Automi e Linguaggi Formali

Parte 9 – Linguaggi non context-free

Davide Bresolin
Ultimo aggiornamento: 16 aprile 2020

Università degli Studi di Padova

Sommario

- Il pumping lemma ci dimostra che esistono linguaggi non regolari
- Vedremo un pumping lemma simile per i linguaggi regolari:
  - per ogni linguaggio context-free
  - esiste una lunghezza del pumping
  - tutte le stringhe più lunghe possono essere “iterate” (pompate)
- Useremo il lemma per dimostrare ci sono linguaggi non context-free

Pumping Lemma per CFL

Theorem (Pumping Lemma per Linguaggi Context-free)

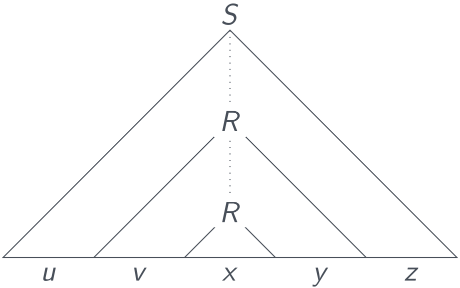
Sia L un linguaggio context-free. Allora

- *esiste una lunghezza $k \geq 0$ tale che*
- *ogni parola $w \in L$ di lunghezza $|w| \geq k$*
- *può essere spezzata in $w = uvxyz$ tale che:*
  1. $|vy| > 0$ *(il secondo o il quarto pezzo non sono la stringa vuota)*
  2. $|vxy| \leq k$ *(il blocco centrale è lungo al max k)*
  3. $\forall i \geq 0$, $uv^ixy^iz \in L$ *(possiamo "pompare" contemporaneamente v e y rimanendo in L)*

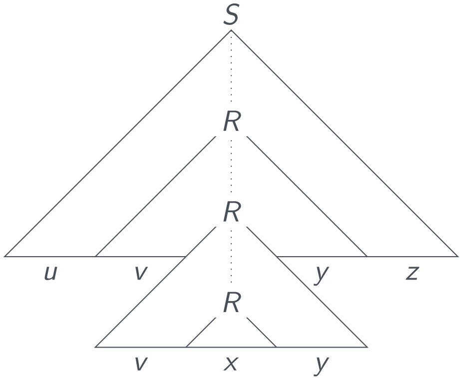
Idea di dimostrazione (1)

- Consideriamo la grammatica G che genera il linguaggio L
- Prendiamo una stringa w "molto lunga" in L
- L'albero sintattico di w deve essere "molto alto"
- Esiste un cammino nell'albero che ripete una variabile R
- Replicando il sottoalbero di R possiamo "pompare" la stringa rimanendo nel linguaggio

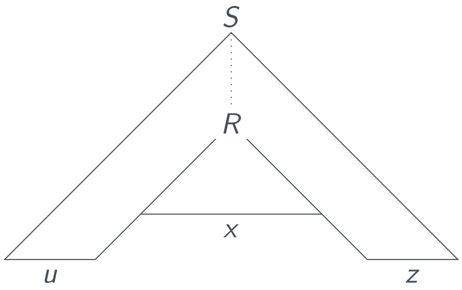
Idea di dimostrazione (2)



Idea di dimostrazione (2)



Idea di dimostrazione (2)



Uso del Pumping Lemma

- Ogni CFL soddisfa il Pumping Lemma context-free.
- Un linguaggio che falsifica il Pumping Lemma non può essere context-free:
  - per ogni lunghezza $k \geq 0$
  - esiste una parola $w \in L$ di lunghezza $|w| \geq k$ tale che
  - per ogni suddivisione $w = uvxyz$ tale che:
    1. $|vy| > 0$ (v e y non entrambi vuoti)
    2. $|vxy| \leq k$ (il pezzo centrale è lungo al max k)
  - esiste un $i \geq 0$ tale che $uv^ixy^iz \in L$ (possiamo "pompare" ed uscire da L)

Attenzione!

Esistono linguaggi non context-free che rispettano il Pumping Lemma: $\{a^ib^jc^k \mid i, j, k \text{ tutti diversi tra loro}\}$

Il Pumping Lemma come Gioco

- L'avversario sceglie la lunghezza k
- Noi scegliamo una parola w
- L'avversario spezza w in $uvxyz$
- Noi scegliamo i tale che $uv^ixy^iz \notin L$
- allora abbiamo vinto

Esempi

Per ognuno dei seguenti linguaggi, dire se rispetta il Pumping Lemma per linguaggi context-free oppure no.

- Il linguaggio $L_1 = \{a^n b^n \mid n \geq 0\}$
- Il linguaggio $L_2 = \{a^n b^n c^n \mid n \geq 0\}$
- Il linguaggio $L_3 = \{a^i b^j c^k \mid 0 \leq i \leq j \leq k\}$
- Il linguaggio $L_4 = \{ww^R \mid w \in \{0,1\}^*\}$
- Il linguaggio $L_5 = \{ww \mid w \in \{0,1\}^*\}$

Dall'idea alla dimostrazione

Idea di dimostrazione del Pumping Lemma per linguaggi context-free:

- Consideriamo la grammatica G che genera il linguaggio L
- Prendiamo una stringa w "molto lunga" in L
- L'albero sintattico di w deve essere "molto alto"
- Esiste un cammino nell'albero che ripete una variabile R
- Replicando il sottoalbero di R possiamo "pompare" la stringa rimanendo nel linguaggio

Per passare dall'idea alla dimostrazione dobbiamo stabilire cosa vuol dire che la stringa è molto lunga e perché l'albero sintattico è molto alto

Proprietà degli alberi

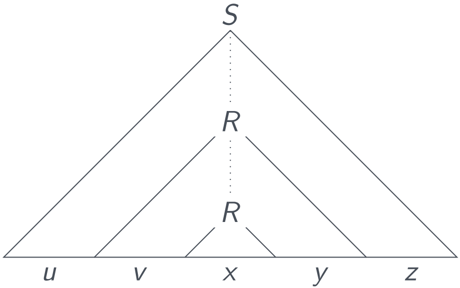
- Sia G una grammatica per il linguaggio L
- Sia b il numero massimo di simboli nel lato destro delle regole
- In un albero sintattico, ogni nodo avrà al massimo b figli:
  - al più b foglie per un albero di altezza 1
  - al più b^2 foglie per un albero di altezza 2
  - al più b^h foglie per un albero di altezza h
- Un albero di altezza h genera una stringa di lunghezza minore o uguale a b^h
- Viceversa: una stringa di lunghezza maggiore o uguale a $b^h + 1$ richiede un albero sintattico di altezza maggiore di h

Stringhe molto lunghe e alberi molto alti

- Prendiamo come lunghezza del pumping $k = b^{|V|+1}$ ($|V|$ numero di variabili in G)
- Presa una stringa $w \in L$, se $|w| \geq k$ allora ogni albero sintattico per w deve avere altezza maggiore o uguale a $|V| + 1$
- Prendiamo più piccolo albero sintattico τ per w
- Prendiamo il cammino più lungo in τ: sarà di lunghezza $\geq |V| + 2$
- Quindi ci sono almeno $|V| + 1$ variabili nel cammino
- Qualche variabile R si ripete nel cammino
- Scegliamo R in modo che si ripeta nei $|V| + 1$ nodi più in basso nel cammino

Suddividiamo w

- La variabile R ci dà la suddivisione $w = uvxyz$:
  - il sottoalbero più in alto genera vxy
  - il sottoalbero più in basso genera x



Pompiamo w

- possiamo sostituire i due sottoalberi tra loro, ottenendo di nuovo un albero sintattico corretto:
  - sostituire ripetutamente il più piccolo con il più grande ci dà uv^ixy^iz per ogni $i > 1$
  - sostituire il più grande con il più piccolo ci dà uxz
- in tutti i casi la nuova stringa appartiene a L
- rimane da dimostrare:
  - che $|vy| > 0$ (v e y non possono essere entrambi la parola vuota)
  - che $|vxy| \leq k$

- supponiamo che $v = \varepsilon$ e $y = \varepsilon$
- allora se sostituiamo il sottoalbero più grande con il più piccolo otteniamo di nuovo w:

$$w = uvxyz = u\varepsilon x\varepsilon z = uxz$$

- Assurdo: avevamo scelto τ come l'albero sintattico più piccolo!

$|vxy| \leq k$

- l'occorrenza più in alto di R genera vxy
- avevamo scelto le occorrenze tra i $|V| + 1$ nodi più in basso
- il sottoalbero che genera vxy è alto al massimo $|V| + 1$
- quindi può generare una stringa di lunghezza al più $b^{|V|+1} = k$

FINE!