# Solution sketch 5 - Computational Models - Spring 2013

1. Reduction Halt$^*$ $\leq_m L$:
   $f(< M >) =< M^{'} >$
   On imput $x$, $M^{'}$ emulates the execution of $M$ on $x$ and accepts (that is, moves to $q_a$) upon $M$ halting (either transition to $q_r$ or $q_a$).

2. A deciding TM will first check that the input is a legal description of a PDA (reject otherwise) and then convert it (as described in class) to a CFG that generates (only) the words accepted by the PDA. Now, the CFG is converted to Chomsky Normal Form - this ensures that for a word $w$ of length $n$, exactly $2n - 1$ steps are needed for its derivation (if possible). Finally, the TM will try all derivations of length $2|w| - 1$ and decide accordingly regarding the derivability of $w$. Merely simulating a PDA using an NTM is wrong since the PDA (and therefore the NTM) might have computation branches that don't halt.

3. (a) Given a TM $M$ and a word $w$, we decide whether $M$ does not accept $w$ by flipping the answer of a TM that decides $A_{TM}$ (forwarding the input $< M >, w$).
   (b) False. We just saw that $\overline{A_{TM}}$ reduces to $A_{TM}$. However, $\overline{A_{TM}} \notin RE$ (otherwise, since $A_{TM} \in RE$, we would get $A_{TM} \in R$) but $A_{TM} \in RE$.
   (c) False. Otherwise it would contradict $(b)$ above (to convince yourself, try constructing an appropriate mapping reduction...).

4. (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) $\mathcal{R}$. Either the TM that always answers $true$, or the TM that always answers $false$ decides this language.

(c) None of the above. $\notin \mathcal{RE}$: $\overline{H_\epsilon} \leq_m L$. Given the TM $\langle M \rangle$, we can compute $\langle M', 0, 1 \rangle$. $M'$ does the following for an input $x$. if $x = 0$ it runs $M$ on $\epsilon$. otherwise, it returns $false$. The reduction is computable. If $M$ halts on $\epsilon$ then $M'$ will halt on both 0 and 1, as required. If $M$ does not halt on $\epsilon$ then $M'$ will only halt on 1. $\notin co-\mathcal{RE}$: $H_\epsilon \leq_m L$. Given the TM $\langle M \rangle$, we can compute $\langle M', 0, 1 \rangle$. $M'$ does the following for an input $x$. if $x = 0$ it runs $M$ on $\epsilon$. otherwise, it starts an infinite loop. The reduction is computable. If $M$ halts on $\epsilon$ then $M'$ halts only on 0, as required. If $M$ does not halt on $\epsilon$ then $M'$ will not halt on 0 or 1.

(d) $Co-RE \setminus R$. To show that the complement lamguage is in $RE$ we will use a universal TM $U$ to run $M$ on all inputs as in section (a). If we encounter 3 different inputs that $M$ accepts, we will return $true$ (prove correctness).$\notin \mathcal{R}$: Rice's theroem.

(e) $\mathcal{R}$. This is trivial. It's the language of all TMs since for any $M$, $L(M) \in RE$.

(f) $\mathcal{R}$. This is a finite language. All finite languages are regular and of course decidable.

(g) $co-\mathcal{RE} \setminus \mathcal{R}$. $\in co-\mathcal{RE}$ : A TM that accepts the complement language runs in parallel on all inputs, and answers yes if it reaches the $|x| + 7$ position on the tape. $\notin \mathcal{R}$: Rice.

(h) $\mathcal{R}$. Given $M$, we run $M$ on $\epsilon$ for $|Q| + 1$ steps (where $Q$ is the set of states of $M$), and accept iff $M$ halts. To prove correctness, notice that the head will only see blanks (since always moves to the right). If the run does not stop in $|Q| + 1$ steps, it means that it entered one of its configurations at least twice and therfore the run will never stop and we can safely reject.

(i) $\mathcal{R}$. Run on all input of length at most $k^2$ for $k^3$ steps and decide. There is a finite number of such inputs, so it's obviously decidable.

(j) $co-\mathcal{RE} \setminus \mathcal{R}$. A TM that accepts the complementing language goes over all strings alphabetically, and for each string check if it's in $L(A_1)$ but not in $L(A_2)$ or vice versa (justify why this can be decided!). If such a string exists, answer yes. $\notin \mathcal{R}$ : We know that $E_{LBA}$ is undecidable. We give a mapping reduction from $E_{LBA}$: $f(A) = (A, loopy)$ and we have that $L(A) = \emptyset$ iff $L(A) = L(loopy)$.

(k) $co-\mathcal{RE} \setminus \mathcal{R}$. A TM that accepts the complementing language goes over all strings alphabetically, and for each string check if it's in $L(G)$ but not in $L(1(0U1)^*)$ or vice versa (justify why this can be decided!). If such a string exists, answer yes.

5. (a) It is easy to construct a series of TMs $\{M_n\}_{n \geq n_0}$ such that $M_n$ has $n$ states and $M_n() = n$. This implies that for every $m$ the set

$$G_m = \{n \mid M() \geq m \;\text{ and }\; M \in S_n\}$$

is not empty (it includes $m$) and has a minimum over the naturals. We conclude that $f$ is defined for every natural $m$ and is therefore total.

(b) If $m_1 \geq m_2$ then $G_{m_2} \subseteq G_{m_1}$ implying (taking a minimum on subset) $f(m_1) \geq f(m_2)$.

(c) If $f$ was computable we could use it to compute $BB(n)$ as follows: using the monotonicity of $f$ we compute in order $f(k)$ until for some $k$ we have $f(k) \leq n$ and $f(k+1) > n$. Such $k$ must satisfy $BB(n) = k$.