Design and Analysis of Algorithms CS218M

NP Complete Problems

Paritosh Pandya

Indian Institute of Technology, Bombay

Autumn, 2022

$\mathbb{NP}$-Complete Problems

A problem $L \subseteq \Sigma^*$ is called $\mathbb{NP}$-complete (denoted $\mathbb{NPC}$) if

- $L \in \mathbb{NP}$.
- For every $L' \in \mathbb{NP}$, we have $L' \leq_P L$.
  (This shows that L is at least as hard as L'.)

If only second condition is satisfied we say that L is $\mathbb{NP}$-hard.

$\mathbb{NP}$-Complete Problems

A problem $L \subseteq \Sigma^*$ is called $\mathbb{NP}$-complete (denoted $\mathbb{NPC}$) if

- $L \in \mathbb{NP}$.
- For every $L' \in \mathbb{NP}$, we have $L' \leq_P L$.
  (This shows that L is at least as hard as L'.)

If only second condition is satisfied we say that L is $\mathbb{NP}$-hard.

Theorem

CIRCUIT_SAT is $\mathbb{NP}$-Complete.

$\mathbb{NP}$-Complete Problems

A problem $L \subseteq \Sigma^*$ is called $\mathbb{NP}$-complete (denoted $\mathbb{NPC}$) if

- $L \in \mathbb{NP}$.
- For every $L' \in \mathbb{NP}$, we have $L' \leq_P L$.
  (This shows that L is at least as hard as L'.)

If only second condition is satisfied we say that L is $\mathbb{NP}$-hard.

Theorem

CIRCUIT_SAT is $\mathbb{NP}$-Complete.

Proving $L \in \mathbb{NPC}$ by Reduction

- To show that L is $\mathbb{NP}$-hard, we reduce in polytime a known $\mathbb{NPC}$ problem L' to L.
- We also show that $L \in \mathbb{NP}$.

$\mathbb{NP}$-Complete Problems

A problem $L \subseteq \Sigma^*$ is called $\mathbb{NP}$-complete (denoted $\mathbb{NPC}$) if

- $L \in \mathbb{NP}$.
- For every $L' \in \mathbb{NP}$, we have $L' \leq_P L$.
  (This shows that L is at least as hard as L'.)

If only second condition is satisfied we say that L is $\mathbb{NP}$-hard.

Theorem

CIRCUIT_SAT is $\mathbb{NP}$-Complete.

Proving $L \in \mathbb{NPC}$ by Reduction

- To show that L is $\mathbb{NP}$-hard, we reduce in polytime a known $\mathbb{NPC}$ problem L' to L.
- We also show that $L \in \mathbb{NP}$.

Theorem

If $L' \leq_P L$ and L' is $\mathbb{NPC}$ then L is $\mathbb{NP}$-hard. Additionally if $L \in \mathbb{NP}$ then L is $\mathbb{NPC}$.

$3CNF\_SAT$ is $\mathbb{NP}$-Complete

$3CNF\_SAT$ is $\mathbb{NP}$-Complete

- Example $(\neg x_v \vee x_u \vee x_z) \wedge (\neg x_v \vee x_w \vee \neg x_z) \wedge (x_v \vee \neg x_u \vee \neg x_w)$

$3CNF\_SAT$ is $\mathbb{NP}$-Complete

- Example $(\neg x_v \vee x_u \vee x_z) \wedge (\neg x_v \vee x_w \vee \neg x_z) \wedge (x_v \vee \neg x_u \vee \neg x_w)$
- A boolean formula ϕ in the form $C_1 \wedge C_2 \wedge \ldots \wedge C_m$ where each clause C_i has the form $(l^i_1 \vee l^i_2 \vee l^i_3)$ where literal l is x or $\neg x$ for a propositional letter x is called $3CNF$ formula.

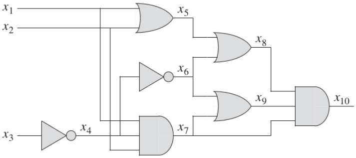
$3CNF\_SAT$ is $\mathbb{NP}$-Complete

- Example $(\neg x_v \vee x_u \vee x_z) \wedge (\neg x_v \vee x_w \vee \neg x_z) \wedge (x_v \vee \neg x_u \vee \neg x_w)$
- A boolean formula ϕ in the form $C_1 \wedge C_2 \wedge \ldots \wedge C_m$ where each clause C_i has the form $(l_1^i \vee l_2^i \vee l_3^i)$ where literal l is x or $\neg x$ for a propositional letter x is called $3CNF$ formula.
- $3CNF\_SAT$ is the collection of satisfiable $3CNF$ formulas. It is easy to see that $3CNF\_SAT \in \mathbb{NP}$ (why?)

$3CNF\_SAT$ is $\mathbb{NP}$-Complete

- Example $(\neg x_v \vee x_u \vee x_z) \wedge (\neg x_v \vee x_w \vee \neg x_z) \wedge (x_v \vee \neg x_u \vee \neg x_w)$
- A boolean formula ϕ in the form $C_1 \wedge C_2 \wedge \ldots \wedge C_m$ where each clause C_i has the form $(l_1^i \vee l_2^i \vee l_3^i)$ where literal l is x or $\neg x$ for a propositional letter x is called $3CNF$ formula.
- $3CNF\_SAT$ is the collection of satisfiable $3CNF$ formulas. It is easy to see that $3CNF\_SAT \in \mathbb{NP}$ (why?)
- We show that $CIRCUIT\_SAT \ \leq_P \ 3CNF\_SAT$.

Reduction: We trasform a circuit C to formula $\phi(C)$



$$\begin{aligned}\phi = x_{10} & \wedge (x_4 \leftrightarrow \neg x_3) \\ & \wedge (x_5 \leftrightarrow (x_1 \vee x_2)) \\ & \wedge (x_6 \leftrightarrow \neg x_4) \\ & \wedge (x_7 \leftrightarrow (x_1 \wedge x_2 \wedge x_4)) \\ & \wedge (x_8 \leftrightarrow (x_5 \vee x_6)) \\ & \wedge (x_9 \leftrightarrow (x_6 \vee x_7)) \\ & \wedge (x_{10} \leftrightarrow (x_7 \wedge x_8 \wedge x_9)) .\end{aligned}$$

Reduction: We trasform a circuit C to formula $\phi(C)$

- Transform circuit C to conjunction of clauses as in previous slide.
- A multi-input AND is replaced by a cascade of 2-input AND with extra variables (Also OR.)

Reduction: We trasform a circuit C to formula $\phi(C)$

- Transform circuit C to conjunction of clauses as in previous slide.
- A multi-input AND is replaced by a cascade of 2-input AND with extra variables (Also OR.)
- Each clause gives rise to a set of $3CNF$ clauses.
  Clause $x_v \leftrightarrow \neg x_u$ gives rise to equivalent set of clauses $(x_v \vee x_u) \wedge (\neg x_v \vee \neg x_u)$.

Reduction: We trasform a circuit C to formula $\phi(C)$

- Transform circuit C to conjunction of clauses as in previous slide.
- A multi-input AND is replaced by a cascade of 2-input AND with extra variables (Also OR.)
- Each clause gives rise to a set of $3CNF$ clauses.
  Clause $x_v \leftrightarrow \neg x_u$ gives rise to equivalent set of clauses
  $(x_v \vee x_u) \wedge (\neg x_v \vee \neg x_u)$.
- Clause $x_v \leftrightarrow x_u \wedge x_w$ is equivalent to
  $(\neg x_v \vee x_u) \wedge (\neg x_v \vee x_w) \wedge (x_v \vee \neg x_u \vee \neg x_w)$

Reduction: We trasform a circuit C to formula $\phi(C)$

- Transform circuit C to conjunction of clauses as in previous slide.
- A multi-input AND is replaced by a cascade of 2-input AND with extra variables (Also OR.)
- Each clause gives rise to a set of $3CNF$ clauses.
  Clause $x_v \leftrightarrow \neg x_u$ gives rise to equivalent set of clauses $(x_v \vee x_u) \wedge (\neg x_v \vee \neg x_u)$.
- Clause $x_v \leftrightarrow x_u \wedge x_w$ is equivalent to $(\neg x_v \vee x_u) \wedge (\neg x_v \vee x_w) \wedge (x_v \vee \neg x_u \vee \neg x_w)$
- Clause $x_v \leftrightarrow (x_u \vee x_w)$ is equivalent to $(x_v \vee \neg x_u) \wedge (x_v \vee \neg x_w) \wedge (\neg x_v \vee x_u \vee x_w)$

Reduction: We trasform a circuit C to formula $\phi(C)$

- Transform circuit C to conjunction of clauses as in previous slide.
- A multi-input AND is replaced by a cascade of 2-input AND with extra variables (Also OR.)
- Each clause gives rise to a set of $3CNF$ clauses.
  Clause $x_v \leftrightarrow \neg x_u$ gives rise to equivalent set of clauses
  $(x_v \vee x_u) \wedge (\neg x_v \vee \neg x_u)$.
- Clause $x_v \leftrightarrow x_u \wedge x_w$ is equivalent to
  $(\neg x_v \vee x_u) \wedge (\neg x_v \vee x_w) \wedge (x_v \vee \neg x_u \vee \neg x_w)$
- Clause $x_v \leftrightarrow (x_u \vee x_w)$ is equivalent to
  $(x_v \vee \neg x_u) \wedge (x_v \vee \neg x_w) \wedge (\neg x_v \vee x_u \vee x_w)$
- Each two literal clause is converted to a set of 3-literal clauses.
  E.g. $(x \vee y)$ is equivalent to $(x \vee y \vee p) \wedge (x \vee y \vee \neg p)$.

Reduction: We trasform a circuit C to formula $\phi(C)$

- Transform circuit C to conjunction of clauses as in previous slide.
- A multi-input AND is replaced by a cascade of 2-input AND with extra variables (Also OR.)
- Each clause gives rise to a set of $3CNF$ clauses.
  Clause $x_v \leftrightarrow \neg x_u$ gives rise to equivalent set of clauses
  $(x_v \vee x_u) \wedge (\neg x_v \vee \neg x_u)$.
- Clause $x_v \leftrightarrow x_u \wedge x_w$ is equivalent to
  $(\neg x_v \vee x_u) \wedge (\neg x_v \vee x_w) \wedge (x_v \vee \neg x_u \vee \neg x_w)$
- Clause $x_v \leftrightarrow (x_u \vee x_w)$ is equivalent to
  $(x_v \vee \neg x_u) \wedge (x_v \vee \neg x_w) \wedge (\neg x_v \vee x_u \vee x_w)$
- Each two literal clause is converted to a set of 3-literal clauses.
  E.g. $(x \vee y)$ is equivalent to $(x \vee y \vee p) \wedge (x \vee y \vee \neg p)$.

Theorem

C is satisfiable iff $\phi(C)$ is satisfiable. Also $|\phi(C)|$ is linear in $|C|$. Hence, $CIRCUIT\_SAT \leq_P 3CNF\_SAT$.

CLIQUE

Given a graph G a subset $V_0 \subseteq V$ is a clique if for every distinct $u, v \in V_0$ we have $(u, v) \in E$.

$$CLIQUE = \{\langle G, k \rangle \mid G \text{ has a clique of size } k\}$$

CLIQUE

Given a graph G a subset $V_0 \subseteq V$ is a clique if for every distinct $u, v \in V_0$ we have $(u, v) \in E$.

$$CLIQUE = \{\langle G, k \rangle \mid G \text{ has a clique of size } k\}$$

Theorem

$CLIQUE \in \mathbb{NPC}$

CLIQUE

Given a graph G a subset $V_0 \subseteq V$ is a clique if for every distinct $u, v \in V_0$ we have $(u, v) \in E$.

$$CLIQUE = \{\langle G, k \rangle \mid G \text{ has a clique of size } k\}$$

Theorem

$CLIQUE \in \mathbb{NPC}$

- $CLIQUE \in \mathbb{NP}$ (How?)

CLIQUE

Given a graph G a subset $V_0 \subseteq V$ is a clique if for every distinct $u, v \in V_0$ we have $(u, v) \in E$.

$$CLIQUE = \{\langle G, k\rangle \mid G \text{ has a clique of size } k\}$$

Theorem

$CLIQUE \in \mathbb{NPC}$

- $CLIQUE \in \mathbb{NP}$ (How?)
- $3CNF\_SAT \leq_P CLIQUE$.

Reduction $3CNF\_SAT \leq_P CLIQUE$

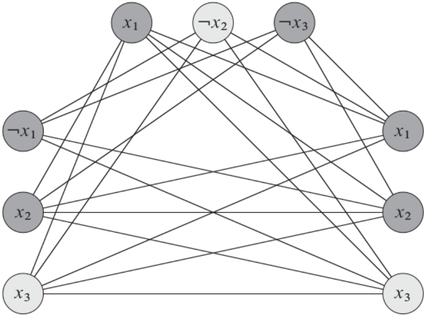
$$\phi = (x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee x_3)$$

Reduction $3CNF\_SAT \leq_P CLIQUE$

$$\phi = (x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee x_3)$$

$C_1 = x_1 \vee \neg x_2 \vee \neg x_3$

$C_2 = \neg x_1 \vee x_2 \vee x_3$



$C_3 = x_1 \vee x_2 \vee x_3$

Figure 34.14 The graph G derived from the 3-CNF formula $\phi = C_1 \wedge C_2 \wedge C_3$, where $C_1 = (x_1 \vee \neg x_2 \vee \neg x_3)$, $C_2 = (\neg x_1 \vee x_2 \vee x_3)$, and $C_3 = (x_1 \vee x_2 \vee x_3)$, in reducing 3-CNF-SAT to CLIQUE. A satisfying assignment of the formula has $x_2 = 0$, $x_3 = 1$, and x_1 either 0 or 1. This assignment satisfies C_1 with $\neg x_2$, and it satisfies C_2 and C_3 with x_3, corresponding to the clique with lightly shaded vertices.

Given a graph G a subset $V_0 \subseteq V$ is

- clique iff for every distinct $u, v \in V_0$ we have $(u, v) \in E$.

$$CLIQUE = \{\langle G, k \rangle \ \mid \ G \text{ has a clique of size } k\}$$

Clique, Independant Set, Vertex Cover

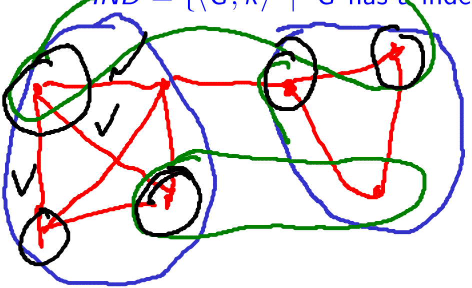
Given a graph G a subset $V_0 \subseteq V$ is

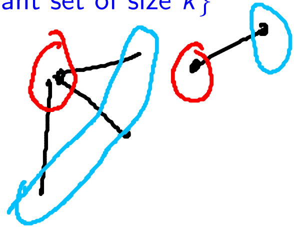
- clique iff for every distinct $u, v \in V_0$ we have $(u, v) \in E$.

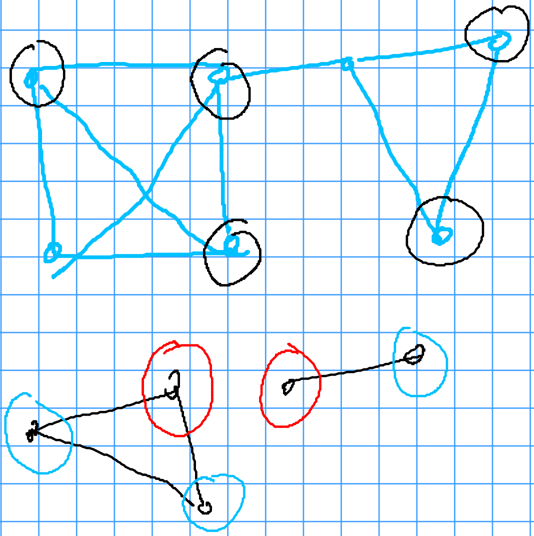
  $$CLIQUE = \{\langle G, k\rangle \mid G \text{ has a clique of size } k\}$$

- Independant Set iff for every $u, v \in V_0$ we have $(u, v) \notin E$.

  $$IND = \{\langle G, k\rangle \mid G \text{ has a independant set of size } k\}$$







Given a graph G a subset $V_0 \subseteq V$ is

- clique iff for every distinct $u, v \in V_0$ we have $(u, v) \in E$.
  $CLIQUE = \{\langle G, k\rangle \mid G \text{ has a clique of size } k\}$
- Independant Set iff for every $u, v \in V_0$ we have $(u, v) \notin E$.
  $IND = \{\langle G, k\rangle \mid G \text{ has a independant set of size } k\}$
- vertex cover iff for every edge $(u, v) \in E$ we have $(u \in V_0 \vee v \in V_0)$.
  $VERTEX\_COVER =$
  $\{\langle G, k\rangle \mid G \text{ has a vertex cover of size } k\}$

Clique, Independant Set, Vertex Cover

Given a graph G a subset $V_0 \subseteq V$ is

- clique iff for every distinct $u, v \in V_0$ we have $(u, v) \in E$.
  $$CLIQUE = \{\langle G, k\rangle \mid G \text{ has a clique of size } k\}$$
- Independant Set iff for every $u, v \in V_0$ we have $(u, v) \notin E$.
  $$IND = \{\langle G, k\rangle \mid G \text{ has a independant set of size } k\}$$
- vertex cover iff for every edge $(u, v) \in E$ we have $(u \in V_0 \vee v \in V_0)$.
  $$VERTEX\_COVER = \{\langle G, k\rangle \mid G \text{ has a vertex cover of size } k\}$$

We explore reductions between these decision problems.

Independant set

Theorem

Let $G = (V, E)$ be a given graph and let the complement graph $G' = (V, \overline{E})$ where $\overline{E} = V^2 - E$. Then, For any $V_0 \subseteq V$, we have V_0 is a clique in G iff V_0 is an independant set in G'.

Independant set

Theorem

Let $G = (V, E)$ be a given graph and let the complement graph $G' = (V, \overline{E})$ where $\overline{E} = V^2 - E$. Then, For any $V_0 \subseteq V$, we have V_0 is a clique in G iff V_0 is an independant set in G'.

- G has clique of size at least k iff G' has independant set of size at least k.

Independant set

Theorem

Let $G = (V, E)$ be a given graph and let the complement graph $G' = (V, \overline{E})$ where $\overline{E} = V^2 - E$. Then, For any $V_0 \subseteq V$, we have V_0 is a clique in G iff V_0 is an independant set in G'.

- G has clique of size at least k iff G' has independant set of size at least k.
- Hence, $CLIQUE \leq_P IND$ and $IND \leq_P CLIQUE$.

Independant set

Theorem

Let $G = (V, E)$ be a given graph and let the complement graph $G' = (V, \overline{E})$ where $\overline{E} = V^2 - E$. Then, For any $V_0 \subseteq V$, we have V_0 is a clique in G iff V_0 is an independant set in G'.

- G has clique of size at least k iff G' has independant set of size at least k.
- Hence, $CLIQUE \leq_P IND$ and $IND \leq_P CLIQUE$.
- $IND \in \mathbb{NPC}$.

Independant set

Theorem

Let $G = (V, E)$ be a given graph and let the complement graph $G' = (V, \overline{E})$ where $\overline{E} = V^2 - E$. Then, For any $V_0 \subseteq V$, we have V_0 is a clique in G iff V_0 is an independant set in G'.

- G has clique of size at least k iff G' has independant set of size at least k.
- Hence, $CLIQUE \leq_P IND$ and $IND \leq_P CLIQUE$.
- $IND \in \mathbb{NPC}$.
- SELF STUDY: There is a nice generalization of independant set in graph to a problem called set packing. (See KT 8.1)

Vertex Cover

Theorem

Let $G = (V, E)$ be a given graph. Then for any $V_0 \subseteq V$, we have V_0 is vertex cover iff $V - V_0$ is an independant set.

Vertex Cover

Theorem

Let $G = (V, E)$ be a given graph. Then for any $V_0 \subseteq V$, we have V_0 is vertex cover iff $V - V_0$ is an independant set.

- G has independant set of size k iff G has a vertex cover of set of size $|V| - k$.

Theorem

Let $G = (V, E)$ be a given graph. Then for any $V_0 \subseteq V$, we have V_0 is vertex cover iff $V - V_0$ is an independant set.

- G has independant set of size k iff G has a vertex cover of set of size $|V| - k$.
- Hence, $IND \leq_P VERTEX\_COVER$ and $VERTEX\_COVER \leq_P IND$.

Vertex Cover

Theorem

Let $G = (V, E)$ be a given graph. Then for any $V_0 \subseteq V$, we have V_0 is vertex cover iff $V - V_0$ is an independant set.

- G has independant set of size k iff G has a vertex cover of set of size $|V| - k$.
- Hence, $IND \leq_P VERTEX\_COVER$ and $VERTEX\_COVER \leq_P IND$.
- $VERTEX\_COVER \in \mathbb{NPC}$.

Theorem

Let $G = (V, E)$ be a given graph. Then for any $V_0 \subseteq V$, we have V_0 is vertex cover iff $V - V_0$ is an independant set.

- G has independant set of size k iff G has a vertex cover of set of size $|V| - k$.
- Hence, $IND \leq_P VERTEX\_COVER$ and $VERTEX\_COVER \leq_P IND$.
- $VERTEX\_COVER \in \mathbb{NPC}$.
- SELF STUDY: There is a nice generalization of vertex cover in graph to a problem called set cover. (See KT 8.1)

Graph Coloring

Given undirected graph $G = (V, E)$ has a k coloring $f : V \to \{1, 2, \ldots, k\}$ if $(u, v) \in E \Rightarrow f(u) \neq f(v)$.

$$GRAPH\_COLORING = \{\langle G, k\rangle \mid G \text{ has a } k \text{ coloring size}\}$$

Given undirected graph $G = (V, E)$ has a k coloring $f : V \rightarrow \{1, 2, \ldots, k\}$ if $(u, v) \in E \Rightarrow f(u) \neq f(v)$.

$$GRAPH\_COLORING = \{\langle G, k\rangle \mid G \text{ has a } k \text{ coloring size}\}$$

Theorem

Whether a graphs is 2-colorable is in $\mathbb{P}$.

Graph Coloring

Given undirected graph $G = (V, E)$ has a k coloring $f : V \to \{1, 2, \ldots, k\}$ if $(u, v) \in E \Rightarrow f(u) \neq f(v)$.

$$GRAPH\_COLORING = \{\langle G, k\rangle \mid G \text{ has a } k \text{ coloring size}\}$$

Theorem

Whether a graphs is 2-colorable is in $\mathbb{P}$.

Let $3COLOR$ be the set of all graphs having 3 coloring.

Given undirected graph $G = (V, E)$ has a k coloring $f : V \to \{1, 2, \ldots, k\}$ if $(u, v) \in E \Rightarrow f(u) \neq f(v)$.

$$GRAPH\_COLORING = \{\langle G, k\rangle \mid G \text{ has a } k \text{ coloring size}\}$$

Theorem

Whether a graphs is 2-colorable is in $\mathbb{P}$.

Let $3COLOR$ be the set of all graphs having 3 coloring.

Theorem

$3COLOR \in \mathbb{NPC}$.

Given undirected graph $G = (V, E)$ has a k coloring
$f : V \to \{1, 2, \ldots, k\}$ if $(u, v) \in E \Rightarrow f(u) \neq f(v)$.

$$GRAPH\_COLORING = \{\langle G, k \rangle \mid G \text{ has a } k \text{ coloring size}\}$$

Theorem

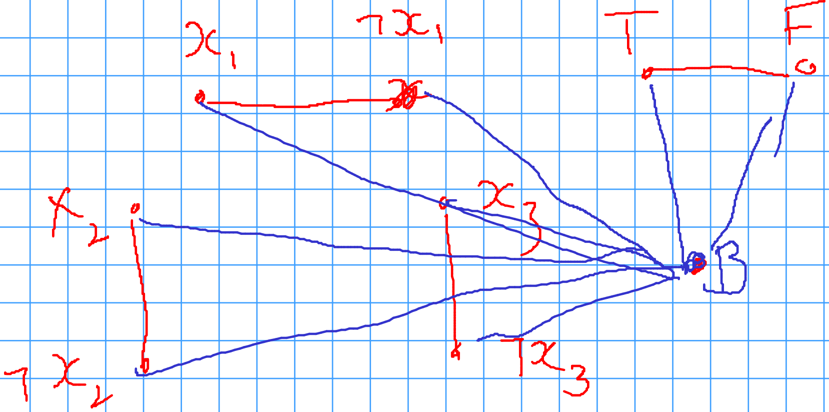
Whether a graphs is 2-colorable is in $\mathbb{P}$.

Let $3COLOR$ be the set of all graphs having 3 coloring.

Theorem

$3COLOR \in \mathbb{NPC}$.

- Whether a graph is 3-colorable is in $3COLOAR \in \mathbb{NP}$.



x_1
$\neg x_1$
T
F
x_2
x_3
B
$\neg x_2$
$\neg x_3$

Graph Coloring

Given undirected graph $G = (V, E)$ has a k coloring $f : V \rightarrow \{1, 2, \ldots, k\}$ if $(u, v) \in E \Rightarrow f(u) \neq f(v)$.

$$GRAPH\_COLORING = \{\langle G, k\rangle \mid G \text{ has a } k \text{ coloring size}\}$$

Theorem

Whether a graphs is 2-colorable is in $\mathbb{P}$.

Let $3COLOR$ be the set of all graphs having 3 coloring.

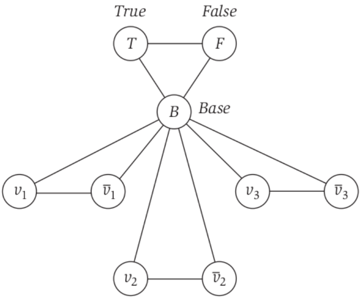
Theorem

$3COLOR \in \mathbb{NPC}$.

- Whether a graph is 3-colorable is in $3COLOAR \in \mathbb{NP}$.
- $3CNF\_SAT \leq_P 3COLOR$

Proof Idea

- For each variable x_i we have nodes v_i and $\overline{v}_i$.
- Encoding valuation by 3-coloring.



$(x_1 \vee x_2 \vee \neg x_3)$

Proof Idea (2)

Enforcing clause $(x_1 \vee \neg x_2 \vee x_3)$.

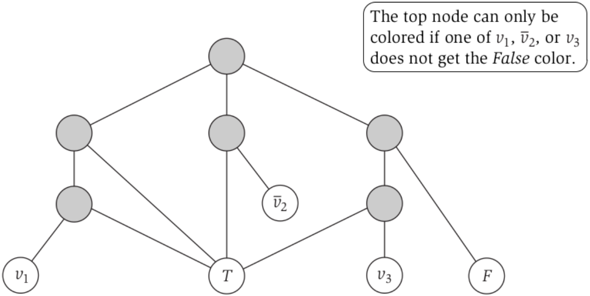
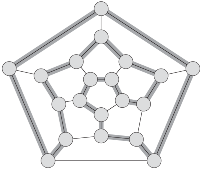


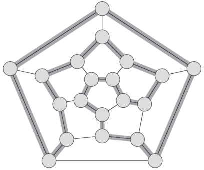
Figure 8.12 Attaching a subgraph to represent the clause $x_1 \vee \overline{x}_2 \vee x_3$.

Hamiltonian Cycle

Given an undirected graph $G = (V, E)$ does there exist a Hamiltonian cycle in G? A Hamiltonian cycle is a simple cycle where each vertex occurs exactly once.



Given an undirected graph $G = (V, E)$ does there exist a Hamiltonian cycle in G? A Hamiltonian cycle is a simple cycle where each vertex occurs exactly once.

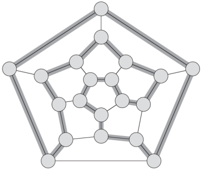


Theorem

$HAM\_CYCLE \in \mathbb{NPC}$

Hamiltonian Cycle

Given an undirected graph $G = (V, E)$ does there exist a Hamiltonian cycle in G? A Hamiltonian cycle is a simple cycle where each vertex occurs exactly once.



Theorem

$HAM\_CYCLE \in \mathbb{NPC}$

Proof that $VERTEX\_COVER \leq_P HAM\_CYCLE$ is in book (CLRS 34.5.3). Students may read it out of interest.

Travelling Salesman Problem

Problem *TSP* Given a complete directed graph $G = (V, V \times V)$ with non-negative integer weights $c : V \times V \rightarrow Z_0$ does there exist a Hamiltonian cycle (called tour) whose weight is at most k?

Problem *TSP* Given a complete directed graph $G = (V, V \times V)$ with non-negative integer weights $c : V \times V \rightarrow Z_0$ does there exist a Hamiltonian cycle (called tour) whose weight is at most k?

Theorem

$TSP \in \mathbb{NPC}$

Problem *TSP* Given a complete directed graph $G = (V, V \times V)$ with non-negative integer weights $c : V \times V \to Z_0$ does there exist a Hamiltonian cycle (called tour) whose weight is at most k?

Theorem

$TSP \in \mathbb{NPC}$

- $TSP \in \mathbb{NP}$

Travelling Salesman Problem

Problem *TSP* Given a complete directed graph $G = (V, V \times V)$ with non-negative integer weights $c : V \times V \to Z_0$ does there exist a Hamiltonian cycle (called tour) whose weight is at most k?

Theorem

$TSP \in \mathbb{NPC}$

- $TSP \in \mathbb{NP}$
- $HAM\_CYCLE \leq_P TSP$.

Show that $HAM\_CYCLE \leq_P TSP$.

Give an instance $G = (V, E)$ of Hamiltonian cycle problem construct an instance of TSP as $G' = (V, V \times V)$ with $c(u, v) = 0$ if $(u, v) \in E$ and $c(u, v) = 1$ otherwise. The aim is to find a tour of weight 0.

Show that $HAM\_CYCLE \leq_P TSP$.

Give an instance $G = (V, E)$ of Hamiltonian cycle problem construct an instance of TSP as $G' = (V, V \times V)$ with $c(u, v) = 0$ if $(u, v) \in E$ and $c(u, v) = 1$ otherwise. The aim is to find a tour of weight 0.

- Instnace $(G', c, 0)$ can be constructed in poly-time.

Show that $HAM\_CYCLE \leq_P TSP$.

Give an instance $G = (V, E)$ of Hamiltonian cycle problem construct an instance of TSP as $G' = (V, V \times V)$ with $c(u, v) = 0$ if $(u, v) \in E$ and $c(u, v) = 1$ otherwise. The aim is to find a tour of weight 0.

- Instnace $(G', c, 0)$ can be constructed in poly-time.
- G has a Hamiltonian cycle iff $(G', c, 0)$ has a tour of weight 0.

Subset Sum Problem

Problem Given a finite set of positive integers S and an integer $t > 0$ is there a subset $S' \subseteq S$ s.t. $(\Sigma_{i \in S'} \; i) \; = \; t$.

$S = \{1, 7, \overline{3}, 9, \overline{8}\}$

$t = 11$

Subset Sum Problem

Problem Given a finite set of positive integers S and an integer $t > 0$ is there a subset $S' \subseteq S$ s.t. $(\Sigma_{i \in S'} i) = t$.

Theorem

$SUBSET\_SUM \in \mathbb{NPC}$

Subset Sum Problem

Problem Given a finite set of positive integers S and an integer $t > 0$ is there a subset $S' \subseteq S$ s.t. $(\Sigma_{i \in S'} \ i) \ = \ t$.

Theorem

$SUBSET\_SUM \in \mathbb{NPC}$

- $SUBSET\_SUM \in \mathbb{NP}$

Subset Sum Problem

Problem Given a finite set of positive integers S and an integer $t > 0$ is there a subset $S' \subseteq S$ s.t. $(\Sigma_{i \in S'}\ i) = t$.

Theorem

$SUBSET\_SUM \in \mathbb{NPC}$

- $SUBSET\_SUM \in \mathbb{NP}$
- $3CNF\_SAT \leq_P SUBSET\_SUM$
  Proof in CLRS 34.5.5 (Only for interested).

Comments on $\mathbb{NP}$ Problems

Comments on $\mathbb{NP}$ Problems

- Not every instance of an $\mathbb{NPC}$ problem is necessarily hard. Heuristics can solve a large number of them.

Comments on $\mathbb{NP}$ Problems

- Not every instance of an $\mathbb{NPC}$ problem is necessarily hard. Heuristics can solve a large number of them.
- Approximation Algorithms.

Comments on $\mathbb{NP}$ Problems

- Not every instance of an $\mathbb{NPC}$ problem is necessarily hard. Heuristics can solve a large number of them.
- Approximation Algorithms.
- Randomized Algorithms.

Class $\mathbb{PSPACE}$

$\mathbb{PSPACE}$ is the class of problems which can be solved by an alogirthm using space polynomial in the size of input.

- $\mathbb{P} \subseteq \mathbb{PSPACE}$.

Class $\mathbb{PSPACE}$

$\mathbb{PSPACE}$ is the class of problems which can be solved by an alogirthm using space polynomial in the size of input.

- $\mathbb{P} \subseteq \mathbb{PSPACE}$.
- $3CNF\_SAT \in \mathbb{PSPACE}$.

Class $\mathbb{PSPACE}$

$\mathbb{PSPACE}$ is the class of problems which can be solved by an alogirthm using space polynomial in the size of input.

- $\mathbb{P} \subseteq \mathbb{PSPACE}$.
- $3CNF\_SAT \in \mathbb{PSPACE}$.
- $\mathbb{NP} \subseteq \mathbb{PSPACE}$.

Class $\mathbb{PSPACE}$

$\mathbb{PSPACE}$ is the class of problems which can be solved by an alogirthm using space polynomial in the size of input.

- $\mathbb{P} \subseteq \mathbb{PSPACE}$.
- $3CNF\_SAT \in \mathbb{PSPACE}$.
- $\mathbb{NP} \subseteq \mathbb{PSPACE}$.
- $Co\mathbb{NP} \subseteq \mathbb{PSPACE}$.

Class $\mathbb{PSPACE}$

$\mathbb{PSPACE}$ is the class of problems which can be solved by an alogirthm using space polynomial in the size of input.

- $\mathbb{P} \subseteq \mathbb{PSPACE}$.
- $3CNF\_SAT \in \mathbb{PSPACE}$.
- $\mathbb{NP} \subseteq \mathbb{PSPACE}$.
- $Co\mathbb{NP} \subseteq \mathbb{PSPACE}$.
- It is not known whether $classP = \mathbb{PSPACE}$?

Class $\mathbb{PSPACE}$

$\mathbb{PSPACE}$ is the class of problems which can be solved by an alogirthm using space polynomial in the size of input.

- $\mathbb{P} \subseteq \mathbb{PSPACE}$.
- $3CNF\_SAT \in \mathbb{PSPACE}$.
- $\mathbb{NP} \subseteq \mathbb{PSPACE}$.
- $Co\mathbb{NP} \subseteq \mathbb{PSPACE}$.
- It is not known whether $classP = \mathbb{PSPACE}$?
- $\mathbb{PSPACE}$-complete problems. Definition?

QSAT : Quantified 3CNF satisfiability

Propositional formula with two additional constructs $\exists x \phi(x, Y)$ and $\forall x \phi(x, Y)$.

QSAT : Quantified 3CNF satisfiability

Propositional formula with two additional constructs $\exists x \phi(x, Y)$ and $\forall x \phi(x, Y)$.

- $$\begin{aligned} \exists x.\phi(x, Y) &= \phi(0, Y) \vee \phi(1, Y) \\ \forall x.\phi(x, Y) &= \phi(0, Y) \wedge \phi(1, Y) \end{aligned}$$

QSAT : Quantified 3CNF satisfiability

Propositional formula with two additional constructs $\exists x \phi(x, Y)$ and $\forall x \phi(x, Y)$.

- $$\begin{aligned} \exists x.\phi(x, Y) &= \phi(0, Y) \vee \phi(1, Y) \\ \forall x.\phi(x, Y) &= \phi(0, Y) \wedge \phi(1, Y) \end{aligned}$$
- Example:
  $\phi(x_1, x_2, x_3) = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3)$

QSAT : Quantified 3CNF satisfiability

Propositional formula with two additional constructs $\exists x \phi(x, Y)$ and $\forall x \phi(x, Y)$.

- $$\begin{aligned} \exists x.\phi(x, Y) &= \phi(0, Y) \vee \phi(1, Y) \\ \forall x.\phi(x, Y) &= \phi(0, Y) \wedge \phi(1, Y) \end{aligned}$$
- Example:
  $\phi(x_1, x_2, x_3) = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3)$
- Satisfiability is asking: $\exists x_1 \exists x_2 \exists x_3.\ \phi(x_1, x_2, x_3)$.
  Compactly written as $\exists x_1, x_2, x_3.\ \phi(x_1, x_2, x_3)$.

QSAT : Quantified 3CNF satisfiability

Propositional formula with two additional constructs $\exists x \phi(x, Y)$ and $\forall x \phi(x, Y)$.

- $$\begin{aligned} \exists x.\phi(x, Y) &= \phi(0, Y) \vee \phi(1, Y) \\ \forall x.\phi(x, Y) &= \phi(0, Y) \wedge \phi(1, Y) \end{aligned}$$
- Example:
  $\phi(x_1, x_2, x_3) = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3)$
- Satisfiability is asking: $\exists x_1 \exists x_2 \exists x_3.\ \phi(x_1, x_2, x_3)$.
  Compactly written as $\exists x_1, x_2, x_3.\ \phi(x_1, x_2, x_3)$.
- Consider the QBF formula $\exists x_1 \forall x_2 \exists x_3.\ \phi(x_1, x_2, x_3)$.