Math Circles – Finite Automata
Question Sheet 3

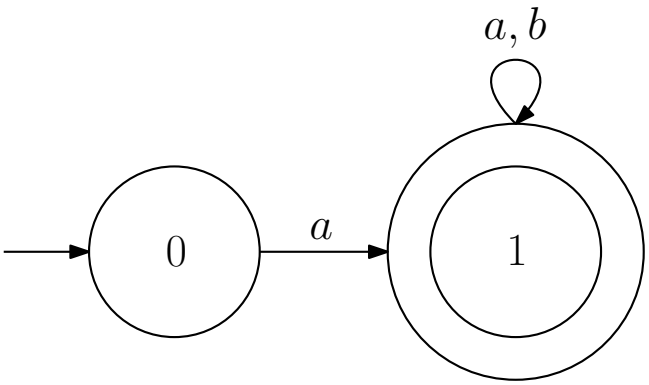
Nickolas Rollick – nrollick@uwaterloo.ca

November 21, 2018

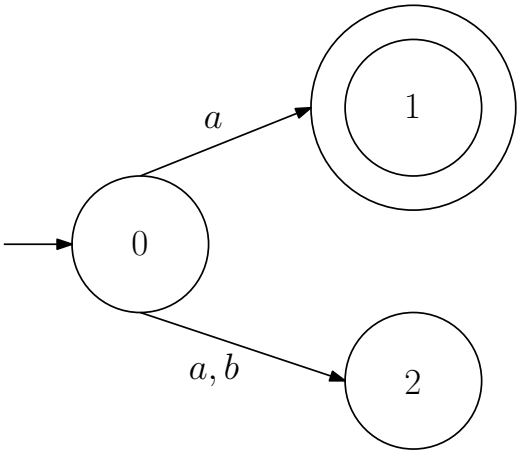
Questions from Lesson

1. Describe all strings accepted by each of the following NFAs:

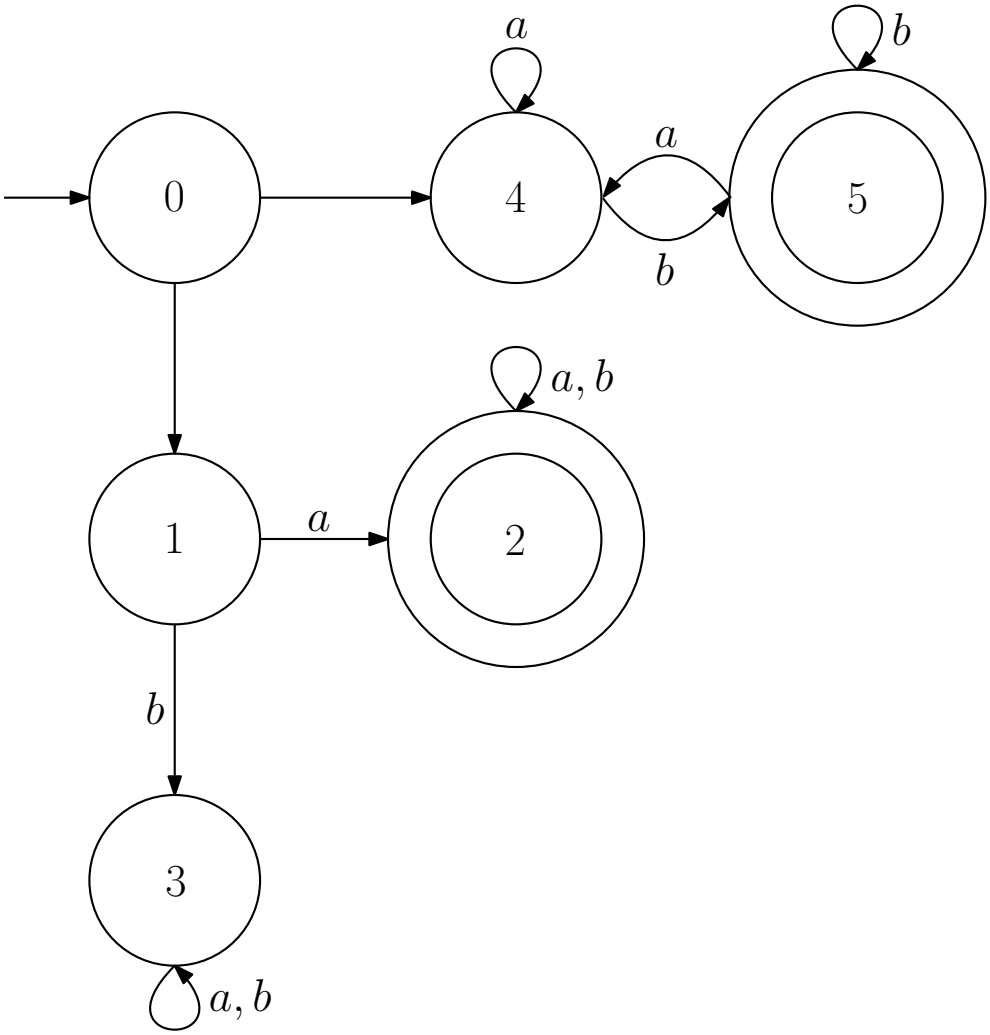
(a)



(b)



(c)



2. Draw an NFA satisfying each set of conditions:

   (a) It has four states, and accepts the language of strings that have no bs, or else have one or more bs followed by a single a.

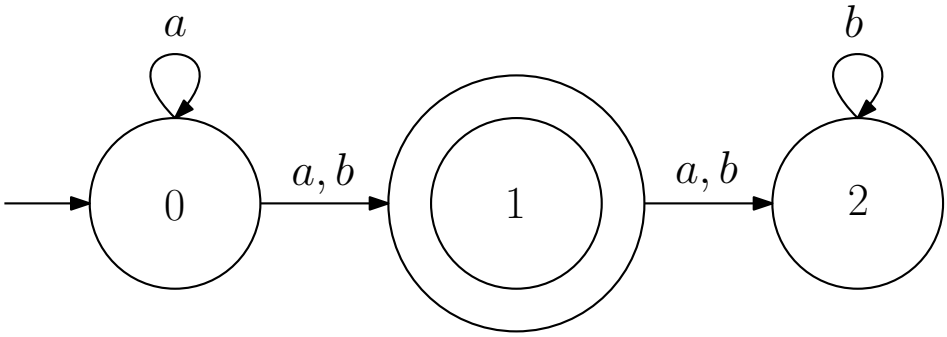
   (b) It has three states and allows three possible symbols: a, b, and c. It accepts the language defined by the following conditions:

   - The empty string is in the language.
   - Given a string in the language, adding ab or abc to the end gives another string in the language.

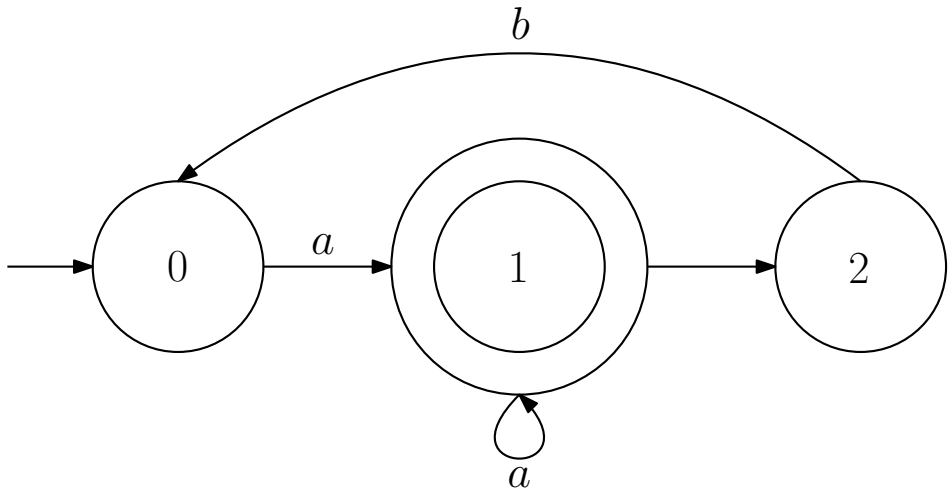
   (For example, ab, abc, $abcab$, $abab$, $abcabc$ all belong to this language.)

3. Given the following NFAs, construct DFAs that accept the same language:

(a)



(b)



Extra Questions

4. Here, we will see that every *finite* language built out of strings of as and bs is regular. In other words, given any finite collection of strings, there is some DFA that accepts exactly those strings and no others.

   (a) Draw an NFA with four states and three arrows that accepts the string *bab* and nothing else. Now draw an NFA with six states and five arrows that accepts the string *aaabb* and nothing else.

   (b) Building on the pattern from the previous part, suppose we are given any string of as and bs (maybe it's *aaababbaababababba*). Describe how to build an NFA that accepts that string, and nothing else.

   (c) Given any single string, explain why there is a **D**FA (not just an **N**FA) accepting that string, and nothing else.

(d) Last time, we saw that for any two DFAs, accepting any two languages, we can build another DFA accepting the strings in either language. Using this, explain how we can take any **two** strings and build a DFA accepting those two strings and nothing else.

(e) Building on the idea from the previous part, explain how we can take any finite number of strings and build a DFA accepting those strings and nothing else.

5. Given any NFA, explain how to construct a new NFA with exactly one accepting state that accepts the same language. (**Hint:** Try this on a few examples first – the NFA in Question 1(c) is a good place to start.)

6. For any string, the *reverse* of that string is obtained by reversing the order of all the symbols. For example, the reverse of *aabab* is *babaa*, and the reverse of *abba* is *abba* again. Given a language, we can define the *reverse language* to be what you get if you reverse all the strings in the original language.

   If a language is regular, explain why the reverse language is also regular. (**Hint:** Starting with a DFA accepting the original language, turn it into an NFA with a single accepting state that accepts the same language, using the previous question. Then convert that NFA into a new NFA accepting the reverse language.)