NP-Complete Reductions 3

Prof. Gregory Provan
Department of Computer Science
University College Cork
**Definition:** A language $B$ is NP-complete if:

1. $B \in \text{NP}$
2. Every $A$ in $\text{NP}$ is polynomial-time reducible to $B$ (i.e. $B$ is NP-hard)

If $B$ is NP-Complete and $P \neq \text{NP}$, then *There is no fast algorithm for $B$.*
A reduction by *restriction* shows that the source problem is a special case of the target problem.

For example, $3\text{SAT} \leq_p \text{CNF-SAT}$ because every satisfiable $3\text{CNF}$ is also a satisfiable $\text{CNF}$.

**Example.** Prove $3\text{SAT} \leq_p 4\text{SAT}$ by restriction.

A $3\text{CNF}$ can be converted to an equivalent $4\text{CNF}$ by repeating one literal in each clause.
COLORING

COLOR = \{ \langle G, k \rangle \mid G \text{ is } k\text{-colorable} \}

3COLOR = \{ G \mid G \text{ is 3-colorable} \}

Prove that 3COLOR \leq_p COLOR
HAMILTONIAN PATHS

HAMPATH = \{ \langle G,s,t \rangle \mid G \text{ has a hamiltonian path from } s \text{ to } t \}
HAM PATH BETWEEN 2 VERTICES (H2V)

- INSTANCE: Graph $G(V,E)$, vertices $u, v \in V$
- QUESTION: Does $G$ contain a HAMILTON PATH from $u$ to $v$?

- Prove that H2V is NP-complete
H2V is NP-complete

- **HAM-PATH** $\leq_p$ **H2V**
- **Proof:**
  - Restrict **HAM-PATH** such that the beginning node is $u$ and the end node is $v$
  - **HAM-PATH** is a special case of **H2V**
LONGEST PATH

- INSTANCE: Graph $G(V,E)$, positive integer $k \leq |V|$
- QUESTION: Does $G$ contain a simple path with $k$ or more edges.
LONGEST-PATH is NP-Complete

- Perform reduction from H2V
- Assume for LONGEST-PATH we have the case where $k = |V|$ where LONGEST-PATH is a special case of H2V where the path is Hamiltonian
VERTEX COVER

VERTEX-COVER = \{ \langle G, k \rangle \mid G \text{ has a vertex cover of size at most } k \}
INDSET = \{ \langle G, k \rangle \mid G \text{ has an independent set of size at least } k \} 

Prove that VERTEX-COVER \leq_p INDSET.
SUBSET SUM

SUBSET-SUM = \{ \langle y_1, \ldots, y_n, t \rangle \mid \exists S \subseteq \{1, \ldots, n\}. \sum_{j \in S} y_j = t \}\}

Which of the following are in SUBSET-SUM?

YES \langle 1, 3, 5, 7, 10 \rangle

NO \langle 19, 11, 27, 4, 13 \rangle

YES \langle 19, 11, 27, 4, 61 \rangle
KNAPSACK = \{ (w_1,v_1), \ldots, (w_n,v_n), W, V \mid \exists S \subseteq \{1 \ldots n\} \text{ so that } \sum_{i \in S} w_i \leq W \text{ and } \sum_{i \in S} v_i \geq V \}\}
Theorem. SUBSET-SUM \leq_p KNAPSACK

Proof.

A subset sum instance is a knapsack where the weights are equal to the values:

Let \( f(y_1, \ldots, y_n, t) = \langle (y_1, y_1) \ldots (y_n, y_n), t, t \rangle \).

Then \( \exists S. \sum_{i \in S} y_i = t \iff \exists S. \sum_{i \in S} y_i \geq t \) and \( \sum_{i \in S} y_i \leq t \), so \( \langle y_1 \ldots y_n, t \rangle \in \text{SUBSET-SUM} \iff f(y_1 \ldots y_n, t) \in \text{KNAPSACK} \)
SET-COVER = \{ \langle S_1, \ldots, S_n, k \rangle \mid \forall i, S_i \subseteq U \text{ and } \exists i[1\ldots k] \text{ so that } S_{i[1]} \cup S_{i[2]} \cup \ldots \cup S_{i[k]} = U \}\}

Which of the following are in SET-COVER?

- YES  \langle \{1\}, \{1,2\}, \{2\}, \{3\}, 2 \rangle
- NO  \langle \{1,4\}, \{1,2\}, \{1,3\}, \{4\}, 2 \rangle
- YES  \langle \{1\}, \{2\}, \{1,2\}, 2 \rangle

Theorem. VERTEX-COVER \leq_p SET-COVER

Proof. A vertex cover instance is just a set cover instance where every node is a set of edges.
A reduction from A to B by local replacement shows how to “translate” between “units” of A and “units” of B.

**Example.** vertex cover “units” are vertices and edges; set cover “units” are elements and sets.

**Example.** CIRCUIT-SAT units are gates, CNFSAT units are clauses, 3SAT units are 3-literal clauses.
GRADUATION

A transcript is a set of course numbers a student has taken

A major consists of:
  Pairs: exactly one of which must be taken
  Lists: at least one course of which must be taken

GRADUATION = \{〈T,M〉 | a subset of T satisfies M\}

For example:
M = [1901A,1901B], [1902A,1902B]
   (4011,4041A,4041B), (4211,4707), (4061)
GRADUATION 2 ∈ NP:
The subset is a proof that (T,M) 2 GRADUATION.

3SAT ≤ P GRADUATION:

\[(x_1 \lor x_2 \lor \neg x_3) \land (\neg x_1 \lor x_2 \lor x_2) \land (\neg x_2 \lor x_3 \lor x_1)\]

\[T = \{101, 102, 201, 202, 301, 302\}\]

\[M = [101, 102], [201, 202], [301, 302]\]

\[(101,201,302), (101,201,201), (101,202,301)\]
GRADUATION 2 NP:
The subset is a proof that $(T,M) \in \text{GRADUATION}$. 

3SAT $\leq_p$ GRADUATION:

Let $\phi = C_1 \land C_2 \land \ldots \land C_m$ have variables $x_1 \ldots x_k$.

For each $x_i$:
- add classes $i01$ and $i02$ to $T$.
- add pair $(i01,i02)$ to $M$.

For each $C_j$, we add a triple to $M$:
- if $x_i$ is a literal in $C_j$, the triple includes $i01$.
- if $\neg x_i$ is a literal in $C_j$, the triple include $i02$.
UHAMPATH

No HAM PATH

Undirected HAM PATH

UHAMPATH = \{ \langle G, s, t \rangle \mid G \text{ is an undirected graph with a Hamiltonian path from } s \text{ to } t \}\}
HAMPATH \leq_p UHAMPATH
HAMPATH $\leq_P$ UHAMPATH
HAMPATH $\leq_p$ UHAMPATH

\[ f(G,s,t) = (G',s',t') \] where:

For each node \( u \in G \):
- add nodes \( u_{in}, u_{out}, u_{mid} \) to \( G' \).
- add edges \( \{u_{in}, u_{mid}\} \) and \( \{u_{mid}, u_{out}\} \) to \( G' \).

For each edge \( (u,v) \in G \), add \( \{u_{out}, v_{in}\} \) to \( G' \).

\( s' = s_{in}, \ t' = t_{out} \).

If \( (G,s,t) \in \text{HAMPATH} \), then \( (G',s',t') \in \text{UHAMPATH} \):

\( (s,u,v,..,t) \neq (s_{in},s_{mid},s_{out},u_{in},u_{mid},u_{out},v_{in},v_{mid},v_{out},...,t_{out}) \)
HAMPATH \leq_p UHAMPATH

If \((G', s', t')\) 2 UHAMPATH, then \((G, s, t)\) 2 HAMPATH:

Let \((s_{in}=v_1, v_2, \ldots, v_{3n}=t_{out})\) be the undirected path.

Claim: for all \(i \geq 0\), there exists \(u \in G\) so that:

\[ v_{3i+1}=u_{in}, \quad v_{3i+2}=u_{mid}, \quad v_{3i+3}=u_{out} \]

\(i=0:\) If \(v_2 \neq s_{mid}\), then no \(v_i=s_{mid}\). So \(v_2=s_{mid}, \quad v_3=s_{out}\)

Induction: if \(v_{3i} = u'_{out}\), then \(v_{3i+1}=u_{in}\), so \(v_{3i+2}=u_{mid}\).

The directed Hamiltonian path is \((u_1, u_2, \ldots, u_n)\)
We transform a 3-cnf formula $\phi$ into $(y_1 \ldots y_n, t)$:

$$\phi \in 3\text{SAT} \Leftrightarrow (y_1 \ldots y_n, t) \in \text{SUBSET-SUM}$$

The transformation can be done in time polynomial in the length of $\phi$. 

**3SAT $\leq_p$ SUBSET-SUM**
Each variable and each clause result in two $y_i$'s. Each $y_i$ will have a digit for each clause and variable.

\[(x_1 \lor x_2 \lor x_2) \land
(\neg x_1 \lor x_2 \lor x_2) \land
(x_1 \lor \neg x_2 \lor x_2)\]
\(3\text{SAT} \leq_p \text{SUBSET-SUM}\)

Let \(\phi = C_1 \lor C_2 \lor \ldots \lor C_m\) have \(k\) variables \(x_1 \ldots x_k\).

We output \(y_1 \ldots y_{2k+2m}\), each a \((k+m)\)-digit number.

for each \(1 \cdot j \cdot k:\)
  for each \(1 \cdot i \cdot m:\)
    the \(i^{\text{th}}\) digit of \(y_{2j-1}\) is 1 if \(x_j \in C_i\), else 0
    the \(i^{\text{th}}\) digit of \(y_{2j}\) is 1 if \(\neg x_j \in C_i\), else 0.
  digit \(j+m\) of \(y_{2j}\), \(y_{2j-1}\) is 1.

For each \(1 \cdot j \cdot m:\) \(y_{2k+2j} = y_{2k+2j-1} = 10^{j-1}\)

Output \(t = 11\ldots1133\ldots3\)