# Math Circles – Finite Automata
# Question Sheet 1 (Solutions)

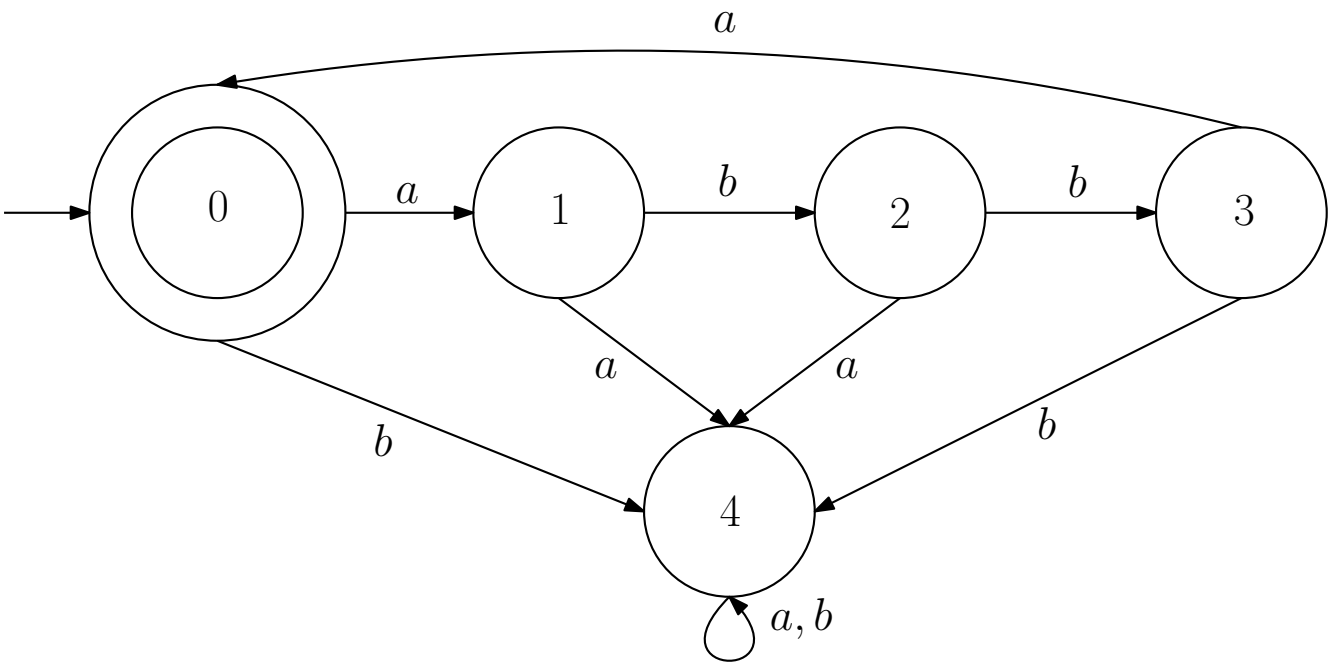
Nickolas Rollick – nrollick@uwaterloo.ca

November 7, 2018

**Note:** These solutions may give you the *answers* to all the problems, but they usually won't tell you how to *get* the answer. All the fun and profit lies in finding the answers for yourself... Also be aware that some questions have more than one solution – this will only provide you with one of them!

## Questions from Lesson

1. Consider the following DFA:

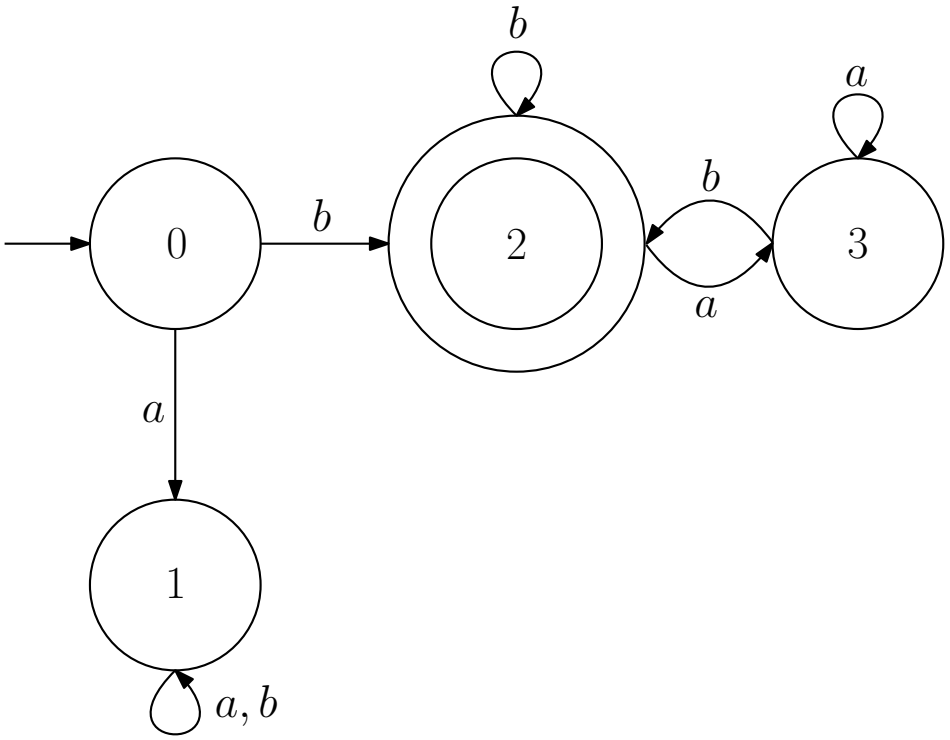



Describe the strings accepted by this machine.

**Solution:** This machine accepts only the strings where *abba* is repeated a certain number of times (including zero). So, it accepts the empty string, *abba*, *abbaabba*, *abbaabbaabba*, and so on. In shorter form, it accepts strings of the form $(abba)^m$, where $m$ can be any whole number bigger than or equal to 0.

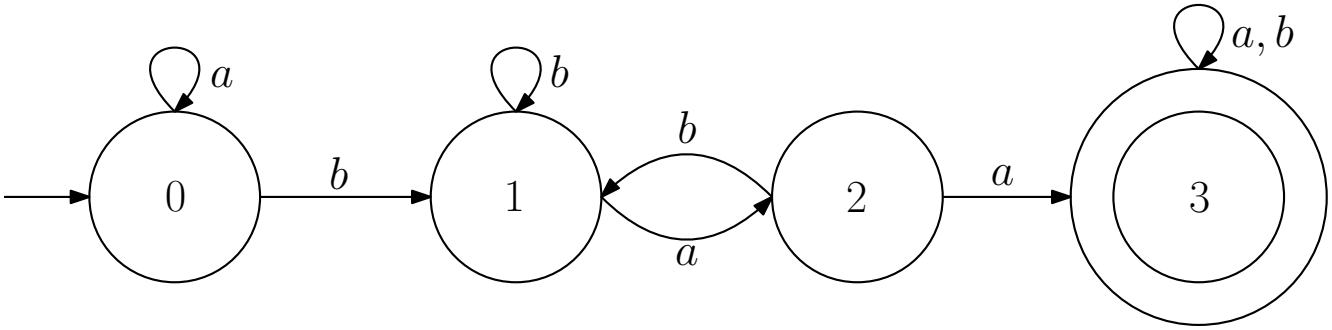
2. Build a DFA that accepts exactly the strings starting *and* ending with $b$.

**Solution:**



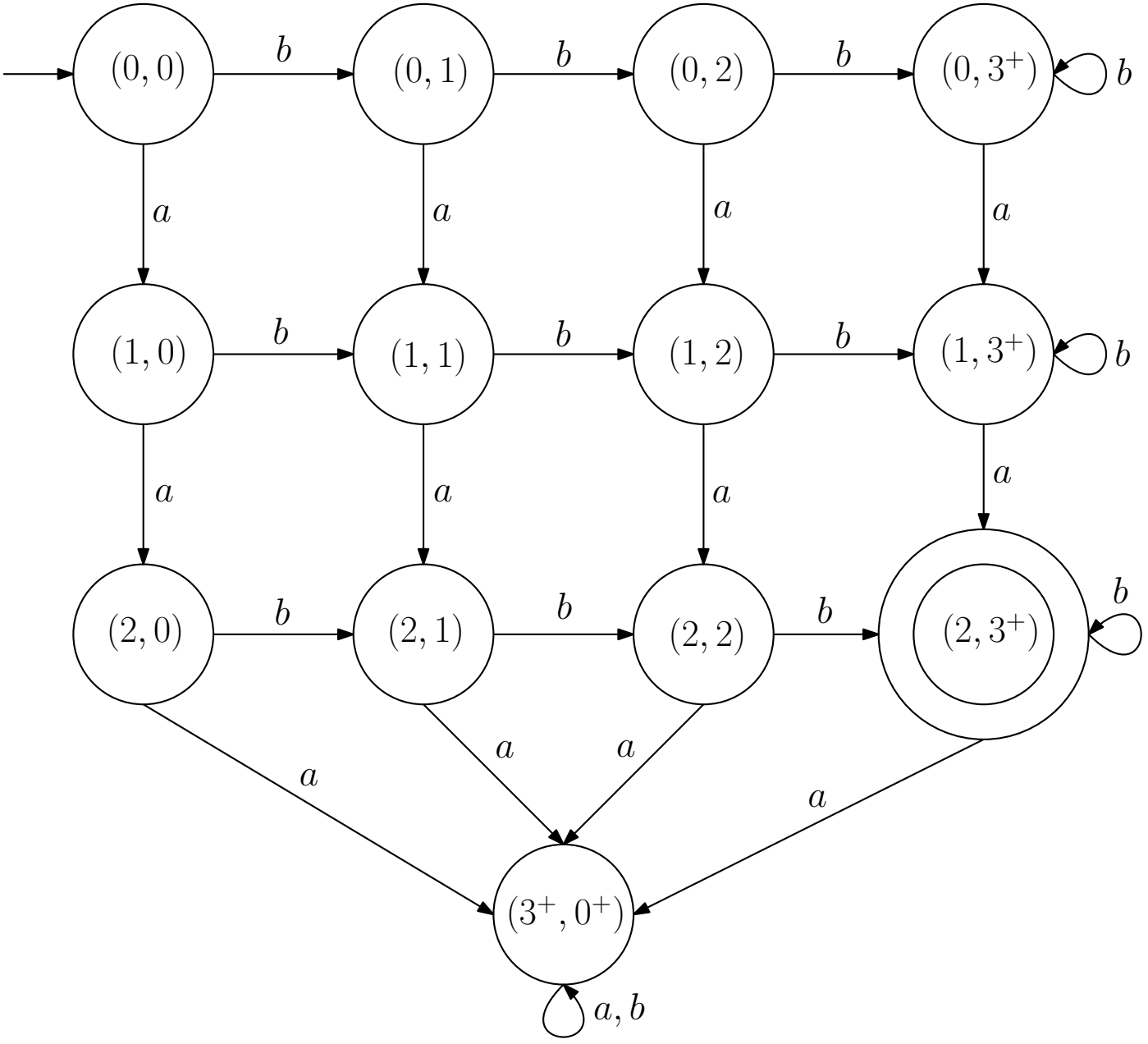

3. Build a DFA that accepts exactly the strings with *baa* inside them. (So *ababaa* and *bbaabbb* work, but *baba* doesn't).

**Solution:**



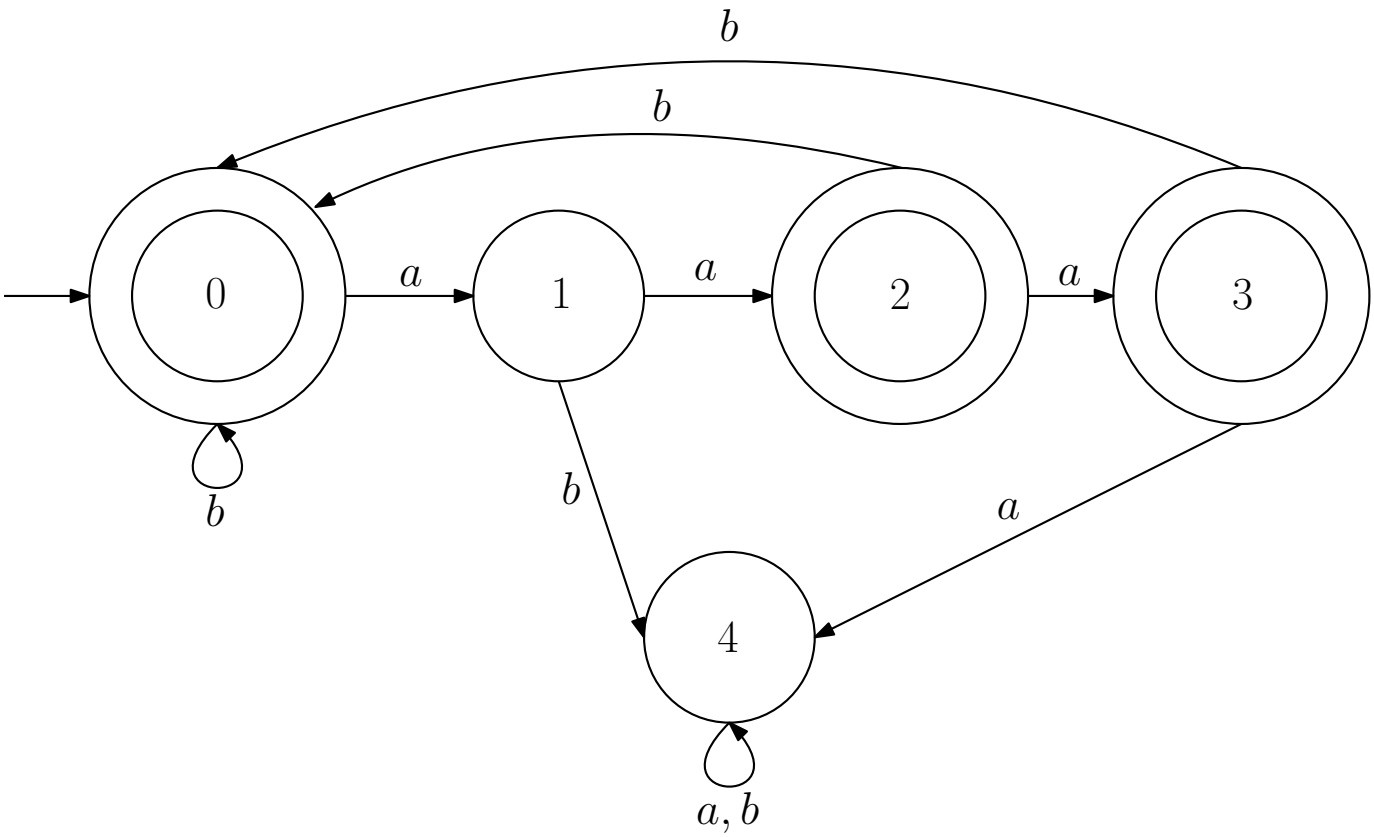
4. Build a DFA that accepts the strings with *exactly* two $a$s and more than two $b$s.

**Solution:** This DFA can be laid out beautifully, using the horizontal direction to track the number of $b$s and the vertical direction to track the number of $a$s:



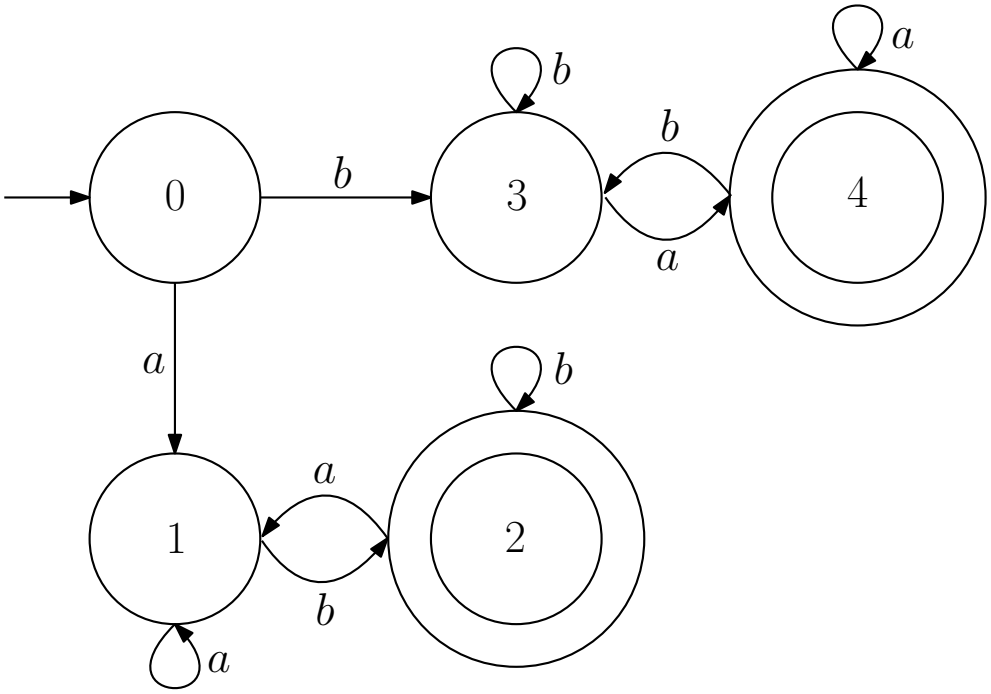
5. Build a DFA that accepts only strings where every run of $a$s has length 2 or 3 (so the $a$s always come 2 in a row or 3 in a row).

**Solution:**



## Extra Questions

6. Can you describe the language accepted by the following DFA, in simple terms?




**Solution:** This DFA accepts the language of strings where the first letter is different from the last one. Another way to describe it is the language of strings that start with $a$ and end with $b$, OR start with $b$ and end with $a$.

7. Consider a language built up from the following rules:

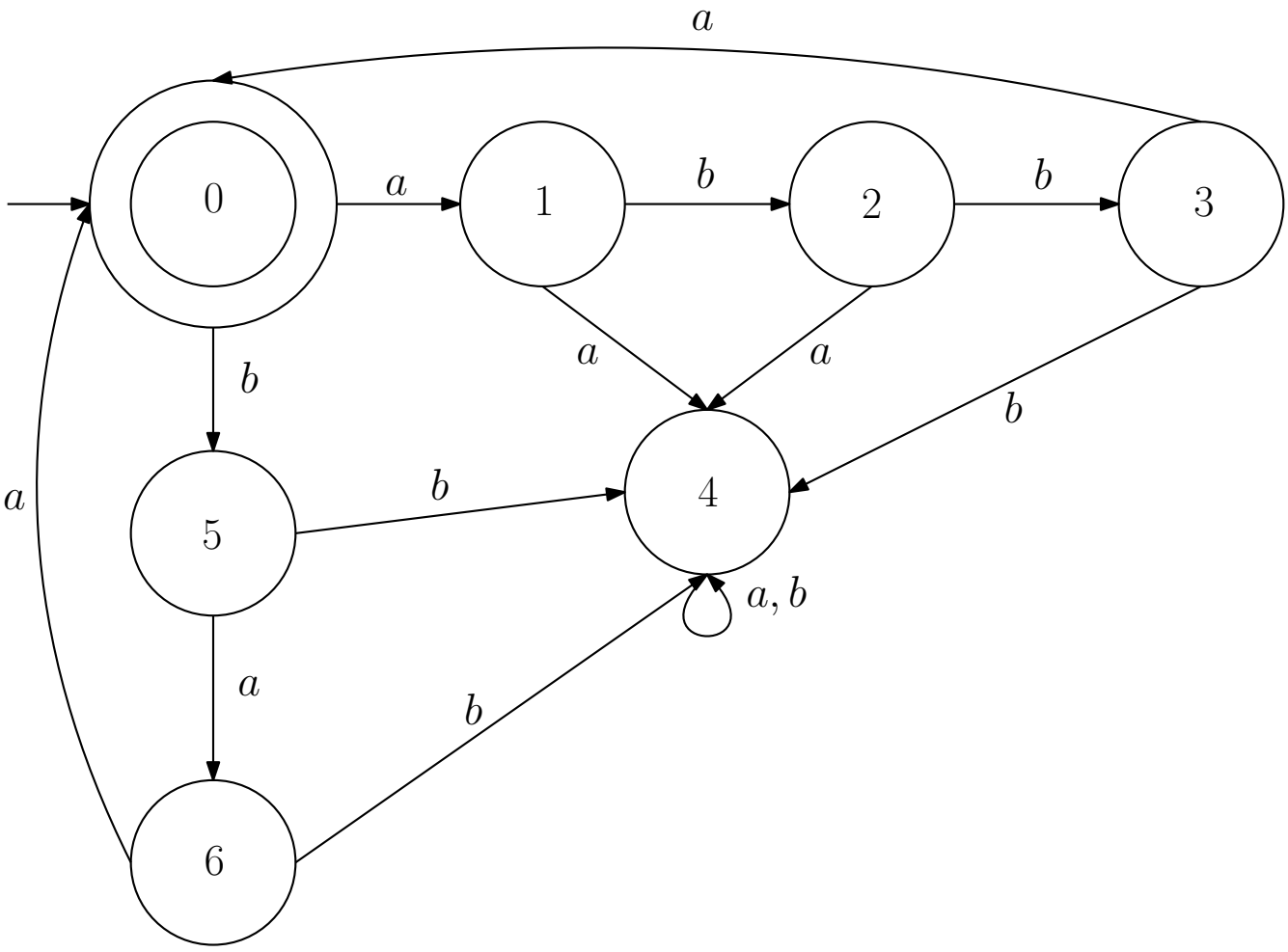
   (a) The empty string is in the language.

   (b) If you know a string is in the language, attaching *abba* or *baa* to the end of the string gives you another string in the language.

   So *abba* and *baa* are in the language, and so are *abbaabba*, *abbabaa*, *baabaa*, and so on. The short way mathematicians use to describe this language is $\{abba, baa\}^*$.

   Can you build a DFA accepting this language?

   **Solution:**



8. Have you ever tried to create a password for a website and been forced to include a bunch of special characters and obey a bunch of other rules? These conditions can be modelled by a DFA. Let's take a look at a simplified version of the situation.

   Suppose users must create a password out of the symbols $a, b, c, 0$, subject to the following conditions:

   (a) The password must contain at least one 0, and at least one letter (one of $a, b, c$).

   (b) The password can never contain *cab* inside it.

   Can you construct a DFA that accepts the legal passwords, and rejects the illegal ones? (Don't worry if the picture gets messy).

   **Solution:** Here is one possible solution, without labels inside the states: