CS 373, Spring 2009. Mock midterm 1 (Based on first midterm in CS 273, Fall 2008.)

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 7 problems, on pages numbered 1 through 7. Make sure you have a complete exam.
- The point value of each problem is indicated next to the problem, and in the table below.
- Points may be deducted for solutions which are correct but excessively complicated, hard to understand, or poorly explained.
- The exam is designed for one hour, but you have the full two 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 273 students **only** after verifying that they have also taken the exam (e.g. they aren't about to take the conflict exam).

Problem	Possible	Score
1	8	
2	6	
3	8	
4	6	
5	6	
6	8	
7	8	
Total	50	

Problem 1: Short Answer (8 points)

The answers to these problems should be short and not complicated.

(a) If an NFA M accepts the empty string (ϵ), does M's start state have to be an accepting state? Why or why not?

(b) Is every finite language regular? Why or why not?

(c) Suppose that an NFA $M = (Q, \Sigma, \delta, q_0, F)$ accepts a language L. Create a new NFA M' by flipping the accept/non-accept markings on M. That is, $M' = (Q, \Sigma, \delta, q_0, Q - F)$. Does M' accept $\overline{L}$ (the set complement of L)? Why or why not?

(d) Simplify the following regular expression $\emptyset^*(a \cup b) \cup \emptyset b^* \cup \epsilon abb$

Problem 2: DFA design (6 points)

Let $\Sigma = \{a, b\}$. Let L be the set of strings in Σ^* which contain the substring *bba* or the substring *aaa*.

For example, $aabba \in L$ and $baaab \in L$, but $babab \notin L$. Strings shorter than three characters are never in L.

Construct a DFA that accepts L and give a state diagram showing **all** states in the DFA.

You will receive **zero** *credit if your DFA uses more than* **10** *states or makes significant use of non-determinism.*

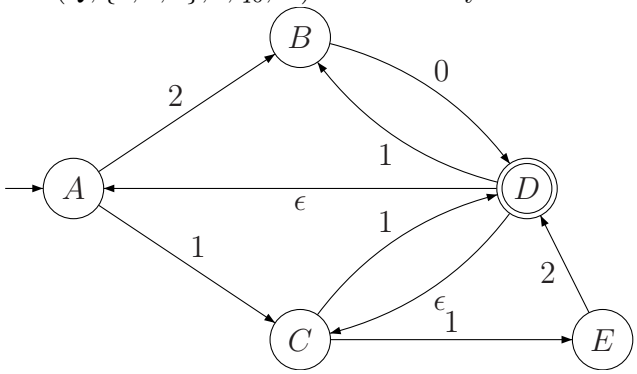
Problem 3: Remembering definitions (8 points)

(a) Define formally what it means for a DFA $(Q, \Sigma, \delta, q_0, F)$ to accept a string $w = w_1 w_2 \dots w_n$.

(b) Let Σ and Γ be alphabets. Suppose that h is a function from Σ^* to Γ^*. Define what it means for h to be a homomorphism.

Problem 4: NFA transitions (6 points)

Suppose that the NFA $N = (Q, \{0, 1, 2\}, \delta, q_0, F)$ is defined by the following state diagram:



Fill in the following values:

(a) $F =$

(b) $\delta(A, 0) =$

(c) $\delta(C, 1) =$

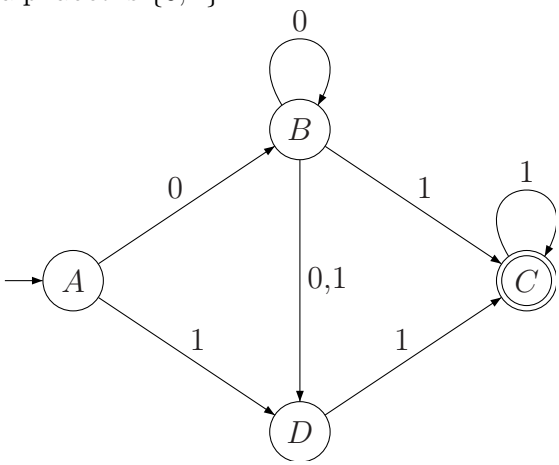
(d) $\delta(D, 1) =$

(e) List the members of the set $\{q \in Q \mid D \in \delta(q, 2)\}$:

(f) Does the NFA accept the word 11120? (Yes / No)

Problem 5: NFA to DFA conversion (6 points)

Convert the following NFA to a DFA recognizing the same language, using the subset construction. Give a state diagram showing all states reachable from the start state, with an informative name on each state. Assume the alphabet is $\{0, 1\}$.



Problem 6: Short Construction (8 points)

(a) Give a regular expression for the language L containing all strings in a^*b^* whose length is a multiple of three. E.g. L contains aaaabb but does not contain abababb or aaabb.

(b) Let $\Sigma = \{a, b, c\}$. Give an NFA for the language L containing all strings in Σ^* which have an a or a c in the last four positions. E.g. bbabbb and abbbcb are both in L, but acabbbb is not. Notice that strings of length four or less are in L exactly when they contain an a or a c.

You will receive **zero** *credit if your NFA contains more than 8 states.*

Problem 7: NFA modification and tuple notation (8 points)

For this problem, the alphabet is always $\Sigma = \{a, b\}$.

Given an NFA M that accepts the language L, design a new NFA M' that accepts the language $L' = \{twt \mid w \in L, t \in \Sigma\}$. For example, if aab is in L, then $aaaba$ and $baabb$ are in L'.

(a) Briefly explain the idea behind your construction, using English and/or pictures.

(b) Suppose that $M = (Q, \Sigma, \delta, q_0, F)$. Give the details of your construction of M', using tuple notation.