

Homework 20

Brian Knotten, Brett Schreiber, Brian Falkenstein

October 14, 2018

3

5

To prove that if an algorithm for one of Undirected Graph Isomorphism, Directed Graph Isomorphism, and Mixed Graph Isomorphism implies that they all do, the following reductions need to be made:

Directed Graph Isomorphism \leq_p Mixed Graph Isomorphism

Assume there exists an algorithm for Mixed Graph Isomorphism called *MISO*. Then it is possible to construct a poly-time algorithm for Directed Graph Isomorphism called *DISO* as follows:

```
DISO(G, H) :  
    return MISO(G, H)
```

A purely directed graph can be thought of as a special case of a mixed graph, so an algorithm for Mixed Graph Isomorphism works just as well on inputs that only have directed edges. Moreover, since *MISO* is poly-time, and since *DISO* makes no transformations from input to input or from output to output, then *DISO* is also poly-time.

Undirected Graph Isomorphism \leq_p Mixed Graph Isomorphism

Assume there exists an algorithm for Mixed Graph Isomorphism called *MISO*. Then it is possible to construct a poly-time algorithm for Undirected Graph Isomorphism called *UIISO* as follows:

```
UIISO(G, H) :  
    return MISO(G, H)
```

A purely undirected graph can be thought of as a special case of a mixed graph, so an algorithm for Mixed Graph Isomorphism works just as well on inputs that only have directed edges. Moreover, since *MISO* is poly-time, and since *UIISO* makes no transformations from input to input or from output to output, then *UIISO* is also poly-time.

Undirected Graph Isomorphism \leq_p Directed Graph Isomorphism

Assume there exists an algorithm for Directed Graph Isomorphism called *DISO*. Then it is possible to construct a poly-time algorithm for Undirected Graph Isomorphism called *UIISO* as follows:

```
UIISO(G, H) :  
    Let G' = Directed(G)  
    Let H' = Directed(H)  
    return DISO(G', H')
```

Where *Directed* is a supplementary function as follows: *Directed*(*G*) :

```
for each undirected edge e between u and v in G:  
    add a directed edge from u to v into G' and  
    add a directed edge from v to u in G'  
return G'
```

An undirected graph can be thought of as a special case of a directed graph where each undirected edge corresponds to two mutual directed edges. Since two directed graphs can only be isomorphic if they both share these pairs of mutual directed edges, the algorithm remains correct after the input has been transformed. Since this transformation is polynomial in the number of edges, and since the output does not need to be transformed, then *UIISO* is also a poly-time algorithm.