Homework 9

Adam Karl

October 20, 2020

1 Problem 27 (4 points)

1.1 SingleFixedHamiltonianPath $\leq$ DoubleFixedHamiltonialPath

To transform the input for SingleFixedHamiltonianPath to an input for DoubleFixedHamiltonialPath:

- Keep the same s
- Create a new vertex t, and add edges from t to every other vertex in G

Then run the algorithm for DoubleFixedHamiltonialPath on the graph and return whatever it returns. DoubleFixedHamiltonialPath will return true if and only if the original graph (before modification) had a hamiltonian path starting from s.

Since the transforms run in polytime, SingleFixedHamiltonianPath $\leq$ DoubleFixedHamiltonialPath.

1.2 DoubleFixedHamiltonianPath $\leq$ SingleFixedHamiltonialPath

To transform the input for DoubleFixedHamiltonianPath to an input for SingleFixedHamiltonialPath:

- Keep the same s
- Create a new vertex t' with one edge connecting to t

Creating t' in this way ensures that any Hamiltonian path MUST end at t' (coming directly from t)

Then run the algorithm for SingleFixedHamiltonialPath on the graph with the original s as the start and return whatever it returns. SingleFixedHamiltonialPath will return true if and only if the original graph (before modification) had a Hamiltonian path starting from s, and ending at t.

Since the transforms run in polytime, DoubleFixedHamiltonianPath $\leq$ SingleFixedHamiltonialPath.

1.3 SingleFixedHamiltonianPath $\leq$ HamiltonianCycle

To transform the input for SingleFixedHamiltonianPath to an input for HamiltonianCycle:

- Add edges from s to every other vertex in the graph (that it doesn't already have an edge to)

Then run the algorithm for HamiltonianCycle on the graph and return whatever it returns. HamiltonianCycle will return true if and only if the original graph (before modification) had a Hamiltonian path starting from s. Adding the edges from s to every other vertex in the graph ensures that at the end of the Hamiltonian path, the last vertex has an edge to return to s to complete the cycle

Since the transforms run in polytime, SingleFixedHamiltonianPath $\leq$ HamiltonianCycle.

1.4 HamiltonianCycle $\leq$ SingleFixedHamiltonianPath

To transform the input for HamiltonianCycle to an input for SingleFixedHamiltonianPath:

- Choose an arbitrary vertex in the graph as s
- Create a new vertex t, and add edges to every vertex that s has an
- Create a new vertex t', and only connect it to t.

Since t' only connects to one vertex, any valid Hamiltonian path MUST end at t'. In order to reach t', there must be a valid Hamiltonian path that starts at s, touches all other vertices before going to t, then t'. remember that s and t were split from the same vertex. Thus, running SingleFixedHamiltonianPath on this graph (with s as the start point), will return true if and only if the original graph had a Hamiltonian cycle.

Since the transforms run in polytime, HamiltonianCycle $\leq$ SingleFixedHamiltonianPath.

1.5 Conclusion

We have established:

- SingleFixedHamiltonianPath $\leq$ DoubleFixedHamiltonialPath
- DoubleFixedHamiltonianPath $\leq$ SingleFixedHamiltonialPath
- SingleFixedHamiltonianPath $\leq$ HamiltonianCycle
- HamiltonianCycle $\leq$ SingleFixedHamiltonianPath

Although we have not done all 6 pairwise reductions, this set is sufficient to conclude that each of the algorithms can be reduced to each other (for instance, to prove HamiltonianCycle $\leq$ DoubleFixedHamiltonianPath we could solve HamiltonianCycle by running the SingleFixedHamiltonianPath on it, but have that SingleFixedHamiltonianPath use the reduction to DoubleFixedHamiltonialPath we showed in 1.1).

Since all 3 algorithms can be reduced to each other in polytime, if we are given a polytime algorithm for any 1 of the three, we can use our reductions to find polytime algorithms for the other 2. Thus, if any one of these problems have a polynomial-time algorithm then they all have polynomial time algorithms.