This homework is due Tuesday September 12 at the start of class. (Homework turned in after class begins will not be accepted.) Questions are drawn from the material in Sections 1.1 and 1.2 of the text on deterministic and nondeterministic finite automata.

The homework is worth 100 points. When questions with several parts do not specify the points for each part, each part has equal weight.

Remember to write on just one side of a page, do not use scrap paper, put your answers in the correct order, and staple your pages together. If you can't solve a problem, state this, and write only what you know to be correct. Neatness and conciseness count.

(1) **(Designing automata)** (20 points) Show that each of the following languages is regular.

(Below, xy denotes the concatenation of strings x and y. Also, c^i where c is a letter and i is a nonnegative integer, denotes the string consisting of i copies of letter c.)

(a) $\left\{ w \ : \ w \text{ is a base-10 number that is a multiple of } 3 \right\}$

(b) $\left\{ xy \ : \ x \text{ and } y \text{ are binary strings with a common substring of length } 3 \right\}$

(c) $\left\{ w \ : \ w \text{ is a binary string containing both substrings } \texttt{010} \text{ and } \texttt{101} \right\}$

(d) For any fixed integer $k \geq 0$,

$$\left\{ \texttt{0}^i w \texttt{1}^i \ : \ w \text{ is a binary string and } 0 \leq i \leq k \right\}$$

(Hint: Construct a DFA or an NFA that recognizes each language. To present an automaton, draw its transition diagram.)

(2) **(Column automata)** (20 points) For the alphabet consisting of all 2-row columns whose entries are 0 or 1,

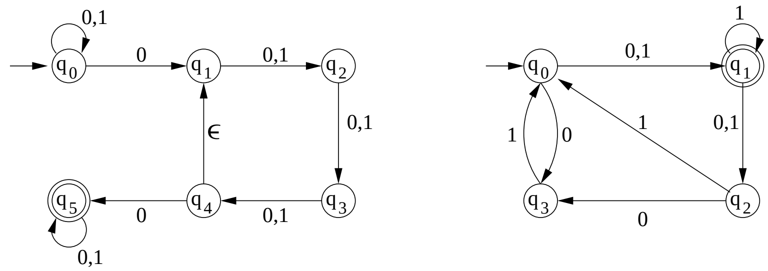
$$\Sigma \ := \ \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\},$$

show that the following languages are regular.

(a) $\left\{ w \ : \ \left(\begin{array}{l} w \text{ is a string over } \Sigma \text{ whose top and bottom rows have the same} \\ \text{parity} \end{array} \right) \right\}.$

(b) $\left\{ w \ : \ \left(\begin{array}{l} w \text{ is a string over } \Sigma \text{ where the binary integers } t \text{ and } b \text{ encoded} \\ \text{by the top and bottom rows, respectively, satisfy } t \leq b \end{array} \right) \right\}.$

(3) **(Removing nondeterminism)** (10 points) Convert each of the following NFAs to an equivalent DFA that recognizes the same language.



(a) (b)

(4) **(Variations on automata)** (20 points) The following questions ask you to consider the effect of altering the definition of an NFA or a DFA.

(a) Suppose we alter the definition of an NFA so that we now identify two types of states in its state set Q: the *good* states $G \subseteq Q$, and the *bad* states $B \subseteq Q$, where $G \cap B = \emptyset$. (Note that a state in Q may be neither good nor bad, but no state is both good and bad.) We also alter the definition of acceptance: the automaton now *accepts* input w if, considering all possible computations on w, *some* computation ends in G and *no* computation ends in B. Call this new type of NFA a **good-bad-NFA**.

Prove that the class of languages recognized by good-bad-NFAs is the class of regular languages.

(b) Suppose we alter the definition of a DFA so that once the automaton leaves its start state, it cannot return to its start state. For such a DFA, if an input w causes it to take a transition from a state p to its start state q, where $p \neq q$, then the DFA immediately halts and rejects w. Call this new type of DFA a **no-return-DFA**.

Prove that the class of languages recognized by no-return-DFAs is the class of regular languages.

(Hint: In each case, show that the variant can be converted to a standard automaton that recognizes the same language.)

(5) **(Closure properties)** (30 points) In the questions below, given a language L we describe how to form a new language from the strings in L. In each question, prove that if L is regular, then the new language is also regular.

(a) $\text{Skip}(L) \ := \ \Big\{ xy \ : \ xcy \in L \text{ where } x, y \text{ are strings and } c \text{ is a letter} \Big\}$

(b) $\text{Suffix}(L) \ := \ \Big\{ y \ : \ xy \in L \text{ where } x, y \text{ are strings} \Big\}$

(Hint: Given an automaton that recognizes L, show how to construct an automaton that recognizes the new language.)

(6) **(Restricting DFA acceptance) (bonus)** (10 points) Suppose we restrict DFAs so they have *at most one accepting state*. Can any regular language L be recognized by this restricted form of DFA? (Prove your answer.)

Note that Problem (6) is a bonus question. Bonus questions are not required (except for the honors section) and their points are not added to regular points.