Harvard University
Computer Science 121

Problem Set 1

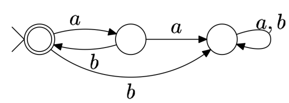
Due Tuesday, September 16 at 11:59 PM.
Submit your solutions electronically to cs121+ps1@seas.harvard.edu with "ps1 submission" in the subject line. The solutions to each part should be attached as separate PDF files, called lastname+ps1a.pdf, lastname+ps1b.pdf, and lastname+ps1c.pdf.
Late problem sets may be turned in until Friday, September 20 at 11:59 PM with a 20% penalty.
See syllabus for collaboration policy.

PART A (Graded by Louis)

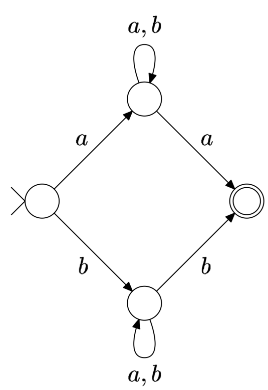
PROBLEM 1 (5+5 points)

Informally describe the languages recognized by the following finite automata:

(A) This DFA:



(B) This NFA:



(A) The set of strings which consist of 0 or more repetitions of ab. More formally, $L = \{ab\}^*$.

(B) The set of strings which begin and end with the same symbol and contain at least two symbols. More formally, $L = \{x : x \in \{a, b\}^*$, x begins and ends with the same symbol, and $|x| \geq 2$ $\}$.

PROBLEM 2 (5 points)

Write the formal description $(Q, \Sigma, \delta, q_0, F)$ for the DFA in Problem 1(A).

$M = (Q, \Sigma, \delta, q_0, F)$, where:
$Q = \{q_0, q_1, q_2\}$,
$\Sigma = \{a, b\}$,
$F = \{q_0\}$,
and δ is given by the table:

δ	a	b
q_0	q_1	q_2
q_1	q_2	q_0
q_2	q_2	q_2

PROBLEM 3 (5 points)

Use the subset construction to make a DFA for the NFA in Problem 1(B). Show your work.

Let the states be labeled 0, 1, 2, 3 as shown above. The powerset of the set of states Q is:

$$\mathcal{P}(Q) = \{\varnothing, \{0\}, \{1\}, \{2\}, \{3\}, \{0,1\}, \{0,2\}, \{0,3\}, \{1,2\}, \{1,3\}, \{2,3\},$$
$$\{0,1,2\}, \{0,1,3\}, \{0,2,3\}, \{1,2,3\}, \{0,1,2,3\}\}$$

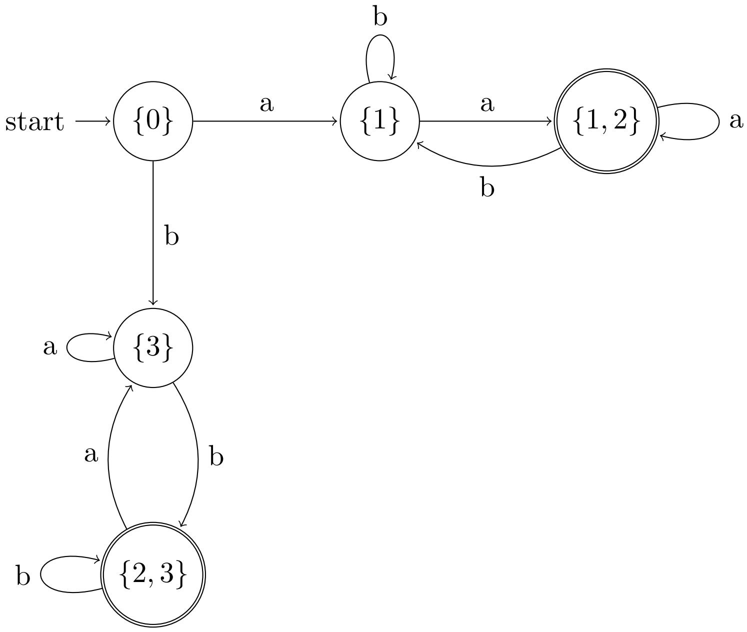
The set of final states in the newly constructed DFA will be any state that contains an accept state in the NFA, which is state 2.

We now examine the NFA M above to construct the following transition function for our newly constructed DFA, M'. This is constructed by taking the union of the transition function for M as described in Sipser, Theorem 1.39 (page 55).

δ	a	b
$\varnothing$	$\varnothing$	$\varnothing$
$\{0\}$	$\{1\}$	$\{3\}$
$\{1\}$	$\{1,2\}$	$\{1\}$
$\{2\}$	$\varnothing$	$\varnothing$
$\{3\}$	$\{3\}$	$\{2,3\}$
$\{0,1\}$	$\{1,2\}$	$\{1,3\}$
$\{0,2\}$	$\{1\}$	$\{3\}$
$\{0,3\}$	$\{1,3\}$	$\{2,3\}$
$\{1,2\}$	$\{1,2\}$	$\{1\}$
$\{1,3\}$	$\{1,2,3\}$	$\{1,2,3\}$
$\{2,3\}$	$\{3\}$	$\{2,3\}$
$\{0,1,2\}$	$\{1,2\}$	$\{1,3\}$
$\{0,2,3\}$	$\{1,3\}$	$\{2,3\}$
$\{0,1,3\}$	$\{1,2,3\}$	$\{1,2,3\}$
$\{1,2,3\}$	$\{1,2,3\}$	$\{1,2,3\}$

We remove any unreachable states, which includes any state including a 0 (except the start state) and states that are only reachable through states containing 0 ($\{1,3\}, \{1,2,3\}$. Looking at the table, we can see that the transition function never yields the any of the states that contain 0, corresponding to the notion that no arrows ever enter the start state in the original NFA.

This gives us the desired NFA M':



PART B (Graded by Francisco)

PROBLEM 4 (5 + 5 + 5 points)

Are the following statements true or false? Justify your answers with a proof or counterexample.

(A) For any languages L_1 and L_2, $(L_1 \cap L_2)^* = L_1^* \cap L_2^*$.

(B) For any languages L_1 and L_2, $(L_1 \cup L_2)^* = L_1^* \cup L_2^*$.

(C) Complementing all states in a DFA M (making the final states non-final and vice-versa) will result in a new DFA M' such that $L(M') = \Sigma^* - L(M)$.

(A) *False.* Let $L_1 = \{a\}$ and $L_2 = \{aa\}$. $(L_1 \cap L_2)^* = \emptyset^* = \{\epsilon\}$. But $aa \in (L_1^* \cap L_2^*)$.

(B) *False.* Let $L_1 = \{a\}$ and $L_2 = \{b\}$. Then $(L_1 \cup L_2)^* = \{a, b\}^*$, or Σ^*. On the other hand, $L_1^* \cup L_2^* = \{\epsilon, a, aa, aaa \ldots\} \cup \{\epsilon, b, bb, bbb \ldots\}$, which is the set of strings consisting of either any number of $a's$ or any number of $b's$, but not strings containing both a's and b's. This set is clearly not equal to Σ^*.

(C) *True.* We need to show that for every $w \in \Sigma^*, w \in L(M)$ if and only if $w \notin L(M')$. In a DFA, each string has only one computational path, which ends in a single state. Suppose that $w \in L(M)$. Then the computation on w ends in a final state s in M. By construction, s is not final in M', and therefore $w \notin L(M')$. Similarly, suppose $w \notin L(M)$. Then the computation on w ends in a state non-final t in M. By construction, t is final in M', and therefore $w \in L(M')$.

PROBLEM 5 (5+5 points)

An NFA M contains a *cycle* if there is a state q and a string x such that if M is in state q and reads string x, M can return to state q. Prove or disprove the following statements:

(A) If M recognizes an infinite language, then M has a cycle.

(B) If M has a cycle, then M recognizes an infinite language.

(A) *Proof by contradiction.* Suppose the NFA M has n states and recognizes an infinite language but does not contain a cycle. Since $L(M)$ is infinite, there exists a string $w \in L(M)$ of length greater than n. (Infinite languages do not have maximum length strings, so for any fixed value there must be a string of greater length in the language.) Now consider an accepting computation path of w in M. That path must contain $|w| > n$ states. By the pigeonhole principle, at least one state q must appear at least twice in the computation path. However, this would require a cycle, a contradiction.

A common source of confusion in this problem was the length of the strings in an infinite language. While an infinite language contains an infinite number of strings, no string is of infinite length (all strings have finite length) and strings of a particular length are not guaranteed to be in the language (consider a language with no strings of length 10 in it). However, there is always a string whose length is greater than some arbitrarily-chosen integer.

(B) *False.* Consider an NFA N consisting of only a single state which is its start state but not a final state. N accepts no strings, not even the empty string. Now construct a transition from this state to itself for any element of the alphabet. This is a cycle but the language of N is unchanged and certainly not infinite.

PART C (Graded by Perry)

PROBLEM 6 (5+15 points)

For any language L, let $ExtraB(L) = \{sbt : s, t \in \{a,b\}^* \text{ and } st \in L\}$.

(A) What is $ExtraB(\{aba\}^*)$?

(B) Show that if L is regular, then so is $ExtraB(L)$. (Hint: Show how, given a DFA for L, you can construct an NFA for $ExtraB(L)$).

(A) $ExtraB(\{aba\}^*) = \{aba\}^*(\{b\} \cup \{abba\})\{aba\}^*$

(B) Suppose L is regular. So, there is some DFA $D = \{Q, \Sigma, \delta, q_0, F\}$ that decides L. Let $N = \{Q', \Sigma, \delta', q_0', F'\}$, where:

$Q' = Q \times \{0,1\}$
$\delta'((q,i),a) = \{(\delta(q,a),i)\}$ for $i \in \{0,1\}$
$\delta'((q,0),b) = \{(\delta(q,b),0),(q,1)\}$
$\delta'((q,1),b) = \{(\delta(q,b),1)\}$

$q'_0 = (q_0, 0)$
$F' = \{(q, 1) : q \in F\}$
NFA N recognizes $ExtraB(L)$:

Informally: N has 2 'copies' of the states of D. When in the starting copy, whenever N reads a b, it can nondeterministically choose to jump to the second copy instead of advancing as normal through the first copy. The final states are just the versions of the final states in the second copy. So, the strings accepted by N are any strings accepted by D, with the insertion of a b at any point in the string.

Formally: Suppose $w \in L(N)$. Call states of the form (q, i) i-states. Since the computation of w on N begins in a 0-state and ends in a 1-state, at some point it must transition from a 0-state to a 1-state, which can only happen from reading a b. Also, by construction, it is not possible to transition back from a 1-state to a 0-state. So, we can partition w into sbt, where s is the portion of the string read while the computation is in a 0-state, t is the portion of the string read while the computation is in a 1-state (and the b is the character that, when read, transitions from a 0-state to a 1-state). Let $q'_i = (q_i, 0)$ be the state that N is in after reading s in the accepting computation of w. Then, by construction, D is in state q_i after reading s, and when in q_i transitions to an accepting state after reading t. So, $st \in L(D) = L$.

Conversely, suppose $w = st \in L$. Let q_i be the state that D is in after reading s. Then an accepting computation of sbt on N is to transition to $(q_i, 0)$ after reading s, $(q_i, 1)$ after reading the next b, and then transition to $(q_f, 1) \in F'$ after reading t. Thus, $sbt \in L(N)$ as desired.

Since $w \in L(N) \iff w = sbt$ and $st \in L$, N is an NFA for $ExtraB(L)$, and so $ExtraB(L)$ is regular if L is, as desired.

PROBLEM 7 (3 bonus points)

CHALLENGE: For any string s and language L, let $P_s(L) = \{t : st \in L\}$. Show that L is regular if and only if there are finitely many unique sets $P_s(L)$.

$\rightarrow$: Suppose L is regular, let $M = (Q, \Sigma, \delta, q_0, F)$ be its DFA. Observe that for any s, $P_s(L) = L(Q, \Sigma, \delta, \delta^*(q_0, s), F)$:

$$w \in P_s(L) \iff sw \in L \iff \delta^*(q_0, sw) \in F \iff \delta^*(\delta^*(s, q_0), w) \in F \iff w \in L(Q, \Sigma, \delta, \delta^*(q_0, s), F)$$

So, $P_s(L)$ is just the language of the same DFA as L, but with the start state changed. So, since there are only a finite number of states in the DFA that can be the start state, the number of unique sets $P_s(L)$ is finite.

$\leftarrow$: Suppose there are a finite number of unique sets $P_s(L)$ for some L. Construct DFA M as follows:

$Q = \{q_{P_s(L)} : s \in \Sigma^*\}$ (note that the subscripts $P_s(L)$ should be interpreted extensionally)
$\delta(q_{P_s(L)}, \sigma) = q_{P_{s\sigma}(L)}$
$q_0 = q_{P_\epsilon(L)}$
$F = \{q_{P_s(L)} : \epsilon \in P_s(L)\}$

Observe that $L(M) = L$:

$$w \in L(M) \iff \delta^*(q_{P_\epsilon(L)}, w) \in F \iff q_{P_w(L)} \in F \iff \epsilon \in P_w(L) \iff \epsilon \in \{t : wt \in L\} \iff w \in L$$

So, L is regular as desired.