

## EECS 336 – Homework 8

---

Weihan Chu

May 31, 2016

### 1 QUESTION 1

Define a certificate for the language graph isomorphism to be a permutation  $y$  that maps vertices of  $G_1$  to vertices  $G_2$ . The verification algorithm first checks whether  $y$  is a permutation. This step takes  $O(V)$ . Then checks that  $G_1$  is identical to  $G_2$ , which means the algorithm checks that  $(u, v)$  is an edge if and only if  $(y(u), y(v))$  is an edge. This step takes  $O(V + E)$ . So the total complexity is  $O(V + E)$ . GRAPH-ISOMORPHISM is in NP.

### 2 QUESTION 2

So for this problem, we need to first find the complement of TAUTOLOGY, which means there is any assignment of 1 and 0 to the input variables that formula evaluates to 0. Now we need to prove the complement of TAUTOLOGY  $\in NP$ . Assume that input is  $x$  and  $y$  is a certificate.

Step1: We need to first verify that  $y$  is a valid input, which means each  $x_i$  is 0 or 1. If the input can satisfy this requirement, we just go to step 2. If else, we will output FALSE. The time complexity of this step is  $O(K)$ .

Step2: We then use certificate as input variable to calculate the output and then evaluate whether the output is 0 or 1. This step takes  $O(1)$ .

So the final time complexity is  $O(K)$ . So this is in polynomial time. So we can get that the complement of TAUTOLOGY  $\in NP$ . So TAUTOLOGY  $\in co - NP$ .

### 3 QUESTION 3

#### 3.1 PART A

We can do this algorithm like follows. First, we take any vertex in the graph as the starting point and paint this vertex using the first color. Then we find all the vertices that are connected to the first vertex and paint all those vertices with the second color. Then we use the BFS method to find the vertices that are not visited and paint them. If all edges' two vertices are painted with different color then it is 2-colorable. If there are any edges whose both end vertices are painted with same color. Then the graph will not be 2-colorable.

#### 3.2 PART B

We can convert the graph coloring problem to the decision problem like: Given an undirected graph  $G(V, E)$  and an integer  $k$ , can we color the graph  $G$  with  $k$  colors such that adjacent vertices have different colors.?

- a. If we can solve this decision problem in polynomial time. What we need to do is only to assign the  $k$  from  $|V|$  to 2 and check whether the output of the decision problem is True. Then we just choose the minimum number from all numbers whose output is true, which actually means that the graph is colorable by this number of colors. So we solve this problem and in linear time. This number is what we want in GRAPH-COLORING problem. So we can know that the time is still polynomial if the decision problem can be solved in polynomial time.
- b. If the GRAPH-COLORING problem can be solved in polynomial time. Then we know the minimum number of colors needed to make the graph colorable. We assume this number is  $m$ . and we assume the given number in decision problem is  $k$ . If  $k < m$ , we can output false in decision problem. If  $k \geq m$ , we can output true in decision problem. So we just use constant time to solve this problem. So we can get that the time is still polynomial if the GRAPH-COLORING problem can be solved in polynomial time.

#### 3.3 PART C

If given a colored graph  $G$  and  $k$ , we can verify it has  $k$  colors and adjacent vertices have different colors in polynomial time. So this problem is in NP.

Then we add  $k - 3$  vertex to  $G$  and they are connected to other vertices to form a new graph  $G_1$ . From the proof in part b, we know that  $G$  will be 3-colorable only when  $G_1$  be painted in  $k$  color. Since this conversion takes polynomial time and the decision problem is in NPC. So the 3-COLOR problem is also in NPC.

## 4 QUESTION 4

### 4.1 PART A

Input to the reduction:

Graph  $G = (V, E)$

Output to the reduction:

Boolean expression  $\phi$  in CNF with variables  $X$  and clauses  $C$

For the description of  $\phi$  in terms of  $G$ :

If  $|V| = [1, 2, 3, \dots, n]$ , then  $X = [x_{iv} | 1 \leq i, v \leq n]$

And  $x_{iv}$  means the node  $v$  is the  $i$ th node of the Hamiltonian circle

We will construct  $C$  as follows:

1. For nodes  $1 \leq v \leq n$ , node  $v$  must appear in the path

$(x_{1v} \vee x_{2v} \vee \dots \vee x_{nv})$ , this step takes  $O(n^2)$

2. For node  $1 \leq v \leq n$ , node  $v$  must appear only once in the path

$(\neg x_{iv} \vee \neg x_{kv}, 1 \leq i, k, v \leq n, i \neq k)$ , this step takes  $O(n^3)$

3. For node  $1 \leq v \leq n$ , some node  $v$  must be the  $i$ th node

$(x_{i1} \vee x_{i2} \vee \dots \vee x_{in})$ , this step takes  $O(n^2)$

4. For node  $1 \leq v \leq n$ , two nodes  $v$  and  $w$  cannot both be the  $i$ th

$(\neg x_{iv} \vee \neg x_{ik}, 1 \leq i, k, v \leq n, v \neq k)$ , this step takes  $O(n^3)$

5. Adjacent nodes in the path must also be adjacent in the graph.

$(\neg x_{ki} \vee \neg x_{k+1,v})$ , for all  $(i, j)$  not in  $G$  and  $1 \leq k \leq n-1$ , this step takes  $O(n^3)$

So we can convert the Hamiltonian circle to CNF-SAT in polynomial time. So we can get that  $HC \leq CNF-SAT$ .

### 4.2 PART B

We still use the same parameter as in part a.

So we can construct as follows:

1. For nodes  $1 \leq v \leq n$ , node  $v$  must appear in the path  
( $x_{1v} + x_{2v} + \dots + x_{nv} \geq 1$ ), this step takes  $O(n^2)$

2. For node  $1 \leq v \leq n$ , node  $v$  must appear only once in the path  
( $x_{iv} + x_{kv} \leq 1, 1 \leq i, k, v \leq n, i \neq k$ ), this step takes  $O(n^3)$

3. For node  $1 \leq v \leq n$ , some node  $v$  must be the  $i_h$  node  
( $x_{i1} + x_{i2} + \dots + x_{in} \geq 1$ ), this step takes  $O(n^2)$

4. For node  $1 \leq v \leq n$ , two nodes  $v$  and  $w$  cannot both be the  $i_h$   
( $x_{iv} + x_{ik} \leq 1, 1 \leq i, k, v \leq n, v \neq k$ ), this step takes  $O(n^3)$

5. Adjacent nodes in the path must also be adjacent in the graph.  
( $x_{ki} + x_{k+1,v} \leq 1$ ), for all  $(i, j)$  not in  $G$  and  $1 \leq k \leq n - 1$ , this step takes  $O(n^3)$

So we can convert the Hamiltonian circle to 01-LP in polynomial time. So we can get that  
 $HC \leq 01\text{-LP}$ .