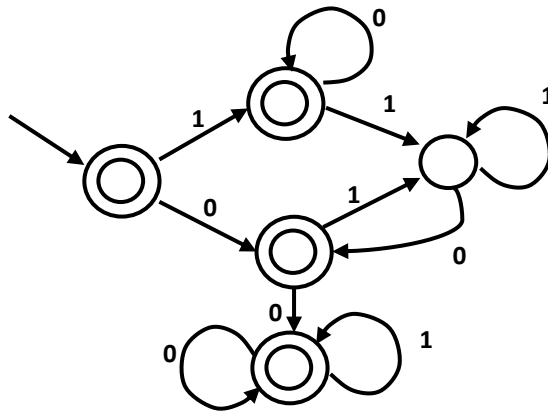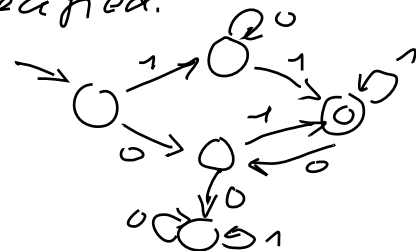1. Find a regular expression that represents the complement of the language represented by the following DFA:
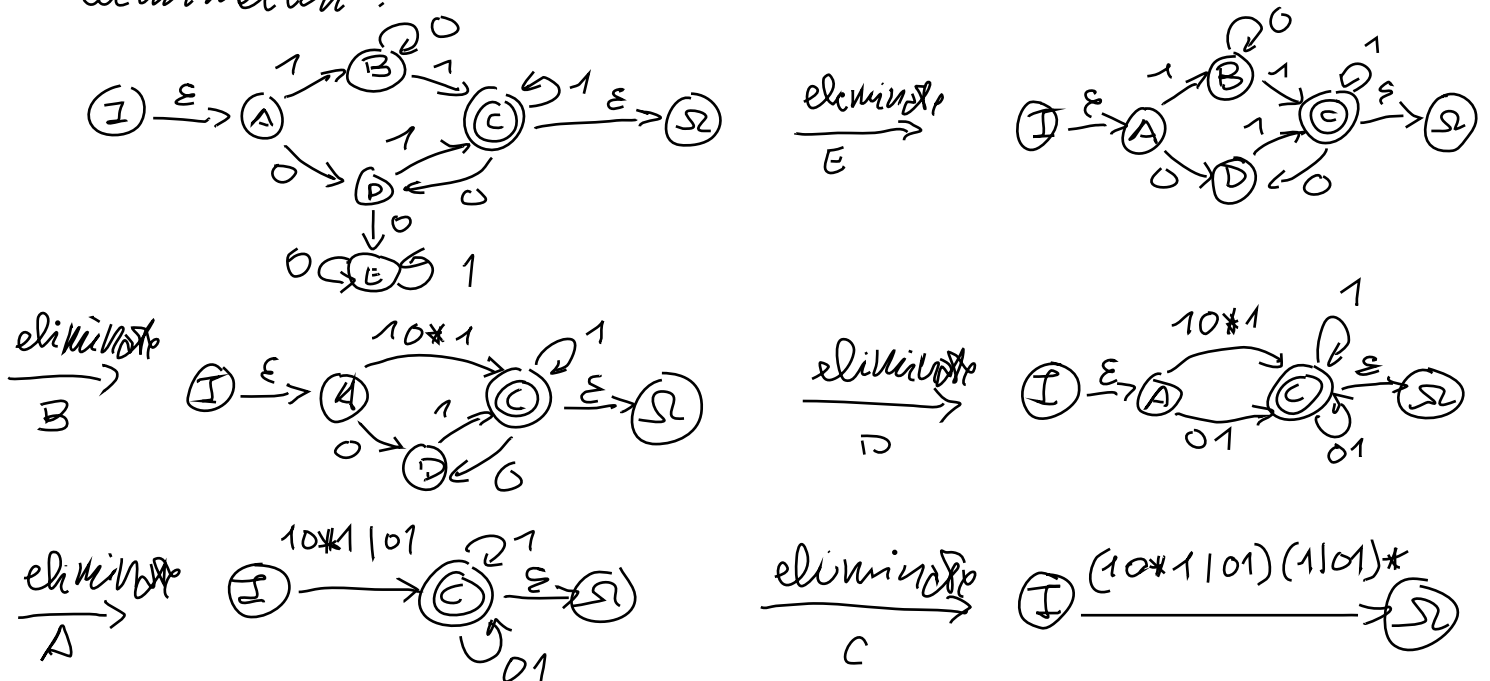

The DFA is already completely specified.
The complement DFA is then

An equivalent regular expression can be obtained by state elimination:

The regular expression is $(10^*1 \mid 01)(1 \mid 01)^*$

2. Write a CFG that generates the union of the following two languages:
   - $\{ 0^n 1^n \mid n > 0 \}$
   - The language of all the strings made of an odd number of 0 followed by an odd number of 1

$$S \rightarrow A \mid BC$$
$$A \rightarrow 0A1 \mid 01$$
$$B \rightarrow 0B0 \mid 0$$
$$C \rightarrow 1B1 \mid 1$$

.

3. Given the following grammar, with terminal symbols {a,c}, find the LR(0) parsing table for it.
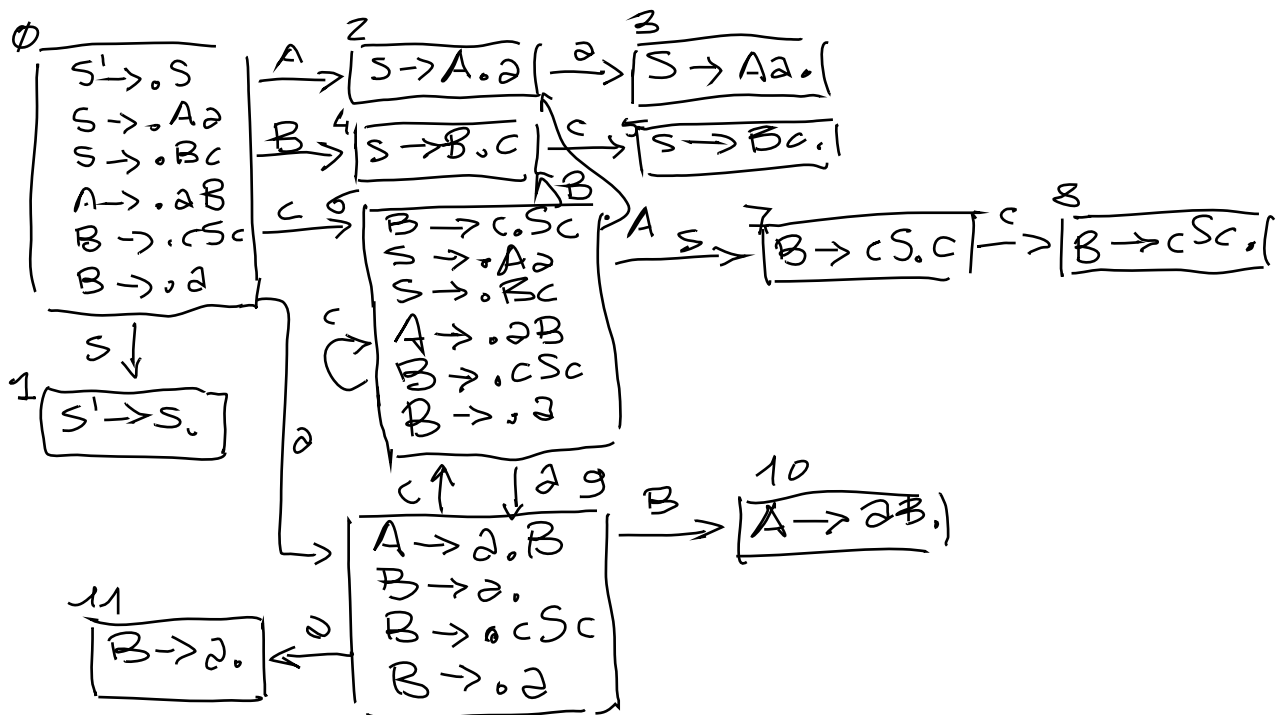
$S \rightarrow Aa \mid Bc$ 1,2
$A \rightarrow aB$ 3   ← rule numbering
$B \rightarrow cSc \mid a$ 4,5

Is the grammar LR(0)? Is it SLR? Motivate your answers

we add n.t. S' and rule $S' \rightarrow S$ (rule 0)

LR(0) parsing table :

	a	c	$	A	B	S
0	s9	s6		2	4	1
1			acc			
2	s5					
3	r1	r1	r1			
4		s5				
5	r2	r2	r2			
6	s9	s6		2	4	7
7		s8				
8	r4	r4	r4			
9	sh/r5	s6/r5	r5			10
10	r3	r3	r3			
11	r5	r5	r5			

As the table contains conflicts, the grammar is NOT LR(0). In order to know if it is SLR we have to compute FOLLOW(B) = {c,a}. As FOLLOW(B) includes the non-terminal symbols for which there are the conflicts, the conflicts remain and the grammar is NOT SLR.

4. In what case is a context-free grammar said to be ambiguous? Is the ambiguity of a context-free grammar decidable? What is the meaning of decidability?

A CFG is ambiguous if there is at least one string in its language having two different parse trees.

The ambiguity of a CFG is not decidable.

Decidability of a problem means there is an algorithm that can always solve it (i.e., for any inputs).