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. $(ab)^*ba$

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

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

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

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.

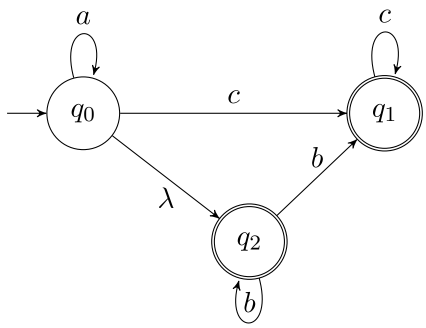
E. $(ab)^* \cup a^*$

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

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

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.

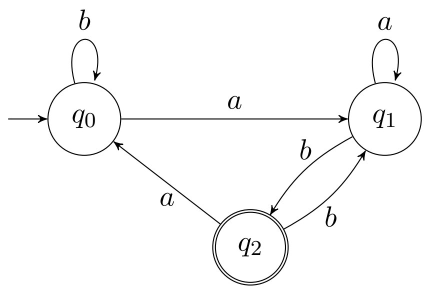
(c) Use the algorithm from the lecture (or Algorithm 5.6.3 in the Sudkamp book) to construct a state diagram of a DFA that is equivalent to M

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

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.

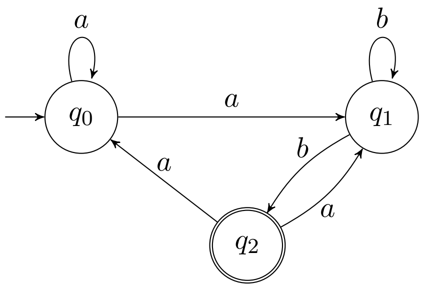
2.4 Complement

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.

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

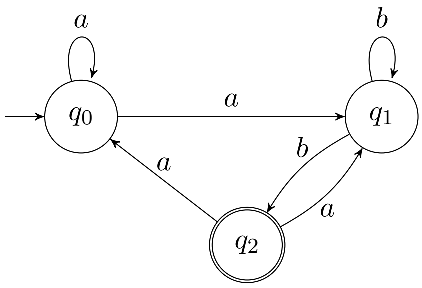


Hint: This is not so easy.

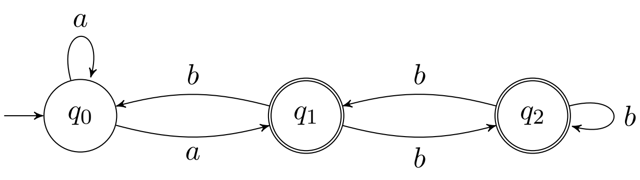
2.5 From NFA to RE

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.

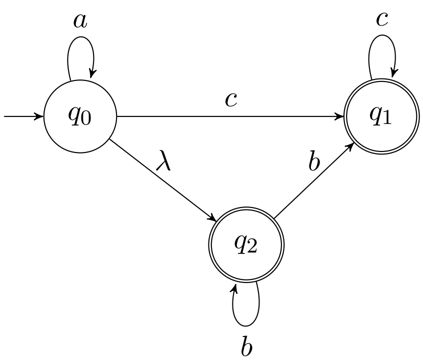


M. The automaton given below.

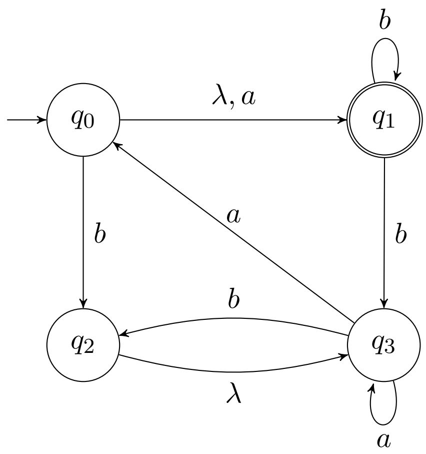


N. *(extra)* The finite automaton given in Exercise M above, but with accepting states q_0 and q_1 (instead of q_1 and q_2).

O. The finite automaton given below.



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!

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.