Solution sketch 4 - Computational Models - Spring 2012

As always, these solutions might contain mistakes. Please let us know if you find any! Note: *Only questions 1,3,5,6 will be graded.*

1. (a) True. If $M = (Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$, decides L then $M = (Q, \Sigma, \Gamma, \delta, q_0, q_r, q_a)$ decides $\overline{L}$.

   (b) False. We know A_{TM} in $R.E.$, but $\overline{A_{TM}}$ isn't.

   (c) True. Given M_1 which decides L_1 and M_2 which decides L_2, we can simply run M_1 on our input, then run M_2 on our input, and accept iff both accepted.

   (d) True. Use definitions, De-Morgan rule and the fact that $R.E$ is closed under union.

   (e) True. Construct a non-deterministic TM that works as follows: it guesses how to split the input to an arbitrary number of parts, and then runs a TM that accepts M on all parts. It accepts iff M accepted all parts.

2. Let M_L be a TM that accepts $L_f \in RE$. A TM that computes f is the following: On input x, it simulates (simultaneously, using an enumerator for Γ^*) the computations of M_L on all inputs (x, y), and outputs y whenever M_L accepts (x, y). For the other direction, let M be a TM that computes f. A TM that accepts L_f is the following: on input (x, y) it simulates the computation of M on input x and accepts if the result of the computation equals y.

3. (a) Not decidable.

   (b) Not decidable. We show $Empty_{TM} \leq_m L$. Given $\langle M \rangle$, $f(\langle M \rangle) = \langle M, M' \rangle$ where M' is a TM that decides the empty language. (Prove correctness).

(c) Decidable. Notice that in less then 100 steps only the first 100 characters of the input can be read. Hence, to decide L: run M for 100 steps on every w such that $|w| \leq 100$, and accept iff one of them accepts.

4. first, let $L_{abc} = \{a^n b^n c^n | n \geq 0\}$ and let M_{abc} be a TM that decides L_{abc}.

   First we show that $\overline{\text{Halt}_{TM}} \leq_m L : f(\langle M, w\rangle) = \langle M'\rangle$. On input x, M' simulates $M(w)$ and if M halts then M' continues and simulates $M_{abc}(x)$. M' accepts iff M_{abc} accepts. Now, if M does not halt on input w then $L(M') = \phi$, and if M does halt on input w then $L(M') = L_{abc}$.

   Now we show that $\overline{\text{Halt}_{TM}} \leq_m \overline{L} : f(\langle M, w\rangle) = \langle M'\rangle$. On input x, M' simulates $M(w)$ for $|x|$ steps. If M halts, M' rejects. Otherwise, M' simulates $M_{abc}(x)$ and accepts iff M_{abc} accepts. Now, if M does not halt on input w then $L(M') = L_{abc}$, and if M does halt on input w then $L(M')$ is finite and therefore CFG.

5. (a) True. $f(\langle A\rangle)$ is a DFA for the complementary language.

   (b) True. Let M be a specific TM that accepts every input. Let M' be a specific TM that rejects every input. $f(\langle x\rangle)$ is $\langle M, x\rangle$ if $x \in L(0^*1^*)$ and $\langle M', x\rangle$ if $x \notin L(0^*1^*)$. (why is it computable?!)

   (c) True. Let M be a TM that semi-decides L. $f(\langle x\rangle) = \langle M, x\rangle$. (why is it computable? why is it correct?)

   (d) True. Idea: use the composition of the mapping reductions.

6. (a) Let E_L be an enumerator for L. A TM E that takes the output of E_L but only prints a word if it is bigger than any previous word output by E_L is a monotone enumerator. Since L is infinite, E outputs an infinite language which is a subset of L and is in R (since E is a monotone enumerator for it).

   (b) We inductively construct an infinite L over $\Sigma = \{0\}$ that does not have the property (of having a decidable infinite subset). First, we consider the enumeration $M_0, M_1, \ldots$ of the TMs that decide infinite languages over $\Sigma = \{0\}$.

   We start by seting w_0 to be the second shortest word accepted by M_0. Inductively, given $w_0, .., w_n$ we set w_{n+1} to be the second shortest word accepted by M_{n+1} of length more than the length of w_n. Finally, we set L to be $\{w_0, w_1, \ldots\}$

   Now, any infinite subset of L cant be decidable. Indeed, if such a subset was decided by a TM M_i then that M_i must also accept a word not in L (the shortest word ... in the construction above) and this is a contradiction.