Midterm 1
CS 373: Theory of Computation

Date: Thursday, October 4, 2012.

Instructions:

- This is a closed book exam. No notes, cheat sheets, textbook, or printed material allowed.
- You have 90 minutes to solve this exam.
- This exam has 5 problems each worth 10 points. However, not all problems are of equal difficulty.
- Please write your name on the top of *every* page in the space provided.
- If your solution does not fit in the space provided, and continues onto one of the back sheets, please indicate clearly where we should look for the solution.
- Unless otherwise stated, "prove that", "show that" for a problem means you need to formally prove what you are claiming.
- You may use, without proof, any result that you were asked to prove in the homework or was proved in the lecture. If you use such a result, please explicitly state the result (like " 'Perfect shuffle of regular languages is regular' was proved in a homework", instead of "this was shown in a homework").

Name	SOLUTIONS
Netid	solutions

Discussion: T 2:00–2:50 T 3:00–3:50 W 1:00–1:50 W 4:00–4:50 W 5:00–5:50

Problem	Maximum Points	Points Earned	Grader
1	10		
2	10		
3	10		
4	10		
5	10		
Total	50		

Problem 1. [Category: Comprehension] **True/False.** Decide for each statement whether it is true or false. Circle **T** if the statement is *necessarily true*; circle **F** if it it is not necessarily true. Each correct answer is worth **1 point**.

(a) Let Σ and Δ be two alphabets. For a set A, let $|A|$ denote the number of elements in A. Then for all n, $|\Sigma^n| = |\Delta^n|$.
False. Just take $\Sigma = \{0\}$ and $\Delta = \{0,1\}$. $|\Sigma^1| = 1 \neq 2 = |\Delta^1|$.

(b) Suppose M is a DFA such that $\epsilon \in \mathbf{L}(M)$. Then the initial state of M must be a final state.
True. The DFA cannot take any steps without reading a symbol from the input. Thus, on ϵ as input, the DFA stays in the initial state, and if ϵ is accepted, then the initial state must be an accepting state.

(c) For language L_1 and L_2 over the alphabet Σ, $L_1 \setminus L_2$ denotes the difference between the two sets, i.e., it is the set of all strings that belong to L_1 but not L_2. If L_1 and L_2 are regular then $L_1 \setminus L_2$ is regular.
True. This was done in the lectures. The reason is $L_1 \setminus L_2 = L_1 \cap (\Sigma^* \setminus L_2)$, and since regular languages are closed under intersection and complementation, the result follows.

(d) There is an NFA N with n states, such that any DFA recognizing $\mathbf{L}(N)$ has at least 2^n states.
True. We have seen in the lecture notes that the language of binary strings that have a 1 n positions from the end can be accepted by an NFA with n states, but any DFA accepting this language must have at least 2^n states.

(e) If $L \subseteq \{0\}^*$ then L is regular.
False. The languages $\{0^p \mid p \text{ is prime}\}$ and $\{0^{n^2} \mid n \geq 0\}$ are two languages that we have shown to be non-regular in the lectures and discussions.

(f) Let $L_1 \subseteq \Sigma^*$ and $L_2 \subseteq \Sigma^*$ be languages. Then $L_1^* \cap \Sigma^0 = L_2^* \cap \Sigma^0$.
True. $\Sigma^0 = \{\epsilon\} \subseteq L^*$ for any language L.

(g) Suppose R_1, R_2 are regular expressions such that $\mathbf{L}(R_1) = \mathbf{L}(R_2)$. Then R_1 and R_2 have the same number of operators.
False. R_1 and R_2 could be very different regular expressions that describe the same language. For example 0^* and $0^* \cup \emptyset \cup \emptyset \cup \emptyset$ describe the same language.

(h) Since regular languages are closed under homomorphism, *non-regular* languages are also closed under homomorphisms. That is, if L is *not* regular and h is a homomorphism then $h(L)$ is not regular.
False. Take h such that $h(0) = \epsilon = h(1)$. Then $h(L_{0n1n}) = \{\epsilon\}$ is regular.

(i) The following is correct proof showing that the language $A = \{a^n b^n \mid n \geq 0\}$ is not regular: Let $h : \{a,b\}^* \to \{0,1\}^*$ be a homomprphism given by $h(a) = 0$ and $h(b) = 1$. Then since $A = h^{-1}(L_{0n1n})$, A is not regular. (Recall that $L_{0n1n} = \{0^n 1^n \mid n \geq 0\}$.)
False. This is an incorrect proof. The inverse homomorphic image of a non-regular language is not necessarily non-regular (see Quiz 8). Hence the fact that A is the inverse homomorphic image of a non-regular language does not mean that A is not regular.

(j) There is a non-regular language L that satisfies the pumping lemma.
True. See quiz 9 for an example.

Problem 2. [Category: Comprehension+Proof] For a binary string $w \in \{0,1\}^*$, let $[\![w]\!]$ denote the number

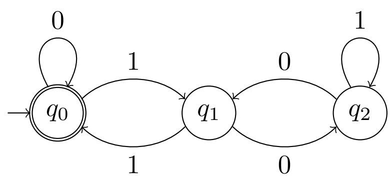

Figure 1: DFA A recognizing L_3

whose binary representation is given by w; here we will assume that the rightmost symbol is the least significant bit. We could define this inductively as

$$[\![\epsilon]\!] = 0 \quad [\![w0]\!] = 2 \times [\![w]\!] \quad [\![w1]\!] = 2 \times [\![w]\!] + 1$$

Thus, for example, $[\![10]\!] = (2^1 \times 1) + (2^0 \times 0) = 2$ and $[\![101]\!] = (2^2 \times 1) + (2^1 \times 0) + (2^0 \times 1) = 5$. Let $L_3 = \{w \in \{0,1\}^* \mid [\![w]\!] \bmod 3 = 0\}$ is the collection of all binary strings w that are multiples of 3. (Recall $a \bmod b = c$ means that c is the remainder when a is divided by b.)

The DFA A (shown in Figure 1) recognizes the language L_3. The states of A keep track of the remainder when the input string is divided by 3; thus, reaching state q_i means that the remainder is i. The transitions of A are defined based on the observation that

$$[\![wa]\!] \bmod 3 = (2([\![w]\!] \bmod 3) + a) \bmod 3$$

(a) Answer the following:

$\hat{\delta}_A(q_0, 111) = \underline{\{q_1\}}$ **[1 point]**

$\hat{\delta}_A(q_2, 101) = \underline{\{q_0\}}$ **[1 point]**

(b) Let us define

$$L_A(q_0, q_1) = \{w \in \{0,1\}^* \mid \hat{\delta}_A(q_0, w) = \{q_1\}\}$$
$$L_A(q_1, q_0) = \{w \in \{0,1\}^* \mid \hat{\delta}_A(q_1, w) = \{q_0\}\}$$

Answer the following questions:

(i) Is $100 \in L_A(q_0, q_1)$? <u>Yes</u> **[1 point]**

(ii) Is $100 \in L_A(q_1, q_0)$? <u>Yes</u> **[1 point]**

(iii) Is $1000 \in L_A(q_0, q_1)$? <u>No</u> **[1 point]**

(iv) Is $1000 \in L_A(q_1, q_0)$? <u>Yes</u> **[1 point]**

(c) Describe formally the strings that belong to $L_A(q_0, q_1)$ and $L_A(q_1, q_0)$. (Don't repeat the definitions in part(b) but rather come up with a description based on how the automaton A works.) **[2 points]**

The language $L_A(q_0, q_1)$ is essentially given by the intuition behind the construction that A remembers the remainder when divided by 3. Thus,

$$L_A(q_0, q_1) = \{w \mid [\![w]\!] \bmod 3 = 1\}$$

The language $L_A(q_1, q_0)$ can be understood as follows. Observe that 1 takes A from q_0 to q_1. So a string w takes A from q_1 to q_0 if $1w$ takes A from q_0 to q_0, i.e.,

$$L_A(q_1, q_0) = \{w \mid [\![1w]\!] \bmod 3 = 0\}$$

There are other ways to describe this language. I list two others without explaining the intuition behind them.

$L_A(q_1, q_0) = \{w \mid (2^{|w|} + [\![w]\!]) \bmod 3 = 0\}$
$L_A(q_1, q_0) = \{w \mid (1 + [\![w]\!]) \bmod 3 = 0 \text{ if } |w| \text{ is even and } (2 + [\![w]\!]) \bmod 3 = 0 \text{ if } |w| \text{ is odd}\}$

(d) Let M with initial state q_0 be *any* DFA that recognizes L_3. Prove that $\hat{\delta}_M(q_0, \epsilon) \neq \hat{\delta}_M(q_0, 1)$. **[2 points]**

If $\hat{\delta}_M(q_0, \epsilon) = \hat{\delta}_M(q_0, 1)$ then $\hat{\delta}_M(q_0, \epsilon 1) = \hat{\delta}_M(q_0, 11)$. That means M either accepts both 1 and 11 or neither. But $\epsilon 1 = 1 \notin L_3$ and $11 \in L_3$.

Problem 3. [Category: Comprehension+Design] For a string $w = a_1 a_2 \cdots a_n \in \Sigma^*$ where each $a_i \in \Sigma$, $w^R = a_n a_{n-1} \cdots a_1$ is the "reverse" of w. For a language $A \subseteq \Sigma^*$, $A^R = \{w^R \mid w \in A\}$.

(a) For $L_1 = \{\epsilon, 01, 11, 100\}$ what is L_1^R? **[1 point]**

$L_1^R = \{\epsilon, 10, 11, 001\}$.

(b) For $L_2 = \mathbf{L}(0^*(10)^*(0 \cup 1)^*)$, give a regular expression describing L_2^R. **[1 point]**

L_2^R is described by the regular expression $(0 \cup 1)^*(01)^*0^*$.

(c) Regular languages are closed under the "reversing" operation. That is, if A is regular then A^R is regular. This can be shown by constructing an NFA M^R recognizing A^R, given a DFA M recognizing A. Essentially, the NFA M^R "reverses" the direction of the transitions of M and has a new initial state that has ϵ-transitions to the final states of M. Complete the formal definition of M^R based on this intuition.

Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA recognizing A. The NFA $M^R = (Q^R, \Sigma, \delta^R, q_0^R, F^R)$ where

(i) $Q^R = \underline{Q \cup \{s\} \text{ where } s \notin Q}$ **[2 points]**

(ii) $q_0^R = \underline{s}$ **[1 point]**

(iii) $F^R = \underline{\{q_0\}}$ **[1 point]**

(iv) Describe the transition function δ^R. **[3 points]**

$$\delta^R(q, a) = \begin{cases} F & \text{if } q = s \text{ and } a = \epsilon \\ \{q' \mid \delta(q', a) = q\} & \text{if } q \in Q \text{ and } a \neq \epsilon \\ \emptyset & \text{in all other cases} \end{cases}$$

Problem 4. [Category: Proof] Complete the following proof by induction that $\mathbf{L}(M^R) = (\mathbf{L}(M))^R$, where the DFA M and NFA M^R are as defined in Problem 3.

(a) The correctness can be established by capturing the relationship between computations of M and computation of M^R. The statement to be proved by induction is **[2 points]**

$$\forall w \in \Sigma^*.\ \forall q \in Q.\ q_0 \xrightarrow{w}_M q \text{ iff } \underline{q \xrightarrow{w^R}_{M^R} q_0}$$

The proof of this statement by induction on $|w|$ is as follows.

(b) Prove the base case. **[2 points]**

Let w be such that $|w| = 0$. Then $w = \epsilon$. Now since M is deterministic, $q_0 \xrightarrow{\epsilon}_M q$ iff $q = q_0$ because a DFA does not take any steps without reading a symbol. Now, M^R has no ϵ-transitions except from the new initial state s, and q is a state of M (i.e., $q \in Q$). Thus, we have $q \xrightarrow{\epsilon}_{M^R} q_0$ (for $q \in Q$) iff $q = q_0$. Putting it all together we have the base case.

(c) State the induction hypothesis. **[1 point]**

For all w such that $|w| < n$, for all $q \in Q$, $q_0 \xrightarrow{w}_M q$ iff $q \xrightarrow{w^R}_{M^R} q_0$.

(d) Prove the induction step. **[3 points]**

Let $w = ua$, where $u \in \Sigma^{n-1}$ and $a \in \Sigma$. The induction step can be established by the following reasoning.

$$\begin{aligned} q_0 \xrightarrow{ua}_M q \quad & \text{iff } \exists q' \in Q.\ q_0 \xrightarrow{u}_M q' \xrightarrow{a}_M q \text{ where } \delta(q', a) = q \\ & \text{iff } \exists q' \in Q.\ q \xrightarrow{a}_{M^R} q' \xrightarrow{u^R}_{M^R} q_0 \text{ because of ind. hyp. and defn. of } \delta^R \\ & \text{iff } q \xrightarrow{w^R}_{M^R} q_0 \end{aligned}$$

(e) Using the statement in part (a), prove that $\mathbf{L}(M^R) = (\mathbf{L}(M))^R$. **[2 points]**

Consider the following sequence of reasoning steps: $w \in \mathbf{L}(M)$ iff there is $q \in F$, s.t. $q_0 \xrightarrow{w}_M q$ (definition of M accepting) iff $q \xrightarrow{w^R}_{M^R} q_0$ (by statement just proved) iff $s \xrightarrow{\epsilon}_{M^R} q \xrightarrow{w^R}_{M^R} q_0$ iff $w^R \in \mathbf{L}(M^R)$. This completes the proof.

Problem 5. [Category: Proof] As in Problem 2, for a binary string $w \in \{0,1\}^*$, let $[\![w]\!]$ denote the number whose binary representation is given by w where the rightmost symbol is the least significant bit; the formal inductive definition of $[\![w]\!]$ is given in Problem 2. Let $L_{m3} \subseteq \{0,1,\#\}^*$ be the language

$$L_{m3} = \{x\#y \mid x, y \in \{0,1\}^* \text{ and } [\![y]\!] = 3 \times [\![x]\!]\}$$

Prove that L_{m3} is not regular. You may use any of the proof techniques discussed in class. **[10 points]**

It is interesting to contrast this result with problem 1 in Homework 2. The non-regularity can be proved in many ways. All the proofs below rely on the following observation. For $w = 10^i$, we have $[\![w]\!] = 2^i$, and $3 \times [\![w]\!] = 2^{i+1} + 2^i$ is represented in binary as 110^i.

Lower bound technique. Suppose for contradition L_{m3} is regular and is recognized by DFA M with initial state q_0. Consider the (infinite) set $W = \{10^i | i \geq 0\}$. Since M has only finitely many states and W is infinite, by pigeon hole principle, there must be two (distinct) strings $x = 10^i, y = 10^j \in W$, $\hat{\delta}_M(q_0, x) = \hat{\delta}_M(q_0, y)$. That means $\hat{\delta}_M(q_0, x\#110^i = 10^i\#110^i) = \hat{\delta}_M(q_0, y\#110^i = 10^j\#110^i)$ and so either both $10^i\#110^i$ and $10^j\#110^i$ are accepted or neither one is. But $10^i\#110^i \in L_{m3}$ and $10^j\#110^i \notin L_{m3}$, which contradicts the assumption that M recognizes L_{m3}.

Closure property proof. Consider the following sequence of languages.

- $L_1 = L_{m3} \cap \mathbf{L}(10^*\#110^*) = \{10^i\#110^i \mid i \geq\}$
- Consider homomorphism $h_1 : \{a, b, c, \#\}^* \to \{0, 1, \#\}^*$ such that $h_1(a) = h_1(b) = 0$, $h_1(c) = 1$, and $h_1(\#) = \#$. Now,
$$L_2 = h_1^{-1}(L_1) \cap \mathbf{L}(ca^*\#ccb^*) = \{ca^i\#ccb^i \mid i \geq 0\}$$
- Consider homomprhism $h_2 : \{a, b, c, \#\}^* \to \{0, 1\}^*$ where $h_2(a) = 0$, $h_2(b) = 1$, and $h_2(c) = h_2(\#) = \epsilon$. Then, $L_3 = h_2(L_2) = \{0^n1^n \mid n \geq 0\} = L_{0n1n}$

If L_{m3} is regular then so are $L_1, L_2, L_3 = L_{0n1n}$. But since L_{0n1n} is not regular, L_{m3} is not regular.

Pumping Lemma proof. Let p be the pumping length. Take $w = 10^p\#110^p \in L_{m3}$. Let x, y, z be such that $w = xyz$, $|xy| \leq p$ and $|y| > 0$. Now there are two possibilities. Either $x = \epsilon$, or $x \neq \epsilon$.

Case 1 If $x = \epsilon$ then $y = 10^r$ (for $r \geq 0$) and $z = 0^s\#110^p$ where $r+s = p$. Now $xy^0z = z = 0^s\#110^p \notin L_{m3}$ since $3 \times [\![0^s]\!] = 0 \neq 2^{p+1} + 2^p = [\![110^p]\!]$.

Case 2 If $x \neq \epsilon$ then $x = 10^r$, $y = 0^s$ and $z = 0^t\#110^p$, where $r \geq 0$, $s > 0$, $t \geq 0$ and $r + s + t = p$; we can assume this form for x, y, and z because $|xy| \leq p$ and so if $x \neq \epsilon$ then y must only contain 0s. Now $xy^0z = 10^{r+t}\#110^p \notin L_{m3}$ because $3 \times [\![10^{r+t}]\!] = 3 \times 2^{r+t} = 2^{r+t+1} + 2^{r+t} \neq 2^{p+1} + 2^p = [\![110^p]\!]$ since $r + t < p$.