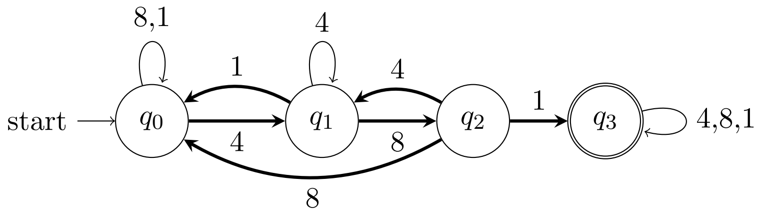
COMS 331: Theory of Computing, Spring 2023
Homework Assignment 2

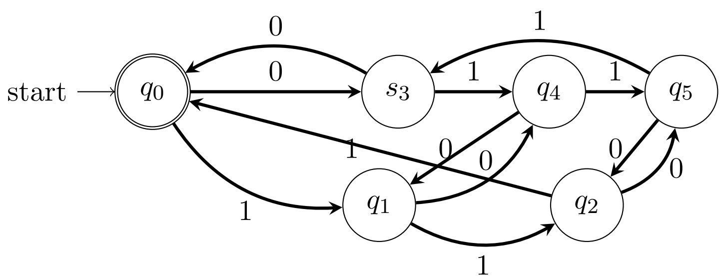
Neha Maddali

Due at 10:00PM, Thursday, February 9, on Gradescope.

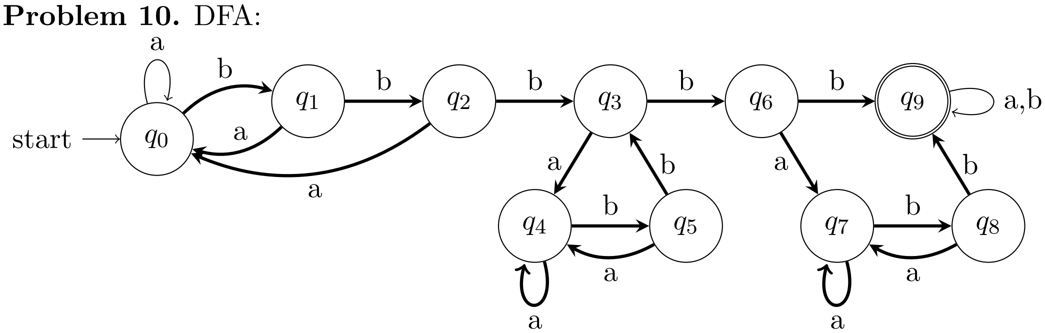
Problem 8. DFA:



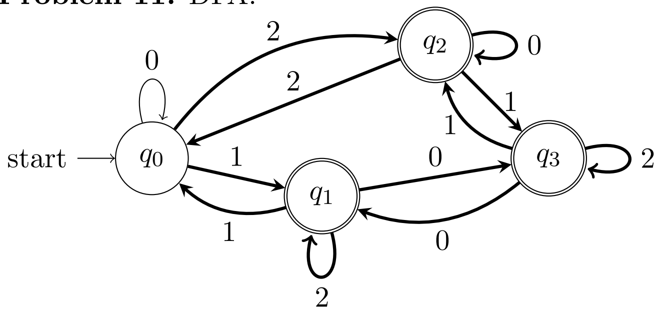
Problem 9. DFA:



Problem 10. DFA:



Problem 11. DFA:



Problem 12.
Proof by induction on y.
BASE CASE: Let $y = \epsilon$. On the left side, we have $\hat{\delta}(q, x\epsilon)$, which is $\hat{\delta}(q, x)$. On the right side, we have $\hat{\delta}(\hat{\delta}(q, x), \epsilon)$ and since the null string means the state doesn't change, we have $\hat{\delta}(q, x)$. Then $\hat{\delta}(q, x) = \hat{\delta}(q, x)$
INDUCTION HYPOTHESIS: Assume $\hat{\delta}(q, xy) = \hat{\delta}(\hat{\delta}(q, x), y)$ holds for $y \in \Sigma^*$.
INDUCTION STEP: Let $x, y \in \Sigma^*, q \in Q$ and assume that the induction hypothesis is true. Then we will show that the result holds for ya where $a \in \Sigma$. Then on the left side we have $\hat{\delta}(q, xya) = \delta(\hat{\delta}(q, xy), a)$ and by the induction hypothesis, $\delta(\hat{\delta}(\hat{\delta}(q, x), y), a) = \hat{\delta}(\hat{\delta}(q, x), ya)$ which is what we are trying to prove. ■

Problem 13. Prove that every full prefix-free language is maximal.

Let A be some full prefix-free language that is not maximal. Since A is full, we have $\sum_{x \in A} 2^{-|x|} = 1$. By the definition of "maximal," this means that A is a proper subset of another prefix-free language, and we can call that prefix-free language B and B is also maximal. Then there must be some string y such that $A \cup \{y\} = B$. Then we have $\sum_{x \in A} 2^{-|x|} + 2^{-|y|}$ which is > 1 because $1 + 2^{-|y|} > 1$. This is a contradiction because for B to be a prefix-free language, $\sum_{x \in B} 2^{-|x|} \leq 1$ by the Kraft inequality, but $\sum_{x \in B} 2^{-|x|} > 1$ because $\sum_{x \in A} 2^{-|x|} + 2^{-|y|} > 1$. Thus every full prefix-free language is maximal. ■

Problem 14. How many DFAs $M = (Q, \Sigma, \delta, s, F)$ are there with $\Sigma = \{0, 1\}$ and $Q = \{1, 2, ..., n\}$?

Number of possible start states: n. This is because there are n states and any one of them can be a start state.
Number of possible final states: 2^n. This is because when we consider the combination of all possible final states (0 final states, then 1 final state, all the way up to n final states), that summation is the n^{th} row of Pascal's triangle, which sums up to 2^n. It's basically the cardinality of the power set of Q.
Number of possible transitions: n^{2n}. This is because δ is defined as a function that maps $Q \times \Sigma \rightarrow Q$ and $|Q| = n$ and $|\Sigma| = 2$. Each of the n states have 2 symbols that could map to any of the n states, resulting in n^{2n}.
Total DFAs: $n * 2^n * n^{2n} = n^{2n+1} * 2^n$

Problem 15. Prove that there is a language $A \subseteq \{0, 1\}^*$ with both of the following properties:

(i) For all $x \in A$, $|x| \leq 5$.

(ii) Every DFA that decides A has more than 8 states.

Let us define a language L where L = $\{z \in \{0,1\}^* |$ for every DFA M that decides z, the # of states in M is $> 8\}$.
Let us show that language L satisfies property i. L allows for us to include strings of length 5 or less. There are $2^0 + 2^1 + 2^2 + 2^3 + 2^4 + 2^5 = 63$ strings of length ≤ 5, so property i is satisfied. 63 strings of length ≤ 5 means there are 2^{63} languages satisfying property i.
Let us show that language L satisfies property ii. Prove using contradiction. Suppose there exists a DFA M with states ≤ 8 that decides L. Using the formula from problem 14, there are $8^{2(8)+1} * 2^8 = 2^{59}$ DFAs with 8 states. Because of $2^{59} < 2^{63}$, there is at least one language that satifies property i that doesn't satisfy property ii, so a contradiction is made and property ii is proved.
There is a language $A \subseteq \{0,1\}^*$ with properties i and ii. ■