CSCI 338: Assignment 6

Nathan Stouffer

Problem 1

A triangle in an undirected graph is a 3-clique. Show that $TRIANGLE \in P$, where $TRIANGLE = \{\langle G \rangle \mid G \text{ contains a triangle}\}$.

Proof: To show that $TRIANGLE \in P$, we must give a polynomial time Turing Machine to decide $TRIANGLE$. Consider the below TM:

$A =$ "On input $\langle G \rangle$ a graph:

1. For each $(u, v, w) \in V \times V \times V$
2. Test if $(u, v) \in E$ and $(v, w) \in E$ and $(u, w) \in E$
3. If test is passed, *accept*
4. No triple passed the test, *reject* "

Does A run in polynomial time? Lines 3 and 4 are constant. Line 2 runs in $O(|E|)$ and line 1 runs $|V|^3$ times. So A is a $|V|^3 * O(|E|) = O(|V|^3 * |E|)$ machine, which is polynomial (note $|E| \leq |V|^2$). Thus $TRIANGLE \in P$.

□

Problem 2

Let G represent an undirected graph.
Also let

$$SPATH = \{\langle G, a, b, k\rangle \mid G \text{ contains a simple path of length at most } k \text{ from } a \text{ to } b\}$$

and

$$LPATH = \{\langle G, a, b, k\rangle \mid G \text{ contains a simple path of length at least } k \text{ from } a \text{ to } b\}$$

Show that $SPATH \in P$ and that $LPATH$ is NP-complete.

Proof: We first show that $SPATH \in P$. To do so, we use a slight variation of breadth first search. Consider the following TM.

$S =$ "On input $\langle G, a, b, k\rangle$:
1. Mark a
2. Repeat k times
3. Mark any node with an edge from an already marked node
4. If b is marked, *accept*; otherwise *reject* "

Lines 1 and 4 run in constant time. Line 2 repeats k times and line 3 runs in $O(|E|)$ time. So S is a $k * O(|E|) = O(|E|)$ machine, and $SPATH$ must be in P.

We now show that $LPATH$ is NP-complete. We must first show that $LPATH \in NP$ by giving a polynomial verifier for $LPATH$:

$A =$ "On input $\langle G, a, b, k, p\rangle$ where p is a sequence of edges:
1. Test if all edges in p are in G
2. Test if p begins at a and terminates at b
3. Test if $|p| \geq k$
4. If all tests are passed, *accept*
5. Otherwise, *reject* "

Line 1 is $O(|E|)$ and the remaining lines are constant. So A is a polynomial time machine and $LPATH \in NP$, but is it NP-complete? In fact it is, consider the reduction $UHAMPATH \leq_p LPATH$

$R =$ " On input $\langle G, a, b\rangle$:

1. If $|V| = 1$, *reject*
2. Match the output of B (a decider for $LPATH$) on $\langle G, a, b, |V| - 1 \rangle$ "

The above mapping reduction takes constant time and $UHAMPATH$ is NP-complete, so $LPATH$ must also be NP-complete.

□

Problem 3

Let $DOUBLE-SAT = \{\langle\phi\rangle \mid \phi \text{ has at least two satisfying assignments}\}$. Show that $DOUBLE - SAT$ is NP-complete.

Proof: We first show that $DOUBLE - SAT$ is a member of NP:

$A =$ "On input $\langle\phi\rangle$:

1. Nondeterministically select two assignments for ϕ

2. If both assignments evaluate to true, *accept*; otherwise *reject* "

The above machine runs on polynomial time since selecting two assignments takes constant time and evaluating statements is polynomial. We now show $DOUBLE-SAT$ is NP-complete by reducing SAT. Consider the following TM: $R =$ "On input $\langle\phi\rangle$:

1. Construct $\alpha = \phi \wedge (y \vee \overline{y})$

2. Match the output of B (a decider for $DOUBLE - SAT$) on input $\langle\alpha\rangle$"

This reduction takes only constant time, so it must be the case that $DOUBLE - SAT$ is a member of NP-complete.

□

Problem 4

A subset of the nodes of a graph G is a dominating set if every other node of G is adjacent to some node in the subset. Let

$$DOMINATING - SET = \{\langle G, k\rangle \mid G \text{ has a dominating set with } k \text{ nodes}\}$$

Show that it is NP-complete by giving a reduction from $VERTEX - COVER$.

Proof: We first show that $DOMINATING - SET \in NP$ by giving it a non-deterministic decider.

$A =$ "On input $\langle G, k\rangle$:

1. Nondeterministically select a subset V' with $|V'| = k$
2. For each node v in $G \setminus V'$
3. If there is no edge (v, v') for any $v' \in V$, *reject*
4. All nodes tested, *accept*"

The above machine is $O(|E|)$, so $DOMINATING - SET \in NP$. We now show, through a reduction of $VERTEX - COVER$, that $DOMINATING - SET$ is a member of NP-complete. Consider the following description of a Turing Machine.

For $\langle G, k\rangle$, an input to VC, we then construct the input to DS. For each edge $(u, v) \in G$, add the node w_{uv} and edges (u, w_{uv}) and (v, w_{uv}). Then match the output of DS.

Note that the reduction takes only polynomial time with the number of edges in G. As far as correctness goes, VC accepts graphs which have subsets of nodes such that each edge contains one of the nodes, while DS accepts graphs where nodes are adjacent. The above TM takes any edge and adds a node in it's place (while keeping the original graph), which would force a DS to contain the new node, and so contain the original edge. The other direction can be shown with the inverse mapping (deleting w_{uv}).

So, $DOMINATING - SET$ is a member of NP and a NP-complete problem can reduce to $DOMINATING - SET$. So $DOMINATING - SET$ must be NP-complete.

□