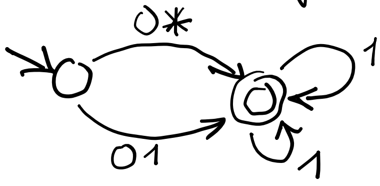
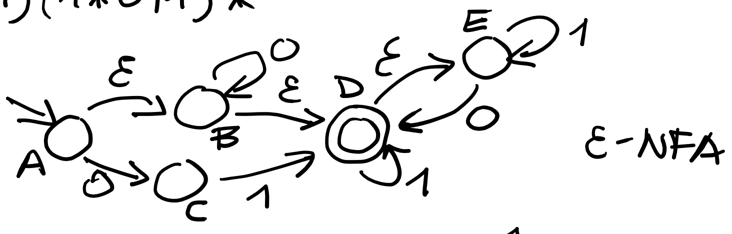
Formal Languages and Compilers Theory test July 17, 2023

1. (9 points) Check if the regular expression (0*|01)(1*0|1)* is equivalent to (0|1).

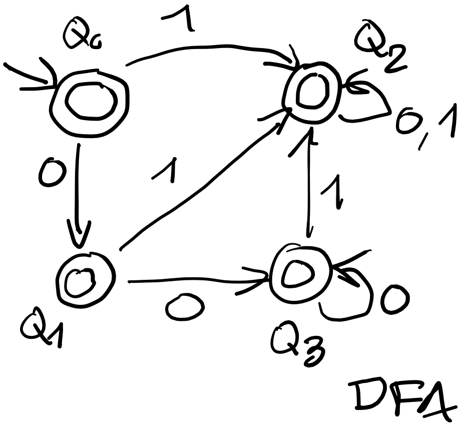
Find DFA for (0*|01)(1*0|1)*





DFA construction

		0	1
Q_0	{A,B,D,E} *	{B,D,E}	{D,E}
Q_1	{C,B,D,E} *	{B,D,E}	{D,E}
Q_2	{D,E} *	{D,E}	{D,E}
Q_3	{B,D,E} *	{B,D,E}	{D,E}



As the DFA has all states final, it is equivalent to (0|1)*, not to (0|1).

An alternative solution is to build the DFA for (0|1) and show that it is not equivalent to the DFA of the other expression.

2. (8 points) Write the definition of a context-free grammar that generates the language

$L = \{ 0^m 1^n 0^{n-m} \mid m \geq 0 , n > m\}$

Let $n - m = k$ (as $m \geqslant 0$ and $n > m$, $k > 0$)

$\Rightarrow L = \{ 0^m 1^{m+k} 0^k \mid m \geqslant 0, k > 0 \}$

$0^m 1^{m+k} 0^k = 0^m 1^m 1^k 0^k$

$S \to AB$

$A \to 0A1 \mid \varepsilon$

$B \to 1B0 \mid 10$

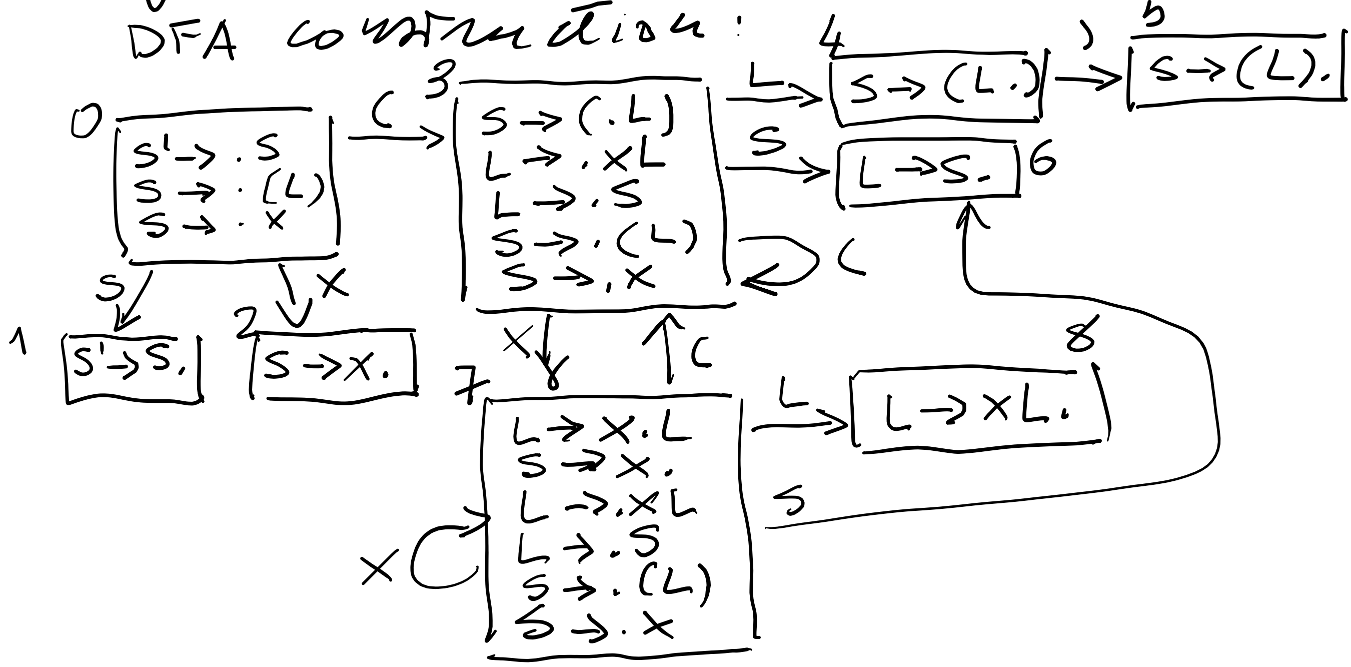
3. (9 points) Build the SLR parsing table for the following grammar whose set of terminal symbols is { x , (,) } and whose start symbol is S

S → (L) | x   (1, 2)
L → x L | S   (3, 4)

Tell if this grammar is SLR or not and tell if this grammar is LR(0) or not. Motivate your answers.

As S has 2 rules we add a new start symbol S' and rule S'->S (0)

DFA construction:



	nullable	FIRST	FOLLOW
S	F	(, x	$,)
L	F	(, x	)

	x	(	)	$	S	L
0	s2	s3			1	
1				acc		
2			r2	r2		
3	s7	s3			6	4
4			s5			
5			r1	r1		
6			r4			
7	s7	s3	r2	r2	6	8
8			r3			

The given grammar is SLR because the SLR parsing table has no conflict.
However it is not LR(0) because the LR(0) parsing table would have shift/reduce conflicts in state 7.

4. (6 points) Explain why it is not possible to obtain a deterministic bottom-up parser for an ambiguous grammar.

If a grammar is ambiguous, an input string may have more than one parse tree. For that reason, a bottom-up parser for that grammar does not have a unique way to build the parse tree for any input string, which produces conflicts in the parsing table and makes the parser nondeterministic. Only if we choose a priori which of the alternative parse trees is the right one and we exclude the others (conflict resolution) the parsing process can be made deterministic.