

# CS 521: Data Structures and Algorithms I

## Homework 5

Dustin Ingram

December 3, 2011

1. **Solution:** Assuming that  $\text{HAM-CYCLE} \in P$ , we can propose an algorithm which either removes or adds each edge  $e \in G.E$  from  $G$  or an empty graph, respectively; at each iteration we can use  $\text{HAM-CYCLE}$  to determine if the graph still contains a hamiltonian cycle and permanently accept/reject that edge as necessary. In either case, once all vertices  $\in G.V$  are added, this results in a graph with a set of edges  $E$  which belong to a Hamiltonian Cycle, produced in polynomial time.
2. **Solution:** If the language  $\text{HAM-PATH} = \{\langle G, u, v \rangle : \text{there is a hamiltonian path from } u \text{ to } v \text{ in graph } G\}$  belongs to NP, then this means that it is possible to verify that there is a hamiltonian cycle in  $G$  given some certificate  $C$  in polynomial time. If  $C$  is given as a graph  $C(V', E')$ , this is verifiable by showing that there is a simple path  $P$  such that  $u \rightsquigarrow v \in C(V', E')$ , that this path uses all vertices  $\in C.V'$  and all edges  $\in C.E'$ , and that  $C.V' = G.V$  and  $C.E' \subset G.E$ , all of which can be done in polynomial time.
3. **Solution:** To show that the  $\leq_p$  relation is a transitive relation on languages, we first assume that  $L_1 \leq_p L_2$  and  $L_2 \leq_p L_3$ . Given that this is true, there must be some polynomial-time reduction function  $f_n(l)$  which reduces  $l$  such that

**if**  $l \in L_1$ , **then**  $f_1(l) \in L_2$

and

**if**  $l \in L_2$ , **then**  $f_2(l) \in L_3$

Since  $f_n(l)$  is a polynomial-time algorithm, then  $f_n(f_n(l))$  is polynomial-time as well, thus

**if**  $l \in L_1$ , **then**  $f_2(f_1(l)) \in L_3$

which thereby satisfies that  $L_1 \leq_p L_3$ .

4. **Solution:** To prove that the subgraph-isomorphism problem is NP-complete, it must be shown to be a reduction of some other known NP-complete

problem, in this case, the clique problem. First, we represent the subgraph-isomorphism problem as a decision problem:

**Given inputs**  $G_1, G_2$   $\begin{cases} \text{TRUE} & \text{if } G_1 \text{ is isomorphic to a subgraph of } G_2 \\ \text{FALSE} & \text{if } G_1 \text{ is not isomorphic to a subgraph of } G_2 \end{cases}$

Next, we represent the clique problem as a decision problem as well:

**Given inputs**  $G, k$   $\begin{cases} \text{TRUE} & \text{if a complete subgraph with } k \text{ vertices} \in G \\ \text{FALSE} & \text{if a complete subgraph with } k \text{ vertices} \notin G \end{cases}$

Finally, to reduce the clique problem into the subgraph-isomorphism problem, we simply let  $G_2 = G$  and let  $G_1$  be a completely connected graph with  $k$  total vertices. Since  $G_1$  is only isomorphic to  $G_2$  if  $G_2$  has a clique of size  $k$ , this is an correct reduction from an NP-complete problem; since the reduction can be performed in polynomial time, it is clear that the subgraph-isomorphism problem is therefore NP-complete as well.

## 5. Solution:

- (b) This case can be solved with a polynomial-time algorithm: after sorting the coins by value and representing them as a stack ordered from greatest to least, pop the first (largest) coin to Bonnie, then continue giving coins off the top of the stack to Clyde until he has the same amount (in which case the process is repeated) or there are no more coins (in which case it is impossible to evenly split the coins).
- (d) This case is clearly reducible from part (c), which is NP-complete as a reduction from the subset-sum problem. In (c), we wish to split the checks directly in half, with the maximum difference between each half being 1. In (d), the maximum difference is now 100; if we multiplied every value in (c) by 100 and solved for a maximum difference of 100, this would be analogous to the original problem statement.