CSCE 551 Midterm II Answers, April 2, 2008

1. (15 points total) Omitted.

2. (25 points) Let A be the language of all strings $\langle M, t\rangle$, where

   - M is a TM,
   - t is a natural number, and
   - there is at least one input string w such that M accepts w after at most t steps.

   Give a decision procedure for A. (High-level algorithmic description of a decider for A.) [Hint: You don't need to search over *all* strings w.]

   Answer: The idea is that a machine can only look at the first t symbols of its input before t steps, so if it accepts any string at all, it will accept a string of length no more than t. Thus we limit our search space to the finitely many strings of length $\leq t$.

   "On input $\langle M, t\rangle$ where M is a TM and $t \in \mathbb{N}$:

   (a) For all strings w such that $|w| \leq t$, do
      i. Run M on input w for t steps.
      ii. If M accepts w within t steps, then accept.

   (b) Reject."

3. (25 points) For this problem, assume that all TMs have input alphabet $\{0, 1\}$. Let

   $$B = \{\langle M\rangle : L(M) = 0^*\}.$$

   In other words, B is the language of all strings encoding TMs M that accept a string just in case it does not contain a 1. Give a specific algorithm computing a function f that mapping reduces A_{TM} to B. (This shows that B is undecidable.)

   Answer: Let

   $f =$ "On input $\langle M, w\rangle$, where M is a TM and w is a string:

   (a) Let $R =$ 'On input x:
      i. If $x \notin 0^*$ (i.e., if x has a 1 somewhere) then reject.
      ii. Run M on input w (and do what M does).'

   (b) Output $\langle R\rangle$.

4. (20 points) Let $g : \Sigma^* \to \Sigma^*$ be a computable function. Suppose that $|w| < |x| \implies |g(w)| < |g(x)|$ for any strings $w, x \in \Sigma^*$ (we say that g is *strictly length-monotone*). Give a decision procedure for the range of g, i.e., give a high-level algorithm for a decider for

   $$\text{range}(g) = \{x \in \Sigma^* : (\exists w \in \Sigma^*)\ g(w) = x\}.$$

   If necessary, explain briefly why your algorithm works.

Answer: It is easy to see (and can be shown by induction) that $|w| \le |g(w)|$ for all w. Thus we can limit our search to strings with length less than or equal to the length of the input. That is, if there is any w such that $g(w) = x$, then we must have $|w| \le |x|$.

"On input x:

(a) For all strings w such that $|w| \le |x|$, do

  i. Compute $y = g(w)$.

  ii. If $y = x$ then accept.

(b) Reject.

Bonus Questions

These are for extra credit. Don't try these unless you have time after doing the rest of the test.

1. (15 points extra credit) Modify your m-reduction in Problem 3 above to get an m-reduction from $\overline{A_{\text{TM}}}$ to B. (Thus B is not Turing-recognizable, either.)

   Answer: Let

   f = "On input $\langle M, w\rangle$, where M is a TM and w is a string:

   (a) Let R = 'On input x:

     i. If $x \in 0^*$ (i.e., if x has no 1s anywhere) then accept.

     ii. Run M on input w (and do what M does).'

   (b) Output $\langle R\rangle$.

2. (15 points extra credit) Let g be as in Problem 4, except that you can only assume that $|w| < |x| \implies |g(w)| \le |g(x)|$. Show that range($g$) is decidable.

   Answer: There are two cases:

   range(g) is finite. Then clearly range(g) is decidable, because all finite languages are decidable (even regular).

   range(g) is infinite. Then the following algorithm decides range(g):

   "On input x:

   (a) Found := FALSE

   (b) For $n = 0, 1, 2, \ldots$, do

     i. For all strings w of length n, do

       A. Compute $y = g(w)$.

B. If $y = x$ then accept.

C. If $|y| > |x|$ then Found := TRUE

ii. If Found = TRUE then reject.

Since g has infinite range (and this is a crucial assumption), eventually Found will be set to TRUE and the algorithm will halt (if it hasn't already). When Found is TRUE, by monotonicity we know that no string longer than w will output x via g, so the search can stop.