COMP 3721

Tutorial 10

P

Definition

The class **P** consists of all decision problems (languages) that are solvable in polynomial time. That is, there exists an algorithm that decides in polynomial time if any given input is a yes-input or a no-input.

Theorem

P *is closed under complement, union, intersection, concatenation, and Kleene star.*

NP

Definition

A nondeterministic TM runs in polynomial time if for any input x, the number of steps of any computation path is $O(n^c)$, where c is a constant and $n = |x|$ is the input size. The class **NP** consists of all decision problems that can be decided by a nondeterministic TM in polynomial time.

Remark: **NP** stands for "nondeterministic polynomial time", not "non-polynomial"!

Theorem

NP *is closed under union, intersection, concatenation and Kleene star.*

NP

Theorem

A decision problem belongs to **NP** *iff for each yes-input, there exists a certificate which allows one to verify in polynomial time that the input is indeed a yes-input.*

P and NP

1) Show that regular languages are in ***P***.

P and NP

1) Show that regular languages are in ***P***.

P is the class of languages that can be **decided** by a **deterministic** Turing machine in **polynomial** time.

Idea:

- DFA can be regarded as a special kind of Turing machine.
- For any given input, after reading a symbol, the reading head moves one square to the right, the finite control enters a new state, which is deterministically dependent on the current state and current input symbol.
- After reading the entire input string, the finite control decides whether the input string is accepted or not.

P and NP

2) Prove that ***P*** is closed under Kleene star.

Hints: Use Dynamic Programming.

P and NP

2) Prove that ***P*** is closed under Kleene star.

Proof:

Let *L* be any language in ***P***. Let *A* be the polynomial time algorithm decides *L*, then $\mathrm{A} \in P$. Assume $A = O(n^k)$.

We want to show $A^* \in P$, where A^* is the algorithm that decides L^*.

Let the input string w=$w_1 . . . w_n$.

$w \in L^*$ if and only if at least one of the following conditions is true.

- w=e
- $w \in L$
- $\exists u, v \colon w = uv \text{ and } u \in L^* \text{ and } v \in L^*$

P and NP

2) Prove that ***P*** is closed under Kleene star.

Proof:

Subproblems:

For each $1 \le i \le j \le n$, we use $f(i, j)$ to indicate whether the substring $w_{i,j} = w_i \ldots w_j$ is in L^*.

If $w_{i,j}$ is in L^*, $f(i, j) = 1$; otherwise $f(i, j) = 0$.

Our goal is to compute $f(1, n)$.

P and NP

2) Prove that ***P*** is closed under Kleene star.

On input w=$w_1 . . . w_n$
if w=e, then accept
else:
    for $l \leftarrow 1$ *to n*:
        for $i \leftarrow 1$ *to n-(l -1):*
            $j \leftarrow \mathrm{i} + l - 1$
            Run A on $w_{i,j}$
                if A accepts $w_{i,j}$, then $f(i,j) \leftarrow 1$
                else:
                    for $k \leftarrow i$ to j-1:
                        if f(i,k)=1 and f(k+1,j)=1
                            then $f(i,j) \leftarrow 1$
if f(1,n)=1, then accept; else reject.

P and NP

2) Prove that ***P*** is closed under Kleene star.

Proof:

Analyze the time complexity of the decider:

There are 3 nested loops in the algorithm, each of which can be traversed at most O(n) time.

In the second loop we run A on an input of length at most n.

The total time is at most $O(n) \cdot \left(O\left(n^k\right) + O(n)\right) \cdot O(n) = O(n^{2+\max(k,1)})$. So $A^* \in P$.

P and NP

3) Prove that graph isomorphism problem is in ***NP***.

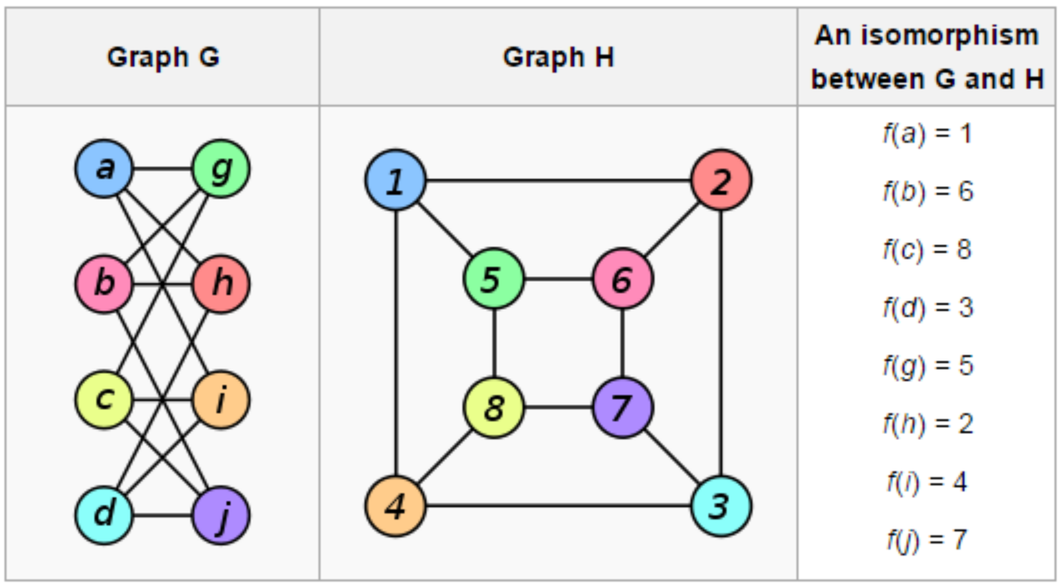
The graph isomorphism problem determining whether two finite graphs, G and H, are isomorphic.

In graph theory, an isomorphism of graphs G and H is a bijection between the vertex sets of G and H

such that *any* two vertices *u* and *v* of G are adjacent in G if and only if *f(u)* and *f(v)* are adjacent in H.

P and NP

3) Prove that graph isomorphism problem is in ***NP***.



https://commons.wikimedia.org/wiki/File:Graph_isomorphism_a.svg#/media/File:Graph_isomorphism_a.svg

P and NP

3) Prove that graph isomorphism problem is in ***NP***.

The graph isomorphism problem determining whether two finite graphs, G and H, are isomorphic.

Theorem

A decision problem belongs to **NP** *iff for each yes-input, there exists a certificate which allows one to verify in polynomial time that the input is indeed a yes-input.*

1. Find the certificate.
2. Prove that it can be verified in polynomial time.

P and NP

3) Prove that graph isomorphism problem is in ***NP***.

Proof: Input: two graphs G and H.
Certificate: a map $f\colon V_G \to V_H$.

Verify:

1. Check if f is a bijection, that is, if $f(V_G)$ is a permutation of V_H. If no, return false; else continue.
2. Permute V_G as given by $f(V_G)$. Verify that the permuted G is identical to H.

Step 1 takes at most $O(n^2)$ where n=#vertices. Step 2 runs in $O(n+e)$ where e=#edges. The algorithm runs in $O(n^2)$. So $graph\ isomorphism\ problem \in NP$.