# COMP0003 Theory of Computation
# Exercises I: DFAs, NFAs, and regexes

Yuzuko Nakamura

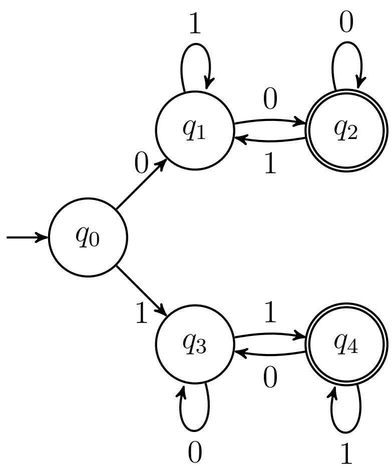
19 & 21 November 2019

## 1 DFAs

**Exercise 1.** Construct a DFA (pictorially and in tuple form) that accepts all strings that start with "`010`" ($\Sigma = \{\texttt{0}, \texttt{1}\}$).

**Exercise 2.** Construct a DFA (pictorially and in tuple form) over the alphabet $\Sigma = \{\texttt{a}, \texttt{b}\}$ that accepts strings with one or more `a`'s followed by one or more `b`'s (for example, should accept "`ab`", "`aaab`", and "`aaaabbbbbb`" but not "`bbb`", "`abba`", or "`ba`".

**Exercise 3.** What is the language of the following DFA?



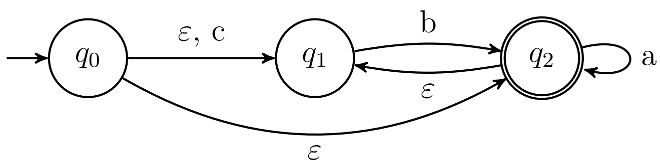

## 2 NFAs

**Exercise 4.** Create an NFA that recognizes simple, properly-formatted numbers. The alphabet is $\Sigma = \{\texttt{-},\ \texttt{.},\ \texttt{0},\ \texttt{1},\ \texttt{2},\ \texttt{3},\ \texttt{4},\ \texttt{5},\ \texttt{6},\ \texttt{7},\ \texttt{8},\ \texttt{9}\}$, and a properly formatted number consists of (optional) a sign (`-`), (optional) zero or more digits in front of a decimal point, followed by one or more digits.

**Exercise 5.** In class, we showed that DFAs are closed under complement by switching the set of accepting and non-accepting states. Would the same technique work with NFAs? Why or why not?

**Exercise 6.** Use NFAs to show that regular languages are closed under reversal. That is, if $L$ is a regular language, then $L^R = \{w \mid w^R \text{ ($w$ written backwards)} \in L\}$ is also a regular language.

**Exercise 7.** Convert the following NFA to an equivalent DFA.




# 3 Regexes

**Exercise 8.** Write the languages featured in Exercises 1-4 and 7 as regular expressions.