Solution sketch 6 - Computational Models - Spring 2017

1. (a) A disjunction is the boolean OR of its clauses, and so it is satisfiable if any of its clauses can be satisfied. A clause is the boolean AND of some number of literal terms. Any clause is satisfiable unless it contains both some literal x and the negation of that literal $\bar{x}$. Thus an algorithm to determine DNF-Satisfiability would loop through the clauses and check if there is a clause which is satisfiable. This is clearly in P.

   (b) There are two methods of proving this:

   - A CNF formula is the boolean AND of its clauses. Thus it is a tautology if and only if every clause is a tautology. A clause is the boolean OR of some number of literals. Thus a clause is only a tautology if and only if it has some literal x and the negation of that literal $\bar{x}$. This can easily be checked by scanning through the clauses, and so $TCNF$ is in P.
   - A more clever method of proving this is to note that by DeMorgan's law the negation of a DNF formula is a CNF formula. Further, note that if X is a DNF tautology then its negation is a CNF contradiction, i.e. it is unsatisfiable. Thus we can solve this problem by negating the input and using our solution to the previous problem.

2. (a) Let M be the nondeterministic TM that decides L in polynomial time. M^* on x:

   - Guess a partition of x to $x = w_1 \cdots w_k$ (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^* runs in polynomial time. Now, if $x \in L^*$ there exists a partition $x = w_1 \ldots w_k$ 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^*. If $x \notin L^*$ there exists no such partition, and every computation path of M^* rejects. Hence, $L^* \in NP$.

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

   - 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 $M(x_i \cdots x_{j-1}) = 1$.
   - 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^* runs in polynomial time. If a partition $x = w_1 \cdots w_k$ 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^* decides L^* in polynomial time and hence $L^* \in P$.

3. (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 $A \in P$, f can be computed in polynomial time, and the correctness easily follows. Also, as $|f(x)| < n_0$ for every x satisfying $|x| = n_0$, 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 $SAT_{n_0} = \{\langle\varphi\rangle \in SAT \mid |\langle\varphi\rangle| \leq n_0\}$. $SAT_{n_0} \in P$ 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 $\varphi_{k'} \in SAT_{n_0}$ or not. As the reductions preserve correctness, it is easy to see that $\varphi_{k'} \in SAT_{n_0}$ iff $\varphi \in SAT$. Also, the above procedure can be done in polynomial time, as f is polynomial and we apply it linear number of times. Therefore, $SAT \in P$, in contradiction to $P \neq NP$.

4. We first prove that:

   - $UpToOneSAT \in coNP$. Consider the language $\overline{UpToOneSAT}$. It is easy to see that it is in NP, as the witness is simply two distinct satisfying assignments.
   - $\overline{UpToOneSAT} \in NPC$. We prove that $SAT \leq_p \overline{UpToOneSAT}$. Given a CNF φ, the reduction outputs $\varphi' = \varphi \wedge (v \vee \neg v)$ where v is a variable that is not in φ. The fact that the reduction is polynomial is trivial. Now, if φ has a satisfying assignment then φ' has at least two satisfying assignments. If φ is not satisfiable then surely neither is φ.

   Now, we prove $NP = coNP$ under the assumption that $UpToOneSAT \in NP$:

   - Let $L \in NP$. Then, $L \leq_p \overline{UpToOneSAT}$ and $\bar{L} \leq_p UpToOneSAT$. As $UpToOneSAT$ is in NP, $\bar{L} \in NP$ so $L \in coNP$
   - Let $L \in coNP$. Then, $\bar{L} \in NP$ and we proved that $\bar{L} \leq_p \overline{UpToOneSAT}$, so $L \leq_p UpToOneSAT$. As $UpToOneSAT$ is in NP, $L \in NP$.

5. We give a polynomial time mapping reduction from *CLIQUE* to *HALF-CLIQUE*. The input to the reduction is a pair $\langle G, k\rangle$ and the reduction produces the graph $\langle H\rangle$ as output where H is as follows. If G has m nodes and $k = \frac{m}{2}$ then $H = G$. If $k < \frac{m}{2}$, then H is the graph obtained from G by adding j nodes, each connected to every one of the original nodes and to each other, where $j = m - 2k$. Thus H has $m + j = 2m - 2k$ nodes. Observe that G has a k-clique iff H has a clique of size $k + j = m - k$ and so $\langle G, k\rangle \in CLIQUE$ iff $\langle H\rangle \in HALF\text{-}CLIQUE$. If $k > 2m$, then H is the graph obtained by adding j nodes to G without any additional edges, where $j = 2k - m$. Thus H has $m + j = 2k$ nodes, and so G has a k-clique iff H has a clique of size k. Therefore $\langle G, k\rangle \in CLIQUE$ iff $\langle H\rangle \in HALF\text{-}CLIQUE$. We also need to show $HALF\text{-}CLIQUE \in$ NP. The certificate is simply the clique.

6. Clearly *VC* is in NP. Given a a set $S \subseteq V$, one can verify in polynomial time if that is a vertex cover. This can be done by taking each edge and checking if it has at least one endpoint in S.

   To show that *VC* is NP-complete, we give a polynomial time mapping reduction f from *IS* to *VC*. Given an instance of *IS*, $\langle G = (V, E), k\rangle$, $f(G, k) = \langle G, |V| - k\rangle$.

   The reduction is polynomial. Let $G = (V, E)$ be an undirected graph, we show that $S \subseteq V$ is an independent set in G if and only if $V \setminus S$ is a vertex cover in G.

Suppose S is an independent set, and let $e = (u, v)$ be some edge. Only one of u, v can be in S. Hence, at least one of u, v is in $V \setminus S$. So $V \setminus S$ is a vertex cover.

Suppose $V \setminus S$ is a vertex cover, and let $u, v \in S$. There can't be an edge between u and v (otherwise, that edge wouldn't be covered in $V \setminus S$). So, S is an independent set.