Computational Models - Exercise #6 solution sketch

1. (a) Correct. For every nontrivial L_1 and L_2 in P, $L_1 \leq_p L_2$ (prove it).

   (b) Equivalent to P vs. NP. Note that $\mathsf{NP} \subseteq \mathsf{R}$, so every two nontrivial languages are mapping-reducable to each-other. Thus, if $\mathsf{P} = \mathsf{NP}$ then the claim is true by the observation in (a), and if $\mathsf{P} \neq \mathsf{NP}$ then the claim is false, say by taking $L_1 = \text{SAT}$ and $L_2 = \{0\}$.

   (c) Correct. The reduction from any languagae in RE to A_{TM} is in fact linear in time (see slide 27 of Lecture 10).

   (d) Equivalent to P vs. NP. If $\mathsf{P} = \mathsf{NP}$ then every language in P is NP-complete (prove it). If $\mathsf{P} \neq \mathsf{NP}$ then, say, $\text{SAT} \notin \mathsf{P}$. Assume towards contradiction that $\text{SAT} \leq_p L$, so $L \notin \mathsf{P}$ – and this is obviously not the case.

   (e) Correct. Let $L \in \mathsf{NP}$. Then, there exists a polynomial-time reduction f such that $x \in L$ iff $f(x) \in \text{SAT}$. Conider the TM M_L, that on input x computes $f(x)$, simulates the TM for SAT and answers accordingly.

   The correcntess follows immediately from the correctness of the reduction. Let $n = |x|$. Assume that f runs in time n^c for some constant c. The length of $f(x)$ is then at most n^c, so the running time of the TM for SAT is at most $(n^c)^{O(\log n^c)} = n^{O(\log n)}$, so the overall running time of M_L is $n^{O(\log n)}$ as well.

2. (a) $L_1 \in \mathsf{P}$. A TM M_1 on input φ: Check that indeed φ is a valid CNF with at most 2016 clauses. If not, reject. Otherwise, φ has at most $3 \cdot 2016$ variables. Go over every possible assignment (there are at most $2^{3 \cdot 2016}$) and accept iff one of them is a satisfying one.

   The correctness is immediate. Also, note that we can do the above in polynomial (in fact, linear) time in $|x|$.

   (b) $L_2 \notin \mathsf{P}$. We will show that $\text{SAT} \leq_p L_2$. The reduction: Given φ, output φ' – the CNF

   $$\varphi \wedge x \wedge \ldots \wedge x$$

   where we added 2016 clauses (of size 1) and a varaible x *not appearing* appear in φ.

   Obviously, the reduction is polynomial and φ is satisfiable iff φ' is satisfiable and has at least 2016 clauses.

   (c) $L_3 \in \mathsf{P}$. A graph where every vertex has degree 2 is a disjoint union of cycles (prove it!). A cycle of length c has a minimal vertex cover of size $\lceil c/2 \rceil$. Thus, a TM that decides L_3: Check if G is indeed a union of disjoint cycles and if so check that k is large enough to cover all cycles. The polynomiality follows immediately.

(d) $L_4 \notin \mathsf{P}$. We will show that HamPath $\leq_p L_4$. Given $\langle G \rangle$ where $G = (V, E)$, output $\langle G, e \rangle$ where $G' = (V', E')$ is

$$\begin{aligned} V' &= V \cup \{v_a, v_b, v_c\}\,,\ v_a, v_b, v_c \notin V \\ E' &= E \cup \{\{v_a, v_b\}, \{v_b, v_c\}\} \cup \{\{v, v_a\} \mid v \in V\} \cup \{\{v, v_c\} \mid v \in V\}\,. \end{aligned}$$

and $e = \{v_a, v_b\}$.

The reduction is polynomial. Now, for the first direction, consider some Hamiltonial path $v_1, \ldots, v_n$ in G. Then, G' will also have a path $v_1, \ldots, v_n$ with each vertex only appearing once in the path. In order to turn this path into a Hamiltonian cycle, the three additional vertices will have to be included. In order to do so, the path has to be extended in either $(v_n, v_a, v_b, v_c, v_1)$ or $(v_n, v_c, v_b, v_a, v_1)$. G' thus have a Hamiltonian cycle that will always include the edge e.

For the second direction, consider some G' with a Hamiltonian cycle along some path $v_1, \ldots, v_n, v_a, v_b, v_c, v_1$. Since G has vertices $V = V' \setminus \{v_a, v_b, v_c\}$, G has a Hamiltonian path along $v_1, \ldots, v_n$.

3. (a) Let M be the TM that decides L in polynomial time. $M^\star$ on $x = x_1 \ldots x_n$:

- Construct a directed graph $G = (V, E)$ where $V = \{1, \ldots, n+1\}$ and for every $i < j$, $(i, j) \in E$ if and only if $M(x_{i,\ldots,j-1}) = 1$.
- Check if there exists a path in G from 1 to $n+1$.
- If such path exists, accept. Otherwise, reject.

There are $O(n^2)$ possible edges in G, and for every possible edge we simulate M. Then, we run a reachability algorithm (say, BFS). Thus, $M^\star$ runs in polynomial time. If a partition $x = w_1 \ldots w_k$ exists such that $w_i \in L$ for every i then there is a path from 1 to $n+1$ in G. Otherwise, there is no path. Overall, $M^\star$ decides $L^\star$ in polynomial time and hence $L^\star \in \mathsf{P}$.

(b) Let M be the nondeterministic TM that decides L in polynomial time. $M^\star$ on $x = x_1 \ldots x_n$:

- Guess a partition of x to $x = w_1 \ldots w_k$ (how?).
- For every $i \in \{1, \ldots, k\}$, simulate $M(w_i)$.
- If all simulations accepted, accept. Otherwise, reject.

The guessing can be done in linear time, and we simulate M at most n times, so $M^\star$ runs in polynomial time. Now, if $x \in L^\star$ there *exists* a partition $x = w_1 \ldots w_k$ exists such that $w_i \in L$ for every i, so there exist computation paths (for every such i) through which M accepts. Thus, there is an accepting computation path for $M^\star$. If $x \notin L^\star$ there exists no such partition, and every computation path of $M^\star$ rejects. Hence, $L^\star \in \mathsf{NP}$.

4. (a) The claim is true. As B is nontrivial there exist $y \in B$ and $z \notin B$. Set $n_0 = \max\{|y|, |z|\} + 1$ and $f(x)$ to be y if $x \in A$ and z otherwise. As $A \in \mathsf{P}$, f can be computed in polynomial time, and the correctness easily follows. Also, as $|f(x)| < n_0$ for every x satisfying $|x| \geq n_0$, the reduction is also shrinking.

(b) The claim is false. Assume to the contrary that there is a shrinking reduction f from SAT to SAT with a constant n_0. Denote $\mathrm{SAT}_{n_0} = \{\langle \varphi \rangle \in \mathsf{SAT} \mid |\langle \varphi \rangle| \leq n_0\}$. $\mathrm{SAT}_{n_0} \in \mathsf{P}$

as it is finite. Deciding SAT in P will be as follows: Given φ, compute the series φ_k such that $\varphi_0 = \varphi$ and $\varphi_k = f(\varphi_{k-1})$. The computation stops when we reach a k' such that $|\langle\varphi_{k'}\rangle| \leq n_0$ and we answer according to whether $\varphi_{k'} \in \mathrm{SAT}_{n_0}$ or not. As the reductions preserve correctness, it is easy to see that $\varphi_{k'} \in \mathrm{SAT}_{n_0}$ iff $\varphi \in \mathsf{SAT}$. Also, the above procedure can be done in polynomial time, as f is polynomial and we apply it linear number of times. Therefore, $\mathsf{SAT} \in \mathsf{P}$, in contradiction to $\mathsf{P} \neq \mathsf{NP}$.

5. (a) The claim is true, by the following chain of reductions:
   - $\mathsf{CLIQUE} \leq_p \mathsf{VC}$: Given $\langle G, k\rangle$ where $G = (V, E)$, output $\langle \bar{G}, |V| - k\rangle$ where $\bar{G} = (V, E')$ and $\{u, v\} \in E$ iff $\{u, v\} \in E'$. Prove that G has a clique of size k iff $\bar{G}$ has a vertex cover of size $|V| - k$.
   - $\mathsf{VC} \leq_p \mathsf{IS}$: Given $\langle G, k\rangle$ where $G = (V, E)$, output $\langle G, |V| - k\rangle$. It is easy to show that G has a vertex cover of size k iff it has an independent set of size $|V| - k$.
   - $\mathsf{IS} \leq_p \mathsf{CLIQUE}$: Given $\langle G, k\rangle$, output $\langle \bar{G}, k\rangle$. It is easy to show that G has an independent set of size k iff it has a clique of size k.

   Now, it is easy to prove that if $A \leq_p B$ and $B \leq_p C$ then $A \leq_p C$, so we are finished.

   (b) The claim is false. If $H_{TM} \leq_p \overline{H_{TM}}$ then $H_{TM} \leq_m \overline{H_{TM}}$ and $H_{TM} \in \mathsf{coRE}$. As $H_{TM} \in \mathsf{RE}$, $H_{TM} \in \mathsf{RE} \cap \mathsf{coRE} = \mathsf{R}$, in contradiction.

   (c) The claim is false. If $\mathrm{SAT} \leq_p \{1\}$ (both languages are in NP) then $\mathrm{SAT} \in \mathsf{P}$, in contradiction to $\mathsf{P} \neq \mathsf{NP}$.

6. (a) Correct. Let $L \in \oplus\mathsf{P}$, so there exists an appropriate non-deterministic TM M. M' on input x: Simulate every computation path of M on x. Keep a counter of the accepting and rejecting paths, and after each simulation erase the simulation tape. Answer according to the counters.

   Every simulation runs in polynomial time (although there are exponentially many) and thus in polynomial space. Other than keeping the counters (which we can do in polynomial space), we re-use the space after each simulation.

   (b) Correct. Let M_1 and M_2 be the TMs that accept L_1 and L_2. M on input x:
   - Simulate M_1 on x.
   - If M_1 accepted, simulate M_2 on x and answer accordingly.

   The polynomial running time is clear. Now, if $x \in L_1 \cap L_2$ then $x \in L_1$ and $x \in L_2$. Thus, the number of accepting paths of both M_1 and M_2 is odd. Thus, the number of accepting paths of M is their product – which is again odd. If $x \notin L$ then there exists $i \in \{1, 2\}$ for which the number of accepting paths of M_i is even, so the product is neccessarily even.

7. (a) $L_1 \notin \mathsf{P}$. Assume towards contradiction that it is, so it has a determinsitic TM M_{21} that decides it in polynomial time, and consider the TM M', that on input $(S_1, \ldots, S_n, T)$:
   - For every m, from 0 to n:
     - Run M_1 on $(S_1, \ldots, S_n, T, m)$.
     - If M_1 accepted, accept.
   - Reject.

Verify to yourself that M' indeed decides SubsetSum in polynomial time. As SubsetSum is NP-complete and we are assuming $\mathsf{P} \neq \mathsf{NP}$, this is a contradiction.

(b) $L_2 \notin \mathsf{P}$, by a very similar trick as (a).[1]

(c) $L_3 \in \mathsf{P}$. Cheking that indeed for every i, $S_i \geq \frac{T}{2016}$ can be done in polynomial time. Now, note that $(S_1, \ldots, S_n, T) \in L_3$ only if the desired set I is of size at most 2016. The total number of subsets of $\{1, \ldots, n\}$ of size at most 2016 is $\sum_{i=0}^{2016} \binom{n}{i} \leq 2016 \cdot n^{2016}$, which is polynomial in n. Thus, we can simply go over all possibilities and check their sum. Every such check is linear in time, so overall our TM that decides L_3 runs in polynomial time.

[1] Note that this solution does not show these problems are NP-complete, as we did not show a many-one reduction (but rather a Turing reduction).