Corso di Linguaggi di Programmazione
Prova scritta del 5 luglio 2018.

Tempo a disposizione: ore 2.
Svolgere gli esercizi 1-4 e 5-8 su due fogli differenti.

1. Si consideri l'espressione regolare $b^*(a|b)$. Si costruisca l'automa NFA M associato, secondo la costruzione vista a lezione. Si trasformi l'NFA M nell'equivalente DFA M', secondo la costruzione per sottoinsiemi vista a lezione. M' è un DFA minimo?

2. Costruire il più semplice automa che riconosca il linguaggio $\{a^{3k+2} \mid k \geq 0\}$. È regolare tale linguaggio?

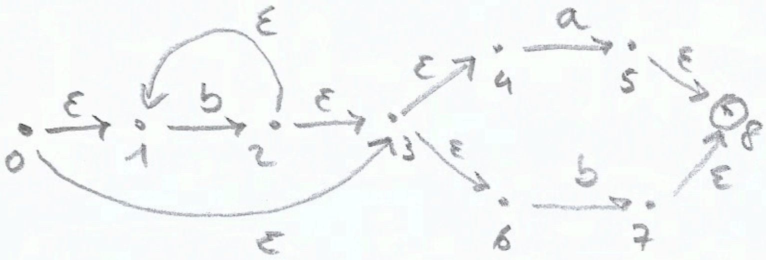
3. Si consideri la grammatica G con simbolo iniziale S:

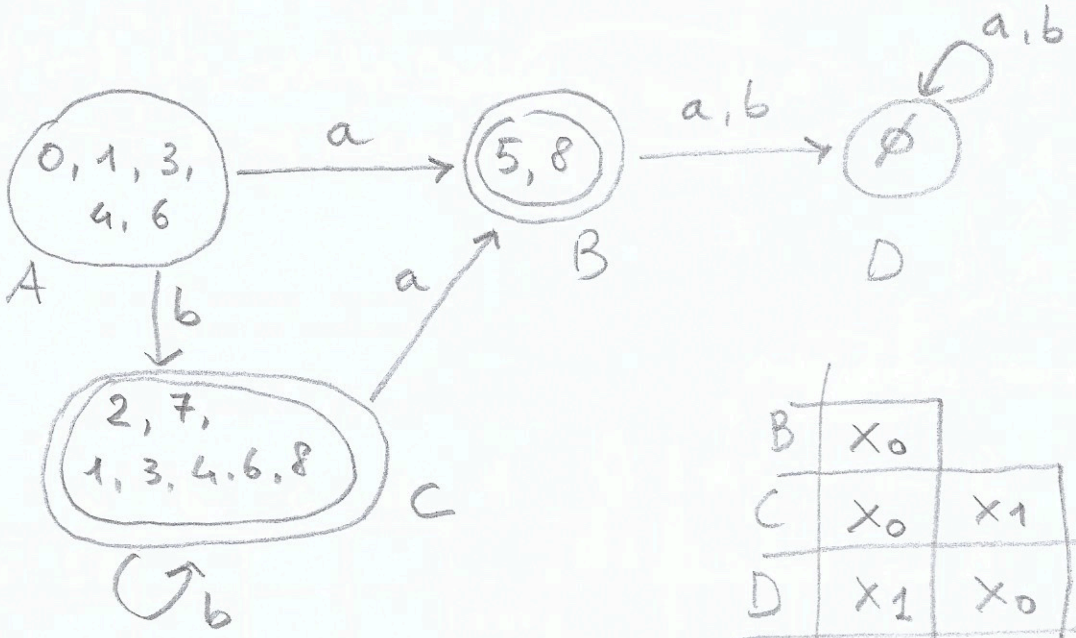
$$\begin{array}{lcl} S & \rightarrow & \mathtt{a}S\mathtt{b} \mid B \mid \epsilon \\ B & \rightarrow & \epsilon \mid \mathtt{c}B \end{array}$$

(*i*) Quale linguaggio genera G? (*ii*) G è ambigua? In caso affermativo, manipolarla per renderla non ambigua. (*iii*) Manipolare G per ottenerne una equivalente senza produzioni unitarie.

4. Si costruisca un parser bottom-up per il linguaggio $L = \{ab, abc, abd\}$ e si mostri il suo funzionamento su input abc. È possibile costruire un parser LL(1) per L?

1) $b^*(a|b)$





B	X_0		
C	X_0	X_1	
D	X_1	X_0	X_0
	A	B	C

$\Rightarrow$ è minimo

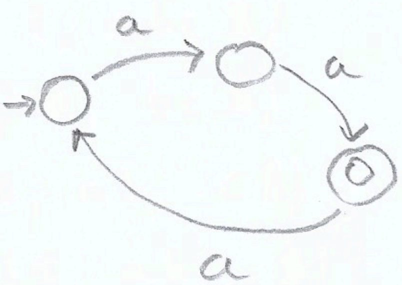
---

2) $L = \{ a^{3k+2} \mid k \geq 0 \}$

$= \{a^2, a^5, a^8, a^{11} \dots\}$

$aa(aaa)^*$

$\Rightarrow$ L è regolare



3)

$$\left.\begin{array}{l} S \to aSb \mid B \mid \varepsilon \\ B \to \varepsilon \mid cB \end{array}\right] G$$

- $L(G) = \{ a^n c^m b^n \mid n \geq 0, m \geq 0 \}$

- G è ambigua

$$\begin{array}{c} S \\ | \\ \varepsilon \end{array} \qquad \begin{array}{c} S \\ | \\ B \\ | \\ \varepsilon \end{array}$$

due alberi di derivazione diversi per ε!

- G' non ambigua

$$\begin{array}{l} S \to aSb \mid B \\ B \to \varepsilon \mid cB \end{array}$$

- G'' da G senza prod. unitarie

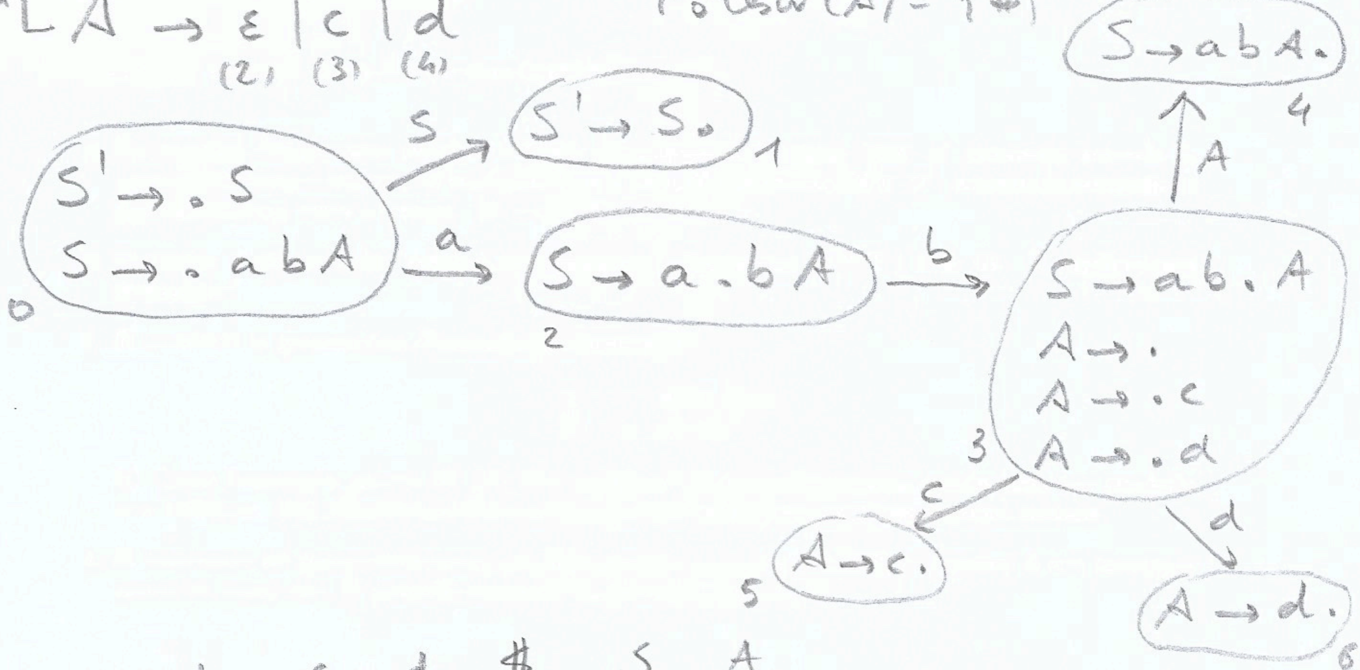
$$\left.\begin{array}{l} S \to aSb \mid \varepsilon \mid cB \\ B \to \varepsilon \mid cB \end{array}\right]$$

N.B. G'' non è ambigua

1) $L = \{ab, abc, abd\}$

$$G \left[\begin{array}{l} S \to abA \quad (1) \\ A \to \varepsilon \mid c \mid d \quad (2)\ (3)\ (4) \end{array}\right.$$

Follow(S) = {\$}
Follow(A) = {\$}



	a	b	c	d	\$	S	A
0	S2					G1	
1					Acc		
2		S3					
3			S5	S6	R2		G4
4					R1		
5					R3		
6					R4		

(0, ε, abc\$)
(02, a, bc\$)
(023, ab, c\$)
$(02\underbrace{35}, ab\underbrace{c}, \$) \to A$
$(0\underbrace{234}, \underbrace{abA}, \$) \to S$
(01, S, \$)
Acc

È possibile costruire un parser LL(1) per L perché L è regolare e tutti i linguaggi regolari sono di classe LL(1)
(comunque G è LL(1))