Lecture L&M 2

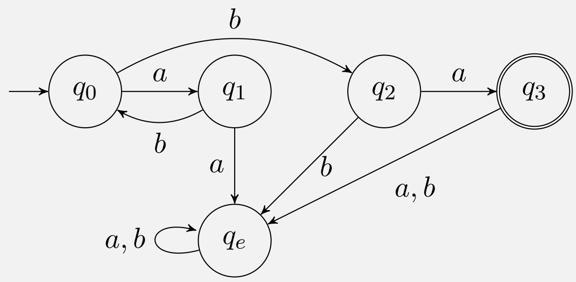
Non-deterministic Finite Automata

2.1 Construct DFA from RE

For each of the following regular expressions E, draw the state diagram of a DFA that accepts $\mathcal{L}(E)$.

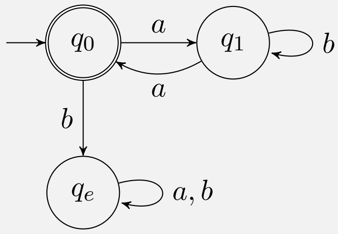
A. $(ab)^*ba$

A complete automaton:



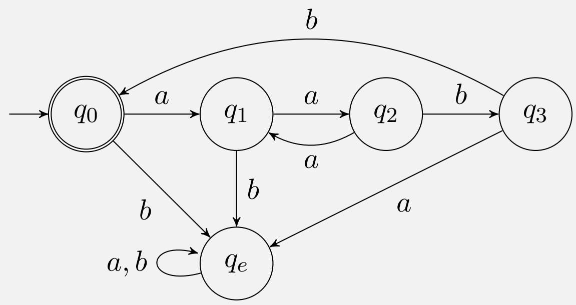
B. $(ab^*a)^*$

A complete automaton:



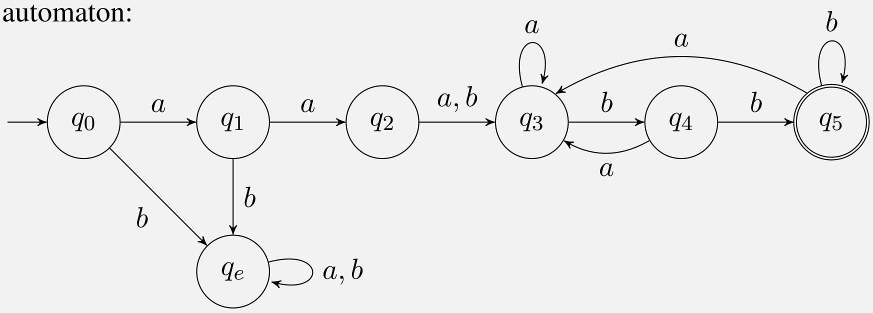
C. $((aa)^+bb)^*$

A complete automaton:



D. $aa(a \cup b)^+bb$

A complete automaton:



Hint: Do only two of these during the tutorial and save the others for practice at home or for the exam. You can use the construction from the lecture (from RE to NFA-λ, then from NFA-λ to DFA), but for these small regular expressions, it may be easier to try to come up with a DFA directly.

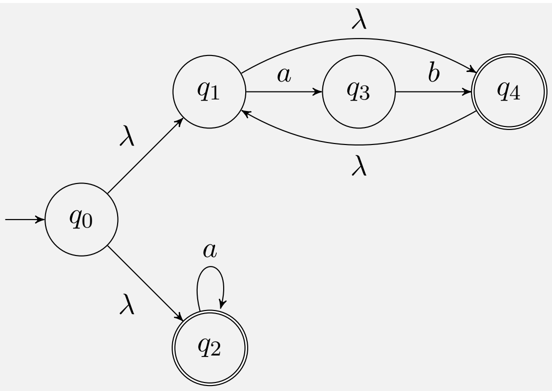
2.2 From RE to NFA-λ

For each of the following regular expressions E, use the construction from the lecture to derive the state diagram of an NFA-λ that accepts $\mathcal{L}(E)$. You may omit obviously unnecessary λ-steps.

The solutions are inspired by the construction from the lecture, but leave out some unnecessary λ-steps.

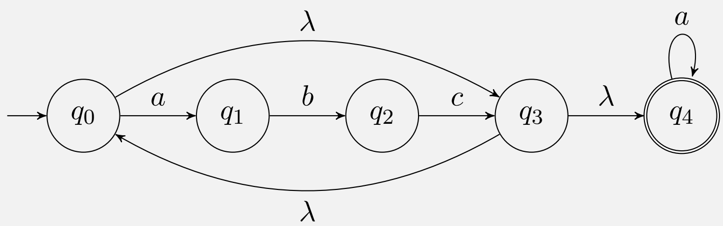
E. $(ab)^* \cup a^*$

The NFA-λ:



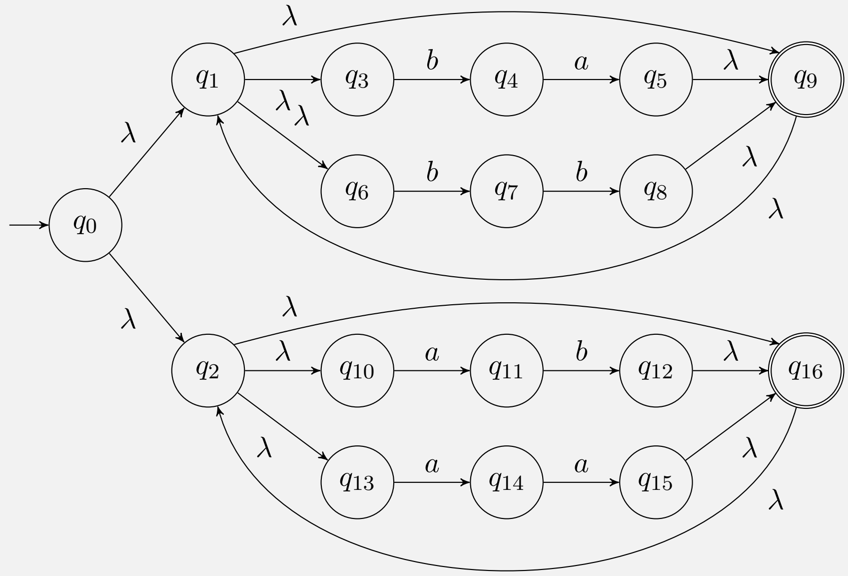
F. $(abc)^*a^*$

The NFA-λ:



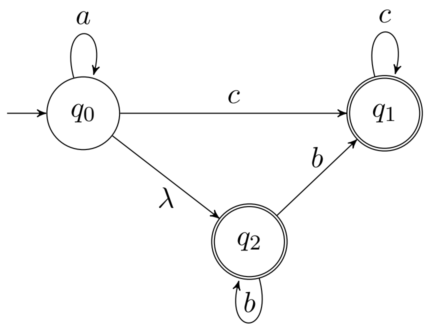
G. *(extra)* $(ba \cup bb)^* \cup (ab \cup aa)^*$

The NFA-λ:



2.3 From NFA to DFA (determinisation)

H. Let M be the following NFA-λ



(a) Compute the $\lambda - closure(q_i)$ for $i = 0, 1, 2$.

(b) Give the inpute trasition function t for M.

(c) Use the algorithm from the lecture (or Algorithm 5.6.3 in the Sudkamp book) to construct a state diagram of a DFA that is equivalent to M

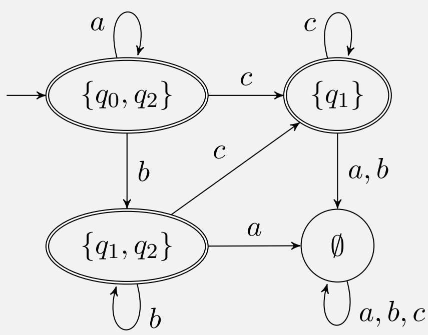
(d) Give a regular expression for $L(M)$.

5.36 a) λ-closure$(q_0) = \{q_0, q_2\}$
λ-closure$(q_1) = \{q_1\}$
λ-closure$(q_2) = \{q_2\}$

b) The input transition function is given by the following table:

t	a	b	c
q_0	$\{q_0, q_2\}$	$\{q_1, q_2\}$	$\{q_1\}$
q_1	$\emptyset$	$\emptyset$	$\{q_1\}$
q_2	$\emptyset$	$\{q_1, q_2\}$	$\emptyset$

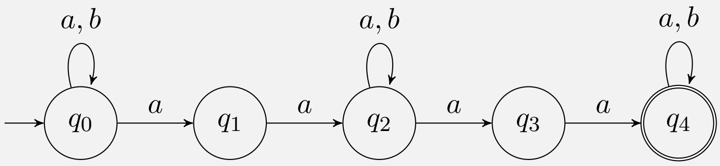
c) The DFA corresponding to this NFA:



d) $a^*b^*c^*$ (you have seen this expression on a previous exercise sheet!)

I. *(extra)* Consider the language L that contains exactly the words over $\{a, b\}$ in which the substring aa occurs at least twice separately (i.e. $aaa \notin L$). Give a regular expression E such that $\mathcal{L}(E) = L$, draw the state diagram of an NFA (with or without λ-steps) that accepts L, and finally determinise this automaton to obtain a DFA that accepts L.

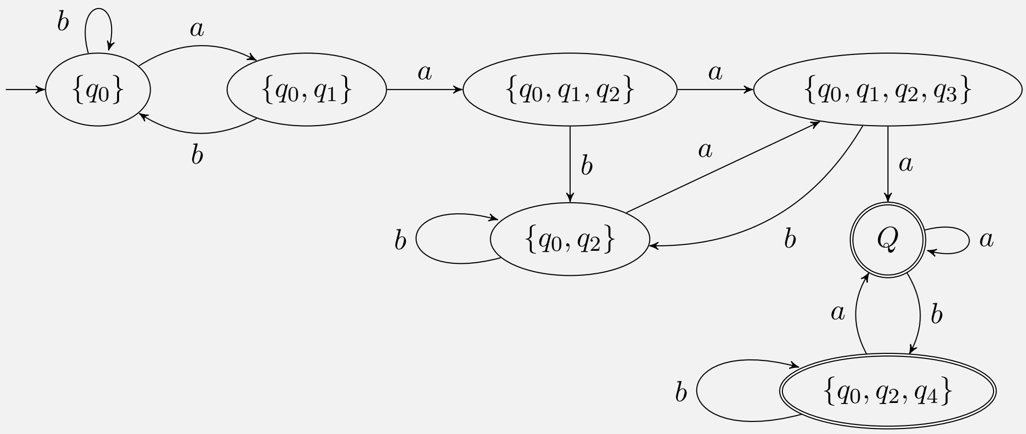
- The regular expression: $(a \cup b)^*aa(a \cup b)^*aa(a \cup b)^*$
- A possible solution here is (without λ-steps):



- The input transition function:

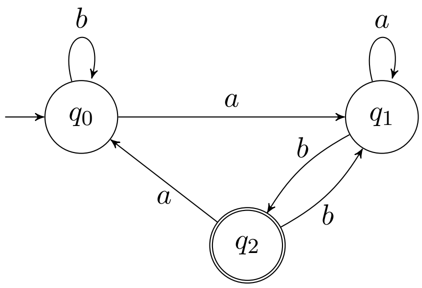
t	a	b
q_0	$\{q_0, q_1\}$	$\{q_0\}$
q_1	$\{q_2\}$	$\emptyset$
q_2	$\{q_2, q_3\}$	$\{q_2\}$
q_3	$\{q_4\}$	$\emptyset$
q_4	$\{q_4\}$	$\{q_4\}$

- The DFA:



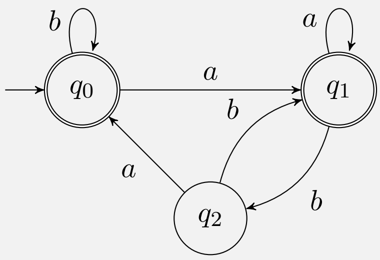
2.4 Complement

J. Construct a finite automaton that accepts the complement of the language accepted by the finite automaton M below. Let M be the DFA defined by the following state diagram:

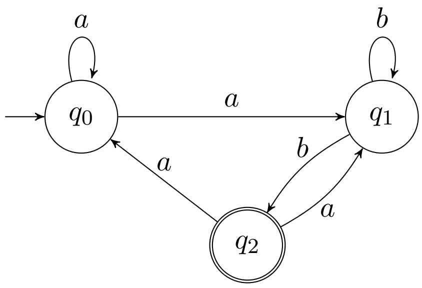


Hint: This is very easy.

The complement is formed simply by making the accepting states non-accepting and the other way around:



K. Construct a finite automaton that accepts the complement of the language accepted by the finite automaton M below.



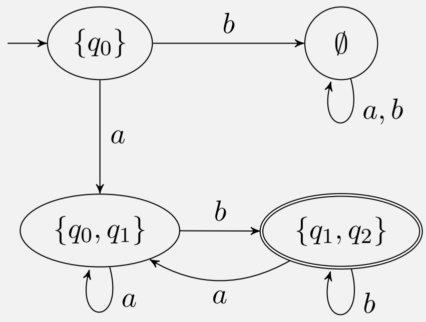
Hint: This is not so easy.

The automaton must first be determinised.

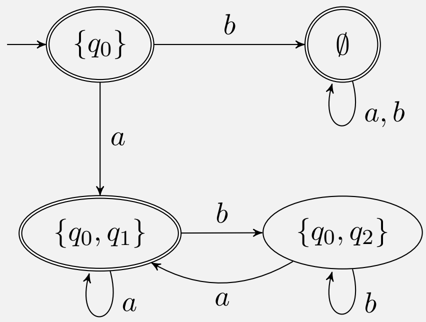
- The input transition function:

t	a	b
q_0	$\{q_0, q_1\}$	$\emptyset$
q_1	$\emptyset$	$\{q_1, q_2\}$
q_2	$\{q_0, q_1\}$	$\emptyset$

- The DFA that follows from this:



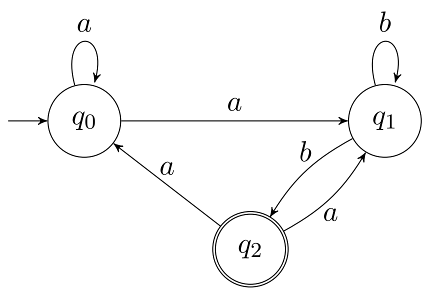
- So the complement is:



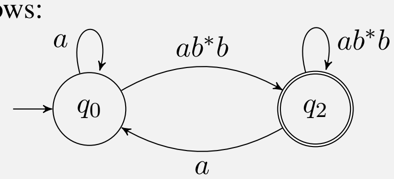
2.5 From NFA to RE

For each of the following finite automata, use the algorithm from the lecture to obtain a regular expression that accepts the same language.

L. The automaton given below.

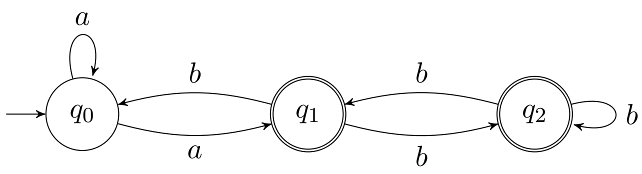


eliminating q_1 results in two arrows:

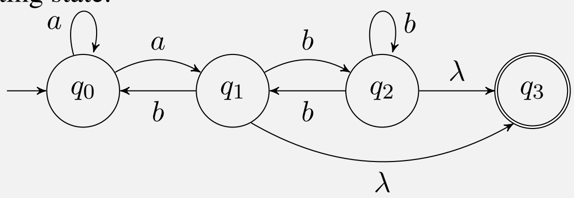


The regular expression then becomes: $a^*ab^*b(aa^*ab^*b \cup ab^*b)^*$. This can be simplified to $a^+b^+(aa^+b^+ \cup ab^+)^*$ and eventually even to $(a^+b^+)^+$.

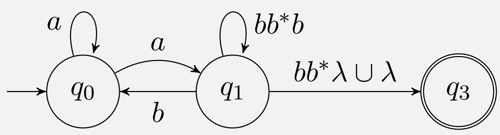
M. The automaton given below.



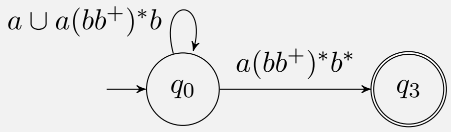
First add q_3 as new accepting state.



Then first eliminate q_2, the state with the least incoming and outgoing arrows:



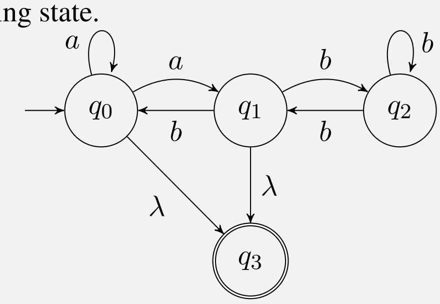
Simplify $bb^*\lambda \cup \lambda$ to b^*, bb^*b to bb^+ and eliminate q_1:



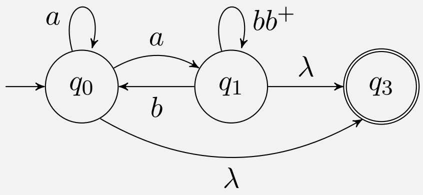
The resulting expression then is $(a \cup a(bb^+)^*b)^*a(bb^+)^*b^*$; this can be simplified to $(a \cup ab \cup abbb^+)^*ab^*$.

N. *(extra)* The finite automaton given in Exercise M above, but with accepting states q_0 and q_1 (instead of q_1 and q_2).

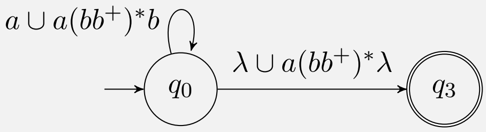
First add q_3 as the new accepting state.



Eliminate q_2:

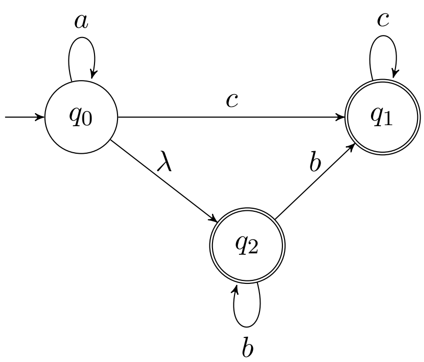


Eliminate q_1:

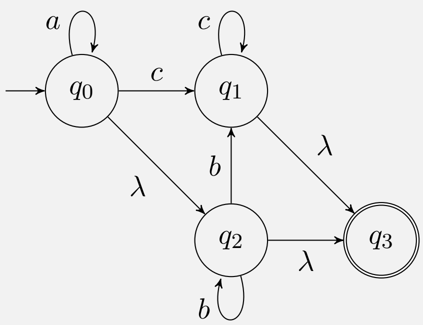


The resulting expression is $(a \cup a(bb^+)^*b)^*(\lambda \cup a(bb^+)^*)$.

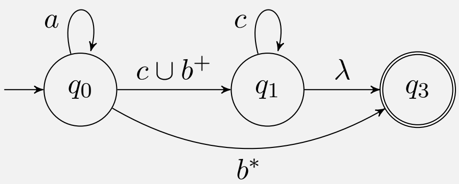
O. The finite automaton given below.



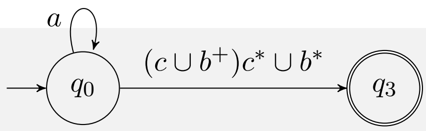
New final state:



Eliminate q_2:

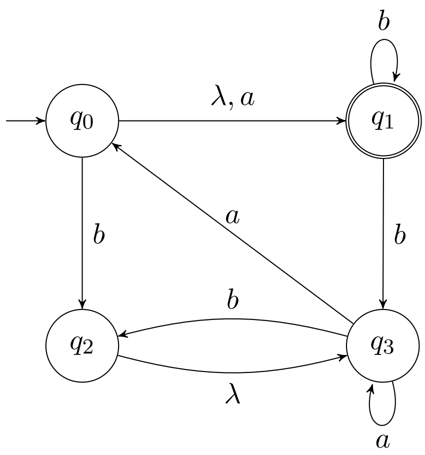


Eliminate q_1:



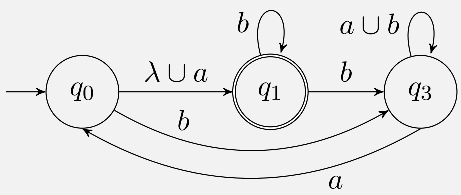
$(c \cup b^+)c^* \cup b^*$ can be simplified to b^*c^*; the complete expression then becomes $a^*b^*c^*$.

P. The finite automaton given below.

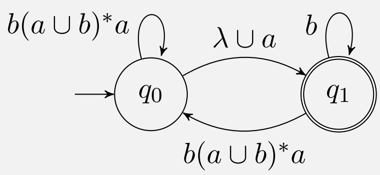


Note: Transition label "λ, a" is a shorthand for two transitions, one labelled with λ and one labelled with a, in the same direction between the same two states. Equivalently, you can replace the label with "$\lambda \cup a$" and get a headstart in your transformation!

Eliminate q_2:



Eliminate q_3:



The complete expression is $(b(a \cup b)^*a)^*(\lambda \cup a)\big(b \cup b(a \cup b)^*a(b(a \cup b)^*a)^*(\lambda \cup a)\big)^*$, which can be simplified to $(b(a \cup b)^*a)^*(\lambda \cup a)\big(b \cup (b(a \cup b)^*a)^+(\lambda \cup a)\big)^*$.

Hint: Exercise P is a bit more interesting than O, so you may want to skip O. But if you then find out that you have trouble with P, go back and do O to get some more practice first.