NP-completeness

11.1 P and NP

Review: definition

- \( P = \bigcup_{k \geq 1} \text{TIME}(n^k) \);
- \( NP = \bigcup_{k \geq 1} \text{NTIME}(n^k) \);
- \( NP \rightarrow \) languages whose YES instance can be verified in (deterministic) polynomial time.

Following are some examples of problem in \( NP \)

Theorem 11.1 \( \text{SUBSET-SUM} \in NP \)

Proof:

\( \text{SUBSET-SUM} = \{ <S, t> : S = \{x_1, x_2, x_3, \ldots, x_k \} \text{ where } \exists N \subseteq S, N = \{n_1, n_2, n_3, \ldots, n_j \} \text{ such that } \sum n_i = t \} \)

(Example: \( S = \{1, 17, 4, 8, 3, 9\} \), \( t = 22 \), then \( <S, t> \) is a YES instance of \( \text{SUBSET-SUM} \).)

Certificate: Given a set of elements in \( S \) \( \{y_1, y_2, \ldots, y_j\} \) such that \( \sum y_i = t \)
Verifier: Check each \( y_i \) is from \( S \)
- Check no duplicates
- Check \( \sum y_i = t \)

\( \Rightarrow \text{SUBSET-SUM} \) is in \( NP \).

Theorem 11.2 \( L \in P \rightarrow L \in NP \)

Proof: Suppose \( L \in P \)
Certificate: 
Verifier: Run the poly time decider for \( L \)
Theorem 11.3 \textit{SAT} \in NP

\textbf{Proof:}

\[
\text{SAT} = \{ < \phi > : \phi \text{ is satisfiable boolean formula} \}
\]

First define satisfiable boolean formula:

Boolean Formula: A formula with variables \(x_1, x_2, \ldots, x_n\) and their negation \(\bar{x}_i\), \(\lor\) and \(\land\)

(For example: \(\phi = (x_1 \lor \bar{x}_2) \land (x_3 \lor x_4) \land (x_2 \land \bar{x}_1)\))

A boolean formula is satisfiable: if \(\exists\) assignments that makes \(\phi\) true (evaluate to be 1).

(Taking the formula from above, assign \(x_1, x_2, x_3, x_4 = 1, \phi \rightarrow 1\), so \(\phi \in \text{SAT}\))

Certificate: Assignment to the variables
Verifier: Check if the assignment satisfies the formula.

\[\Rightarrow \text{SAT} \in NP\]

Theorem 11.4 \textit{3SAT} \in NP

\textbf{Proof:}

\[
\text{3SAT} = \{ < \phi > : \phi \text{ is a 3CNF that’s satisfiable} \}
\]

3CNF:
Definition: \(\phi = C_1 \land C_2 \land C_3 \cdots \land C_N\) where \(C_i = x_i \lor y_i \lor z_i\), \((x_i, y_i \text{ and } z_i \text{ are literals}(\rightarrow x_i \text{ and } \bar{x}_i))\)

(Example of 3CNF: \((x_2 \lor x_1 \lor \bar{x}_4) \land (x_2 \lor x_1 \lor \bar{x}_3) \land (x_1 \lor \bar{x}_2 \lor x_3))\)

From the fact that SAT \(\in NP\),

\[\Rightarrow \text{3SAT} \in NP\]

11.2 Reduction

Definition 11.5 computable
\(a\ function \ f : \Sigma^* \rightarrow \Sigma^* \ is \ computable \ if \ there\ is\ a\ poly-time\ TM\ on\ input\ w\ write\ f(w)\ on\ the\ tape\ then\ HALTS.\)

Definition 11.6 reduciable
\(a\ language\ A\ is\ poly\ time\ reducible\ to\ language\ B\ if\ \exists\ poly\ time\ computable\ function\ f : \Sigma^* \rightarrow \Sigma^* \ such\ that\ \forall w \in \Sigma^*, w \in A \iff f(w) \in B\)

Denoted by \(A \leq_P B\)

Lemma 11.7 If \(A \leq_P B\) and \(B\) has a poly-time algorithm then \(A\) has a poly-time algorithm

\textbf{Proof:} For any input \(w\), to show \(w \notin A\)
1. use $A \leq_P B$, map $w$ to $f(w) \in B$;

2. from the fact that $B \in P$, use the ploy time algorithm to compute $f(w)^2 \in B$ and use the result;

Then $A$ has a poly-time algorithm.

**Theorem 11.8** 3SAT $\leq_P$ CLIQUE

**Proof:**

Reminder:

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

TO show that 3SAT $\leq_P$ CLIQUE, we want to show $\phi \iff\ f(\phi) \iff\ f(\phi)$ where $f(\phi)$ is some instance $\langle G, k \rangle \phi = C_1 \land C_2 \cdots \land C_m$ where $C_i = x_i \lor y_i \lor z_i$

$\phi \rightarrow f(\phi)$:

Map $\phi$ to $f(\phi)$ by add all possible edges excepts the ones between $x_i$ and $\bar{x}_i$

Suppose $\phi$ is satisfiable, then each $C_j$ has at least 1 true literal. Pick one true literal $x_i$ from $C_i$.

For all $j \neq i$, there must be a $x_j$ from $C_j$ that’s true and since $x_i$ and $\bar{x}_i$ can not both be true, $x_i \neq \bar{x}_i$. So for each $x_i$ where $i \in [1 \ldots m]$, there’s a path between $x_i$ and $x_j$.

Thus we obtain a clique

**Example:**

P Let $\phi = (x_1 \lor x_2 \lor x_4) \land (\bar{x}_2 \lor x_3 \lor x_1) \land (x_4 \lor x_3 \lor x_1)$

Assign $x_2, x_3, x_4 = 1$, the map will look like

![Diagram of a graph showing a clique]

$f(\phi) \rightarrow \phi$:

Suppose $G$ has a k clique.

To obtain $\phi$, must choose one node for each $C_j$.

Decode each node to get a partial assignment then fill in the rest of the assignment arbitrarily.

Since there is no edge between $x_i$ and $\bar{x}_i$, they must not be in the clique together, thus there’s no inconsistency.

$\Rightarrow \phi$ is satisfied.

**11.3 COOK-LEVIN THEOREM**

**Definition 11.9** NP-complete

A language $A$ is NP-complete if:

1. $A \in NP$;
2. \( \forall \text{ problem } B \in NP, B \leq_P A \)

**Theorem 11.10 Cook-Levin Thm**
\( P = NP \iff SAT \in P \)

**Fact** If \( A \) is NP-complete and \( A \in P \), then \( P = NP \)
Suppose \( A \in P \), \( A \) is NP-complete, then \( \forall B \in NP, B \leq_P A \), thus \( B \in P \) then we have \( P = NP \)

**Theorem 11.11** If \( NP \neq P \), \( \exists L \text{ such that } L \in NP \land L \notin NP\text{-complete} \)

**Theorem 11.12 Cook-Levin Thm (restate)**
\( SAT \text{ is } NP\text{-complete} \)

We will prove this theorem next time.