9 - Esercizi

Esercizi sulla generazione

Sia dato il linguaggio $L = \{a^n b^{2n+1} | n \geq 0\}$.
Determinare una grammatica generativa per L.
Alcune parole:
L={$b, ab^3, a^2b^5, a^3b^7, ..$}

Parola più piccola possibile:
S $\rightarrow$ b
Un esempio di produzione potrebbe essere:

- S $\rightarrow$ b | aSbb
  Esempio:
  S $\implies$ aabbbbb
  S $\rightarrow$ aSbb
  S $\rightarrow$ aaSbbbb
  S $\rightarrow$ aabbbbb
  Avendo quindi la grammatica:
  G = (X, V, S, P):
  X={a,b}
  V={S}
  S
  P = {S $\rightarrow$ b | aSbb}

Sia dato il linguaggio $L = \{a^n b^{n+m} c^m | n, m > 0\}$.
Determinare una grammatica generativa per L.
Alcune parole
L = {$ab^2c, ab^3c^2, a^2b^3c, a^2b^4c^2$}
Per facilitare la comprensione possiamo riscrivere il linguaggio:
$L = \{a^n b^n b^m c^m | n, m > 0\}$
Notiamo che vi è un legame tra a e b, ma anche tra b e c. Le coppie hanno infatti tra loro lo stesso esponente. Inoltre, siccome le due coppie hanno esponente diverso, possiamo generarle indipendentemente tra loro.
Possiamo pensare dunque ad una produzione:
S $\rightarrow$ aAbbBc
A $\rightarrow$ λ | aAb
B $\rightarrow$ λ | bBc
Con la conseguente grammatica.

Un'altra possibile produzione sarebbe:

S $\rightarrow$ AB

A $\rightarrow$ ab | aAb

B $\rightarrow$ bc | bBC

Avendo quindi la grammatica:

G = (X, V, S, P):

X={a,b,c}

V={S, A, B}

S

P = {S $\rightarrow$ AB, A $\rightarrow$ ab | aAb, B $\rightarrow$ bc | bBc}

Sia dato il linguaggio $L = \{a^n b^{2k+1} | n, k \geq 0\}$.

Determinare una grammatica generativa per L.

...

Possibile produzione:

S $\rightarrow$ ABb

A $\rightarrow$ aA | λ

B $\rightarrow$ bbB | λ

...

Sia dato il linguaggio $L = \{a^n b^n c^n | n > 0\}$.

Determinare una grammatica generativa per L.

...

Possibile produzione

S $\Rightarrow$ aSBC $\Rightarrow$ aaSBCBC $\Rightarrow$ aaaSBCBCBC $\Rightarrow$ aaaaSBCBCBCBC $\Rightarrow$ aaaaaBCBCBCBCBC $\Rightarrow$ aaaaaBBCCBCBCBC $\Rightarrow$ aaaaaBBCBCCBCBC $\Rightarrow$ aaaaaBBBCCCBCBC $\Rightarrow$ aaaaaBBBCCBCCBC $\Rightarrow$ aaaaaBBBCBCCCBC $\Rightarrow$ aaaaaBBBBCCCCBC $\Rightarrow$ aaaaaBBBBCCCBCC $\Rightarrow$ aaaaaBBBBCCBCCC $\Rightarrow$ aaaaaBBBBCBCCCC $\Rightarrow$ aaaaaBBBBBCCCCC $\Rightarrow$ aaaaabBBBBCCCCC $\Rightarrow$ aaaaabbBBBCCCCC $\Rightarrow$ aaaaabbbBBCCCCC $\Rightarrow$ aaaaabbbbBCCCCC $\Rightarrow$ aaaaabbbbbCCCCC

Regole di produzione:

S $\rightarrow$ aSBC|aBC

CB $\rightarrow$ BC

aB $\rightarrow$ ab

bB $\rightarrow$ bb

bC $\rightarrow$ bc

cC $\rightarrow$ cc

Sia dato il linguaggio $L = \{a^n b^m c^n | n, m > 0\}$.
Determinare una grammatica generativa per L.

...

S $\rightarrow$ aSc | aBc

B $\rightarrow$ bB | b

---

1) Sia dato il seguente linguaggio L sull'alfabeto $X = \{0, 1\}$

$$L = \{w \in X^* \mid w = 0^n 1 0^m , m > n > 0\}$$

Determinare una grammatica G libera da contesto che generi L(G).

(PUNTI 10)

G = (X, V, S, P):

X={0,1}

V={S,A,B}

S

P = {S $\rightarrow$ A1B

A $\rightarrow$ 0A|0

B $\rightarrow$ 0B|0BB|0

}

Esercizi sul Pumping Lemma

Sia dato il linguaggio $L = \{a^n b^n c^n | n > 0\}$.
Dimostrare che L non è C.F..

L libero $\implies \exists p \in N \; \forall z \in L |z| > p$

1. $|vwx| \leq p$
2. $(vx \neq \lambda)$
3. $\forall i, \; i \geq 0 : uv^i wx^i y \in L$

Studiamo una stringa $z \in L \; |z| > p$ (scegliendo la stringa con lunghezza maggiore di p più comoda per i calcoli)

$z = a^p b^p c^p \implies |z| = 3p > p$

$$\underbrace{a \dots a}_{\text{p}}\underbrace{b \dots b}_{\text{p}}\underbrace{c \dots c}_{\text{p}}$$

In questa stringa abbiamo un p numero di a, un p numero di b ed un numero p di c.

Casi:

1. vwx formato solo da a

2. vwx formato solo da b
3. vwx formato solo da c
4. vwx formato a cavallo tra a e b
5. vwx formato a cavallo tra b e c
6. vwx non può contenere a b e c, poiché non sufficientemente lunga

Caso 1:
Prendiamo una stringa a caso (quella più semplice)
uv^2wx^2y
Prendendo la stringa pompata, possiamo aggiungere solo delle a.
Ora, il numero di a aumenta: $p+1 \leq \#(a) \leq p+p$. Tuttavia il numero di c e b rimane invariato, il che non rispecchia le regole del linguaggio. Quindi
$uv^2wx^2y \notin L \text{ poichè } \#(a) \neq \#(b) \neq \#(c)$

Caso 2 e 3:
Stessa cosa del caso 1, ripetuto però con le b e con le c.

Caso 4:
il punto vwx è a cavallo tra a e b.
Caso 4.1: $v \neq \lambda \; x = \lambda$
$v \neq \lambda \implies \text{ v contiene solo delle a}$, il che vuol dire che $wx = ab\dots b$. Pompando andremmo ad aumentare solo il numero delle a (non delle b poichè dovrebbero essere contenute nelle x, il quale è però vuoto). Anche in questo caso quindi avremmo un numero di a pari a:
$p+1 \leq \#(a) \leq p+p-1$ (p-1 poichè almeno una a è contenuta nelle w). Quindi
$uv^2wx^2y \notin L \text{ poichè } \#(a) \neq \#(b) \neq \#(c)$
Caso 4.2: $v = \lambda \; x \neq \lambda$
$x \neq \lambda \implies \text{ x contiene solo delle b}$. In questo caso quindi avremmo un numero di b pari a:
$p+1 \leq \#(b) \leq p+p-1$ (p-1 poichè almeno una b è contenuta nelle w). Quindi
$uv^2wx^2y \notin L \text{ poichè } \#(a) \neq \#(b) \neq \#(c)$
Caso 4.3: $v \neq \lambda \; x \neq \lambda$
$v \neq \lambda \quad x \neq \lambda \implies \text{ v contiene solo delle a e x contiene solo delle b}$. In questo caso quindi avremmo un numero di b pari al numero a: $p+1 \leq \#(b) = \#(a) \leq p+p-1$ (p-1 poichè almeno una b è contenuta nelle w). Quindi $uv^2wx^2y \notin L \text{ poichè } \#(a) \wedge \#(b) \neq \#(c)$

Caso 5:
Analogo al caso 4.

Come fare il pumping lemma ("algoritmo")

1. Assumiamo per assurdo che L è libero
2. Andiamo a definire una costante p e una stringa z che appartiene al linguaggio con lunghezza maggiore di p

2.1) $\exists p \in \mathbb{N} \quad \forall z \in \quad L|a| > p \quad z = uvwxy$

Nota bene: se il linguaggio è formato da 3 caratteri il pumping lemma è quello classico, se invece è di meno bisogna fare lo studio della lunghezza della stringa pompata (guardare esempi dal professore come a^{n^2})
3) Definiamo le proprietà del pumping lemma:
- $|vwx| \leq p$
- $vx \neq \lambda$
- $\forall i, \ i \geq 0 : uv^i wx^i y \in L$

Proviamo a studiare un linguaggio:
Sia $L = \{a^i b^j c^k \quad i > j > k > 0 \implies \#(a) > \#(b) > \#(c) > 0\}$.
Determinare se è libero da contesto.

Generiamo delle parole del linguaggio $L = \{a^3b^2c, a^4b^3c^2, a^4b^2c, a^5b^4c^3, \ldots\}$

4. Calcoliamo la lunghezza della stringa:
   Consideriamo la stringa: $z = a^{p+2}b^{p+1}c^p \quad |z| = 3p + p > p$
5. Consideriamo tutti i casi possibili
   Caso 1: vwx formata solo da a
   Caso 2: vwx formata solo da b
   Caso 3: vwx formata solo da c
   Caso 4: vwx formata a cavallo tra a e b
   Caso 5: vwx formata a cavallo tra b e c
   Osservazione: vwx non è abbastanza lunga per contenere a, b, c insieme
6. Studiamo i casi
   Nota bene: la stringa pompata e quella depompata hanno due formule diverse per poter essere descritte (oltre a dover essere applicate in base alle regole dettate dal linguaggio).

- **Pompare**: $p + 1 \leq \#(carattere) \leq p + p$
- **Depompare**: $p - p \leq \#(carattere) \leq p - 1$

Caso 1) Studiamo uv^0wx^0y
(Perchè studiamo ora la de-pompata? Dalla regola, se le a non sono maggiori di b e c sappiamo che non rispettiamo le regole del linguaggio)
$uv^0wx^0y \rightarrow$

- $p - p \leq \#(a) \leq p - 1$
- $\#(b) = p$
- $\#(c) = p$

$uv^0wx^0y \notin L$ poiché $\#(a) \leq (\#(b), \#(c))$

Caso 2) Uguale al caso 1 (ovviamente all'esame bisogna ricopiarlo)

Caso 3) uv^2wx^2y

(Qui conviene pompare la stringa, poiché nella de-pompata dimostreremmo esattamente il contrario di quello che vogliamo smentire, cioè il numero delle c non dovrebbe essere superiore a quello delle a)

$uv^2wx^2y \rightarrow$

- $\#(a) = p$
- $\#(b) = p$
- $p + 1 \leq \#(c) \leq p + p$

$uv^2wx^2y \notin L$ poiché $\#(c) \leq (\#(a), \#(b))$

Caso 4) Creiamo diversi casi

Caso 4.1): $v \neq \lambda \; x = \lambda$; v contiene solo delle a (de-pompiamo per poter dimostrare che togliendo il numero delle a andiamo a rompere la regola):

$uv^0wx^0y \rightarrow$

- $p - p \leq \#(a) \leq p + 1$
- $\#(b) = p + 1$
- $\#(c) = p \Rightarrow \#(a) \leq \#(b) \Rightarrow uv^0wx^0y \notin L$

Caso 4.2: $v = \lambda \; x \neq \lambda$; v contiene solo delle b:

$uv^0wx^0y \rightarrow$

- $\#(a) = p + 2$
- $p - p \leq \#(b) \leq p$
- $\#(c) = p$
  $\Rightarrow \#(b) \leq \#(c)$
  $\Rightarrow uv^0wx^0y \notin L$

Caso 4.3: $v \neq \lambda \; x \neq \lambda$; v contiene solo delle a e x contiene solo delle b.

$uv^0wx^0y \rightarrow$

- $\#(a) = p + 2 - |v|$
- $\#(b) = p + 1 - |x|$
- $\#(c) = p$

$\Rightarrow \#(a), \#(b) \leq \#(c) \Rightarrow uv^0wx^0y \notin L$

Caso 5) Creiamo diversi casi

Caso 5.1): $v \neq \lambda \; x = \lambda$; v contiene solo delle b (de-pompiamo per poter dimostrare che togliendo il numero delle b andiamo a rompere la regola):

$uv^0wx^0y \rightarrow$

- $\#(a) = p + 2$
- $p - p \leq \#(b) \leq p$

- $\#(c) = p$
  $\Rightarrow \#(b) \leq \#(c) \Rightarrow uv^0wx^0y \notin L$

Caso 5.2): $v = \lambda$ $x \neq \lambda$; x contiene solo delle c (pompiamo per aumentare il numero delle c e rompere la regola):
$uv^2wx^2y \rightarrow$

- $\#(a) = p + 2$
- $\#(b) = p + 1$
- $p + 1 \leq \#(c) \leq p + p$
  $\Rightarrow \#(c) \geq \#(b) \Rightarrow uv^2wx^2y \notin L$

Caso 5.3): $v \neq \lambda$ $x \neq \lambda$; v contiene solo delle b e x contiene solo delle c.
$uv^2wx^2y \rightarrow$

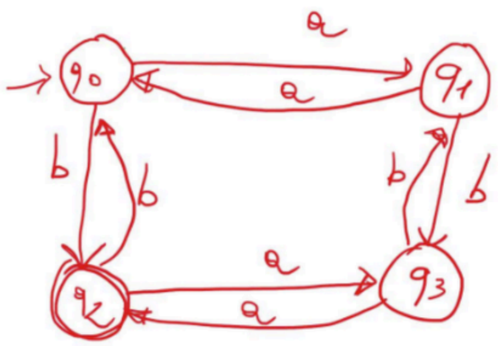
- $\#(a) = p + 2$
- $\#(b) = p + 1 + |v|$
- $\#(c) = p + |x|$
  $\Rightarrow \#(b) \geq \#(a)$ oppure $\#(c) \geq \#(b) \Rightarrow uv^2wx^2y \notin L$

7. Conclusioni: Concludiamo che il linguaggio non è C.F.

Esercizi su automi

$L = \{ w \in \{a,b\}^* \mid w$ ha num pari a e num dispari $b \}$

$L = \{ b, a^2b, a^4b, a^6b, \dots, b^3, a^2b^3, a^4b^3, \dots \}$



q_0 riconosce le w con $P(a)\ P(b)$
q_1 $D(a)\ P(b)$
q_2 $P(a)\ D(b)$
q_3 $D(a)\ D(b)$

$Q = \{q_0, q_1, q_2, q_3\}$

$M = (Q, \delta, q_0, F)$

$\delta: Q \times X \rightarrow Q$ $\quad F = \{q_2\}$

$\delta(q_0, a) = q_1$

$\delta(q_0, b) = q_2$

Gerarchia di Chomsky

Linguaggio C.F. o C.S:

Conviene generare la grammatica o applicare il pumping lemma

Lineari destri:

Bisogna costruire l'automa