CS 170 DIS 14

Released on 2019-04-29

1 A Reduction Warm-up

In the Rudrata path problem (aka the Hamiltonian Path Problem), we are given a graph G and want to find if there is a path in G that uses each vertex exactly once.

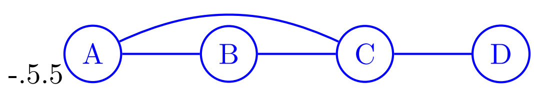
Is the following argument correct? Please justify your answer.

We will show that Undirected Rudrata Path can be reduced to Longest Path in a DAG. Given a graph G, use DFS to find a traversal of G and assign directions to all the edges in G based on this traversal (i.e. edges will point in the same direction they were traversed and back edges will be omitted). This gives a DAG. If the longest path in this DAG has $|V| - 1$ edges then there is a Rudrata path in G since any simple path with $|V| - 1$ edges must visit every vertex.

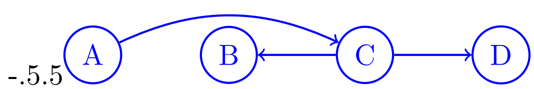
Solution: It is incorrect.

It is true that if the longest path in the DAG has length $|V| - 1$ then there is a Rudrata path in G. However, to prove a reduction correct, **you have to prove both directions**. That is, if you have reduced problem A to problem B by transforming instance I to instance I' then you should prove that I has a solution **if and only if** I' has a solution. In the above "reduction," one direction doesn't hold. Specifically, if G has a Rudrata path then the DAG that we produce does not necessarily have a path of length $|V| - 1$–it depends on how we choose directions for the edges.

For a concrete counterexample, consider the following graph:



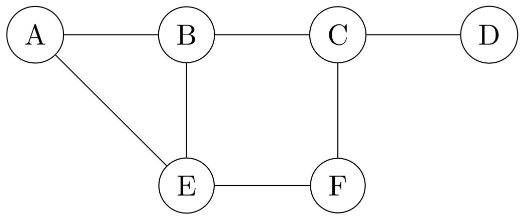
It is possible that when traversing this graph by DFS, node C will be encountered before node B and thus the DAG produced will be



which does not have a path of length 3 even though the original graph did have a Rudrata path.

2 Reducing Vertex Cover to Set Cover

In the minimum vertex cover problem, we are given an undirected graph $G = (V, E)$ and asked to find the smallest set $U \subseteq V$ that "covers" the set of edges E. In other words, we want to find the smallest set U such that for each $(u, v) \in E$, either u or v is in U (U is not necessarily unique). For example, in the following graph, $\{A, E, C, D\}$ is a vertex cover, but not a minimum vertex cover. The minimum vertex covers are $\{B, E, C\}$ and $\{A, E, C\}$.



Recall the following definition of the minimum Set Cover problem: Given a set U of elements and a collection $S_1, \ldots, S_m$ of subsets of U, what is the smallest collection of these sets whose union equals U? So, for example, given $U := \{a, b, c, d\}$, $S_1 := \{a, b, c\}$, $S_2 := \{b, c\}$, and $S_3 := \{c, d\}$, a solution to the problem is the collection of S_1 and S_3.

Give an efficient reduction from the Minimum Vertex Cover Problem to the Minimum Set Cover Problem.

Solution: Let $G = (V, E)$ be an instance of the Minimum Vertex Cover Problem. Create an instance of the Minimum Set Cover Problem where $U = E$ and for each $u \in V$, the set S_u contains all edges adjacent to u. Let $C = \{S_{u_1}, S_{u_2}, \ldots, S_{u_k}\}$ be a set cover. Then our corresponding vertex cover will be $u_1, u_2, \ldots, u_k$. To see this is a vertex cover, take any $(u, v) \in E$. Since $(u, v) \in U$, there is some set S_{u_i} containing (u, v), so u_i equals u or v and (u, v) is covered in the vertex cover.

Now take any vertex cover $u_1, \ldots, u_k$. To see that $S_{u_1}, \ldots, S_{u_k}$ is a set cover, take any $(u, v) \in E$. By the definition of vertex cover, there is an i such that either $u = u_i$ or $v = u_i$. So $(u, v) \in S_{u_i}$, so $S_{u_1}, \ldots, S_{u_k}$ is a set cover.

Since every vertex cover has a corresponding set cover (and vice-versa) and minimizing set cover minimizes the corresponding vertex cover, the reduction holds.