SOLUZIONI APPELLO 2016-01

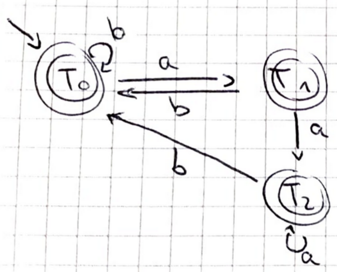
1.
NO.
NON esiste un automa stati finiti in grado di riconoscere L_1.

2.
NO.
È infatti possibile scegliere $z=a^pb^pa^pb^p$ per dimostrare la validità del NEGATO del Pumping Lemma.

3.
SÌ.

4.
4.

	a	b
$T_0 = \{A, B, C, E\}$	$\{D,E\} = \boxed{T_1}$	$\{E, A, B, C\} = T_0$
$T_1 = \{D, E\}$	$\{E\} = \boxed{T_2}$	$\{B, A, C, E\} = T_0$
$T_2 = \{E\}$	T_2	T_0



$\boxed{Q = Q_0[\![ab]\!] = \{A, B, C, E\}}$

5.
$P_0 = P[[abab]] = \{B\}$.

6.
NO.

7.
follow(B) = {a,c,\$}.

8.
8.

	a	b	c	\$
B	$B \to \varepsilon$		$B \to \varepsilon$	$B \to \varepsilon$

9.
Sia T la tabella di parsing SLR(1) per G_1.
$T[I_7,a]$ contiene un conflitto REDUCE/REDUCE.
Le riduzioni coinvolte sono:
-r4: $A \rightarrow \varepsilon$;
-r5: $B \rightarrow \varepsilon$.

10.
$J_{11} = J_0[[Aa]] = \{[S \rightarrow Aa\cdot B, \{\$\}],$
$[B \rightarrow \cdot, \{\$\}]\}.$

PARTE B.

$states_K(G) \leq 2^{|G|^{K+1}} \rightsquigarrow$ AUTOMA LR(K)

[1] $2^{|G|} = \boxed{2^6} (= 64)$

[2] $2^{|G|^2} = \boxed{2^{36}}$

[3]
È facile osservare come i body delle produzioni 'A→ab' e 'B→abb' differiscano per un solo terminale 'b' (in coda).
Supponendo di aver consumato dall’input la sequenza 'aab' e che il prossimo carattere in lettura sia 'b', il parser non riuscirà a decidere se ridurre sulla produzione 'A→ab' o spostarsi (con shift) verso uno stato dove sarà invece possibile ridurre su 'B→abb'.
Per questo motivo, la grammatica certamente NON è LR(1).