CMPT 432 - Spring 2025 - Dr. Labouseur

# Lab 2 – Finite Automata

Ryan Munger
Ryan.Munger1@marist.edu

February 13, 2025

## 1 Crafting a Compiler

### 1.1 Exercise 3.3 - Regular Expressions

Create Regex for the following DFAs:

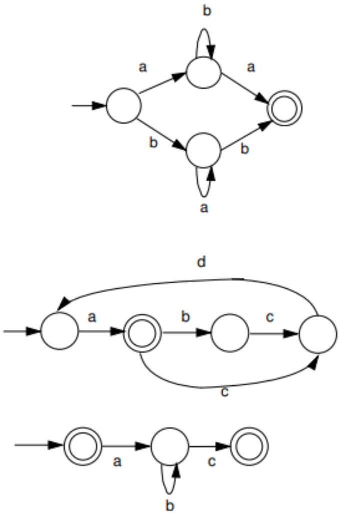


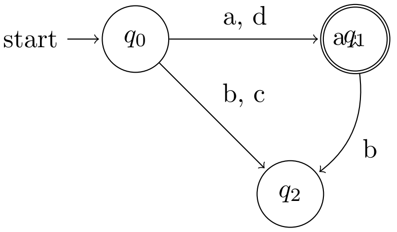
Figure 1: DFAs for exercise 3.3

**DFA #1:** (ab*a)|(ba*b)
**DFA #2:** a((bcda)|(cda))*
**DFA #2:** $\epsilon$ |(ab*c)

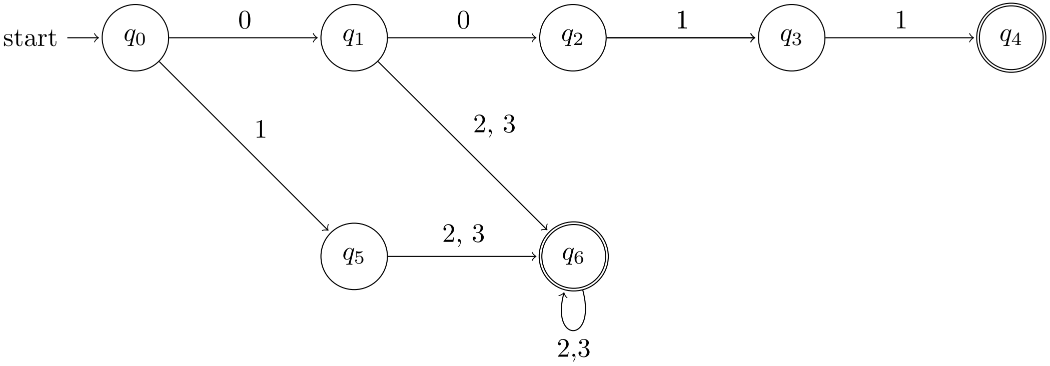
## 1.2 Exercise 3.4 - Deterministic Finite Automatas

Write DFAs that recognize the tokens defined by the following regex:
On a side note, the TikZ and Automata packages are awesome!

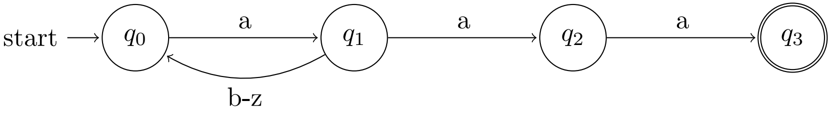
1. $(a|(bc)*d)+$



2. ((0|1)(2|3)+)|0011



3. $(a\neg(a))*aaa$



---

Some say girlfriends are like DFAs: *if you don't start off right, you can't escape the error state...*

---

# 2 Dragon Book

## 2.1 Exercise 3.3.4 - Case Insensitivity in Regex

*Most languages are case sensitive, so keywords can be written only one way, and the regular expressions describing their lexeme is very simple. However, some languages, like SQL, are case insensitive, so a keyword can be written either in lowercase or in uppercase, or in any mixture of cases. Thus, the SQL keyword SELECT can also be written select, Select, or SELECT, for instance. Show how to write a regular expression for a keyword in a case-insensitive language. Illustrate the idea by writing the expression for "select" in SQL.*

We can use OR (|) or regex flags to accomplish this. Most regex engines will accept flags such as /select/i to ignore case. This aside, we could also do: $(s|S)(e|E)(l|L)(e|E)(c|C)(t|t)$

---

**Insert corny joke about having two problems here**

---