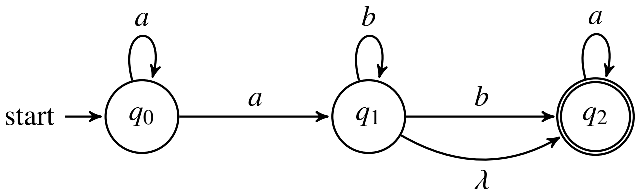
Languages and Computation (COMP 2049) Lab 03

Non-deterministic Finite Automata, Regular Languages, Regular Expressions

(1) Construct a non-deterministic finite automaton (NFA) that accepts the language $\{ab, abc\}^*$.

- Try to use as few states as possible. It is indeed possible to construct one with only three states.
- Use JFLAP to test your design.

(2) Convert the following NFA into an equivalent deterministic finite automaton (DFA):



(3) Is it true that for every non-deterministic finite automaton $M = (Q, \Sigma, \delta, q_0, F)$, the complement $\overline{L(M)}$ satisfies the following?

$$\overline{L(M)} = \{w \in \Sigma^* \mid \delta^*(q_0, w) \cap (Q - F) \neq \emptyset\}$$

If yes, then you must write down a proof. If not, then you must present a counterexample.

(4) What languages are denoted by the expressions $r_1 = (\emptyset^*)^*$ and $r_2 = a\emptyset$?

(5) Consider the language:

$$L = \{a^n b^m \mid n < 3, m \leq 3\}.$$

(a) Write down a regular expression r such that $L = L(r)$.

(b) Write down a regular expression r' for the complement of L, i. e., such that $\overline{L} = L(r')$.

(6) Optional self study: The syntax that we use in this module for regular expressions is suitable for educational purposes, but quite restrictive for practical applications. A good source for finding common regular expressions is:

https://regexlib.com/

Try to find regular expressions for:

(a) Integers in hexadecimal notation;

(b) Floating-point numbers.