# Midterm 2
## CS 373: Theory of Computation

Date: Thursday, November 4, 2010.

**Instructions:**

- This is a closed book exam. No notes, cheat sheets, textbook, or printed material allowed.
- You have 120 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, recall that "prove that", "show that" for a problem means you need to formally prove what you are claiming.
- Answering "I don't know" for a problem *does not receive any points.*
- 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 you are using (like " 'Reverse of a regular language is regular' was proved in a homework", rather than saying "this was shown in a homework").

| | |
|---|---|
| **Name** | SOLUTIONS |
| **Netid** | solutions |

| 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 $M = (Q, \Sigma, \delta, q_0, F)$ be the minimal DFA recognizing the language $L(M)$. Suppose $M'$ is same as $M$ except the initial state is changed to $q \neq q_0$, i.e., $M' = (Q, \Sigma, \delta, q, F)$. Assuming all states in $Q$ are reachable from $q$, $M'$ is the minimal DFA recognizing $L(M')$.
**T** **F**
**True.** Observe the if two states are distinguishable in $M$ they are also distinguishable in $M'$.

(b) Let $M = (Q, \Sigma, \delta, q_0, F)$ be the minimal DFA recognizing the language $L(M)$. Since every pair of states of $M$ is distinguishable, if $w$ is accepted from state $q$ then $w$ is not accepted from state $q'$ ($\neq q$).
**T** **F**
**False.** Distinguishability only requires the states to behave differently on *some* string, and not all.

(c) If $L$ is regular then suffix$(L, x)$ is always regular, no matter what $x$ is. (For a definition of suffix$(L, x)$ see problem 2.)
**T** **F**
**True.** Suppose DFA $M$ recognizes $L$, then suffix$(L, x)$ is the set of strings accepted by $M$ from the state $q$ reached on input $x$. Thus, suffix$(L, x)$ is recognized by the DFA $M'$ which is the same as $M$, except that it has initial state $q'$.

(d) For a decidable language $L$, $L^R$ may or may not be decidable. ($L^R$ denotes the reverse of language $L$.)
**T** **F**
**False.** On input $x$, the algorithm for $L^R$, will reverse $x$, and then run the algorithm for $L$.

(e) If $L \subseteq \{0\}^*$ then $L$ is decidable.
**T** **F**
**False.** TMs can be encoded in unary; just take the binary string, and translate that to a unary string. So all the languages that we know to be undecidable (like $L_d$) have a unary encoding that will also be undecidable.

(f) If $L \leq_m \{0^n 1^n \mid n \geq 0\}$ then $L$ is decidable.
**T** **F**
**True.** Since $\{0^n 1^n \mid n \geq 0\}$ is decidable, the observation follows from properties of reductions.

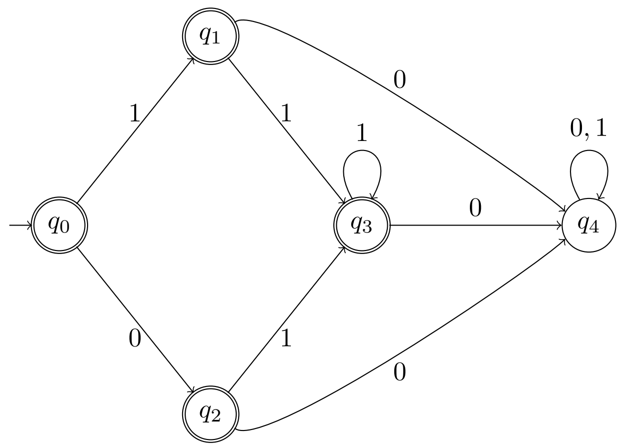
(g) If $L$ is not recursively enumerable then $\overline{L}$ must be recursively enumerable.
**T** **F**
**False.** There are languages like REGULAR which are not r.e., and their complement is also not r.e.

(h) $L_k = \{M \mid M \text{ halts after at most } k \text{ steps on } \epsilon\}$ is not decidable because of Rice's theorem.
**T** **F**
**False.** First $L_k$ is not a property of languages and so Rice's theorem does not apply to it. Second $L_k$ is decidable — simply run input $M$ on $\epsilon$ for $k$ steps, and accept or reject based on whether $M$ halts.

(i) If $L$ is recursively enumerable and $L' \subseteq L$ then $L'$ is recursively enumerable because the enumerator for $L$ also enumerates $L'$.
**T** **F**
**False.** Take $L = \{0, 1\}^*$ (which is decidable and r.e.) and $L' = L_d$ which is not r.e. The enumerator for $L$ outputs all the strings in $L'$ but it also outputs additional strings that may not be in $L'$, and so it is not an enumerator for $L'$.

(j) If $A \leq_m B$ then $\overline{A} \leq_m \overline{B}$.
**T** **F**
**True.** See solutions to homework 8.

**Problem 2**. [Category: Comprehension+Design] Consider the language $L = L(\epsilon \cup (1 \cup 0)1^*)$ and a DFA $M$ that accepts $L$:




(a) Recall that for a DFA $M = (Q, \Sigma, \delta, q_0, F)$, $\text{suffix}(M, q) = \{w \in \Sigma^* \mid q \xrightarrow{w}_M q' \text{ and } q' \in F\}$. In other words, it is the collection of all words accepted if $q$ were the initial state.

For each state $q$ of $M$ describe the language $\text{suffix}(M, q)$, using either regular expressions or formal set notation. **[2.5 Points]**

(b) Recall that for a language $L \subseteq \Sigma^*$, and a string $x \in \Sigma^*$, suffix language of $L$ with respect to $x$, is defined as

$$\text{suffix}(L, x) = \{y \in \Sigma^* \mid xy \in L\}$$

In other words, $\text{suffix}(L, x)$ is the collection of strings $y$ which when prefixed by $x$, result in a string in $L$.

For each of the following values of $x$, describe $\text{suffix}(L, x)$. *(Hint: You may use the DFA $M$ and the previous problem to simplify your calculations.)* **[3.5 Points]**

(a) $x = \epsilon$

(b) $x = 0$

(c) $x = 1$

(d) $x = 00$

(e) $x = 01$

(f) $x = 10$

(g) $x = 11$

(c) Give a minimal DFA for $L$. **[4 Points]**

**Solution:**

(a)

$$\begin{array}{ll} \text{suffix}(M, q_0) = L & \text{suffix}(M, q_1) = L(1^*) \\ \text{suffix}(M, q_2) = L(1^*) & \text{suffix}(M, q_3) = L(1^*) \\ \text{suffix}(M, q_4) = \emptyset & \end{array}$$

(b) (a) $x = \epsilon$: $\text{suffix}(L, x) = L$

(b) $x = 0$: $\text{suffix}(L, x) = 1^*$

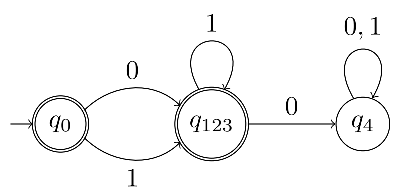
(c) $x = 1$: $\text{suffix}(L, x) = 1^*$

(d) $x = 00$: $\text{suffix}(L, x) = \emptyset$

(e) $x = 01$: $\text{suffix}(L, x) = 1^*$

(f) $x = 10$: $\text{suffix}(L, x) = \emptyset$

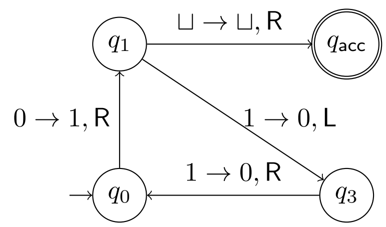
(g) $x = 11$: $\text{suffix}(L, x) = 1^*$




(c)

■

**Problem 3**. [Category: Comprehension] Consider the following Turing machine $M$ on input alphabet $\{0,1\}$. All transitions not shown in the diagram below are assumed to go to the reject state $q_{\mathsf{rej}}$.




(a) Give the formal definition of $M$ as a tuple. **[3 Points]**

(b) Describe the computation of $M$ on the input 0111 formally, as a sequence of instantaneous descriptions/configurations. **[5 Points]**

(c) Is there any input on which $M$ does not halt? If so, give an example string. **[1 Point]**

(d) What is the language recognized by $M$? **[1 Point]**

**Solution:**

(a) $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\mathsf{acc}}, q_{\mathsf{rej}})$ where $Q = \{q_0, q_1, q_3, q_{\mathsf{acc}}, q_{\mathsf{rej}}\}$, $\Sigma = \{0,1\}$, $\Gamma = \{0, 1, \sqcup\}$, and $\delta$ is given as follows.

$$\delta(q_0, 0) = (q_1, 1, \mathsf{R}) \quad \delta(q_1, 1) = (q_3, 0, \mathsf{L}) \quad \delta(q_1, \sqcup) = (q_{\mathsf{acc}}, \sqcup, \mathsf{R})$$
$$\delta(q_3, 1) = (q_0, 0, \mathsf{R}) \quad \delta(q, a) = (q_{\mathsf{rej}}, \sqcup, \mathsf{R}) \text{ in all other cases}$$

(b) The computation is as follows:

$$q_0 0111 \vdash 1 q_1 111 \vdash q_3 1011 \vdash 0 q_0 011 \vdash 01 q_1 11 \vdash 0 q_3 101 \vdash 00 q_0 01 \vdash 001 q_1 1 \vdash 00 q_3 10 \vdash 000 q_0 0$$
$$\vdash 0001 q_1 \sqcup \vdash 0001 \sqcup q_{\mathsf{acc}} \sqcup$$

(c) The machine halts on all inputs.

(d) The language recognized by this machine is given by the regular expression $01^*$.

■

**Problem 4**. [Category: Comprehension+Proof]

(a) Suppose $A$ and $B$ are recursively enumerable languages such that $A \cup B$ and $A \cap B$ are both decidable. Prove that $A$ is decidable. **[5 Points]**

(b) Suppose $A$ is recursively enumerable and $A \leq_m \overline{A}$. Prove that $A$ is decidable. **[5 Points]**

**Solution:**

(a) Let $M_A$ and $M_B$ be TMs recognizing $A$ and $B$, respectively. Let $M_{A\cup B}$ and $M_{A\cap B}$ be decision procedures for $A \cup B$ and $A \cap B$, respectively. An algorithm for $A$ is as follows.

On input $x$
Run $M_{A\cup B}$ on $x$
If $M_{A\cup B}$ rejects then reject (and halt)
else /*** $x$ is in $A \cup B$ ***/
    Run $M_{A\cap B}$ on $x$
    If $M_{A\cap B}$ accepts then accept (and halt)
    else /*** $x \in (A \cup B) \setminus (A \cap B)$ ***/
        Run $M_A$ and $M_B$ in parallel (using dovetailing) on $x$
        If $M_A$ accepts then accept (and halt)
        else if $M_B$ accepts then reject (and halt)

The main observation is that if on $x$, $M_{A\cup B}$ accepts and $M_{A\cap B}$ rejects, then $x$ belongs to exactly one out of $A$ and $B$. Thus, exactly one of the simulations of $M_A$ and $M_B$ will accept, and whichever one terminates first, we know whether $x$ belongs to $A$ or not.

(b) Observe that in homework 8, we showed that if $A \leq_m B$ then $\overline{A} \leq_m \overline{B}$. Thus, if $A \leq_m \overline{A}$ then $\overline{A} \leq_m \overline{\overline{A}} = A$. Therefore, since $A$ is r.e., from properties of reductions, it follows that $\overline{A}$ is r.e. Finally, since $A$ and $\overline{A}$ are r.e., from a theorem proved in class, we can conclude that $A$ is decidable.

■

**Problem 5**. [Category: Proof] Let $L = \{M \mid M$ is a TM and $L(M)$ has at least 11253 strings$\}$. Prove the following facts.

(a) $L$ is undecidable. **[2 Points]**

(b) $L$ is recursively enumerable. **[4 Points]**

(c) $\overline{L}$ is not recursively enumerable. **[4 Points]**

**Solution:**

(a) $L$ is a non-trivial property of languages, and so by Rice's theorem $L$ is undecidable.

(b) $L$ is recursively enumerable because the following nondeterminstic TM recognizes $L$

On input $M$
  Guess 11253 different strings
  Run $M$ on all the strings guessed
  If $M$ (halts and) accepts n each of the guesed strings then accept else reject

Since nondeterminstic TMs are equivalent to deterministic TMs, we know that $L$ is recursively enumerable.

(c) Since $L$ is undecidable (part 1), and recursively enumerable (part 2), $\overline{L}$ must be not recursively enumerable.

■