NP-completeness

Sipser 7.4 (pages 271 – 283)

The classes P and NP

\[ P = \bigcup \text{TIME}(n^k) \]

\[ \text{NP} = \bigcup \text{NTIME}(n^k) \]
A famous NP problem

• CNF satisfiability (CNFSAT):
  Given a boolean formula $B$ in conjunctive normal form for (CNF), is there a truth assignment that satisfies $B$?

$$ (x_1 \lor x_2) \land (\overline{x_1} \lor \overline{x_2}) \land (x_1 \lor \overline{x_2}) $$

A graph theory NP problem

• CLIQUE:
  Given a graph $G = (V, E)$ and an integer $k$, does $G$ contain $C_k$ as a subgraph?
  – Is $<G,3> \in CLIQUE$?
A graph theory NP problem

• \text{CLIQUE}:
  Given a graph \( G = (V, E) \) and an integer \( k \), does \( G \) contain \( C_k \) as a subgraph?
  – Is \( <G,3> \in \text{CLIQUE} \)?

\[ \text{Graph diagram} \]

A graph theory NP problem

• \text{CLIQUE}:
  Given a graph \( G = (V, E) \) and an integer \( k \), does \( G \) contain \( C_k \) as a subgraph?
  – Is \( <G,4> \in \text{CLIQUE} \)?

\[ \text{Graph diagram} \]
A graph theory NP problem

- CLIQUE:
  Given a graph $G = (V, E)$ and an integer $k$, does $G$ contain $C_k$ as a subgraph?
  – Is $<G,4> \in \text{CLIQUE}$?

A graph theory NP problem

- CLIQUE:
  Given a graph $G = (V, E)$ and an integer $k$, does $G$ contain $C_k$ as a subgraph?
  – Is $<G,5> \in \text{CLIQUE}$?
CLIQUE ∈ NP

• **Verifier:**
  V = “On input \(<G,k,c>\):
  1. Test whether \(c\) is a set of \(k\) nodes of \(G\)
  2. Test whether \(G\) contains all edges connecting nodes in \(c\)
  3. If both pass, accept; otherwise, reject.”

• **NTM:**
  N = “On input \(<G,k>\):
  1. Nondeterministically select a subset \(c\) of \(k\) nodes of \(G\)
  2. Test whether \(G\) contains all edges connecting nodes in \(c\)
  3. If yes, accept; otherwise, reject.”

Which problem is harder?
Recall...

- Definition 5.17: A function $f: \Sigma^* \rightarrow \Sigma^*$ is a **computable function** if $\exists$ some Turing machine $M$, on every input $w$, halts with just $f(w)$ on its tape.

Polynomial time computable functions

- Definition 7.28: A function $f: \Sigma^* \rightarrow \Sigma^*$ is a **polynomial time computable function** if $\exists$ some polynomial time Turing machine $M$, on every input $w$, halts with just $f(w)$ on its tape.
Recall...

• Definition 5.20:

Language $A$ is **mapping reducible** to language $B$, written $A \leq_m B$, if there is a computable function $f: \Sigma^* \rightarrow \Sigma^*$, where for every $w$,

$$w \in A \iff f(w) \in B$$

Polynomial time mapping reducibility

• Definition 7.29:

Language $A$ is **polynomial time mapping reducible** to language $B$, written $A \leq_p B$, if there is a polynomial time computable function $f: \Sigma^* \rightarrow \Sigma^*$, where for every $w$,

$$w \in A \iff f(w) \in B$$
Intuitively, $A$ is no harder than $B$

- Theorem 7.31:
  If $A \leq_p B$ and $B \in P$, then $A \in P$.
- Proof:

  ![Diagram]

  - $w$ is input
  - $f(w)$ is output
  - Reduction algorithm $F$
  - Algorithm for deciding $A$
  - Algorithm for deciding $B$

CNF $\leq_p$ CLIQUE

- Given a boolean formula $B$ in CNF, we show how to construct a graph $G$ and an integer $k$ such that $G$ has a clique of size $k \iff B$ is satisfiable.
- Given $(x_1 \lor x_2) \land (\overline{x_1} \lor \overline{x_2}) \land (x_1 \lor \overline{x_2})$
  the construction would yield

  ![Graph]
NP’s hardest problems

• Definition 7.34:
  A language $B$ is **NP-complete** if
  1. $B \in \text{NP}$
  2. $A \leq_p B$, for all $A \in \text{NP}$

P=NP?

• Theorem 7.35: If $B$ is NP-complete and $B \in P$, then $P = NP$. 
Cook–Levin Theorem

- \textit{SAT} is NP-complete.
  (If $A \in \text{NP}$, then $A \leq_p \text{SAT}$.)

But that’s not the only one!

- \textit{CLIQUE} is NP-complete (why?)