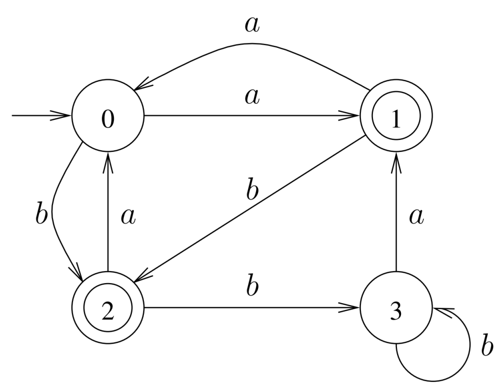
Nathan Bickel, Jackson Ginn, Jack Hyatt, Chance Storey
CSCE 355: Section 001
Professor: Dr. Fenner
January 25, 2024

CSCE 355 Homework 2

Problem 1 Consider the following DFA:



(a) For each of the strings below, say which state the DFA is in after reading the string, and say whether or not the DFA accepts the string.

aaa $\quad bb$ $\quad bbb$ $\quad abab$ $\quad bbbbbbbbbbbbbbbaaa$ $\quad \varepsilon$ $\quad aabbbbababbaaabbaabbababbbb$

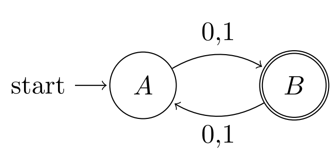
(b) Give two different strings of length 4 that each make the DFA go from state 0 to state 1.

(a)
- aaa: the DFA ends in state 1, where the DFA accepts the string.
- bb: the DFA ends in state 3, where the DFA rejects.
- bbb: the DFA ends in state 3, where the DFA rejects.
- $abab$: the DFA ends in state 2, where the DFA accepts.
- $bbbbbbbbbbbbbbbaaa$: the DFA ends in state 1, so DFA accepts.
- ε: the DFA ends in state 0, so the DFA rejects.
- $aabbbbababbaaabbaabbababbbb$: the DFA ends in state 3, so the DFA rejects.

(b) The strings $abaa$ and $abba$ both end in state 1 and are length 4.

Problem 2 Draw a DFA with alphabet $\{0, 1\}$ that accepts a binary string x iff x has odd length, i.e., iff $|x|$ is odd.

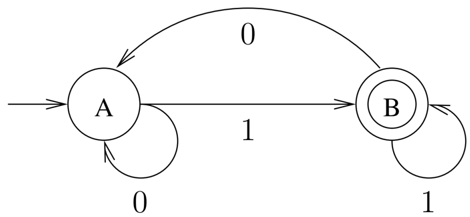
The following DFA works:



Problem 3 Let A be the DFA given by the following tabular form:

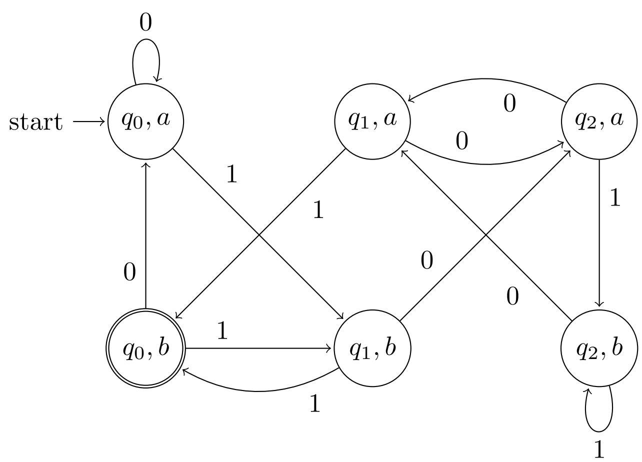
	0	1
$\rightarrow *q_0$	q_0	q_1
q_1	q_2	q_0
q_2	q_1	q_2

(A accepts a binary string iff it represents a multiple of 3.) Recall the DFA described in class (here we'll call it B) that accepts a binary string iff the string ends with 1:



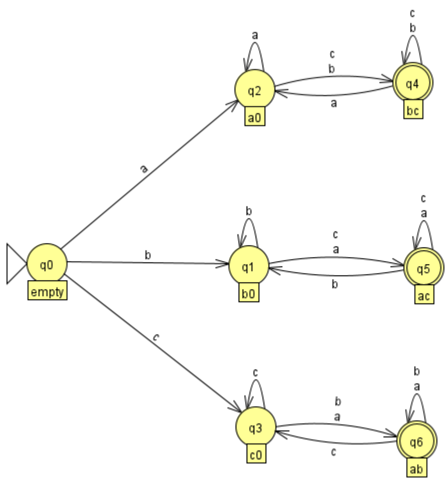
Recall the product construction from class. Draw the diagram for the product of A and B so the resulting DFA recognizes the language $L(A) \cap L(B)$.

Here is the product of DFAs A and B, that will accept a binary string iff it is a multiple of 3 and ends in a 1:



Problem 4 Describe a DFA B that accepts a string over the alphabet $\{a, b, c\}$ iff its first and last symbols are different.

The following DFA works:



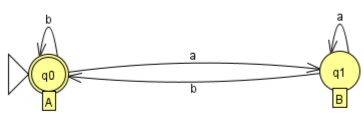
Problem 5 Consider the following two languages over the alphabet $\{a, b\}$:

$$L_1 = \{w \mid w \text{ is either the empty string or ends with } b\} ,$$
$$L_2 = \{w \mid \text{there is a } b \text{ followed by an } a \text{ somewhere in } w\} .$$

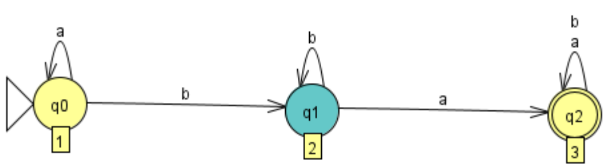
(a) Draw a 2-state DFA recognizing L_1 and a 3-state DFA recognizing L_2.

(b) Using your answer and the product construction, draw a DFA recognizing $L_1 \cap L_2$. Do *not* perform any optimizations (e.g., removing unreachable states or transitions, or merging indistinguishable states).

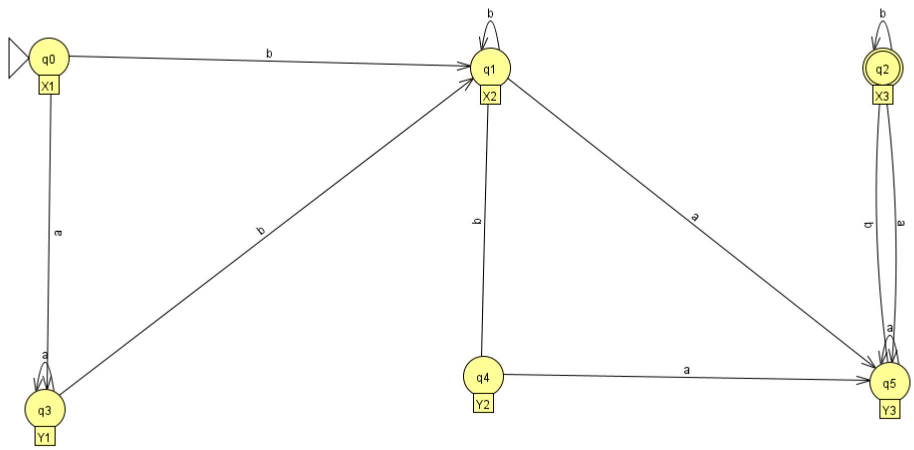
(a) This DFA recognizes L_1:



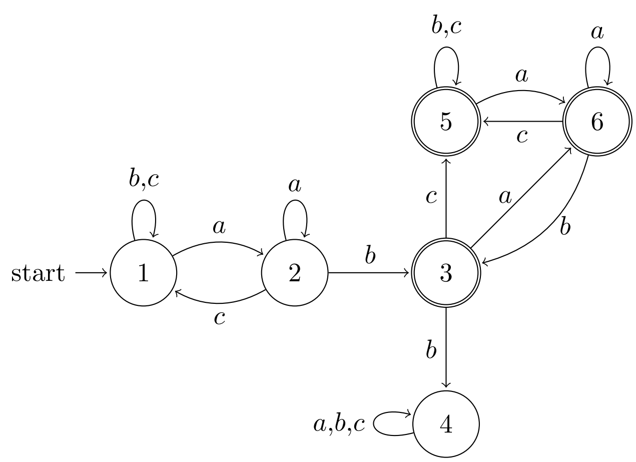
And this DFA recognizes L_2:



(b) We use the product construction to draw the DFA recognizing the intersection:

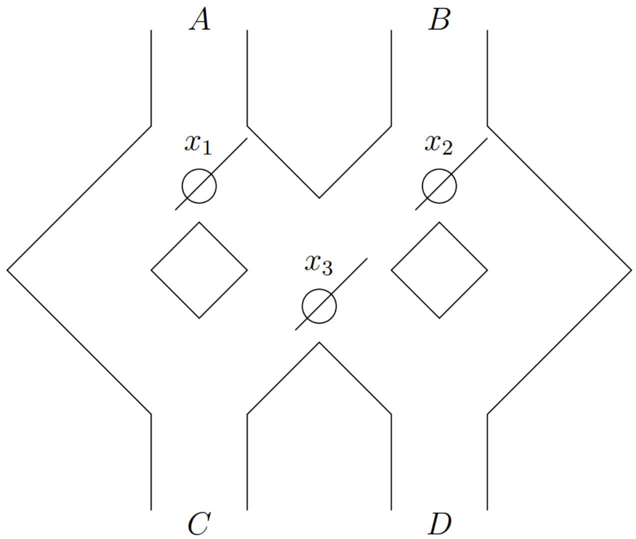


Problem 6 Give the transition diagram for a DFA over the alphabet $\Sigma = \{a, b, c\}$ that accepts a string w iff w contains ab as a substring but does not contain abb as a substring. What is the least number of states you need?



The least number of states you need is 6 because you need 3 accepting states for when you've seen *ab* and are still checking to see if you see any *abb*s, you need two states for finding the first *ab*, and you need a *trash* state for when you've seen *abb* in the string.

Problem 7 This exercise is adapted from Exercise 2.2.1 on pages 52–53, which is formulated somewhat vaguely. Consider the marble-rolling toy (redrawn from Figure 2.8):



A marble is dropped at A or B. Levers x_1, x_2, and x_3 cause the marble to fall either to the left or to the right. whenever a marble encounters a lever, it causes the lever to reverse after the marble passes, so the next marble will take the opposite branch.

Model this toy as a finite automaton. An input to the automaton is a string over the alphabet $\{A, B\}$, which represents a sequence of marbles being dropped into the toy. The toy is initially in the configuration above before any marbles are dropped (so that the first ball will exit at C regardless of where it is dropped). Say that a sequence of marble drops is *accepted* exactly in the case that if one additional marble were to be dropped in, it would go out through D regardless of where it was dropped.

Each of the three switches can either point left or right, so we will have $2^3 = 8$ states. We call the states $d_1d_2d_3$, where d_i is R if x_i is pointing right or L if it is pointing left. Then, the tabular form is:

	A	B
$\rightarrow RRR$	LRR	RLL
RRL	LRL	RLR
RLR	LLR	RRR
RLL	LLL	RRL
LRR	RRL	LLL
LLR	RLL	LRR
$*LRL$	RRR	LLR
$*LLL$	RLR	LRL