CS3800: Theory of Computation — Summer II '22 — Drew van der Poel

Homework 1
Due Friday, July 15 at 11:59pm via Gradescope

Name: Steve Liu
Collaborators: Jamie Lin, Robert Rau

- Make sure to put your name on the first page. If you are using the LaTeX template we provided, then you can make sure it appears by filling in the `yourname` command.
- This assignment is due Friday, July 15 at 11:59pm via Gradescope. No late assignments will be accepted. Make sure to submit something before the deadline.
- Solutions must be typeset. If you need to draw any diagrams, you may draw them by hand as long as they are embedded in the PDF. I recommend using the source file for this assignment to get started.
- I encourage you to work with your classmates on the homework problems. *If you do collaborate, you must write all solutions by yourself, in your own words.* Do not submit anything you cannot explain. Please list all your collaborators in your solution for each problem by filling in the `yourcollaborators` command.
- Finding solutions to homework problems on the web, or by asking students not enrolled in the class is strictly forbidden.

Problem 1. *Automata: Element Ranges (*4 points*)*

(a) **[2 points]** In a DFA with 10 states and an alphabet of size 6, what is the range on the number of transitions? What is the range on the number of start states and accept states? A transition is defined as a state-symbol pair mapped to a state $(q \in Q, c \in \Sigma) \rightarrow (r \in Q)$. Briefly justify your answer.

Solution:

A DFA with 10 states and an alphabet of size 6 will have ONLY 60 transitions, every ”letter” in the alphabet will have a transition with each of the 10 states in the DFA or $6 * 10$ number of transitions. Anymore or any less transitions and the DFA will be have to be considered a NFA.

The number of start states per definition of a DFA has to be 1.

The number of accept states is in the range of $[0, 10]$ since there can be as many all of the states being accept states and as a few as none of the states being accept states.

(b) **[2 points]** In a NFA with 10 states and an alphabet of size 6, what is the range on the number of transitions? What is the range on the number of start states and accept states? A transition is defined as a state-symbol pair mapped to a state $(q \in Q, c \in \Sigma \cup \epsilon) \rightarrow (r \in Q)$. You can assume there are no ϵ-transitions from a state to itself. Briefly justify your answer.

Solution:

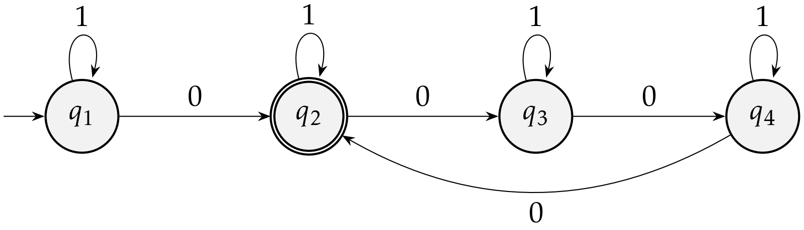
The number of transitions in a NFA with 10 states and an alphabet size of 6 is in the range of $[0, 690]$. The lower bound is 0 because it’s entirely possible for the entire NFA to have 0 transitions for each character in the alphabet. The upper bound is 690 because for each given state, each character can have a transition from that state to every state (including itself) in the NFA so that gives (1 character $*10$ possible transition states) $*$ (6 characters in the alphabet) * 10 states = 600. However there are also ϵ-transitions and given the assumption that ”there are no ϵ-transitions from a state to itself” then each state can have an ϵ-transition from itself to every other state which means that in total, there are ((10 - 1) ϵ-transition per state) * (10 states) = 90. Adding it all up, the upper-bound number of transitions in this NFA would be 690.

The number of starts states per definition of a NFA has to be 1.

The number of accept states is in the range of $[0, 10]$ since there can be as many all of the states being accept states and as a few as none of the states being accept states.

Problem 2. *What Language?* (4 points)

Determine the language that the following DFA recognizes. Briefly justify your answer.



Solution:
Example Inputs:
”0” → ACCEPT

”00” → REJECT

”0000” → ACCEPT

”011111111111” → ACCEPT (number of 1’s doesn’t matter)

”01000” →ACCEPT

”00000” → REJECT

Defining the above DFA as M then the $L(M) = \{ w \mid \text{the number of zeroes in } w \text{ modulo 3 is 1} \}$

Justification: The DFA does nothing with 1s so those can be ignored. If we ignore q1 then the DFA basically recognizes the language of all strings in which the number of 0s is divisible by 3 because of the q4 to q2 loop. Thus, the transition from q1 to q2 means that the language now recognizes all strings in which the number of 0s divided by 3 will leave you with an extra 0, in other words the number of zeroes modulo 3 is one.

Problem 3. *Draw a DFA* (6 points)

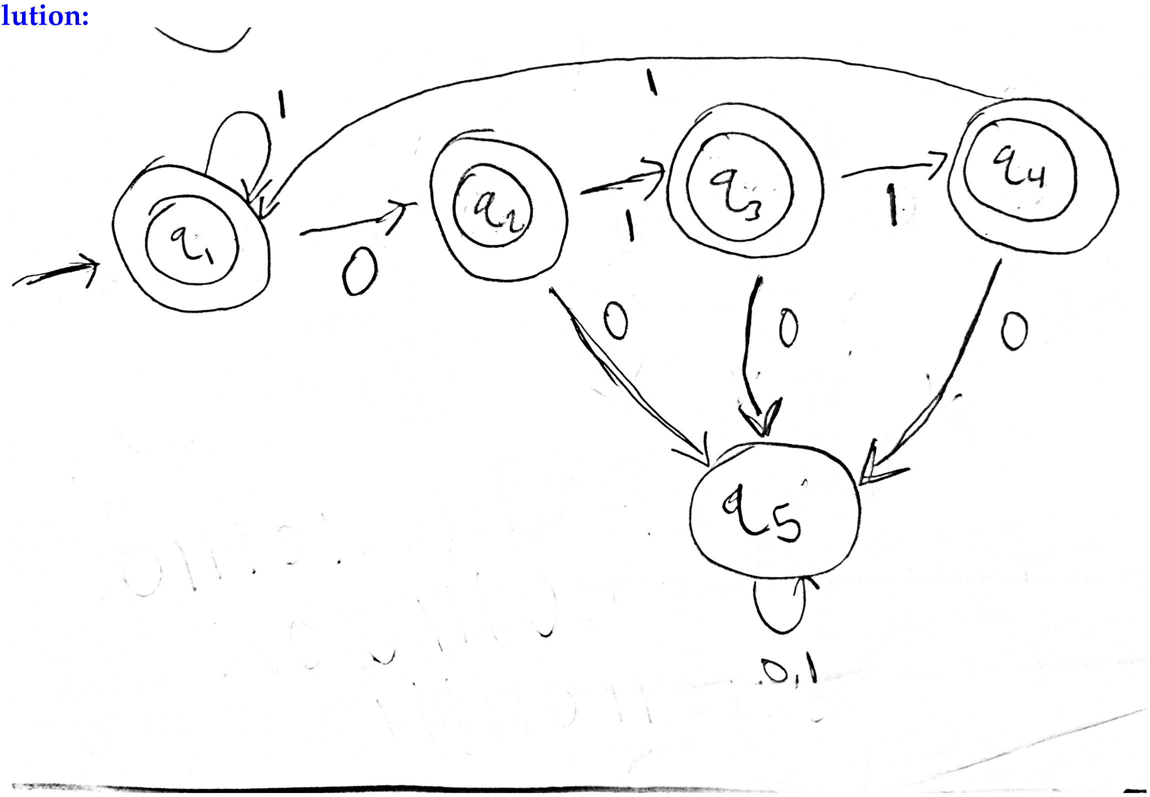
Consider the following language L over $\Sigma = \{0, 1\}$.

$L = \{x|$ any two 0s in x are separated by at least three 1s $\}$

For example, $01110, 01111011101111101, 11101, 1111$ are all in L, but $00, 0110, 01110110$ are not.

Draw a DFA which recognizes L. Be sure to include all states and transitions and to denote the start and accept states. Your DFA should accept the empty string. Your DFA should be as simple as possible.

Solution:



Problem 4. *Union* (10 points)

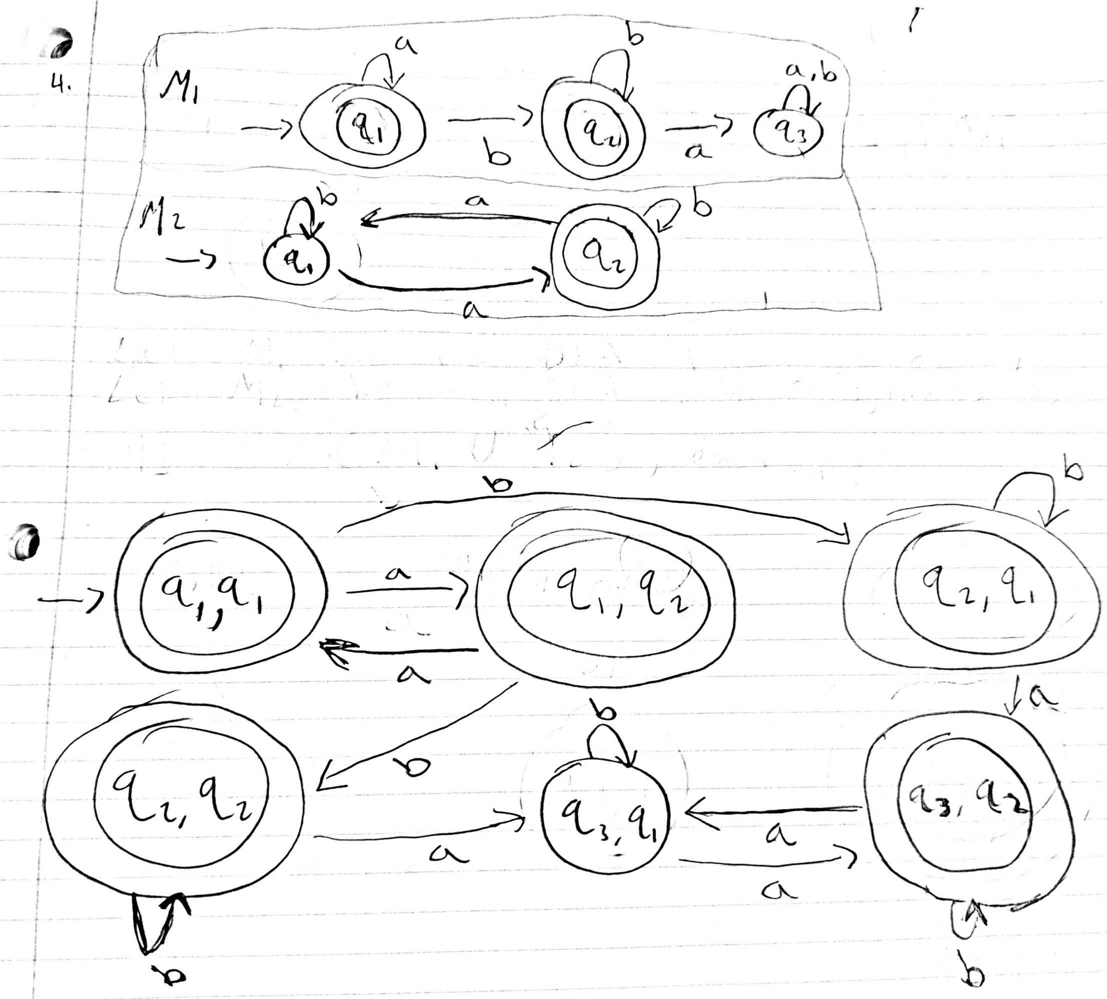
Let $\Sigma = \{a, b\}$. Let $L1$ be the language consisting of strings where all a's are before all b's. Let $L2$ be the language consisting of strings where the number of a's is odd. Draw DFAs recognizing each language, then use the construction seen in class to draw a DFA recognizing $L1 \cup L2$.

Solution:

Let M_1 be the DFA that recognizes $L1$

Let M_2 be the DFA that recognizes $L2$

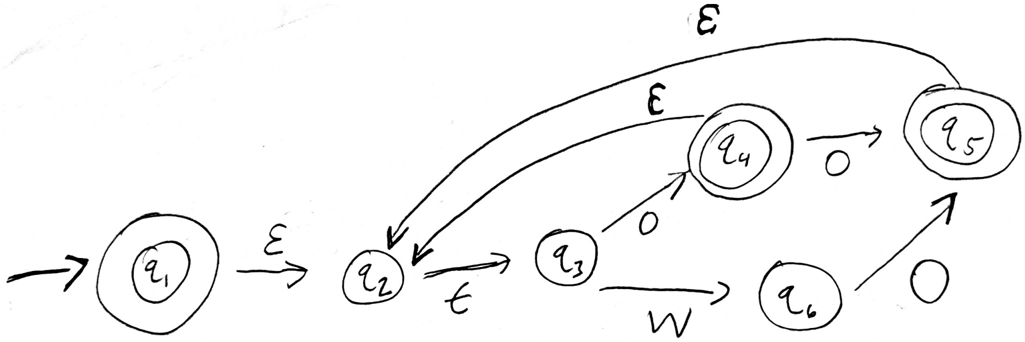
Let M_3 be the DFA that recognizes $L1 \cup L2$ where state, (e, r), is a state-pair where e is from M_1 and r is from M_2



Problem 5. *NFAs (6 points)*

Consider the alphabet $\Sigma = \{o, t, w\}$. Let L be the language of strings which are concatenations of "to", "too", and "two". Draw a NFA which recognizes L. Your NFA should make use of ϵ-transitions and/or have state-character pairs with 0 or more than 1 transitions (to truly take advantage of using NFAs!). Your NFA should accept the empty string.

Solution:



ε
ε
q_1
ε
q_2
t
q_3
o
q_4
o
q_5
w
q_6
o