Solution sketch 5 - Computational Models - Fall 2017

1. (a) Correct. Let f be the reduction from A to B and let g be the reduction from B to C. Define $h : \Sigma^* \to \Sigma^*$ such that $h(x) = g(f(x))$. As f and g are computable, h is computable as well (how?). Also, $x \in A$ if and only if $f(x) \in B$ if and only if $g(f(x)) \in C$ if and only if $h(x) \in C$, as desired.

   (b) Incorrect. Let $A = H_{TM}$ and $B = H_{TM,\varepsilon}$. Clearly, $A \neq B$. Prove yourself that indeed $A \leq_m B$ and $B \leq_m A$.

   (c) Incorrect. Let $A = H_{TM,\varepsilon}$ and $B = \Sigma^*$. Obviously, $A \subseteq B$, but it is not the case that $A \leq_m B$, as $B \in R$ and $A \notin \mathcal{R}$

   (d) Incorrect. Let $A = A_{TM}$ and $B = \overline{A_{TM}}$. It is not the case that $A \leq_m B$ since $B \in$co-$\mathcal{RE}$ but $A \notin$co-$\mathcal{RE}$. Similarly, it is not the case that $B \leq_m A$.

   (e) Correct. A is context-free, so $A \in \mathcal{R}$. Let M_A be the TM that decides A. Let $f(x) = \langle M, x\rangle$, be such that M on input x: Run M_A on x and answer the same. Clearly, f is computable. Now, obviously, $x \in A \iff M$ accepts x.

2. (a) $\mathcal{RE} \setminus \mathcal{R}$. $\mathcal{RE}$: Let $x_1, x_2, ...$ be a lexical ordering of all strings. Use a universal TM U. for $i = 1...\infty$ run M on the inputs $x_1, ..., x_i$ for i steps. accept if M halts on any input (prove correctness). $\notin \mathcal{R}$: $H_{TM} \leq_m L$. Given $\langle M, w\rangle$, we can compute a TM $\langle M'\rangle$ in the following manner. M' will ignore its input x and simply run M on w and return *true*. This reduction is computable. If M halts on w then M' halts on all inputs as required. If M does not halt on w then M' does not halt on any input, as required.

   (b) $co - \mathcal{RE} \setminus \mathcal{R}$. A non-deterministic TM for $\overline{L}$ would guess some word x not in $L(1(1 \cup 0)^*)$, run M on x, and accept iff M accepted x. $\notin \mathcal{R}$: by Rice's theorem.

(c) Not in $\mathcal{RE} \cup$ co-$\mathcal{RE}$. Not in $\mathcal{RE}$ by the following reduction from $\overline{A_{TM}}$: Given $\langle M, w\rangle$, return a description of a TM M' that on input x simulate the run of M on w for $|x|$ steps, if M accepts M' rejects. Otherwise, accept if x has the form $1x'$ and reject otherwise. If $\langle M, w\rangle \in \overline{A_{TM}}$, then M accepts w. Let k be the number of steps it takes to M to accept w, then $L(M') = \{x \in 1\{0,1\}^* \mid |x| < k\} \neq L(1(0 \cup 1)^*)$. If $\langle M, w\rangle \notin \overline{A_{TM}}$ then $L(M') = L(1(0 \cup 1)^*)$. Not in co-$\mathcal{RE}$ by the following reduction from A_{TM}: Given $\langle M, w\rangle$, return a description of a TM M' that on input x rejects if x is not of the form $1x'$, otherwise, simulate the run of M on w for and accept if M accepts. If $\langle M, w\rangle \in A_{TM}$, then $L(M') = L(1(0 \cup 1)^*)$. If $\langle M, w\rangle \notin A_{TM}$ then $L(M') = \emptyset \neq L(1(0 \cup 1)^*)$.

(d) $\mathcal{R}$. M on input $\langle M\rangle$:

- Run M on all inputs of length at most 50 for at most 50 steps.
- Accept if at least one string was accepted.

Obviously, M always halts. Since we bound the number of steps that M runs on an input, then there is no point on looking at any strings that are longer than that number, since if a TM is allowed to run for at most c steps, it is not possible for that TM to process any input symbol beyond the c-th symbol.

(e) $\mathcal{R}$. If $\langle M\rangle$ is indeed a TM then $L(M) \in \mathcal{RE}$ by definition, and RE is closed under union. Thus, a simple TM that decides L: Accept if and only if the input is a legal encoding.

3. (a) Incorrect. We saw in class that we cannot decide the problem for $A = \Sigma^*$.

(b) Incorrect. Let $\mathcal{C}$ be the class of all finite context-free languages. $\mathcal{C}$ is non-trivial and we saw an algorithm that decides whether a given CFG generates a finite language.

4.
- $L \notin \mathcal{RE}$

  We show that $\overline{H_{TM}} \leq_m L$: Let $L_{abc} = \{a^n b^n c^n \mid n \geq 0\}$ and let M_{abc} be a TM that decides L_{abc}. We define $f(\langle M, w\rangle) = \langle M^{'}\rangle$. 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 $L(M^{'}) = L_{abc}$.

- $L \notin$co-$\mathcal{RE}$

  We show that $\overline{H_{TM}} \leq_m \overline{L}$: Let M_{abc} be a TM that decides L_{abc} as before. $f(\langle M, w\rangle) = \langle M^{'}\rangle$. 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) Let $L \notin \mathcal{R}$ we can show that:

   - $L \leq_m A(L)$ by the mapping reduction $f_0(x) = 0x$
   - $\overline{L} \leq_m A(L)$ by the mapping reduction $f_1(x) = 1x$

   Assume by contradiction that $A(L) \in \mathcal{RE}$, thus by the above reductions $L \in \mathcal{RE}$ and $\overline{L} \in \mathcal{RE}$ and therefore $L \in \mathcal{R}$, a contradiction.

   (b) False. Let $L \notin \mathcal{R}$. We know that $A(L) \notin \mathcal{RE}$, and thus $A(L) \notin \mathcal{R}$. We can show that the following is a mapping reduction from $A(L)$ to $\overline{A(L)}$:

$$f(x) = \begin{cases} 0y & x = 1y \text{ for some } y \in \Sigma^* \\ 1y & x = 0y \text{ for some } y \in \Sigma^* \\ 0 & x = \varepsilon, \varepsilon \in L \\ 1 & x = \varepsilon, \varepsilon \notin L \end{cases}$$

   Note that given L the decision about the value of $f(\varepsilon)$ is pre-determined and not part the computation, thus f is computable.