ECE1762 Algorithm LEC17

Computability & Complexity

- \( TM = \) Algorithm = Program

- Problem \( \rightarrow \) 01001...

Translation
(Poly. Time)

\[ x \in L \rightarrow TM \xrightarrow{\text{Yes}} w \in L \quad \xrightarrow{\text{No}} w \notin L \]

- Optimization vs Decesion Problem

Example: Max Flow = what is Max Flow in a graph? /Optimization

\( \checkmark \)

Max Flow, = Does \( G \) have a Flow of size \( k \)?

\( \checkmark \)

Not Flow \( k \) and

Not \( G \)

\( E = \{ e_0, e_1, \ldots \} \)

A TM decides a problem if after a finite of \( t \) steps returns w/ answer yes or no.
A TM accepts a problem if when it returns \( w \) yes or no the answer is correct, but as long as machine runs you don't know whether it will terminate or not.

Some things to prove

1. If \( L \) is decidable, then \( \overline{L} \) is also decidable

\[
P \cap \text{co-NP} = P \text{ is closed under complementation}
\]

\[
\begin{array}{ccc}
\text{TM} & \xrightarrow{\text{yes}} & \overline{w} \\
\xrightarrow{\text{no}} & \overline{w} & \overline{w} \\
\end{array}
\]

2. \( \exists \) problem that are undecidable

Halting: Is there a TM that can decide whether given any machine \( M \) & input \( x \) \((M, x)\), \( M \) has H's on \( x \)?

Proof: Diagonalization
Complexity Class \( P \)

\[ P \equiv \{ x \in \{0,1\}^* : \exists \text{TM} \text{ that decides } x \text{ in } \text{poly-time} \} \]

**THM:** \( P = \{ x \in \{0,1\}^* : \exists \text{TM} \text{ to accept } x \text{ in } \text{poly } O(n^c) \text{ time} \} \)

Verification: Given an instance to a problem, and a candidate solution to problem, how "easy" is to verify if it is indeed a solution?

Algorithm \( A \) verifies language \( L \) off given instance \( x \) there exists a certificate (witness) \( y \) s.t. \( A(x,y) = 1 \) or 0 (candidate solution)

Complexity Class \( NP \)

\[ NP \equiv \{ x \in \{0,1\}^* : \exists \text{certificate } y \in O(|x|^c) \text{ and poly-time algorithm } A \text{ s.t. } A(x,y) = 1 \} \]

**THM:** \( P \subseteq NP \)
\( \text{co-NP} \equiv \text{if } L \in \text{NP} \text{ then } \overline{L} \in \text{co-NP} \)

\[ \exists L \in \{0,1\}^*: \forall y = O(1 \times 1) \in \text{poly } A(x,y) = \emptyset \]

**HAM-CYCLE (NP Class)**

Given a connected, undirected graph a simple cycle that traverses all vertices?

\[ \exists 0,1\}^* \]

\[ L \]

\[ \text{NP} \][\[ \text{co-NP} \]

\[ \text{integer factorization} \]

\[ \text{NP} = P \quad \text{NP} \cap \text{co-NP} = P \]

\[ \text{NP} = \text{co-NP} \]
Reducibility (informal)

Problem A can be reduced to problem \( B \) if there exists a translation algorithm that maps every instance of A into an instance of B, and there mapped instances back to instances of A.

\[
\begin{array}{c}
\text{A} \\
\downarrow \\
\text{B}
\end{array}
\quad f(a) \\
\downarrow \\
f(A) \\
\downarrow \\
f(a)
\]

Polynomial Time Reducibility

we say that \( L \) is poly-reducible to \( L' \) if \( f(x) \) s.t.

\[
x \in L \iff f(x) \in L'
\]

denoted as \( L \leq_p L' \)

THM: if \( L_1 \leq_p L_2 \) and \( L_2 \in \mathsf{P} \) then \( L_1 \in \mathsf{P} \)

\[
x \rightarrow f(x) \rightarrow TM_{L_2} \rightarrow f(x) \in L_2 \rightarrow f^{-1} \rightarrow \text{solution for } x
\]
Language $L \in$ NP-Complete (NPC) off

- $L \in$ NP (takes poly-time to verify)
- $\exists L' \in$ NP, $L' \leq_{p} L$ (NP-hard)

$\text{NP}$

$\text{NPC}$

$\text{P}$

**THM:** If $\exists L \in$ NPC & $L \in$ P $\Rightarrow P = \text{NP}$

**THM:** $\text{NP} = \text{co-NP}$ off $\exists L \in$ NPC st $\overline{L} \in$ NP

$\Rightarrow$ Easy

$\Rightarrow$ Let $L \in$ NPC st $\overline{L} \in$ NP

Peek any $L' \in$ NP, $L' \leq_{p} \overline{L} \Rightarrow \overline{L'} \leq_{p} \overline{L}$

$\Rightarrow \overline{L'} \in$ NP because $\overline{L} \in$ NP

$\in$ co-NP

**NP-Complete Methodology**

To prove that $L \in$ NPC, do the following
- Show that $L \in$ NP (usually easy to show)
- Peek any known $L' \in$ NPC and show $L' \leq_{p} L$
- anybody $\leq_{p} L'$ $\leq_{p} L$
Cook's Thm (1971) Circuit - SAT ∈ NPC

Given circuit w/ OCN AND/OR/NOT gates & single output. Find an input vector that makes output "1"

(A) Circuit - SAT ∈ NP
(B) If NP problem ≤p Circuit - SAT

Outline of Proof:

\[
\begin{align*}
\text{Circuit - SAT} & \quad \downarrow \\
\text{Formula - SAT} & \quad \downarrow \\
\text{3-SAT} & \\
\downarrow & \\
\text{CLIQUE} & \quad \text{HAM-CYCLE} \\
\downarrow & \quad \downarrow \\
\text{Vertex Cover} & \quad \text{travelling salesman}
\end{align*}
\]
**Formula-SAT:** Given a formula $\phi$ with Boolean variables $\land, \lor, \lnot, \iff, (, ),$ and $O(n)$ symbols all together, is there an assignment to the variables that make $\phi = 1$?

$$\phi = ((x \lor y) \rightarrow z) \iff (a \land b \land c \rightarrow d)$$

A) Show Formula SAT $\in$ NP

Verification in poly time verified

B) Circuit-SAT can reduce to Formula-SAT

Circuit-SAT $\leq_P$ Formula SAT

Good enough to show an example

$$\phi = x_9 \land ((x_1 \land x_2) \iff x_4) \land (x_5 \iff x_4) \land (x_6 \iff x_4) \land (x_6 \lor x_3) \iff x_7) \land (x_8 \iff \lnot x_5) \land (x_9 \iff (x_8 \lor x_7))$$
ECE1762 Algorithm LEC18

To prove language $L$ is NPC
- Show that $L \in$ NP (verifed in poly-time)
- Pick known NPC language $L'$ and show that $L' \leq_p L$ (NP-hard)

3-SAT
Given a set of clauses with three variables which is a conjunction of disjunctions. Find a satisfying assignment

$$\emptyset = (x_1 \lor x_2 \lor \overline{x}_3) \land (x_4 \lor \overline{x}_1 \lor x_5) \land (\overline{x}_2 \lor x_3 \lor \overline{x}_5) \land \ldots$$

2-SAT = poly time

CNF = Conjunctee Normal Form

Verification
QBF: PSPACE $\geq$ UP
Proof:

A) \( 3\text{-SAT} \in \text{NP} \) easy

B) Formula-SAT \( \leq_p \) 3-SAT

Given a Formula \( \phi \) I will create a 3-SAT instance:

\[
\phi = ((x_1 \rightarrow x_2) \lor \neg (\neg x_1 \leftrightarrow x_3) \lor x_4) \land \neg x_2
\]

1) Build a parse tree for formula

\[
\phi = y_1 \lor (y_2 \lor (y_3 \lor y_4))
\]

\[
\begin{align*}
(y_2 \lor (y_3 \lor y_4)) \lor \\
(y_3 \lor (x_1 \rightarrow x_2)) \lor \\
(y_4 \lor (y_5 \lor x_4)) \lor \\
(y_5 \lor \neg y_6) \lor \\
(y_6 \lor (\neg x_1 \leftrightarrow x_3))
\end{align*}
\]

\[
\text{replace with clauses}
\]
2) Build Characteristic Function $y_i$ and Find Maxterms

<table>
<thead>
<tr>
<th>$y_1$</th>
<th>$y_2$</th>
<th>$x_2$</th>
<th>$y_1 \leftrightarrow (y_2 \land \overline{x_2})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>$\emptyset$</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

$\overline{\phi}_{\text{PART}} = (\overline{y_1} \land y_2 \land \overline{x_2}) \lor (y_1 \land \overline{y_2} \land \overline{x_2}) \lor (y_1 \land y_2 \land x_2) \lor (y_1 \land \overline{y_2} \land x_2)$

$\overline{\phi}_{\text{PART}} = (y_1 \land \overline{y_2} \land x_2) \lor (\overline{y_1} \land y_2 \land x_2)$

$(x \land y) = (x \lor y \lor \overline{p}) \land (x \lor y \lor \overline{p})$

- You have to show reduction
- You have to show solution to 1 $\iff$ solution to 2
- You have to show reduction takes polynomial time
CLIQUE (Complete Graph)

Decesion Problem

Given a graph G does it have a clique of size k?

A) CLIQUE ∈ NPC (Easy)
B) 3-SAT ≤p CLIQUE

\[ \phi = (x_1 \lor \bar{x}_2 \lor \bar{x}_3)^\lor (\bar{x}_1 \lor x_2 \lor x_3)^\lor (x_1 \lor \bar{x}_2 \lor \bar{x}_3) \]

\( \phi \) has a solution \iff G has clique of size (\# clauses)

1) Introduce a vertex \( \theta \) literal & "group" according to clauses
2) Connect variables between different groups of not complementing each other
**Vertex Cover**: Given a graph, a VC is a set of vertices such that every graph edge is adjacent to at least one vertex from the cover.

Decision Problem: does $G$ have VC of size $k$?

A) $VC \in NP$ (Easy)

B) $CLIQUE \leq_p UC$

\[ G \quad \bar{G} \]

Claim: $G$ has a clique of size $k$ iff $\bar{G}$ a VC of size $n-k$

The reduction is poly-time.
Assume Ham Cycle for the following Problem

\[ \text{NP Complete} \]

Travelling Salesman Problem

Given a complete graph undirected w/ weights what's the lowest cost simple cycle?

Decision: Does G have a TSP of weight w?

A) TSP \( \in \text{NP(early)} \)

B) HAM CYCLE \( \leq_p \) TSP

1) Given G to find a Ham Cycle, introduce the remaining edges to make Gnew a complete graph

In \( E_{\text{new}} = \begin{cases} \text{weight} \; \phi \; \text{of edge existed in } G \\ \text{weight} \; 1 \; \text{of edge new introduced} \end{cases} \)

Does Gnew have TSP of weight \( \phi \)?