There are a total of five problems. You have to solve the first four. Problem 5 is optional.

Problem 1 (CO1): DFA and Regular Languages (10 points)

Let $\Sigma = \{\texttt{0}, \texttt{1}\}$. Consider the following languages over Σ.

$$L_1 = \{w : w = \texttt{1}^m\texttt{0}^n, \text{ where } m, n \geq 0\}$$
$$L_2 = \{w : \texttt{1} \text{ does not appear at any even position in } w\}$$
$$L_3 = L_1 \cap L_2$$

Now solve the following problems.

(a) **Give** the state diagram for a DFA that recognizes L_1. (3 points)

(b) **Give** the state diagram for a DFA that recognizes L_2. (3 points)

(c) If you were to use the "cross product" construction shown in class to obtain a DFA for the language L_3, how many states would it have? (1 point)

(d) **Find** all four-letter strings in L_3. (1 point)

(e) **Give** the state diagram for a DFA that recognizes L_3 using only three states. (2 points)

Problem 2 (CO1): Regular Expressions (10 points)

Consider the following languages over $\Sigma = \{\texttt{0}, \texttt{1}\}$.

$$L_1 = \{w : w \text{ does not contain } \texttt{00}\}$$
$$L_2 = \{w : \text{every } \texttt{0} \text{ in } w \text{ is preceded by at least one } \texttt{1}\}$$
$$L_3 = \{w : \text{the number of times } \texttt{0} \text{ appears in } w \text{ is even}\}$$

Now solve the following problems.

(a) **Give** a regular expression for the language L_1. (2 points)

(b) Your friend claims that $L_1 = L_2$. **Prove** him wrong by writing down a five-letter string in $L_1 \setminus L_2$. Recall that $L_1 \setminus L_2$ contains all strings that are in L_1 but not in L_2. (2 points)

(c) **Give** a regular expression for the language $L_1 \setminus L_2$. (2 points)

(d) **Give** a regular expression for the language L_3. (2 points)

(e) **Give** a regular expression for the language $L_2 \setminus L_3$. (2 points)

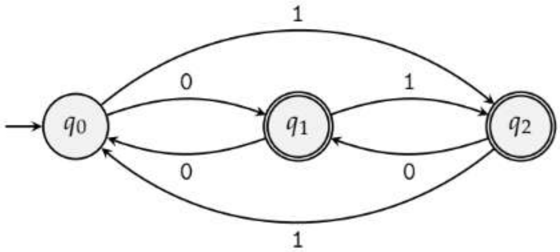
Problem 3 (CO3): Converting Regular Expressions to NFAs (10 points)

Convert the following regular expression over $\Sigma = \{\texttt{e}, \texttt{f}, \texttt{g}, \texttt{h}\}$ into an equivalent NFA. Note that $R_1 + R_2$ is the same as $R_1 \cup R_2$.

$$(\texttt{g}^* + \texttt{h})^* + \texttt{e}(\texttt{f}^*\texttt{g})^* + \texttt{eg}(\texttt{f}^*\texttt{g})^*$$

Problem 4 (CO3): Converting Finite Automata to Regular Expressions (10 points)

Convert the following DFA into an equivalent regular expression using the state elimination method. First eliminate q_2, then q_1, and finally q_0. You must show work.



Problem 5 (Bonus): Even-Odd Runs (5 points)

Disclaimer: This is a bonus problem. Attempt it only after you are done with everything else. Even if you do not attempt it, you can get a perfect score. So, do not worry if you find it too hard!

A *run* in a string is a maximal non-empty substring consisting entirely of the same letter. For example, the string `111001111011` consists of five runs: a run of three `1`s, followed by a run of two `0`s, a run of four `1`s, a run of one `0`, and a run of two `1`s. Consider the following language over $\{0, 1\}$.

$$L = \{w : \text{every run of } 0\text{s in } w \text{ has odd length and every run of } 1\text{s in } w \text{ has even length}\}$$

Give a six-state DFA that recognizes L.

After you are done with the test, please indicate where you stand on the smiley face spectrum.