Mangiotis Arcis 00046001

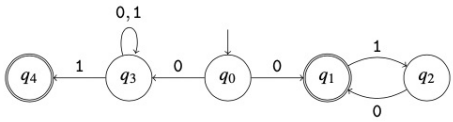
MA1- INFO

Language theory and Compiling

January 2021

Question 1 — 3 points

Here is a finite automaton on the alphabet $\Sigma = \{0, 1\}$:



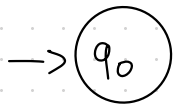
1. Is this automaton an ε-NFA? an NFA? a DFA? Justify your answers.
2. If the automaton is not a DFA, convert it into an equivalent DFA. Explain the technique you have used to perform this conversion and give the intermediate steps.
3. Convert this automaton into an equivalent regular expression, using the state elimination technique. Give the intermediate steps of the construction.

1. This automaton is a NFA

- There are some states that can have more than one transition when reading a symbol (ex. q_0 has two states when reading a 0. we can go to state q_1 or q_3)

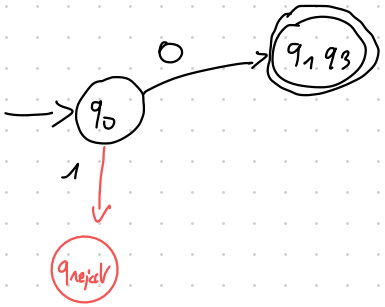
This is not a ε-NFA since there is no ε transition

2)



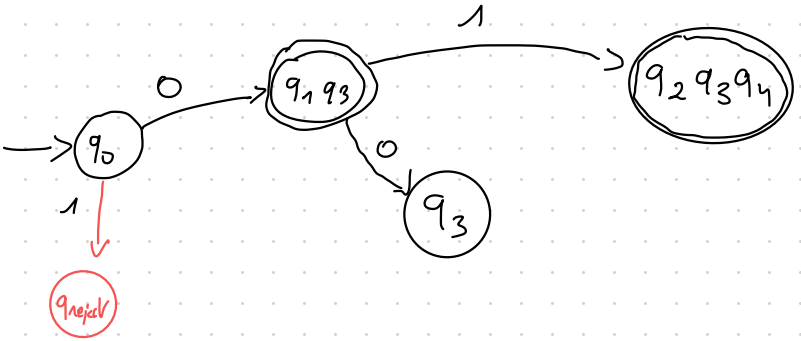
read a 1 -> reject

read a 0 -> go to $q_1 q_3$



read a 0 -> go to q_3

read a 1 -> go to $q_2 q_3 q_4$



from $q_2 q_3 q_4$

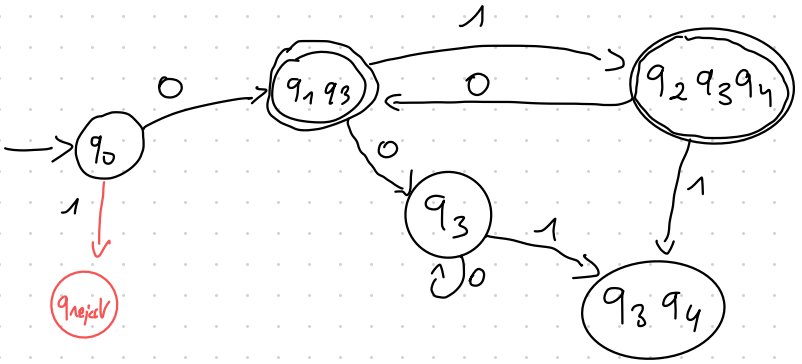
read a 0 -> $q_1 q_3$

read a 1 -> $q_3 q_4$

from q_3

read a 0 -> q_3

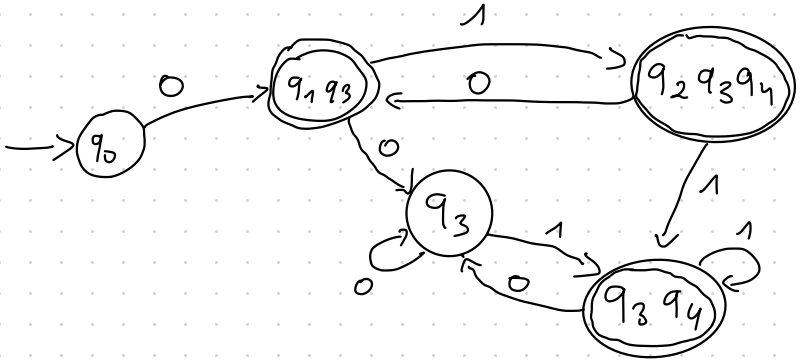
read a 1 -> $q_3 q_4$



from $q_3 q_4$

read a 0 -> q_3

read a 1 -> $q_3 q_4$



it is a DFA

(q reject state is not shown)

3) Regular expression

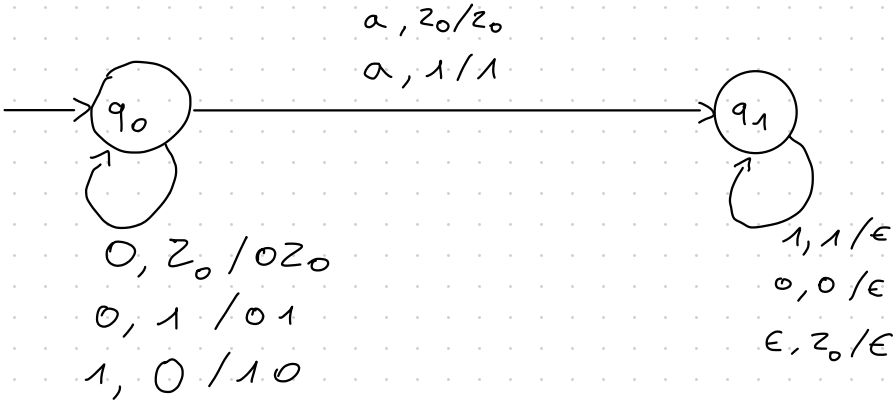
No time

Q2) I had no time for this question

Question 3 — 3 points

Give the diagram of a *deterministic* pushdown automaton, on the alphabet $\Sigma = \{0, 1, \mathtt{a}\}$, that accepts the language $L = \{(01)^n \mathtt{a} (10)^n \mid n \geq 0\}$ using the empty stack acceptance condition. Then, give the run of the automaton on the word `0101a1010` and explain why it is accepting.

- The language L is the language of palindromic sequence with 01 and 10. (we cannot accept 11 a 11 for instance)



- Execution on 0101a1010

$(q_0, 0101a1010, z_0) \vdash (q_0, 101a1010, 0z_0)$

$\vdash (q_0, 01a1010, 10z_0) \vdash (q_0, 1a1010, 010z_0)$

$\vdash (q_0, a1010, 1010z_0) \vdash (q_1, 1010, 1010z_0)$

$\vdash (q_1, 010, 010z_0) \vdash (q_1, 10, 10z_0)$

$\vdash (q_1, 0, 0z_0) \vdash (q_1, \epsilon, z_0) \vdash (q_1, \epsilon, \epsilon)$

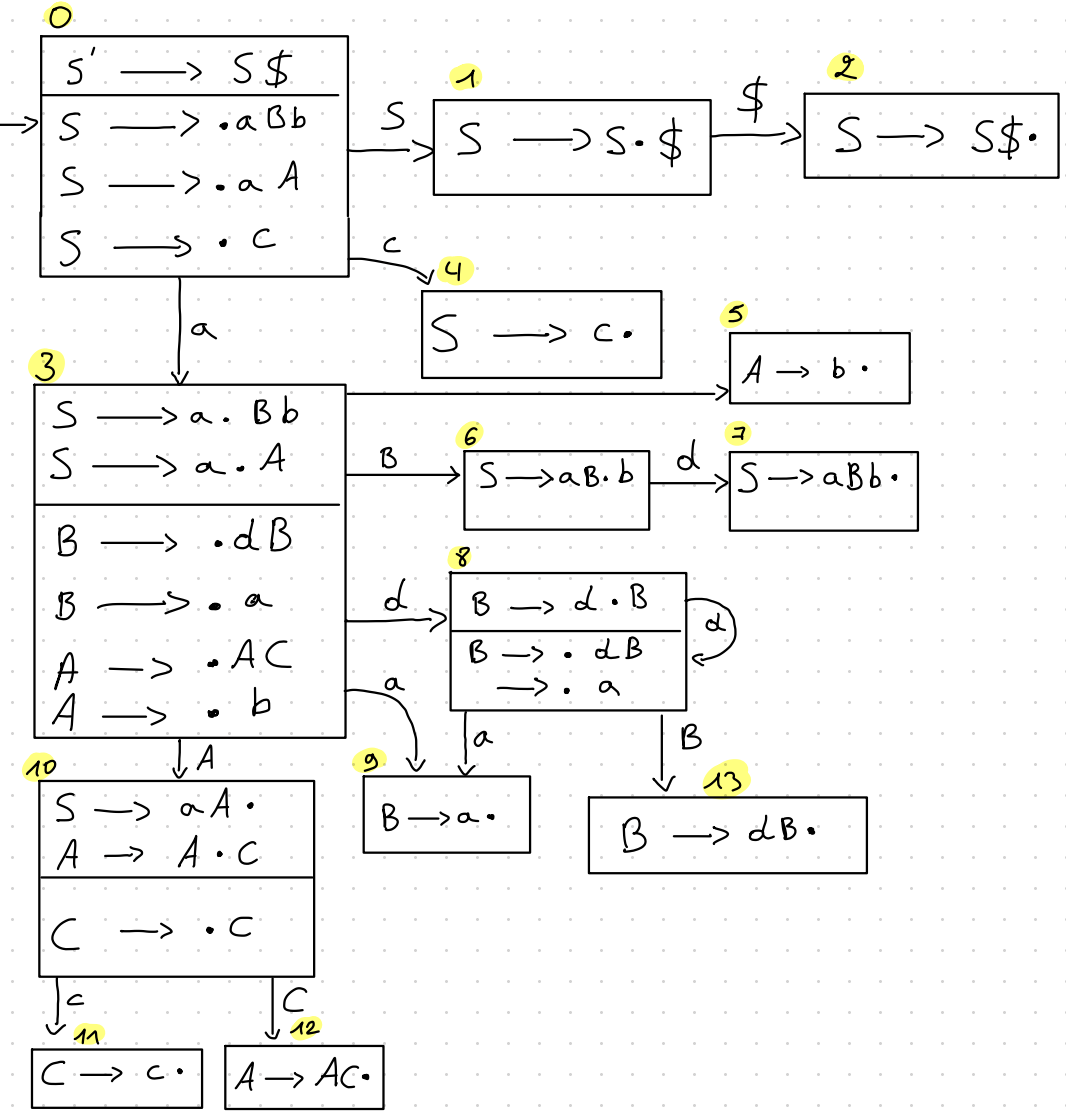
It is accepting because the stack is empty and we've read all the symbols

Question 4 — 3 points

Give the LR(0) canonical finite state machine (CFSM) of the following grammar (where the set of variables is $\{S', S, A, B, C\}$ and the set of terminals is $\{a, b, c, d\}$). **Important**: In your answer, make sure to clearly mark the items that are part of the kernel and those that are part of the closure, as we have done in the practicals and in the lectures. This is to ensure that you have actually built the answer by hand, and not used a tool to do it for you!

(1)	S'	$\rightarrow$	$S\$$
(2)	S	$\rightarrow$	aBb
(3)		$\rightarrow$	aA
(4)		$\rightarrow$	c
(5)	B	$\rightarrow$	dB
(6)		$\rightarrow$	a
(7)	A	$\rightarrow$	AC
(8)		$\rightarrow$	b
(9)	C	$\rightarrow$	c

Is the grammar LR(0)? Is it SLR(1)? Justify your answers by giving action tables without conflicts when your answer is positive, or by pointing out the conflict(s) when your answer is negative.



The grammar is not LR(0) because in this action table here below: there is a conflict

Action	
0	S
1	S
2	Accept
3	S
4	R_2
5	R_8
6	S
7	R_2
8	S
9	R_6
10	S, R_3
11	R_9
12	R_7
13	R_5

conflict