Solutions for Problem Set 7
CS 373: Theory of Computation

Assigned: October 19, 2010   Due on: October 26, 2010 at 10am

---

Homework Problems

Problem 1. [Category: Proof] Solve problem 3.19. *Hint:* Use the result of problem 3.18, which was solved in discussion 9.

Solution: Let L be an infinite recursively enumerable language. Since L is recursively enumerable, L has an enumerator M such that $E(M) = L$. Problem 3.18 (and the discussion section problem) provide a characterization of decidable languages in terms of enumeration: L_1 is decidable iff there is an enumerator M_1 that enumerates the strings of L_1 in lexicographic order. We will use this result to identify an infinite, decidable subset of L.

Consider the following enumerator M_1

```
last-string = ⊥
Run M
Whenever M outputs a string (say) w
    if ((last-string = ⊥) or (w > last-string) then
        output w
        last-string = w
```

In the above algorithm, the check "$w >$ last-string" means that w is after last-string in the lexicographic ordering. Observe that, by construction, any string output by M_1 is in L (as it must have been first output by M which enumerates L), and M_1 outputs strings in lexicographic ordering (because we check that the new string to be output is after the last string that M_1 output). Thus, $E(M_1) \subseteq L$ and $E(M_1)$ is decidable (by problem 3.18).

All that is left to show is that $E(M_1)$ is infinite. Suppose (for contradiction) $E(M_1)$ is finite. Let last-string be the last string output by M_1. This means that last-string is the largest (according to the lexicographic ordering) of strings output by M_1, and is also the largest string output by M. That means L only consists of strings that are lexicographically smaller than last-string. But then L is finite, which gives contradicts the fact that L is infinite. ■

Problem 2. [Category: Proof] Solve problem 4.17.

Solution: We need to prove the following two statements.

1. If D is decidable then $C = \{x \mid \exists y.\ \langle x, y\rangle \in D\}$ is recursively enumerable.
2. If C is recursively enumerable then there is a decidable language D such that $C = \{x \mid \exists y.\ \langle x, y\rangle \in D\}$.

We will prove these in order.

Suppose D is a decidable language, with M_D a TM that always halts and recognizes D. Let $C = \{x \mid \exists y.\ \langle x, y\rangle \in D\}$. Consider the following TM M_C.

On input x
  for any string y do
    Run M_D on $\langle x, y\rangle$
    If M_D accepts then accept and halt

Observe that if M_C accepts x then there is some y such that M_D accepts $\langle x, y\rangle$, and so $L(M_C) \subseteq C$. On the other hand, suppose x is such that for some y, $\langle x, y\rangle \in D$, then M_C will accept x. Thus, $L(M_C) = C$. Hence, C is recursively enumerable.

Conversely, suppose C is recursively enumerable and M_C is a TM that recognizes C. Consider D defined as follows.

$$D = \{\langle x, y\rangle \mid M_C \text{ accepts } x \text{ within } |y| \text{ steps}\}$$

Observe that $C = \{x \mid \exists y.\ \langle x, y\rangle \in D\}$ because $x \in C$ if and only if there some k such that M_C accepts x within k steps. Moreover, D is decided by the following Turing machine M_D

On input $\langle x, y\rangle$
  Run M_C on x for $|y|$ steps
  If M_C accepts x within $|y|$ steps then
    accept $\langle x, y\rangle$
  else reject $\langle x, y\rangle$

Observe that M_D halts on all inputs because all of its statements halt. ■

Problem 3. [Category: Proof] Consider Inf $= \{M \mid M \text{ is a TM and } L(M) \text{ is infinite}\}$. Using reductions, prove that Inf is not recursively enumerable (i.e., Turing recognizable). *Hint:* Reduce a known non-recognizable language like $\overline{A_{\text{TM}}}$ or L_d to Inf.

Solution: Observe that Inf is undecidable by Rice's theorem. However, that does not establish anything about whether Inf is recursively enumerable. We will solve the problem by reducing $\overline{A_{\text{TM}}}$ to Inf as follows.

Given input $\langle M, w\rangle$ let $f(\langle M, w\rangle)$ be the following program:

On input x
  Run M on w for $|x|$ steps
  If M accepts w within $|x|$ steps then
    reject x
  else accept x

Observe that if M does not accept w (i.e., $\langle M, w\rangle \in \overline{A_{\text{TM}}}$) then since M will not accept w no matter how long we run M, all inputs will be accepted. Hence $L(f(\langle M, w\rangle)) = \Sigma^*$ and so $f(\langle M, w\rangle) \in \text{Inf}$. On the other hand, if M accepts w (or $\langle M, w\rangle \notin \overline{A_{\text{TM}}}$) then M accepts w in k steps, for some k. Now, on any string x of length $\geq k$, $f(\langle M, w\rangle)$ will reject, and on any input x of length $< k$, $f(\langle M, w\rangle)$ will accept. Thus,

$$L(f(\langle M, w\rangle)) = \bigcup_{i<k} \Sigma^i$$

which is a finite set. Thus, $f(\langle M, w\rangle) \notin \text{Inf}$. Hence, $\overline{A_{\text{TM}}} \leq_m \text{Inf}$ and since $\overline{A_{\text{TM}}}$ is not recursively enumerable, Inf is not recursively enumerable. ■