a, b\}\} \cup \{\varepsilon\}$$

Nota. *Non esiste una grammatica libera che genera lo stesso linguaggio.*

Esercizio

Dato la parola

$$\{abab\}$$

riuscite a generare una grammatica libera da contesto che produca lo stesso linguaggio?

$$S \longrightarrow abab$$
$$S \longrightarrow aSb \mid \varepsilon$$
$$S \longrightarrow aSa \mid \varepsilon$$

$$S \longrightarrow aSb \mid bAa$$
$$A \longrightarrow aAb$$

$$S \longrightarrow AB$$
$$A \longrightarrow aA \mid aB \mid \varepsilon$$
$$B \longrightarrow bB \mid bA\varepsilon$$

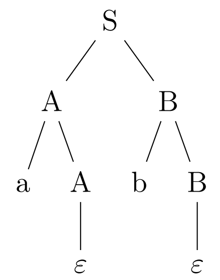


Figura 7: Albero di derivazione

Nota. *Non è un linguaggio libero, quindi non esiste una grammatica libera che lo generi. (?)*

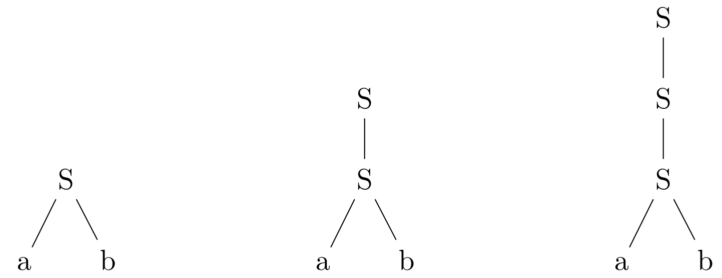
2.9 Pumping Lemma

Lemma 3 (Pumping lemma per linguaggi liberi)**.** *(ipotesi) Sia L un linguaggio libero.*
(tesi) Allora $\exists p \in \mathbb{N}^+$ (esiste una costante strettamente maggiore di zero) tale che $\forall z \in L : |z| > p$. (per ogni parola appartenente al linguaggio maggiore di quella costante)

$$
\begin{aligned}
\exists uvwxy : (z = uvwxy \wedge & \qquad \textit{esistono 5 sotto-stringhe ordinate tali che valgono le seguenti condizioni} \\
|vwx| \leqslant p \wedge & \qquad \textit{queste componenti sono strettamente maggiori di } p \\
|vx| > 0 \wedge & \qquad \textit{almeno una delle due non è } \varepsilon \\
\exists i \in \mathbb{N} : uv^i wx^i y \in L) & \qquad \textit{vale questa condizione}
\end{aligned}
$$

Dimostrazione. L è un linguaggio libero.
$\Longrightarrow$ esiste una grammatica G in *forma normale di Chowsky* tale che $L = L(G)$.
Definiamo p come la lunghezza della parola più lunga che può essere derivata usando un albero di derivazione i cui cammini dalla radice sono lunghi al più $|V \subset T|$ (il no. di simboli non-terminali della grammatica). ■

$$S \longrightarrow aSb \mid ab$$



(a) 1 passo **(b)** 2 passi **(c)** 3 passi

Figura 8: Più passi di derivazione

$$S \longrightarrow A_1 \longrightarrow A_2 \longrightarrow \ldots \longrightarrow A_k \longrightarrow a$$

Ora dimostriamo che L_1 non sia un linguaggio libero sfruttando la dimostrazione di cui sopra

$$L1 = \{ww \mid w \in a, b^*\} \text{non è libero}$$

Dimostrazione. Supponiamo che L_1 sia libero.
Sia p un no. naturale e positivo qualunque.
Sia $z = a^p b^p a^p b^p$ allora $z \in L_1, |z| > p$

$$z = \underbrace{a, \ldots, a}_{p}\underbrace{bb, \ldots, b}_{p}\underbrace{a, \ldots, a}_{p}\underbrace{b, \ldots, b}_{p}$$

Siano $uvwxy$ tali che $z = uvwxy \wedge |vwx| \wedge |vx| > 0$.

$\Longrightarrow$ distinguiamo varie possibilità

1. vwx è composto solo da a in w_1;
2. vwx contiene sia a che b in w_1;
3. vwx contiene solo b di w_1;

4. vwx contiene b di w_1 e a di w_2;
5. vwx contiene solo a di w_2;
6. vwx contiene a e b di w_2;
7. vwx contiene b di w_2.

$\Longrightarrow$ nei casi **1**, **3**, **5**, **7** considero la parola $z' = uv^0wx^0y$

1. $z' = \underline{a^k}b^pa^pb^p$ con $k < p, z' \notin L$;

3. $z' = a^p\underline{b^k}a^pb^p$ con $k < p$;

5. $z' = a^pb^p\underline{a^k}b^p$ con $k < p$;

7. $z' = a^pb^pa^p\underline{b^k}$ con $k < p$.

$\Longrightarrow$ nei casi 2, 4, 6 considero la parola $z' = uv^0wx^0y$

2. z' ha una delle tre possibili forme:
   $z' = \underline{a^k}b^pa^pb^p$ con $k < p$ oppure
   $z' = a^p\underline{b^k}a^pb^p$ con $k < p$ oppure
   $z' = a^j\underline{b^k}a^pb^p$ con $j, k < p$

analogo per 4. e 6.

■

2.10 Usi scorretti del *Pumping lemma*

Ricorda. *Se devo dimostrare la negazione della tesi del p.l. devo dimostrare un asserto che vale* $\forall p \in \mathbb{N}^+$.

Consideriamo l'equazione

$$\{ww \mid w \in \{a, b\}^*\}$$

Sia $z = a^pa^p$.
Sia $z = (ab)^p(ab)^p$.
Prendo $p = 4$

Ricorda. *In questa dimostrazione l'unica che si può scegliere è la* z*, nessun altro parametro può essere scelto.*

2.11 Esercizi

Il linguaggio $L_{17} = \{a^nb^nc^m \mid n, m > 0\}$ è libero?

Ricorda. *Alla luce delle nostre conoscenze per dimostrare che un linguaggio è libero dobbiamo trovare una grammatica che lo genera, mentre per dimostrare che* ***non*** *lo sia dobbiamo usare il pumping lemma per contraddizione.*

$$\begin{aligned} G_1: \quad & S \longrightarrow aSb \mid B \\ & B \longrightarrow cB \mid \varepsilon \end{aligned}$$

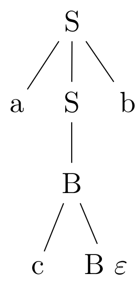


Figura 9: Albero di derivazione

$acb \notin L_{17}$

$$
\begin{aligned}
G_2: \qquad & S \longrightarrow AB \\
& A \longrightarrow aAb \mid \varepsilon \\
& B \longrightarrow Bc \mid \varepsilon
\end{aligned}
$$

$$
\begin{aligned}
G_3: \qquad & S \longrightarrow AB \\
& A \longrightarrow aAb \mid ab \mid \varepsilon \\
& B \longrightarrow Bc \mid c \mid \varepsilon
\end{aligned}
$$

è ambigua perchè non in *Chomsky normal form*

Ricorda. *Non bisogna prendere un'unica istanza di un oggetto dove è definita la quantificazione universale.*

Ricorda. *Per dire che un linguaggio non è libero usiamo il pumping lemma per contraddizione.*

2.12 Come dimostrare che un linguaggio non è libero

Dimostrazione che L_1 non è libero.

1. Supponiamo che L_1 sia libero;
2. Dimostriamo la negazione della tesi del pumping lemma per L_1;
3. Concludiamo con la frase "Questo contraddice il pumping lemma, quindi L_1 non è libero".

Ricorda. *La negazione della tesi del pumping lemma*

$$\forall p \in \mathbb{N}^+ \exists z \in L : |z| > p.$$
$$\dots$$

Ricorda. *tutto ciò che scriviamo nel punto 2 deve essere indipendente dai valori attuali di p, u, v, w, x, y (vuol dire che deve valere per valori arbitrari di p, u, v, w, x, y)*

2.13 Esercizi

$$L_1 = \{a^n b^n c^m \mid n, m \geqslant 0\} \qquad \text{LIBERO}$$

è composto dai linguaggi $\{a^n b^n \mid n \geqslant 0\} \wedge \{c^m \mid m \geqslant 0\}$.
Grammatica non ambigua di L_1

$$\begin{aligned} G_1 : \qquad S &\longrightarrow AB \mid A \mid B \mid \varepsilon \\ A &\longrightarrow aAb \mid ab \\ B &\longrightarrow cB \mid c \end{aligned}$$

$$\begin{aligned} L_2 &= \{a^m b^n c^n \mid n, m \geqslant 0\} && \text{LIBERO} \\ L_3 &= \{a^n b^n c^n \mid m \geqslant 0\} && \text{NON LIBERO} \end{aligned}$$

Nota. L_3 *non è libero per il* pumping lemma.

Supponiamo L_3 libero.
Sia $p \in \mathbb{N}^+$. Sia $z = a^p b^p c^p$.
Allora $z \in L_3, |z| > p$.

$$z = \underbrace{\overbrace{aa, \dots a}^{A}}_{p} \underbrace{\overbrace{bb, \dots b}^{B}}_{p} \underbrace{\overbrace{cc, \dots c}^{C}}_{p}$$

Siano $uvwxy$ tali che $z = uvwxy \wedge |vwx| \leqslant p \wedge |vx| > 0$ allora distinguiamo i casi:

1. vwx è composto solo da a in A;
2. vwx è composto sia da a in A che da b in B;
3. vwx è composto solo da b in B;
4. vwx è composto sia da B in B che da c in C;
5. vwx è composto solo da c in C.

$z' = uv^0wx^0y$

1. $a^k b^p c^p \qquad k < p \implies z' \notin L_3$;
2. $a^k b^j c^p \qquad k < p \vee j < p \implies z' \notin L_3$;
3. $a^p b^k c^p \qquad k < p \implies z' \notin L_3$;
4. $a^p b^k c^j \qquad k < p \vee j < p \implies z' \notin L_3$;
5. $a^p b^p c^k \qquad k < p \implies z' \notin L_3$;

2.14 Intersezione di linguaggi

Lemma 4 (intersezione)**.** *La classe di linguaggi liberi* ***non*** *è chiusa rispetto all'intersezione.*

Dimostrazione. L_1 è libero
$\wedge$
L_2 è libero
$\wedge$
L_3 non è libero. ∎

Ricorda. *L'intersezione di due linguaggi liberi può essere che non sia un linguaggio libero.*

2.15 Esercizi

$$L_4 = \{a^n b^m c^{n+m} \mid n, m > 0\}$$

è equivalente a

$$L_4 = \{a^n b^m c^n c^m \mid n, m > 0\}\text{LIBERO}$$

una grammatica che genera questo linguaggio è il seguente

$$\begin{aligned} S &\longrightarrow \underline{aSc} \mid aBc \\ B &\longrightarrow bBc \mid \underline{bc} \end{aligned}$$

che produce il seguente albero di derivazione

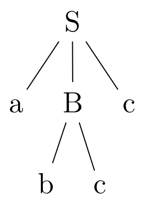


Figura 10: Albero di derivazione

Altro giro, altra corsa

$$L_5 = \{a^n b^m c^n d^m \mid n, m > 0\} \qquad \text{NON LIBERO}$$

$$L_6 = \{wcw^R \mid w \in \{a, b\}^+\}$$

w^R sta per REVERSE di w cioè la parola uguale a w letta da destra verso sinistra.

$$\begin{aligned} S &\longrightarrow aSb \mid bSa \mid a && \text{NON LIBERO} \\ S &\longrightarrow aSa \mid bSb \mid c && \text{NON LIBERO} \\ S &\longrightarrow aSa \mid bSb \mid bcb && \text{LIBERO} \end{aligned}$$