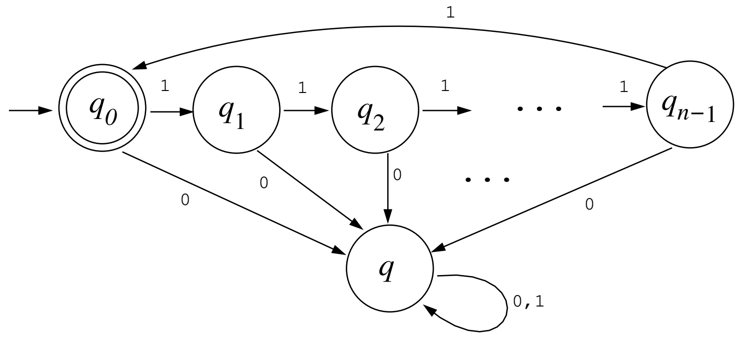
Problem Set 2 Solutions

Problem 1.

Part A. *Show that there is a deterministic finite automata with $n+1$ states that recognizes the language $(1^n)^*$. (The alphabet is $\Sigma = \{0,1\}$.)*



Part B. *Show that there does not exist a smaller deterministic finite automaton for this language.*

Assume to the contrary that there exists a DFA $M = (Q, \Sigma, \delta, q_0, F)$ with n or fewer states such that $L(M) = L_n$. Consider the $n+1$ strings $x_{-1} = 0$, $x_i = 1^i$, for $0 \le i \le n-1$. By the pigeonhole principle, $\delta^*(q_0, x_i) = \delta^*(q_0, x_j)$ for some $-1 \le i < j \le n-1$. But then $\delta^*(q_0, x_i 1^{n-j}) = \delta^*(x_j 1^{q_0, n-j})$. However, $x_j 1^{n-j} \in L_n$ while $x_i 1^{n-j} \notin L_n$. Thus the machine M has "made a mistake" on either $x_j 1^{n-j}$ or $x_i 1^{n-j}$, since it either accepts both of these strings or neither of them. This contradicts the assumption that $L(M) = L_n$.

Problem 2. *Suppose that L is DFA-acceptable. Show that the following language is DFA acceptable, too:*

$$Max(L) = \{x \in L : \text{ there does not exist a } y \in \Sigma^+ \text{ for which } xy \in L\}.$$

Given a finite automaton $M = (Q, \Sigma, \delta, q_0, F)$ for L, a finite automaton $M = (Q, \Sigma, \delta, q_0, F')$ is constructed for $Max(L)$ by "pruning" the final state set; we define F' to be the set of all states $q \in F$ such that there exists *no* nontrivial path from q to some final state of M. Then $x \in L(M')$ iff $x \in L$ and there is no $y \in \Sigma^+$ such that $xy \in L(M)$.

Problem 3. *Same instructions as the last problem, for:*

$$Echo(L) = \{a_1 a_1 a_2 a_2 \cdots a_n a_n \in \Sigma^* : a_1 a_2 \cdots a_n \in L\}.$$

The idea is to add a state "in the middle of each arrow," to ensure that a symbol $a \in \Sigma$ is always followed by another symbol a, and the same destination is then reached. A dead state is also added, in case the symbol a is not followed by a. More formally, let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA accepting L. Then a DFA $M' = (Q', \Sigma, \delta', q_0', F')$ is constructed for $Echo(L)$ by setting

$$Q' = Q \cup Q \times \Sigma \cup \{dead\},$$

$$\delta'(q, \sigma) = (q, \sigma) \text{ for } q \in Q,$$

$$\delta'((q,\sigma),\tau) = \begin{cases} \delta(q,\tau) & \text{if } \sigma = \tau \\ dead & \text{if } \sigma \neq \tau \end{cases},$$

$$q_0' = q_0, \text{ and}$$

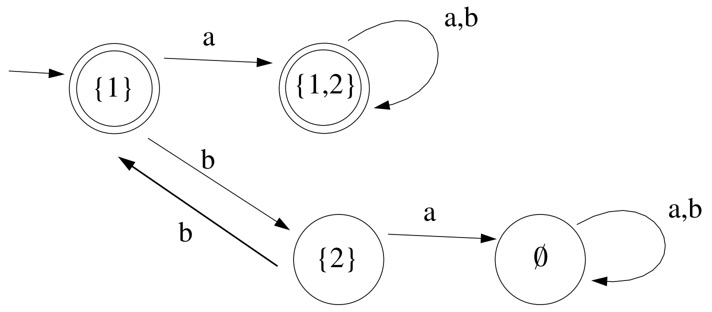
$$F' = F.$$

It easy to see that this construction is correct; it can be formally argued by induction.

Problem 4. *Page 85, Exercise 1.12.*

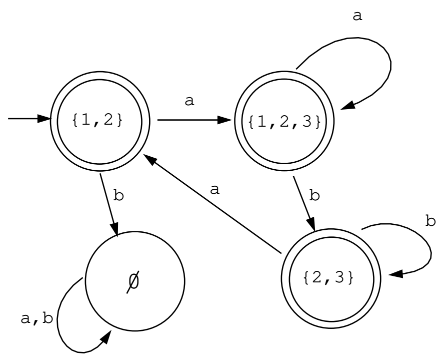
Part A.

state	a	b
$\{1\}$	$\{1,2\}$	$\{2\}$
$\{1,2\}$	$\{1,2\}$	$\{1,2\}$
$\{2\}$	$\emptyset$	$\{1\}$
$\emptyset$	$\emptyset$	$\emptyset$



Part B.

Similarly ...



Problem 5. *Construct a regular expression for each of the languages from Problem Set 1, problem 1:*

1. The set of all strings with 010 as a substring

This one is easy! — $(0 \cup 1)^*010(0 \cup 1)^*$.

2. The set of all strings which do not have 010 as a substring

I think it's easiest to do this one by looking at the DFA in the solution to PS #1 and converting it to a regular expression using the method described in class. Don't think too much! Of course you will get different answers depending on the order in which you kill states, combine parallel arcs, and eliminate obvious redundancies. Here is one answer:

$(1 \cup 00^*11)^*(\varepsilon \cup 00^* \cup 00^*1)$

3. The set of all strings which have an even number of 0's or an even number of 1's

$(1^*01^*01^*)^* \cup (0^*10^*10^*)^*$

4. The complement of $\{1, 10\}^*$

Again, consult the solution to PS #1 for a DFA for this language. By inspection, we can turn it into a regular expression:

$(11^*0)^*0(0 \cup 1)^*$

5. The binary encodings of numbers divisible by 3

Consult the solution to PS#1 for a DFA. Then convert to get:

$(0 \cup 1(01^*0)^*1)^*$