<Discrete Mathematics>

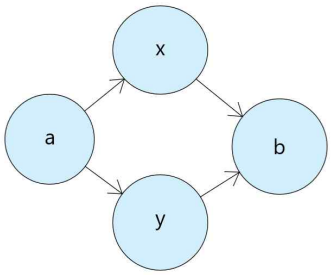
작성자_2018320161_송대선

작성일_05_31

NP={x|x is a decision problem such that a verification is done efficiently}
verification: a yes-instance(size: n), a certificate에 대하여
a yes-instance가 정말로 yes를 반환하는지를 a certificate를 이용하여 증명하는 것.
NP는 verification algorithm이 yes-instance에 대하여 a certificate를 이용하여 n^k의 step 안에 그 yes-instance가 정말로 yes를 반환한다는 것을 보여줄 수 있는 문제들의 집합이다.

instance: input들의 후보
yes-instance: output이 yes가 나오는 input들의 후보
certificate: yes-instance가 output이 yes가 나오게 만든다는 증거

-reachability problem-
input: a directed graph G, node a, node b
output:
YES, if there exists a path from a to b
NO, otherwise



certificate: (a, x, b), (a, y, b)

NP: a certifiacte가 a yes-instance가 정말로 yes를 반환하게 한다는 것을 효율적으로 검증 가능한 문제들의 집합

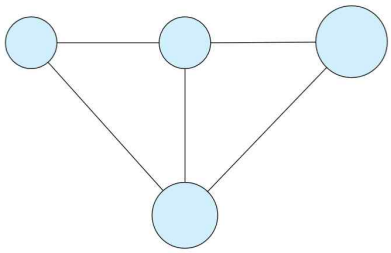
Definition: NPC
(1)
If there exists an efficient algorithm “A” which solves any one of problems in NPC. Then P=NP
≡every problem in NP is solvable efficiently
decision problem들의 집합의 임의의 문제 x를 효율적으로 해결하는 알고리즘이 존재하는데, 그 알고리즘이 모든 NP문제를 효율적으로 해결하면 그 집합이 NPC이다.

(2) a decision problem x ∈ NPC if
a. x ∈ NP
b. All problems in NP can be reduced to x efficiently

HAM, TSP, 3 CNF SAT, SAT ∈ NPC

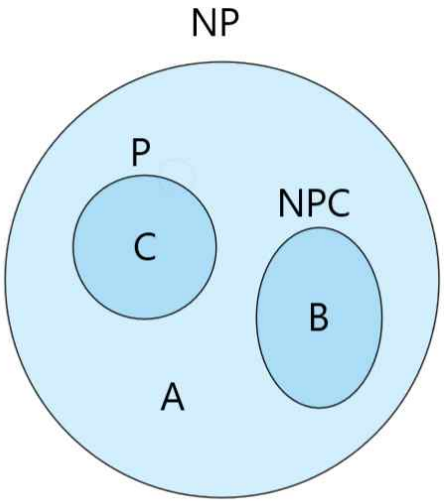
-3-coloring problems-
input: an undirected graph
output:
YES, if G is 3-colorable
NO, otherwise

3-colorable:
all nodes are assigned a color such that adjacent nodes are colored differently



-Fagin`s theorem-
NP Problem -> ESO로 표현가능

-Sudoku problem-
input: 완성되지 않은 $n^2 \times n^2$짜리 스토쿠 판
output:
YES, if
every row must have all elements,
every col must have all elements,
every box must have all elements
NO, otherwise



C ≤ B
C can not be harder than B

A ≤ B
A can not be harder than B

A VS C ?

(1) a proof VS a validation of a proof
누가 더 어려운가?

a proof와 a validation of a proof보다 더 어려운 것이 증명되면, P≠NP이다.
a proof와 a validation of a proof이 동일한 난이도인 것이 증명되면, P=NP이다.

(2) solving VS verifying
누가 더 어려운가?

solving이 verifying보다 더 어려운 것이 증명되면, P≠NP이다.
solving과 verifying이 동일한 난이도인 것이 증명되면, P=NP이다.

solving이 효율적이다. ≡ 문제해결 algorithm의 step이 n^k이다. -> P
verifying이 효율적이다. ≡ verify algorithm의 step이 n^k이다. -> NP

사실 a proof를 문제를 prove하는 solve라고 생각할 수 있고,
그럼 a validation of a proof은 verifying이라고 생각 가능하다.

a property P(x)= solving이 verifying보다 어렵다.
{x|x is a world in which P(x) is true}
{x|x is a world in which P(x) is not true}-> 이걸 찾으면 P≠NP이다.