18/10/20

TOC - Assignment 4

Kousshik Raj
17CS30022

2\. a) • HAMCYCLE is in NP

Proof:- Here, the certificate x will be a sequence of vertices s.t $|x| = n$. It is easy to see that this certificate can be verified by checking if all the vertices are distinct and there is an edge between 2 consecutive vertices and the first and last vertices. All of these can be done in polynomial time.

• HAMCYCLE $\in$ NP-hard

Proof:- Here, we will show a reduction HAMPATH $\leq_p$ HAMCYCLE and since we know HAMPATH is NP-hard, we can say HAMCYCLE is also NP-hard.

Reduction:-

Let $G = (V, E)$ be an instance of HAMPATH. Construct $G' = (V', E')$ where $V' = V \cup \{v_0\}$ & $v_0 \notin V$,

$E' = E \cup \{(v_0, u) \mid \forall u \in V\}$.

Claim:- $\langle G \rangle \in$ HAMPATH $\iff \langle G' \rangle \in$ HAMCYCLE

$\Rightarrow$ Let $v_1 v_2 \dots v_n$ be the hamiltonian path. Since (v_0, v_1) & $(v_n, v_0) \in E'$, the sequence $v_0 v_1 v_2 \dots v_n$ is an hamiltonian cycle of G'

$\Leftarrow$ Let $a\, b\, c \dots z$ be the hamiltonian cycle. Since this is an hamiltonian cycle of G', we can write the cycle as $a\, b\, c \dots v_0 \dots z$. If we disconnect v_0 from its neighbours, we get an hamiltonian path of G with v_0's neighbours as terminal vertices

Kaushik Raj
17CS30022

Thus, the reduction is valid

$\boxed{\therefore \text{HAMCYCLE is NP-complete.}}$

2. b)

- TSP $\in$ NP

Proof:- ~~Again~~ Here as well, the certificate x will be a sequence of vertices such that $|x| = n$. It is easy to verify this certificate by ensuring that all the vertices are distinct and sum of the weights between consecutive vertices and the first & last vertices is less than k. All of these can be done in polynomial time

- TSP $\in$ NP-hard

Proof:- We will show this as a reduction HAMCYCLE $\leq_p$ TSP. Since, we have just seen that HAMCYCLE $\in$ NP-hard, then TSP also $\in$ NP-hard.

Reduction:- Given an instance $G = (V, E)$ of HAMCYCLE. Construct an instance $\langle V', w, k \rangle$ of TSP. Here,

$$V' = V$$

$$w(i,j) = \begin{cases} 0 & \text{if } (i,j) \text{ or } (j,i) \in E \\ 1 & \text{otherwise} \end{cases}$$

$$k = 0$$

Claim:- $\langle G \rangle \in$ HAMCYCLE $\iff \langle V', w, k \rangle \in$ TSP

Kausshik Raj
17CS30022

$\Rightarrow$ If $\langle G \rangle \in$ HAMCYCLE, then there exists a cycle in G that visits all vertex exactly once. Since, the weight of the edges in G is 0 in the corresponding TSP instance, then there exists a tour of all vertices that visits each exactly once and the sum of weights is 0. Thus $\langle V', w, k \rangle \in$ TSP.

$\Leftarrow$ If $\langle V', w, k \rangle \in$ TSP, then there exists a tour of all vertices that goes through only 0 weight edges, as $k = 0$. This means that, there exists a sequence of edges that has a corresponding edge in $\langle G \rangle$, travelling through which each vertex in V', (and thereby in V) is visited exactly once and returns to the start.

Thus, $\langle G \rangle \in$ HAMCYCLE.

$\therefore$ The reduction is valid.

Hence, TSP is NP-complete

1. NAE-3SAT $\in$ NP

Proof:- Here, the certificate will be the truth assignment and in polynomial time we can verify whether for each clause there is one truth literal and one false one atleast.

- NAE-3SAT $\in$ NP-hard

Proof:- We will show this as a reduction 3-SAT $\leq_p$ NAE-4SAT $\leq_p$ NAE-3SAT.

Koushik R
17CS3002

Since, we know that 3-SAT is NP-hard, by the reduction we show that NAE-4SAT and thereby NAE-3SAT is NP-hard.

i) <u>Red I:-</u> 3-SAT $\leq_p$ NAE-4SAT

Given a instance Φ of 3-SAT, construct an instance Φ^* of NAE-4SAT. For each clause $c_i \in \Phi$ of the form $c_i = (x_i \vee x_j \vee x_k)$ construct a new clause $c_i^* \in \Phi^*$ such that $c_i^* = (y_i \vee y_j \vee y_k \vee z)$. That is for each literal x_i in Φ, create a corresponding literal y_i in Φ^*. Moreover, add a common variable z in Φ^* as well.

claim:- $\Phi \in$ 3-SAT $\Longleftrightarrow$ $\Phi^* \in$ NAE-4SAT

~~=> If there Assign z=1, if all other liter~~

~~=> Assign z=1. If $x_i \in$~~

=> Let there be a satisfying assignment f for Φ. Construct a satisfying assignment f^* for Φ^* in the following way:-

$$f^*(z) = 1$$

$$f^*(y_i) = \begin{cases} 0 & \text{if } f(x_i) = 0 \\ 1 & \text{else} \end{cases}$$

All clauses have $z=1$ as the truth assignment, and there will be one false assignment because one of the corresponding literal in Φ has to be true.

Kousshik Roy
17CS30022

Thus, $\Phi^* \in$ NAE-4SAT

$\Leftarrow$ Let the satisfying assignment be f^* for Φ^*. Construct f for Φ in the following way:-

$$f(x_i) = \begin{cases} 1 & \text{if } f^*(y_i) \neq f(z) \\ 0 & \text{else} \end{cases}$$

We can see f satisfies Φ because there will atleast be one pair in a clause that are not the same. Thus,

$$\Phi \in 3\text{-SAT}.$$

Thus, NAE-4SAT is NP-hard.

ii) Red 2:- NAE-4SAT $\leq_p$ NAE-3SAT

Given an instance Φ of NAE-4SAT, construct an instance Φ^* of NAE-3SAT. For each $C_i \in \Phi$ of the form $C_i = (y_i, y_j, y_k, z)$, construct 2 new clauses $C_{i_1}, C_{i_2} \in \Phi^*$, where $C_{i_1} = (y_i, y_j, a)$, $C_{i_2} = (y_k, z, \bar{a})$ and a is a new variable corresponding to the clause. That is, there will be m new variables, where m is the number of clauses.

claim:- $\Phi \in$ NAE-4SAT $\Leftrightarrow \Phi^* \in$ NAE-3SAT

$\Rightarrow$ Let f be a satisfying assignment for Φ. Construct f^* for Φ^* by:-

$$f^*(y_i) = f(y_i)$$
$$f^*(z) = f(z)$$
$$f(a) = \begin{cases} 1 & f(y_i) \wedge f(y_j) = 0 \\ 0 & \text{else} \end{cases}$$

Its easy to see that f^* satisfies Φ^* as long as $y_i = y_j = y_k = z$ is not true.

Scanned with CamScanner

And we know that, it is not possible

$\Leftarrow$ Let f^* be a satisfying assignment for Φ^*. Construct f that satisfies Φ :-

$$f(y_i) = f^*(y_i)$$
$$f(z) = f^*(z)$$

We can see that f satisfies Φ because, $f(y_i) = f(y_j) = f(y_k) = f(z)$ is not possible as otherwise f^* won't be a satisfying assignment for Φ^*.

Thus, the reduction is valid

$\boxed{\therefore \text{ NAE} - \text{3SAT is NP} - \text{complete}}$

Kousshik Raj
17CS30022

Scanned with CamScanner