How to Prove NP-Completeness

A problem \( B \) is NP-complete if
- (membership) \( B \in \text{NP} \)
- (NP-hard) For all \( A \in \text{NP} \), \( A \leq_p B \)

Theorems
- SAT is NP-complete.
- If \( B \) is NP-complete and \( B \leq_p C \) then \( C \) is NP-hard.
- SAT \( \leq_p \) 3SAT.
- 3SAT \( \leq_p \) CLIQUE.
- CLIQUE \( \leq_p \) INDEPENDENT-SET

22c:135 Theory of Computation

Hantao Zhang

http://www.cs.uiowa.edu/~hzhang/c135

The University of Iowa
Department of Computer Science

7.5 Additional NP-Complete Problems

VERTEX-COVER is NP-complete

- VERTX-COVER is in NP.
- VERTX-COVER is in NP-hard.
  - Proof1: Using graph properties.
  - Proof2: Reduce 3SAT to VERTX-COVER.

Other NP-complete Languages

To show that \( C \) is NP-complete, we provide a polynomial time reduction from 3SAT to \( C \).

\[ \text{VERTEX-COVER} = \{ (G, k) \mid G \text{ is an undirected graph that has a } k\text{-node vertex cover} \} \]
Proof 1: VERTEX-COVER is NP-hard

Given a graph $G = (V, E)$,

- $X \subseteq V$ is a vertex-cover if for each edge $(u, v) \in E$, $u \in X$ or $v \in X$.
- $I \subseteq V$ is an independent set if for each edge $(u, v) \in E$, $u \notin I$ or $v \notin I$.
- $X$ is a vertex-cover iff $V - X$ is an independent set of $G$.
- $X$ is a clique of $G$ iff $X$ is an independent set of $\overline{G}$, which is the complement of $G$.

NP-hardness proof:

- $\text{CLIQUE} \leq_p \text{INDEPENDENT-SET}$: $f(\langle G, k \rangle) = \langle \overline{G}, k \rangle$.
- $\text{INDEPENDENT-SET} \leq_p \text{VERTEX-COVER}$: $f(\langle G, k \rangle) = \langle G, |V| - k \rangle$.

Proof 2: VERTEX-COVER is NP-hard

Theorem 7.44 3SAT $\leq_P$ VERTEX-COVER.

- $f(\Phi) = \langle G, m + 2n \rangle$, where $\Phi$ is a set of $n$ clauses on $m$ variables.
- $G = (V, E)$, where $V = \{x_i, \overline{x}_i | 1 \leq i \leq m\} \cup \{l_{i,j} | 1 \leq i \leq 3, 1 \leq j \leq n\}$ and $E = \{(l_{1,j}, l_{2,j}), (l_{1,j}, l_{3,j}), (l_{2,j}, l_{3,j}) | 1 \leq j \leq l\} \cup \{(x_i, l_{j,k}) | 1 \leq i \leq m, 1 \leq j \leq 3, 1 \leq k \leq n\}$,
  the $j^{th}$ literal of clause $k$ is $x_i$.

NP Completeness Proofs

Theorem 7.46 HAM-PATH is NP-complete.

To show that $C$ is NP-complete, we can provide a polynomial time reduction from 3SAT to $C$.

Theorem 7.55 UHAM-PATH is NP-complete.

Other NP-complete Languages

Suppose $G$ is a directed graph:

- HAM-PATH = \{\langle G, s, t \rangle | G has a Hamiltonian path from $s$ to $t$ \}.
- HAM-PATH0 = \{\langle G \rangle | G has a Hamiltonian path \}.
- HAM-CYCLE = \{\langle G \rangle | G has a Hamiltonian cycle \}.

$G$ can be also an undirected graph (UHAM-PATH, UHAM-PATH0, UHAM-CYCLE).
**SUBSET-SUM is NP-complete**

Theorem 7.56 $3\text{SAT} \leq_P \text{SUBSET-SUM}$.  

$f(\Phi) = (X,t)$, where $\Phi$ is a set of $n$ clauses on $m$ variables, $X$ contains $2(m+n)$ numbers of up to $m+n$ digits, and $t$ is a $(m+n)$-digit number whose first $n$ digits are 1's and the rest $m$ digits are 4's.

**SUBSET-SUM**

$\text{SUBSET-SUM} = \{ \langle S, t \rangle \mid S = \{x_1, ..., x_k\}$ and for some $
\{y_1, ..., y_t\} \subseteq \{x_1, ..., x_k\}$ we have $\sum y_i = t \}$.  

**BIN-PACKING**

$\text{BIN-PACKING} = \{ \langle S, t \rangle \mid S = \{x_1, ..., x_k \mid 0 \leq x_i \leq 1\}$ and $S$ can be partitioned into $t$ subsets, $S = S_1 \cup S_2 \cup \cdots \cup S_t$ such that for each $S_i$, $\text{sum}(S_i) \leq 1 \}$.

Theorem BIN-PACKING is NP-complete.

**PARTITION**

$\text{PARTITION} = \{ \langle S \rangle \mid S = \{x_1, ..., x_n\}$ and there exists a subset $Y \subseteq S$ such that $\text{sum}(Y) = \text{sum}(S)/2$, $\}$, where $\text{sum}(X) = \sum_{x \in X} x$.

Theorem PARTITION is NP-complete.
KNAPSACK

\[ \text{KNAPSACK} = \{ \langle V, W, v, w \rangle \mid V = \{v_1, \ldots, v_n\}, W = \{w_1, \ldots, w_n\}, v_i, w_j, v, w \text{ are numbers, and there exists a subset } X \subseteq \{1, 2, \ldots, n\} \text{ such that } \sum_{i \in X} v_i \geq v \text{ and } \sum_{i \in X} w_i \leq w. \] 

Theorem KNAPSACK is NP-complete.