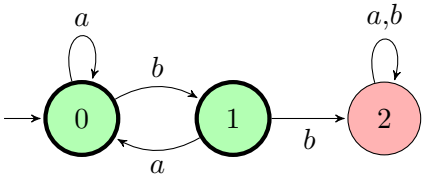
COSC341 TUTORIAL 5

Checking to make sure we understand how to convert between all the different versions of regular languages and a bit more on closure properties. Unless otherwise specified use the alphabet $\Sigma = \{a, b\}$.

1. A *closure property* for the regular languages is a construction that applies to a regular language (or languages) and is guaranteed to produce another regular language. For example "the union of two regular languages is regular". What closure properties have we mentioned so far? What others can you think of?

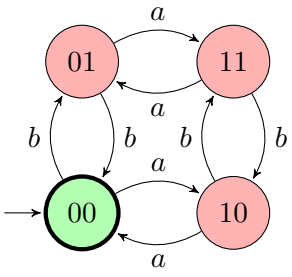
2. You told a friend about the result that the language of strings of a's whose length is a power of two is not regular. They said "I wonder whether there is a regular language of strings from a 17-letter alphabet where the length of every string in the language is a power of two, and all powers of two occur?" What do you think the answer to their question is? How is this related to the previous question? Does this give you a proof?

3. Near the end of Lecture 1 (on the slide titled "What does this machine do?") we introduced the following DFA over the alphabet $\{a, b\}$ that accepts strings containing no pair of consecutive b's:



   Design an NFA for the language of strings that contain either no pair of consecutive b's or no pair of consecutive a's. Convert it to to a DFA.

4. Tutorial 4 asked you to develop a DFA that accepts the language EVEN-EVEN, which consists of all strings containing both an even number of a's and an even number of b's. This was the suggested solution:



   Now design an NFA in standard form that accepts this language and then use the state elimination technique to derive a regular expression for it. Compare your result to the one presented (without use of the state elimination technique) in the solution to Tutorial 4, Question 4. Was this surprising?

5. Recall the language MULTI-6-b (introduced in Tutorial 4), which consists of all strings with the following property: for some a in the string, the number of b's that follow it is a multiple of 6. We saw an NFA with seven states that recognises it (9 states in "standard form"). Convince yourself that to convert it to DFA will result in something like 64 states ("off by several" errors are possible here) and that this is somehow necessary.