
INSTRUCTIONS

Upload a **single file** to Gradescope for each group. All group members' names and PIDs should be on **each** page of the submission.

Your assignments in this class will be evaluated not only on the correctness of your answers, but on your ability to present your ideas clearly and logically. You should always explain how you arrived at your conclusions, using mathematically sound reasoning. Whether you use formal proof techniques or write a more informal argument for why something is true, your answers should always be well-supported. Your goal should be to convince the reader that your results and methods are sound.

READING Sipser Sections 1.1, 1.2

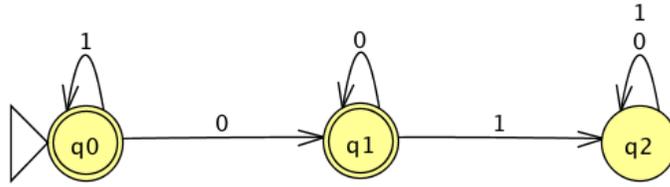
KEY CONCEPTS Deterministic finite automata (DFA), state diagram, computation trace, accept / reject, language of an automaton, regular language, union of languages, concatenation of languages, star of a language, closure of the class of regular languages under certain operations, nondeterministic finite automata (NFA), nondeterministic computation, ε arrows, equivalence of NFA and DFA.

1. (6 points) Draw the state diagram of a DFA over the alphabet $\{0, 1\}$ that recognizes the language

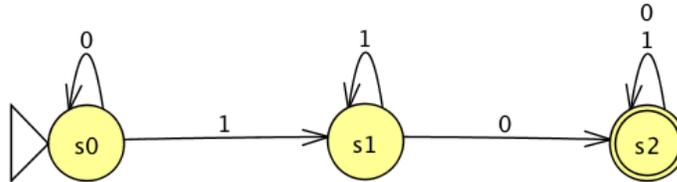
$$\{0^n 1^m \mid n + m \text{ is an even non-negative integer}\}.$$

Your machine should have at most six states. Along with the JFLAP diagram, include a brief description (in English) of why each state is included and why you chose to make each state either accepting or non-accepting.

2. (9 points) Consider the DFA M_1



and the DFA M_2



- (a) If we use the general constructions discussed in class and in the book for building a DFA whose language is

$$L(M_1) \cap \overline{L(M_2)}$$

how many states would be in this DFA? Briefly justify your calculation.

To confirm your answer, run the provided haskell program `HW2.hs`, which implements the closure construction for this question using as arguments the files specifying the DFAs `hw2M1.jff` and `hw2M2.jff`. (For this, you need the files `DFA.hs`, `HW2.hs` and the input DFAs files in your working directory, as well as a running haskell implementation with the `xml` package installed. Then, run `runhaskell1 HW2.hs hw2M1.jff hw2M2.jff >out.jff` to combine the automata in the two input files, and write the result to `out.jff`.)

For full credit for this part of the question, include the image from JFLAP of the output file `out.jff` as part of your assignment submission PDF file and also explain why this DFA has the number of states that it does by relating it to the machines M_1 and M_2 and the construction that was used.

- (b) Construct a DFA whose language is $L(M_1) \cap \overline{L(M_2)}$ and which has at most 4 states. Along with the state diagram, include a brief description (in English) of why each state is included and why you chose to make each state either accepting or non-accepting. You may consider which states of the DFA produced in part (a) of this question are inaccessible or redundant to help you solve this part of the problem.

3. (12 points) Show that the class of regular languages over an alphabet Σ is closed under the operation $EvenOnly(L)$, defined as

$$EvenOnly(L) = \{w \in L \text{ such that } |w| \text{ is even}\}$$

Your proof should have three stages:

Setup Consider an arbitrary DFA $M = (Q, \Sigma, \delta, q_0, F)$, and call the language of this DFA L .

Construction Build a new DFA whose language is $EvenOnly(L)$. To do so, fill in the blanks

$$M' = (Q', \Sigma, \delta', q', F')$$

where

$Q' =$ _____	This will be the set of states for your new machine.
$\delta'(\text{____}, x) =$ _____	For each possible input to the transition function, specify the output. The blank input is a state in Q' and $x \in \Sigma$.
$q' =$ _____	What is the initial state of M' ? Make sure you choose an element of Q' .
$F' =$ _____	What is the set of accepting states of M' ? Choose a subset of Q' .

Justification To prove that the construction of correct, you will need to prove that $L(M') = EvenOnly(L)$ for any L . Fix an arbitrary but unknown language L . Prove two things:

- (1) Assume that some string, call it w , is accepted by M' . Prove that w is in $EvenOnly(L)$.
- (2) Assume that some string, call it y , is in $EvenOnly(L)$. Prove that y is accepted by M' .

4. (8 points) Consider the language over the English (lower-case) alphabet

$$\{bought, brought, douse, fought, house, mouse, ought, sought, thought, wrought\}.$$

Design an NFA recognizing this language. Your NFA should have no more than 20 states. For full credit,

- Draw the state diagram of your machine using JFLAP. Include both a screenshot of your machine **and** a printout of the .jff file in your submission.
- Explain your design in one or two sentences. For each state (or group of states), explain what role they play in recognizing the given language.

Extra credit for correct solutions with 14 states or less.

Optional; not for credit: Is it possible that adding a **single** word to the language could increase the size of the automaton more than if we added **several words at once**? Why or why not?