Introduction to Algorithms

Topic 9-1 : NP-Completeness

Xiang-Yang Li and Haisheng Tan

School of Computer Science and Technology
University of Science and Technology of China (USTC)

Fall Semester 2021

Outline of Topics

3-SAT

- **literals**
  - x = variable.
  - $\neg x$ = negation of a variable.
- **clause** = disjunction of literals.
- **Boolean Formula**
  a collection of clauses, where each clause have exactly 3 literals connected with $\vee$ and each clause is connected with $\wedge$.
  $(x_1 \vee \neg x_1 \vee \neg x_2) \wedge (x_3 \vee x_2 \vee x_4) \wedge (\neg x_1 \vee \neg x_3 \vee \neg x_4)$

$x \vee y \vee z.$

$x \vee y \vee \neg z.$

$x \vee \neg y.$

$\neg x.$

3-SAT

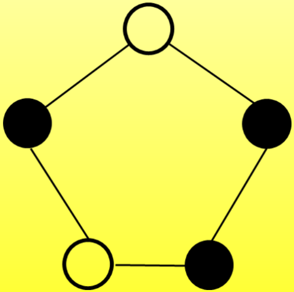
For a particular collection of clauses.
$(x_1 \vee \neg x_1 \vee \neg x_2) \wedge (x_3 \vee x_2 \vee x_4) \wedge (\neg x_1 \vee \neg x_3 \vee \neg x_4)$
Given an instance(one input to a particular problem) of 3-SAT, for example:

- $x_1 = 1, x_2 = 1, x_4 = 0$
- $x_1 = 0, x_2 = 0, x_4 = 1$

We can verify in polynomial time that whether an instance is correct to the clauses.
Whereas, could we get a solution in polynomial time?

Independent Set

- **Independent Set** subset S of vertices such that no two vertices in S are connected



Independent Set Decision/Optimizaiton version

- **IS OPTIMIZATION VERSION**
  - **instance**: graph G
  - **solution**: independent set S in G
  - **measure**: maximize the size of S
- **IS DECISION VERSION**
  - **instance**: graph G, number K
  - **question**: does G have independent set of size $\geq K$
- For an optimizaiton problem, the related decision problem is in a sense "easier", or at least "no harder".

Independent Set

For a particular graph G and a number k.

the YES answer can be certified as an instance - an independent set, we can verify in polynomial time that whether the instance satisfies G and k .

Whereas, could we get a solution in polynomial time?

P & NP

- **P** = decision problems that can be solved in polynomial time.
- **NP(Non-deterministic Polynomial)** = decision problems for which the YES answer can be certified and this certificate can be verified in polynomial time.
- if we can solve a problem in polynomial time, then we can verify the problem in polynomial time.
- $P \subseteq NP$
- 3-SAT problem and Independent Set problem are NP problems.

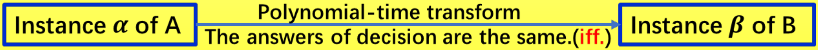
3-SAT vs. Independent Set

Which one is "easier" to solve?

- Independent Set.
- 3-SAT.

Reduction

* Given a decision problem A.
* Given a different decision problem B.
* **an instance**: the input to a particular problem.



a polynomial-time reduction algorithm

Reduction

If we can solve problem B effectively, and A can reduce to B in polynomial time, then we can solve problem A effectively.

In some sense, if A can reduce to B in polynomial time, then we can say problem A is "easier", or at least "no harder" than B. Denoted by $A \leq_p B$

3-SAT $\leq_p$ Independent Set

If we have a black-box for Independent Set then we can solve 3-SAT problem using the black-box in polynomial time.

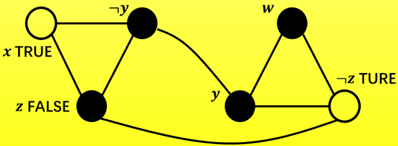
Give me a graph G and a number k and I will tell you if G has independent set of size $\geq K$

3-SAT $\leq_p$ Independent Set

3-SAT with k clauses reduces to Independent Set with number k in graph G.
Every literal in the clauses corresponds to a vertice on the graph. If two literals are in the same clause or one literal are negation of the other, then the related two vertices have an edge.

For example:

- $x \vee \neg y \vee z$
- $w \vee y \vee \neg z$
- two clauses($k = 2$)



3-SAT $\leq_p$ Independent Set

- Independent Set $\Rightarrow$ 3-SAT:
  If the graph has an independent set of k vertices, then each vertex must come from a different clause. To obtain a satisfying assignment, we assign the value TRUE to each literal in the independent set. Since contradictory literals are connected by edges, this assignment is consistent. There may be variables that have no literal in the independent set; we can set these to any value we like. The resulting assignment satisfies the original 3-SAT formula.
- 3-SAT $\Rightarrow$ Independent Set:
  If we have a satisfying assignment, then we can choose one literal in each clause that is TRUE. Those literals form an independent set in the graph.

3-SAT & Independent Set

- 3-SAT $\leq_p$ Independent Set. If Independent Set is in P then 3-SAT is in P.

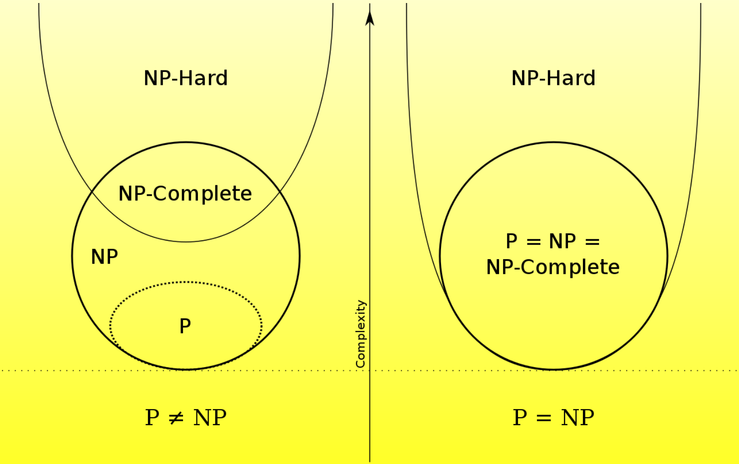
- We can also prove Independent Set $\leq_p$ 3-SAT . If 3-SAT is in P then Independent Set is in P.

Clique
Subset-Sum
3-COL
Planar 3-COL
Hamiltonian path
3-SAT
Independent Set

more reduction

NPC & NP-hard

- **P:** decision problems that can be solved in polynomial time.
- **NP:** decision problems for which the YES answer can be certified and this certificate can be verified in polynomial time.
- B is **NP-hard**:
  if every problem $A \in NP$
  $$A \leq_p B$$
- B is **NP-complete (NPC)**:
  if B is NP-Hard, and
  $$B \in NP$$



P=NP? One of 7 Millennium Prize Problems

NPC / NP-hard: How to prove

- **P** decision problems that can be solved in polynomial time.
- **NP** decision problems for which the YES answer can be certified and this certificate can be verified in polynomial time.

- B is **NP-hard**:
  if every problem $A \in NP$,
  $$\text{A} \leq_p \text{B}$$

- B is **NP-complete (NPC)**:
  if B is NP-Hard, and
  $$B \in NP$$

NPC / NP-hard: How to prove

- **P** decision problems that can be solved in polynomial time.
- **NP** decision problems for which the YES answer can be certified and this certificate can be verified in polynomial time.

- B is **NP-hard**:
  if we have already had a probem A ∈NP-hard, then we only need to prove
  $$A \leq_p B$$

- B is **NP-complete (NPC)**:
  if B is NP-Hard, and
  $$B \in NP$$

COOK's theroem

NP Decision problems for which the YES answer can be certified and this certificate can be verified in polynomial time.

COOK's theroem:

Every problem $A \in NP$,

$$\text{A} \leq_p \text{SAT}$$

Recall:

SAT is the Boolean satisfiability problem, and SAT $\leq_p$ 3-SAT.

NPC / NP-hard: How to prove

NP Decision problems for which the YES answer can be certified and this certificate can be verified in polynomial time.

> COOK's theroem:
> Every problem $A \in NP$,
> $$A \leq_p \text{SAT}$$

So, we have:
IS is NPC, since 3-SAT $\leq_p$ IS, and IS $\in$ NP.