Assignment 3
ϵ-NFA and RE

Josefin Ulfenborg
940806-5960
yunalescca@gmail.com

2017-04-25

1 Task 1

In order to convert the ϵ-NFA into a DFA I'll need to eliminate all ϵ-transitions. As stated in the lecture notes, given an ϵ-NFA E = (Q_E, Σ, δ_E, q_E, F_E), the DFA D will be defined as D = (Q_D, Σ, δ_D, q_D, F_D), with definitions as in the slides.

The first thing I will have to do is to compute the ϵ-closure of each state in the transition table. This will help me define each attribute of the DFA. In order to do this I will for each state see to which states I can go to by reading nothing. That is, by following the arcs labeled ϵ. Also, for each state, the state itself will always be in its own ϵ-closure. Also, since the empty set is a subset of the NFA, I will have to include that as well and simply use the definition in order to define its ϵ-closure.

- ECLOSE($\varnothing$) = $\varnothing$
- ECLOSE($\{q_0\}$) = $\{q_0\}$, because if we read nothing from the start state it will take us to the empty set.
- ECLOSE($\{q_1\}$) = $\{q_1, q_4\}$
- ECLOSE($\{q_2\}$) = $\{q_1, q_2, q_4\}$
- ECLOSE($\{q_3\}$) = $\{q_3, q_5\}$
- ECLOSE($\{q_4\}$) = $\{q_4\}$
- ECLOSE($\{q_5\}$) = $\{q_5\}$

Now that this is done, I can resume my task to define the DFA. The first thing I will do is define the starting state.

- $q_D = ECLOSE(\{q_E\}) = ECLOSE(\{q_0\}) = \{q_0\}$.

Now I will move on to defining the delta transitions. As said in lecture 7, this construction is similar to the subset construction. The difference now is that I need to ϵ-close the set of states after each step. So just as quick example, in order to define $\delta(\{q_0\}, 0)$, I'll have to see where I can go from q_0 by reading a 0, **and** ϵ-close it (which in this case will be $\{q_0, q_1, q_4\}$. This is where I'll have use of the ϵ-closures defined above.

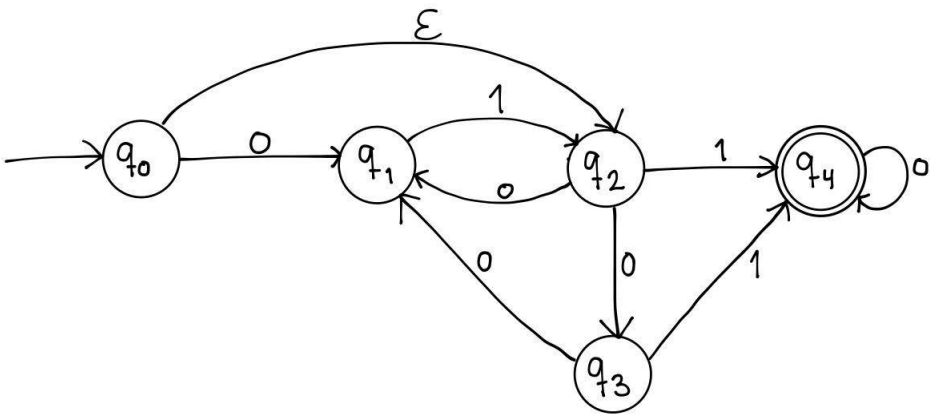
The way I'll gather this information is to put it in a transition table at once, and fill each row in the table when a new set of states appear when transitioning. I will mark the final states with a star as usual, where the final states will be any state containing at least one of the accepting states in E.

	0	1
$\rightarrow \{q_0\}$	$\{q_0, q_1, q_4\}$	$\{q_1, q_2, q_4\}$
$\{q_0, q_1, q_4\}$	$\{q_0, q_1, q_4, q_5\}$	$\{q_1, q_2, q_3, q_4, q_5\}$
$\{q_1, q_2, q_4\}$	$\{q_1, q_2, q_4, q_5\}$	$\{q_1, q_3, q_4, q_5\}$
$\star\{q_0, q_1, q_4, q_5\}$	$\{q_0, q_1, q_4, q_5\}$	$\{q_1, q_2, q_3, q_4, q_5\}$
$\star\{q_1, q_2, q_3, q_4, q_5\}$	$\{q_1, q_2, q_4, q_5\}$	$\{q_1, q_3, q_4, q_5\}$
$\star\{q_1, q_2, q_4, q_5\}$	$\{q_1, q_2, q_4, q_5\}$	$\{q_1, q_3, q_4, q_5\}$
$\star\{q_1, q_3, q_4, q_5\}$	$\{q_4, q_5\}$	$\{q_1, q_3, q_4, q_5\}$
$\star\{q_4, q_5\}$	$\{q_4, q_5\}$	$\{q_3, q_5\}$
$\star\{q_3, q_5\}$	$\{q_5\}$	$\{q_3, q_5\}$
$\star\{q_5\}$	$\{q_5\}$	$\varnothing$

2 Task 2

The regular expression that's given is $(01 + 010)^\star 1(\epsilon + 0)^\star$. What this means is that from the starting state, you can either read a 0 followed by a 1 zero or more times, or read a 0 followed by a 1 followed by a 0 zero or more times, *or* read nothing.

No matter what you choose to read (or not read) from the starting state, exactly one 1 has to follow afterwards. After you have read this 1, you should be in an accepting state, where you can either choose to continue reading nothing, or you can read zero or more 0's. Below is my solution of the automaton.

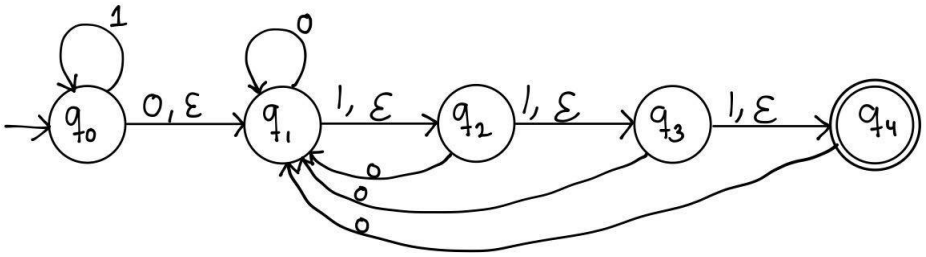


The way I have designed this is that from the starting state you have the option of reading nothing and moving on to q_2.Or if you do not go along this path, you have to read a 0 followed by a 1 and since both 01 and 010 start with 01, I chose to only have that one path from q_0. After you have read 01, you either have the option of reading the mandatory 1 which will bring you to an accepting state, or read more 01's by going back to q_1. And this is where I will use non-determinism, because ,a part from reading the 1, you can either read a 0 which will bring you back to q_1, or you can read a 0 which will take you to q_3 and in this way you have now read 010 instead. And in order to continue reading 010's, or 01's for that matter, you will go back to q_1 where you can again choose to either read 01 or 010. And if you do not want to read any more of either of these, you can read a 1 and go to the accepting state.

3 Task 3

Now the assignment is to construct a regular expression over $\Sigma = \{0, 1\}$ that accepts those words that contains at most three consecutive 1's after each 0. The language in that case consists of words such as ϵ, 1, 11,111,..., 10,100,...,101,1011,10111,101110,...

The way I solved this problem is that I first designed an automaton with the same condition, and from there created the regular expression, since it made it easier to see if the solution was correct. Here is the automaton I designed:



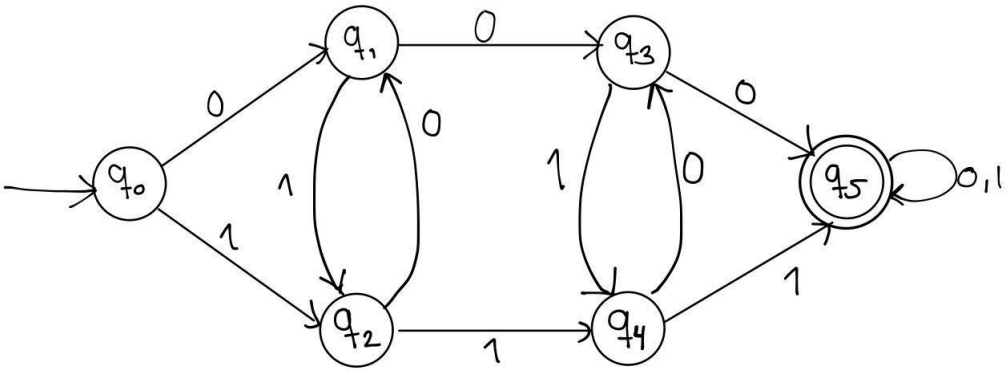
From the starting state, before you have read any 0, you are allowed to read as many 1's as you want (including none at all). After this, you either move on by reading nothing or reading a 0. If you read a 0, you are still allowed to continue to read 0's, or start reading more 1's. Here, however, you are only allowed to read three *consecutive* 1's, which means you can either read none up to three 1's, or/and at any time go back to read more 0's.

This will give me the regular expression

$$1^\star(0^\star(1+\epsilon)(1+\epsilon)(1+\epsilon))^\star$$

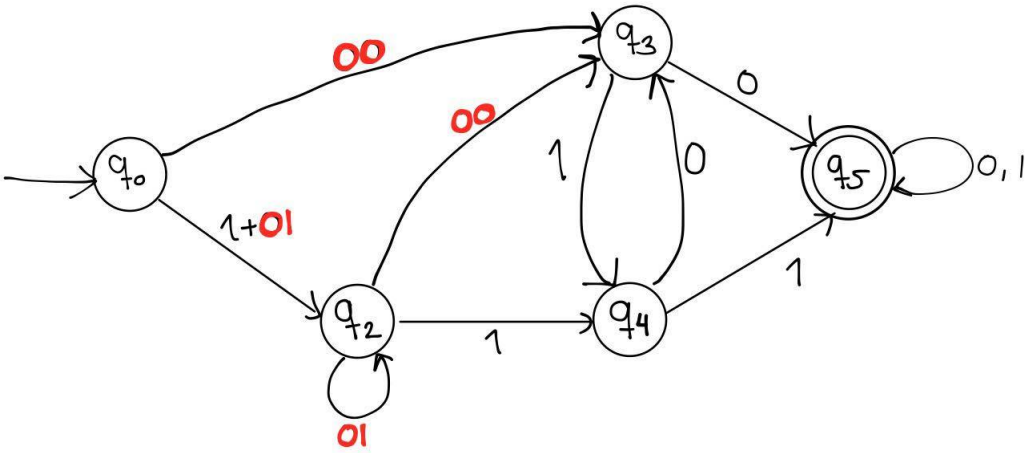
4 Task 4

a) Here is how the automata looks like from the start:



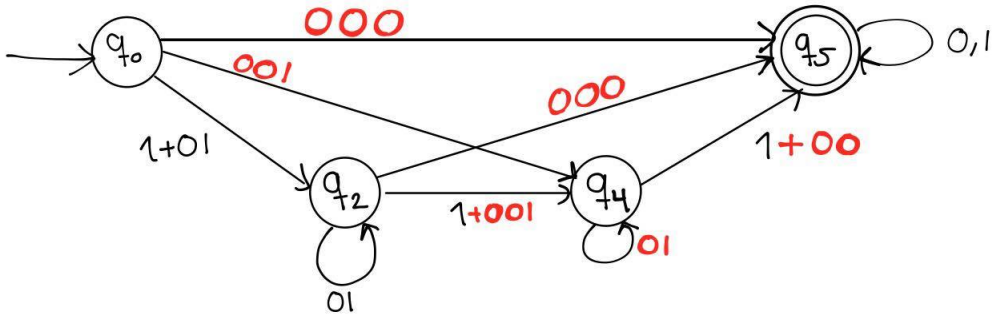
After removing q_1, a few arcs will be removed as well and need to be replaced. I still need to be able to get from q_0 to q_3, from q_2 back to itself, from q_0 to q_2 through q_1, and from q_2 to q_4 through q_1 , which means I will add new arc between the mentioned states, with a regular expression as the label on the arc. And without listing all of the changes, one change I made on the already existing arc between q_0 and q_2 was to change the label from just **1** to **1 + 01**. This means I can either go to state q_2 from q_0 by reading 1, or I can read a 0 followed by a 1.

All labels I added are marked with red, however the arcs are still in black.

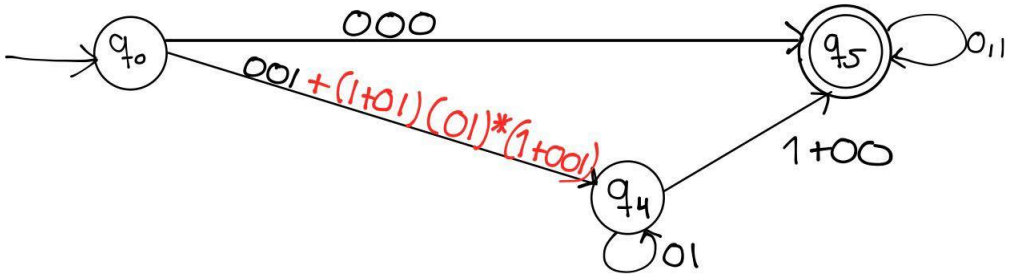


The next state to remove is q_3, which means I now need to add new arcs

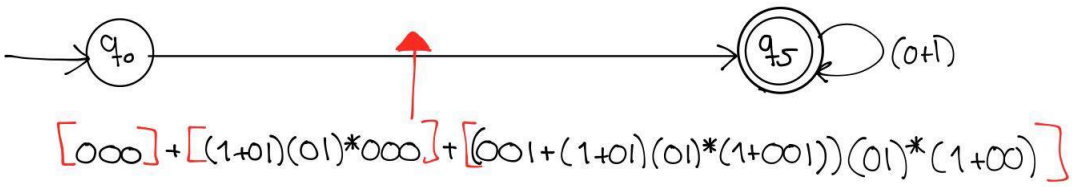
between q_0 and q_5, q_0 and q_4, q_4 to itself and q_2 to q_5. Besides from this I also make changes to some of the already existing arcs.



After removing q_2 I need to add an arc between q_0 and q_4. This arc will have a rather long label, since these are all of the possibilities to from the starting state to q_4.



Lastly I remove q_4, meaning I'm only left with one single arc between the starting and the accepting state. This label I put a bit below the arc in order to fully fit the whole name. And just as before, all the options are created from following each path and writing down the possible ways to take. I put the different options in brackets to better distinguish them.



That is, the resulting regular expression is the following:

$$\Big(000 + (1+01)(01)^\star 000 + \Big(001 + (1+01)(01)^\star(1+001)\Big)(01)^\star(1+00)\Big)(0+1)^\star$$

b) This time I'm supposed to eliminate states by solving the equations. Firstly, the equations I have are the following:

- $E_0 = 0E_1 + 1E_2$
- $E_1 = 0E_3 + 1E_2$
- $E_2 = 0E_1 + 1E_4$
- $E_3 = 0E_5 + 1E_4$
- $E_4 = 0E_3 + 1E_5$
- $E_5 = 0E_5 + 1E_5 + \epsilon = (0+1)E_5 + \epsilon$

From the slides: A solution to $X = RX + S$ is $X = R^\star S$. I begin by solving E_5, and if (0+1) is R, E_5 is X and ϵ is S, then the solution is

$$E_5 = (0+1)^\star\epsilon = (0+1)^\star$$

and by eliminating this, I need to replace this expression with whenever I find E_5 in another equation. This gives me only a change to E_3 and E_4

- $E_3 = 0(0+1)^\star + 1E_4$
- $E_4 = 0E_3 + 1(0+1)^\star$

Now I move on to eliminating E_1, which results E_0 and E_2 having to be rewritten as well:

- $E_0 = 0(0E_3 + 1E_2) + 1E_2 = 00E_3 + (01+1)E_2$
- $E_2 = 0(0E_3 + 1E_2) + 1E_4 = 00E_3 + 01E_2 + 1E_4$

Now I eliminate E_3, which can be found in the remaining E_0, E_2, E_4.

- $E_0 = 00(0(0+1)^\star + 1E_4) + (01+1)E_2 = 000(0+1)^\star + (01+1)E_2 + 001E_4$
- $E_2 = 00(0(0+1)^\star + 1E_4) + 01E_2 + 1E_4 = 000(0+1)^\star + (001+1)E_4 + 01E_2$
- $E_4 = 0(0(0+1)^\star + 1E_4) + 1(0+1)^\star = 00(0+1)^\star + 01E_4 + 1(0+1)^\star = (00+1)(0+1)^\star + 01E_4$

The next step is to solve E_2, which I can do with the solution $X = R^\star S$.

$E_2 = (01)^\star(000(0+1)^\star + (001+1)E_4) = (01)^\star 000(0+1)^\star + (01)^\star(001+1)E_4$

And now eliminating it, replacing the above expression for E_2 in E_0. Since E_4 does not have any term with E_2 it remains unchanged.

- $E_0 = 000(0+1)^\star + 001E_4 + (01+1)((01)^\star 000(0+1)^\star + (01)^\star(001+1)E_4)$ and through some simplifications this gives me

  $E_0 = 000(0+1)^\star + (01+1)(01)^\star 000(0+1)^\star + ((01+1)(01)^\star(001+1) + 001)E_4$

Now I solve E_4

- $E_4 = (01)^\star(00+1)(0+1)^\star$

And finally I eliminate E_4 from E_0 and by doing so, solving the equation.

- $E_0 = 000(0+1)^\star + (01+1)(01)^\star 000(0+1)^\star + ((01+1)(01)^\star(001+1) + 001)(01)^\star(00+1)(0+1)^\star$

And by simplifications you can get this expression to look exactly as the one in a) by first doing distributivity in reverse and factoring out the $(0+1)^\star$ term. After this is done it is easy to see that the expressions are equivalent by looking at the 001 term, because if you either factor it out from b), or use distributivity in a), then you would get one or the other.