Formal Languages and Compilers Theory test September 14th, 2021

1. Find a regular expression that represents the complement of the language generated by the following grammar:

   $S \rightarrow 0\,S \mid 1$

2. Write a context-free grammar that generates the context-free language on the alphabet {0,1,(,)} made of all the non-empty strings having balanced non-empty parentheses, i.e. for each left parenthesis that occurs in the string there is a single distinct right parenthesis that occurs after it in the string, and in between each left and right corresponding parentheses there is at least one symbol.
   For example, 01(001(1))0, (0), 110, and (10)(0) are in the language, while 01(0011))0, (), and) are not.

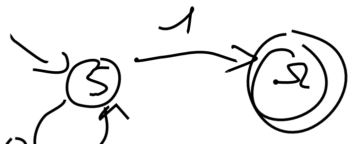
3. Build the LR(1) parsing table for the following grammar

   $S \rightarrow 0\,S\,1 \mid A$
   $A \rightarrow 0\,A \mid \varepsilon$

   Is the grammar LR(1)? Explain why.
   Is the language generated by the grammar a deterministic CFL? Explain why.

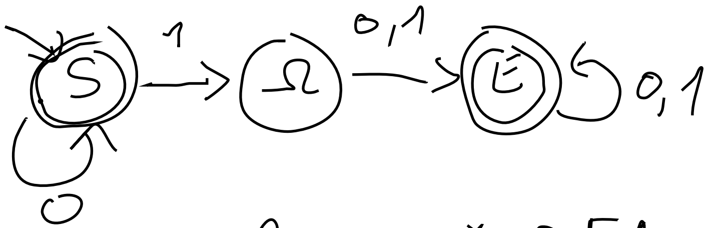
1. Find a regular expression that represents the complement of the language generated by the following grammar:

$S \rightarrow 0S \mid 1$

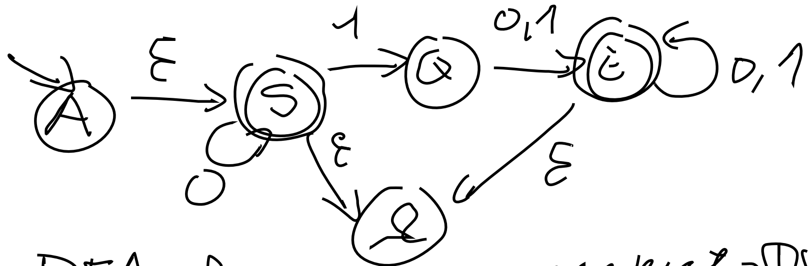


DFA

Completely specified DFA

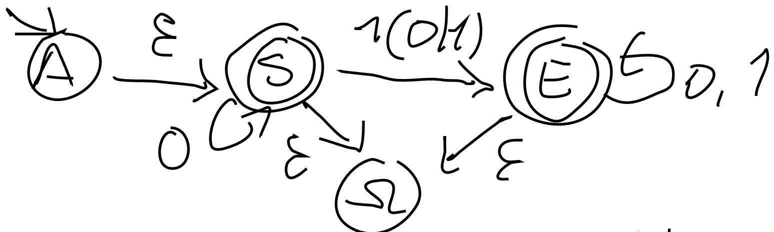


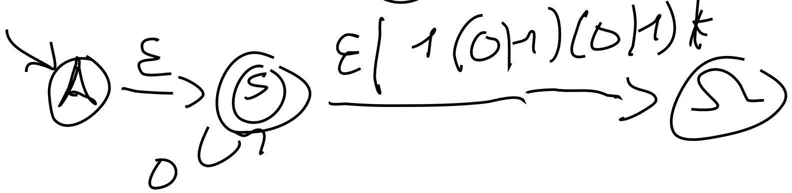
complement DFA



DFA for R.E. computation (Ω renamed to Q)

state elimination:





$\Longrightarrow \; 0^*(\varepsilon \mid 1(0|1)(0|1)^*)$

2. Write a context-free grammar that generates the context-free language on the alphabet {0,1,(,)} made of all the non-empty strings having balanced non-empty parentheses, i.e. for each left parenthesis that occurs in the string there is a single distinct right parenthesis that occurs after it in the string, and in between each left and right corresponding parentheses there is at least one symbol.
For example, 01(001(1))0, (0), 110, and (10)(0) are in the language, while 01(0011))0, (), and) are not.

$S \to 0S \mid 1S \mid S0 \mid S1 \mid (S) \mid 0 \mid 1$

3. Build the LR(1) parsing table for the following grammar

S → 0 S 1 | A

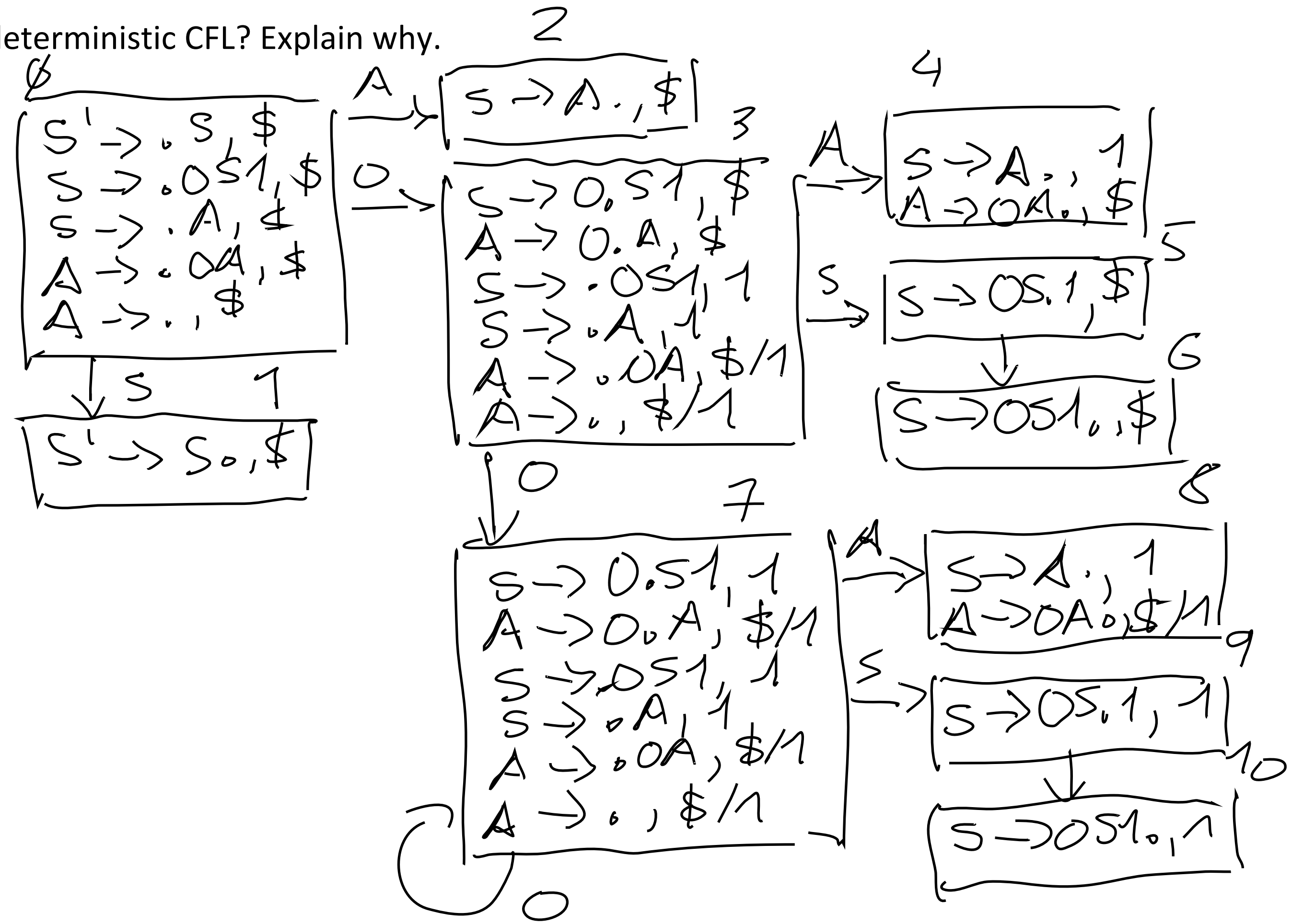
A → 0 A | ε

Is the grammar LR(1)? Explain why.

Is the language generated by the grammar a deterministic CFL? Explain why.

Add rule

S' -> S	0
S -> 0S1 | A	1, 2
A -> 0A | ε	3, 4



	0	1	$	S	A
0	S3		r4	1	2
1			acc		
2			r2		
3	S7	r4	r4	5	4
4		r2	r3		
5		S6			
6			r1		
7	S7	r4	r4	9	8
8		r2/r3	r3		
9		S10			
10		r1			

The grammar is not LR(1) because the merging table has a conflict. For the same reason, the language is not deterministic