INSTRUCTIONS (read carefully)

- Fill in your name, netid, and discussion section time below. Also write your netid on the other pages (in case they get separated).

NAME:	
NETID:	DISC:

- There are 6 problems. Make sure you have a complete exam.
- The point value of each problem is indicated next to the problem, as well as in the table below.
- Points may be deducted for solutions which are correct but excessively complicated, hard to understand, or poorly explained. Please keep your solutions ***short*** and crisp.
- The "I DON'T KNOW" rule does apply. If you do not know the answer to a problem, you can simply write "I DON'T KNOW" and you will get 20% of credit for that problem. The number of points you get in this manner cannot exceed 10 points across the whole exam.
- The exam is designed for one hour and thirty minutes, but you have the two full hours to finish it.
- It is wise to skim all problems and point values first, to best plan your time.
- This is a closed book exam. No notes of any kind are allowed. Do all work in the space provided, using the backs of sheets if necessary. See the proctor if you need more paper.
- Please bring any apparent bugs to the attention of the proctors.
- After the midterm is over, discuss its contents with other CS 373 students **only** after verifying that they have also taken the exam (e.g. they aren't about to take the conflict exam).
There are many people taking the conflict exam— so please make sure before you discuss!
- We indicate next to each problem how much time we suggest you spend on it. We also suggest you spend the last 25 minutes of the exam *reviewing* your answers.

Problem	Possible	Score
1	10	
2	15	
3	15	
4	20	
5	20	
6	20	
Total	100	

Problem 1: Yea or Nay (10 points)

[10 minutes]

The answers to these problems should be just choosing True or False; no other explanation is necessary.

(A) For any regular languages L_1 and L_2 over Σ, $L_1 \setminus L_2$ is also regular?

☐ True ☐ False

Solution:

True. $L_1 \setminus L_2 = L_1 \cap \overline{L_2}$ and we know regular sets are closed under intersection and complementation.

(B) Let $L \subseteq \Sigma^*$ be a finite language. Then $\overline{L}$ is finite.

☐ True ☐ False

Solution:

False. $L = \varnothing$ is a counter-example.

(C) Let $L \subseteq \Sigma^*$ be an infinite language. Then $\overline{L}$ is finite.

☐ True ☐ False

Solution:

False. $\Sigma = \{0\}$ and $L = \{0^{2i} \mid i \geq 0\}$ is a counter-example.

(D) Let a DFA have a transition function $\delta : Q \times \Sigma \to Q$. Then $\delta^* : Q \times \Sigma^* \to Q$.

☐ True ☐ False

Solution:

True by definition of δ^*.

(E) The set of regular languages over $\Sigma = \{a\}$ is finite.

☐ True ☐ False

Solution:

False. All of these singleton sets are regular: $\{a\}, \{aa\}, \{aaa\}, \dots$.

(F) If $X, Y \subseteq Z$ and $X \subseteq Y$, then $(Z \setminus Y) \subseteq (Z \setminus X)$.

☐ True ☐ False

Solution:

True. $\overline{Y} \subseteq \overline{X}$. Now take intersecion of both sides with Z.

(G) The language of the regular expression $\emptyset a^*(a+b)$ is empty.

☐ True ☐ False

Solution:

True. Since $\emptyset$ concatenated to any language is empty.

(H) $\emptyset^* = \emptyset$

☐ True ☐ False

Solution:

False. $\epsilon \in \emptyset^*$.

(I) Let L be a regular language and $L' \subseteq L$. Then L' is also regular.

☐ True ☐ False

Solution:

False. Pick $L = \Sigma^*$ and L' to be a non-regular language.

(J) Let $L = L_1 \cup L_2$ and let L and L_1 be regular languages. Then L_2 is a regular language.

☐ True ☐ False

Solution:

False. Pick $L = L_1 = \Sigma^*$ and L_2 to be a non-regular language.

Problem 2: DFA design (8+7 points)

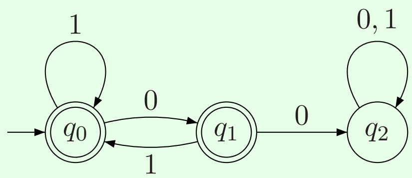
[10 minutes]

Design a DFA for the following language L over the alphabet $\Sigma = \{0, 1\}$:

$$L = \{w \in \{0,1\}^* \mid \ between\ any\ two\ 0s\ of\ w,\ there's\ at\ least\ one\ 1\}$$

(a) Draw a diagram depicting a DFA for L

Solution:



(b) Give a formal description of the DFA as the tuple $A = (Q, \Sigma, \delta, q_0, F)$. Formally define all the 5 components of this tuple.

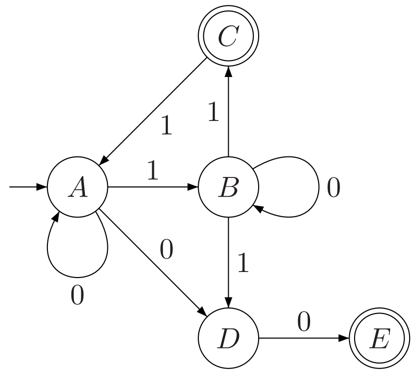
Solution:

$A = (Q, \Sigma, \delta, q_0, F)$
$Q = \{q_0, q_1, q_2\}$
$\Sigma = \{0, 1\}$
$q_0 = q_0$
$F = \{q_0, q_1\}$
δ is defined as:

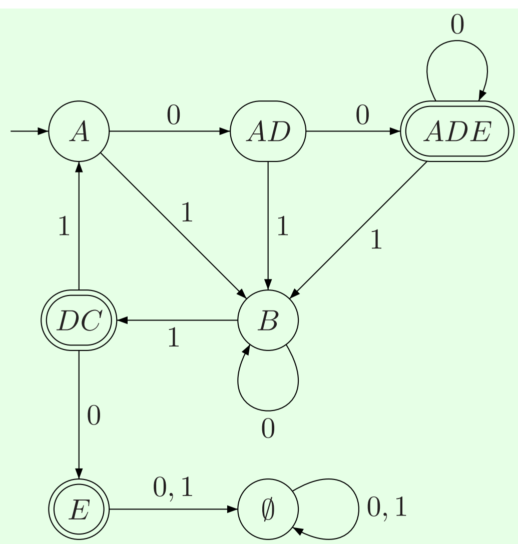
$$\delta(q, c) = \begin{cases} q_1 & \text{if } c = 0 \text{ and } q = q_0 \\ q_2 & \text{if } c = 0 \text{ and } q = q_1 \\ q_0 & \text{if } c = 1 \text{ and } q \neq q_2 \\ q_2 & \text{if } q = q_2 \end{cases}$$

Problem 3: NFA to DFA (15 points)

[15 minutes]
Convert the following NFA (over the alphabet $\Sigma = \{0, 1\}$) to a DFA using the subset construction. Label each state of the DFA appropriately to indicate which states of the NFA it corresponds to. (You just need to draw a diagram for the DFA; and you need not depict states that are unreachable from the initial state.)



Solution:



Problem 4: Automata Transformation (20 points)

[15 minutes]
Let $\Sigma = \{a_0, a_1, b_0, b_1\}$ and let $\Sigma' = \{a, b\}$.
If x is a string in Σ^*, then we define $p(x)$ to be the string obtained from x by replacing every a_0 and every a_1 in x with an a, and every b_0 and b_1 in x with a b. Hence for every $x \in \Sigma^*$, $p(x) \in \Sigma'^*$.

For example, $p(a_0a_1b_0a_1b_1) = aabab$.

Let L be a regular language over the alphabet Σ. Prove that the following language L' (over the alphabet Σ') is regular:

$$L' = \{p(x) \mid x \in L\}$$

(Give formal constructions and a proof.)

Proof:

Solution:

The idea is that we take a DFA of L and modify it such that every transition with label x will get a new label: $p(x)$. This modification generates an NFA. We note that if string x has a path that travels from the initial state of the DFA to its final state, then the same path in the NFA will accept $p(x)$. The reverse is also true. So let's make these observations precise.

Let D be a DFA such that $L(D) = L$ (L is regular so D exists by definition). We prove the claim by explicitly building an NFA N such that $L' = L(N)$.
Let $N = (Q, \Sigma', \delta', q_0, F)$ where:

$$\forall q \in Q, \quad \delta'(q, a) = \{\delta(q, a_0), \delta(q, a_1)\} \qquad \delta'(q, b) = \{\delta(q, b_0), \delta(q, b_1)\}$$

First we prove $L' \subseteq L(N)$: Let y be an arbitray member of L'. By definition of L' we have $y = p(x_0x_1 \ldots x_k) = p(x_0)p(x_1) \ldots p(x_k)$, where $x_0 \cdots x_k \in L$. Since $x_0 \cdots x_k \in L'$, we know that D accepts $x_0 \cdots x_k$. By definition of acceptance in a DFA, this means that we have a sequence of states $q_0, q_1, \cdots, q_{k+1}$ in D such that $\delta(q_i, x_i) = q_{i+1}$ for $0 \leq i \leq k$ and $q_{k+1} \in F$. Note that by definition of δ', we have $q_{i+1} \in \{q_{i+1}\} \subseteq \delta'(q_i, p(x_i))$. Therefore the same sequence $q_0, \ldots, q_{k+1}$ in N proves that N accepts $p(x_0) \ldots p(x_k) = y$ (by definition of acceptance in NFA's).

Finally we prove $L(N) \subseteq L'$: Let y be an arbitray member of $L(N)$. Let $y = y_0 \ldots y_k$ where $y_i \in \Sigma'$. Since N accepts y, by the definition of acceptance in NFA's, we have a sequence of $k+1$ states $q_0, \ldots, q_{k+1}$ where $q_{i+1} \in \delta'(q_i, y_i)$ for $0 \leq i \leq k$ (Note that by construction N contains no ϵ-transitions, so each state transition consumes exactly one symbol of y) and $q_{k+1} \in F$. Since $q_{i+1} \in \delta'(q_i, y_i)$, by definition of δ', there exists $x_i \in \Sigma$ such that $p(x_i) = y_i$ and $q_{i+1} = \delta(q_i, x_i)$ (we can prove this by taking two cases: Case 1 where $y_i = a$ and Case 2 where $y_i = b$.
In Case 1, $q_{i+1} = \delta'(q_i, a) = \{\delta(q, a_0), \delta(q, a_1)\}$, therefore either $q_{i+1} = \delta(q, a_0)$ or $q_{i+1} = \delta(q, a_1)$. Set $x_i = a_0$ in the former and $x_i = a_1$ otherwise.
Case 2 is similar to Case 1).
Now if we look at the sequence of states $q_0, \ldots, q_{k+1}$ in D, by definition of acceptance in DFA's we observe that D accepts $x = x_0 \ldots x_{k+1}$ (key observations: $q_{i+1} = \delta(q_i, x_i)$ and $q_{k+1} \in F$) and therefore $x \in L(D) = L$. But since for all i we have $p(x_i) = y_i$, we know that $p(x) = y$ (by definition of p for strings). Therefore by definition of L', we have $y \in L'$.

Problem 5. (20)

[20 minutes]
Let Σ be an alphabet with $\mathtt{a} \in \Sigma$.
Let L be a regular language over Σ.
Show that $L' = \{w \in \Sigma^* \mid \exists w_1, w_2 \in L, w = w_1 w_2,\ \textit{and } w \textit{ has at least two } \mathtt{a}'s\}$ is regular.

You need to give a formal proof of the above.

Solution:

Note that we can write

$$L' = \{w \in \Sigma^* \mid \exists w_1, w_2 \in L, w = w_1 w_2\} \cap \{w \in \Sigma^* \mid w \text{ contains at least two } a\text{'s}\}.$$

By definition the first set is just $L \cdot L$, which is regular since L is regular and regular languages are closed under concatenation. It is easy to see that the second set is just $L_1 = L(\Sigma^* a \Sigma^* a \Sigma^*)$ and therefore is regular. But we know that regular languages are closed under intersection too, therefore $(L \cdot L) \cap L_1 = L'$ is a regular language.

Problem 6. Proofs, proofs, proofs... (20 points)

[20 minutes]
Let L be the language of the regular expression $a^* b^*$.
Prove that *any* DFA for L must have at least two final states.

(A formal proof is required.)

Solution:

Let A be a DFA for $a^* b^*$, with transition function δ and initial state p_0.

Since $\epsilon \in L$, the initial state p_0 of A must be a final state. Also, since ab is accepted by the DFA A (as $ab \in L$), A must reach a final state after reading ab, say p_1 (i.e. $\delta^*(p_0, ab) = p_1$). We argue that p_0 must be different from p_1.

Assume, by way of contradiction, that $p_0 = p_1$. Consider the word $abab$. Then, since from p_0, reading ab, A reaches p_0, it follows that A would reach p_0 after reading $abab$ as well, and hence A will accept $abab$. This is a contradiction since $abab$ is not accepted by A (as $abab \notin L$).

Since p_0 and p_1 are different and are both final states, we have proved that every DFA for L has at least two final states.