Problem Set 7

Spring 10

Due: Friday April 2nd, by 4pm in 3229 SC.

Extra credit problem (Problem 5): Due on Thursday, April 8th, at 2pm in class.
Please follow the homework format guidelines posted on the class web page:

`http://www.cs.uiuc.edu/class/sp10/cs373/`

1. The way to Tbilisi [**Category**: Puzzle, **Points**: 10]

   On your way to Tbilisi, you come across a fork in the road, one going left and one going right, and you wonder which one goes to Tbilisi. Fortunately, a woman from the neighboring village of Truthli sits on a boulder at the fork, eating peanuts, who surely knows the way. The people of Truthli are peculiar— a Truthli citizen either speaks always the truth or always lies (how this choice is made at birth is still mysterious). Of course, you do not know this woman's orientation to truth.

   Come up with one question you can ask the woman from Truthli such that from her answer you can infer which road leads to Tbilisi.

2. Reductions [**Category**: Proof, **Points**: 30]

   (a) **Reductions for decidability:**
   C is a restricted class of Turing machines for which `cs373.com` provides a very useful Turing machine M_C that takes in a pair $\langle M, D\rangle$, where M is a Turing machine in class C, and D is a DFA, and *accepts* iff $L(M) \cap L(D) \neq \emptyset$. In other words, M_C can decide whether the language of a Turing machine in class C has a common word with the language of a DFA D.

   More precisely,
   $L(M_C) = \{\langle M, D\rangle \mid M \textit{ is a TM in } C \textit{ and } D \textit{ is a DFA and } L(M) \cap L(D) \neq \emptyset\}$.

   We would like to now solve the membership problem for Turing machines in class C: given a TM M in C and a word w, we want to decide whether $w \in L(M)$. More precisely, we want to show that
   $L_{memb} = \{\langle M, w\rangle \mid M \textit{ is a TM in } C \textit{ and } w \in L(M)\}$ is decidable.
   Prove that L_{memb} is decidable by using a reduction.

   (b) **Reductions for undecidability:**
   A Turing machine *rejects* a word w if, when started with input w, it halts and reaches the reject state. (Note: If a TM does not accept w, it does not mean it rejects w, as it may not halt on w).
   Let $L_{rej} = \{\langle M, w\rangle \mid M \textit{ is a TM and } M \textit{ rejects } w\}$.
   Show that L_{rej} is undecidable, using a reduction, and by using the fact that the universal language L_u (or A_{TM} in Sipser) is undecidable.
   Recall that $L_u = \{\langle M, w\rangle \mid M \textit{ is a TM and } M \textit{ accepts } w\}$.

3. TM to NFA [**Category**: Proof., **Points**: 20]

   A Right-move Turing Machine (RTM) is a Turing machine that can only move its head only to the right. It can not move it to the left nor leave it in the same position. More formally, an RTM is a tuple $M = (Q, \Sigma, T, \delta, q_0, q_a, q_r)$ where Q is a set of states, Σ is the input alphabet, T is the tape alphabet, q_0, q_a and q_r are the initial, accept and reject states respectively and $\delta : Q \times T \to Q \times T \times \{R\}$ is a transition function. Prove that the languages accepted by RTMs are *regular* by showing, given any RTM, that there exists a DFA/NFA accepting the same language as the RTM. Your construction has to be formal. Give an informal argument on how your simulation works.

4. Enumerate lexicographically [**Category**: Proof., **Points**: 20]

   Read the definition of an enumerator (Sipser p. 152). Prove the following theorem (this actually is a variant of Theorem 3.21 in Sipser).

   A language is *Turing-decidable* if and only if some enumerator enumerates it in *lexicographic* order.

5. (ExtraCredit)Two-Phase TM [**Category**: Comprehension, **Points**: 20]

   We define a *Two-Phase TM*, $M = (Q, \Sigma, \Gamma, \delta_0, \delta_1, q_0, q_1, F_0, F_1)$ as follows:
   Q is the set of states of this machine (a finite set).
   Σ and Γ are input and tape alphabet respectively ($\sqcup \in \Gamma$ and $\Sigma \subseteq \Gamma - \{\sqcup\}$).
   $q_0, q_1 \in Q$ and $F_0, F_1 \subseteq Q$.
   $\delta_0 : Q \times \Gamma \to Q \times \Gamma - \{\sqcup\}$ and $\delta_1 : Q \times \Gamma \to Q$.
   M has two tapes: $\circledast_1$ and $\circledast_2$. At the beginning of the computation, the input string is written at the start of $\circledast_1$ and the two other tapes are blank. The head of each tape is initially located at the beginning of that tape. The computation has two phases (starting with phase 1 and in state q_0). The following recipe tells you how M operates in each phase:
   Phase 1: Assume that M is in state q and the head of $\circledast_1$ is on top of symbol a. Assume $\delta_0(q, a) = (p, b)$. If $p \notin F_0$, then the head of $\circledast_1$ just goes one step to the right, the head of $\circledast_2$ first writes b on $\circledast_2$ and then goes one step to the right, and finally state of M changes to p and M continues to operate in Phase 1. If $p \in F_0$, then the head of $\circledast_2$ writes b on the tape and immediately jumps to the beginning of the tape, state of M changes to q_1 and Phase 1 ends (which means that Phase 2 starts).
   Phase 2: Assume that M is in state q and the head of $\circledast_2$ is on top of symbol a. If $a = \sqcup$, then the computation stops, M accepts iff $q \in F_1$. Otherwise assume $\delta_1(q, a) = p$. Then the head of $\circledast_2$ progresses one step to the right and M switches to state p. M will continue to operate in Phase 2.

   As usual, we define $L(M)$ to be the set of all those strings in Σ^* that M accepts. Prove that $L(M)$ is regular.