Assignment 44

Alex Clemmer
Student number: u0458675

1

A problem is said to be $\in NP$ if we can solve it in nondeterministic polynomial time, or in other words, if there exists some algorithm $\in P$ that checks the solution of said problem. So all we really need to do is to show that there is some algorithm $\in P$ that checks the answer for this problem.

Fortunately, checking is stupid-simple: look through the k-tree we've generated, and check that no node has a degree $> k$. This is pretty clearly a polynomial-time deterministic algorithm.

2

The challenge of reduction is to show that every problem in some class T is expressible as a problem in some other class Q.

Fortunately, this is not a complicated reduction: if we have a Rudrata path, then each vertex must have 2 or fewer degrees, because each node is touched exactly once; if it had 3, at some point another path must pass through that vertex. Further, if we have a Rudrata path, it is a spanning tree, because we're touching every vertex *exactly* once.

So really it's the same problem. Since Rudrata is NP-complete, so is k-SpanningTree.

3

This is the first part of the induction step in our inductive proof that deciding k-SpanningTree is NP complete for $k \geq 2$. Basically our challenge is to show that any k is reducible to $k+1$ for $k \geq 2$.

The classical proof? Given some graph G with some 2-SpanningTree, add $k-2$ leaf vertices to every vertex, call this graph G'. If G has a 2-SpanningTree, then G' has a k-SpanningTree, since each node had 2 or fewer connected vertices, and we added $k-2$ to each of them. This shows that if there is no solution to the k problem, there is also no solution to the corresponding $k-1$ problem.

In the other direction, given the k-SpanningTree G', we can pull the $k-2$ leaf nodes from each and get a 2-SpanningTree back. This shows that if k has a solution, we can determine it using a corresponding $k+1$ SpanningTree.

This shows that there is a k to $k+1$ reduction: that is, for every k problem, it can be mapped to a $k+1$ problem. This is the definition of a mapping reduction.

4

Trivially follows from the above. We've shown that $k = 2$ is NP-complete, and that every k problem can be fully mapped and solved in terms of some $k+1$. Thus $\forall k \geq 2$, k-SpanningTree is NP-complete.