

# Linguaggi e Traduttori – Tempo: 2 ore

Prof. Marco Gavanelli

25 luglio 2017

## Esercizio 1 (3 punti)

Si consideri il linguaggio  $L = \{a^n b^n | n > 0\} \cup \{a^+ b^+\}$ .

Si classifichi il linguaggio secondo Chomsky.

Si scriva una grammatica non ambigua che genera il linguaggio  $L$ ; si fornisca la grammatica di livello più basso possibile nella classificazione secondo Chomsky (intendendo il livello 3 come minimo e il livello 0 come massimo).

Se è possibile, si mostri l'albero di derivazione della stringa  $aaab$ .

## Esercizio 2 (6 punti)

Si consideri la grammatica  $G = \langle \{a, b, c, d\}, \{S, A, B, C\}, P, S \rangle$ , dove:

$$P = \begin{array}{l} S \rightarrow ABC \\ A \rightarrow aAd \mid \epsilon \\ B \rightarrow bBd \mid \epsilon \\ C \rightarrow c \mid dA \end{array}$$

1. Si classifichi la grammatica secondo Chomsky.
2. La grammatica è LL(1)? Se sì, si scriva la parsing table del PDA riconoscitore. Se no, si motivi il perché.
3. Se possibile, si scriva una grammatica  $G'$  equivalente a  $G$  che non abbia produzioni con la stringa vuota. Se non è possibile, si motivi il perché.
4. Qualora al punto 2 si sia ottenuto un riconoscitore, si mostri il riconoscimento delle stringhe  $daadd$ ,  $adc$ ,  $abdd$ , mostrando l'evoluzione dello stack.

## Esercizio 3 (3 punti)

Sia data la seguente grammatica:  $G = \langle \{a, b\}, \{A, B, C\}, P, A \rangle$ , dove

$$P = \begin{array}{l} A \rightarrow bB \mid aC \\ B \rightarrow a \mid aA \mid bC \\ C \rightarrow b \mid aC \end{array}$$

Si scriva un'espressione regolare equivalente, motivando i passaggi.

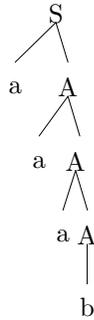
## Soluzione 1

Si può notare che l'insieme  $\{a^n b^n | n > 0\}$  è un sottoinsieme di  $\{a^+ b^+\}$ , quindi il linguaggio  $L$  può essere semplicemente descritto come  $L = \{a^+ b^+\}$ .

Il linguaggio è regolare (tipo 3), può essere infatti generato dalla grammatica  $G = \langle \{a, b\}, \{S, A, B\}, P, S \rangle$

$$P = \begin{aligned} S &\rightarrow aA \\ A &\rightarrow aA \mid bB \mid b \\ B &\rightarrow b \mid bB \end{aligned}$$

Siccome la grammatica è regolare, è anche context-free, per cui si può scrivere l'albero di derivazione:



## Soluzione 2

1. La grammatica è di tipo 2 (context-free).
2. Per verificare se la nuova grammatica è LL(1) e scrivere la parsing table, scriviamo gli insiemi FIRST e FOLLOW dei simboli non terminali

	<i>FIRST</i>	<i>FOLLOW</i>
<i>S</i>	$\{a, b, c, d\}$	$\{\$ \}$
<i>A</i>	$\{a, \epsilon\}$	$\{b, c, d, \$ \}$
<i>B</i>	$\{b, \epsilon\}$	$\{c, d\}$
<i>C</i>	$\{c, d\}$	$\{\$ \}$

La parsing table risulta:

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>\$</i>
<i>S</i>	$S \rightarrow ABC$	$S \rightarrow ABC$	$S \rightarrow ABC$	$S \rightarrow ABC$	
<i>A</i>	$A \rightarrow aAd$	$A \rightarrow \epsilon$	$A \rightarrow \epsilon$	$A \rightarrow \epsilon$	$A \rightarrow \epsilon$
<i>B</i>		$B \rightarrow bBd$	$B \rightarrow \epsilon$	$B \rightarrow \epsilon$	
<i>C</i>			$C \rightarrow c$	$C \rightarrow dA$	

Non presenta conflitti, quindi  $G$  è LL(1).

3. La gramamtica originaria:

$$P = \begin{array}{l} S \rightarrow ABC \\ A \rightarrow aAd \mid \epsilon \\ B \rightarrow bBd \mid \epsilon \\ C \rightarrow c \mid dA \end{array}$$

I non terminali che possono risciversi nella stringa vuota sono  $A$  e  $B$ .  
Togliamo le regole  $A \rightarrow \epsilon$  e  $B \rightarrow \epsilon$  e sostituiamo  $A$  e  $B$  rispettivamente con  $(A \mid \epsilon)$  e  $(B \mid \epsilon)$ :

$$P = \begin{array}{l} S \rightarrow (A \mid \epsilon)(B \mid \epsilon)C \\ A \rightarrow a(A \mid \epsilon)d \\ B \rightarrow b(B \mid \epsilon)d \\ C \rightarrow c \mid d(A \mid \epsilon) \end{array}$$

distribuendo:

$$P = \begin{array}{l} S \rightarrow ABC \mid BC \mid AC \mid C \\ A \rightarrow aAd \mid ad \\ B \rightarrow bBd \mid bd \\ C \rightarrow c \mid dA \mid d \end{array}$$

Riconoscimento:

Input	Stack
daadd\$	S\$
daadd\$	ABC\$
daadd\$	BC\$
daadd\$	C\$
daadd\$	C\$
daadd\$	dA\$
aadd\$	A\$
aadd\$	aAd\$
add\$	Ad\$
add\$	aAdd\$
dd\$	Add\$
dd\$	dd\$
d\$	d\$
\$	\$

Stringa riconosciuta.

Input	Stack
adc\$	S\$
adc\$	ABC\$
adc\$	aAdBC\$
dc\$	AdBC\$
dc\$	dBC\$
c\$	BC\$
c\$	C\$
c\$	c\$
\$	\$

Stringa riconosciuta.

Input	Stack
adbdd\$	S\$
adbdd\$	ABC\$
adbdd\$	aAdBC\$
dbdd\$	AdBC\$
dbdd\$	dBC\$
bdd\$	BC\$
bdd\$	bBdC\$
dd\$	BdC\$
dd\$	dC\$
d\$	C\$
d\$	dA\$
\$	A\$
\$	\$

stringa riconosciuta

### Soluzione 3

Riscriviamo le produzioni della grammatica in termini di equazioni

$$\begin{cases} A = bB + aC \\ B = a + aA + bC \\ C = b + aC \end{cases}$$

Eliminiamo la ricorsione diretta per  $C$  e sostituiamo per  $C$ :

$$\begin{cases} A = bB + aa^*b \\ B = a + aA + ba^*b \\ C = a^*b \end{cases}$$

Sostituiamo  $A$  nell'equazione per  $B$ :

$$\begin{cases} A = bB + aa^*b \\ B = a + abB + aaa^*b + ba^*b \end{cases}$$

Eliminiamo la ricorsione diretta per  $B$ :

$$\begin{cases} A = bB + aa^*b \\ B = (ab)^*(a + aaa^*b + ba^*b) \end{cases}$$

Sostituiamo in  $A$ :  $L(A) = b(ab)^*(a + aaa^*b + ba^*b) + aa^*b$