A catalog of \textbf{NP}-complete problems

Starting point: a special problem called 3SAT

\textit{for the time being, we shall assume that 3SAT is NP-complete}

We will show that a variety of problems are \textbf{NP}-complete, via sequences of reductions from 3SAT

Building the catalog:

To prove that a decision problem \( D \) is \textbf{NP}-complete:

Prove that \( D \in \textbf{NP} \) (usually trivial)

Prove that \( D \) is \textbf{NP}-hard by showing that \( D' \leq_P D \) for some known \textbf{NP}-complete problem \( D' \)
Satisfiability (SAT)

A Boolean formula $\phi$:

- variables $x_1, \ldots, x_n$
- constants 0, 1
- operators $\lor, \land, \neg$
- Parentheses

A satisfying assignment: an assignment to the variables $x_1, \ldots, x_n$ such that $\phi(x_1, \ldots, x_n) = 1$

$\phi$ is called satisfiable if it has a satisfying assignment

The SAT problem: given a Boolean formula $\phi$, is $\phi$ satisfiable?

Clearly, SAT $\in$ NP: a solution is a satisfying assignment, which is short and easy to verify
3SAT: a special case of SAT

Conjunctive Normal Form:

• a conjunction (\(\land\)) of clauses
• each clause is a disjunction (\(\lor\)) of literals
• each literal is a variable \(x\) or its complement \(\overline{x}\)

Examples:

\[x \land y, \quad \overline{x} \land (y \lor z), \quad (x \lor y \lor \overline{z}) \land (w \lor \overline{x} \lor z)\]

A special form: 3-CNF

• Each clause consists of 3 distinct literals

The 3SAT problem: given a 3-CNF formula \(\phi\), is \(\phi\) satisfiable?
Example:

\[ \phi = (x_1 \lor x_2 \lor x_3) \land (x_1 \lor \overline{x}_2 \lor x_3) \land (\overline{x}_1 \lor \overline{x}_2 \lor x_3) \]

\[ x_1 = 0, \ x_2 = 0, \ x_3 = 0 \] is not a satisfying assignment
\[ x_1 = 1, \ x_2 = 0, \ x_3 = 0 \] is a satisfying assignment

therefore, \( \phi \) is satisfiable
An \textbf{NP}-complete graph problem

Let $G$ be an undirected graph

\textbf{Def’n:} an \textit{independent set} in $G$ is a subset $S$ of vertices such that no two vertices in $S$ are connected by an edge in $G$

Examples:
An optimization problem: given an undirected graph $G$, find an independent set of maximum size

A scheduling application:

vertices represent tasks to be performed
an edge $(u, v)$ represents a “conflict”: we cannot perform both tasks $u$ and $v$
a maximum independent set schedules as many tasks as possible

A coding theory application:

vertices represent “code words”
an edge $(u, v)$ means that code word $u$ could get garbled and become code word $v$, or vice versa
a maximum independent set yields a code with as many code words as possible
We recast this as a decision problem

The IS problem: given \((G, k)\), where \(G\) is an undirected graph and \(k\) is a positive integer, does \(G\) have an independent set of size (at least) \(k\)?

Proof that IS is NP-complete

\textit{IS} \in \textbf{NP}: clear (the independent set itself is the solution, which is short and easy to verify)

Need to show IS is NP-hard

Reduction: 3SAT \leq_p IS

Let \(\phi\) be a 3-CNF formula:

\[
\phi = (a_1 \lor b_1 \lor c_1) \land \cdots \land (a_m \lor b_m \lor c_m)
\]

Goal: construct a graph \(G\) such that

\(\phi\) is satisfiable \iff \(G\) has an indep. set of size \(m\)
Proof (cont’d):

$G$ has a $3m$ vertices, one for each literal

There is an edge between two vertices if either

1. the corresponding literals belong to the same clause, or

2. the corresponding literals are *contradictory* (i.e., $x_i$ and $\overline{x}_i$)

Example:

$$\phi = (x_1 \lor x_2 \lor x_3) \land (x_1 \lor \overline{x}_2 \lor x_3) \land (\overline{x}_1 \lor \overline{x}_2 \lor x_3)$$
Proof (cont’d):

Need to show

\[ \phi \text{ is satisfiable } \iff G \text{ has an indep. set of size } m \]

\[ \implies : \]

suppose \( \phi \) is satisfiable
choose a satisfying assignment
for each clause, choose a vertex corresponding to a true literal
verify: no two vertices are connected
this gives an indep. set of size \( m \)
Proof (cont’d):

\[ \iff : \]

suppose \( G \) has an indep. set \( S \) of size \( m \)
by Rule 1, each triple can have at most one
element in \( S \)

so \( S \) has exactly one element from each triple

\begin{itemize}
  \item if this corresponds to a literal \( x_i \), set \( x_i := 1 \)
  \item if this corresponds to a literal \( \overline{x}_i \), set \( x_i := 0 \)
\end{itemize}

by Rule 2, we will never attempt to assign both 0 and 1 to a variable

so this yields a consistent assignment to the variables that satisfies \( \phi \)
Variations on IS

Let $G = (V, E)$ be an undirected graph.

An clique is a subset $S \subseteq V$ such that every pair of vertices in $S$ are connected by an edge in $E$.

A vertex cover is a subset $S \subseteq V$ such that every edge in $E$ has at least one endpoint in $S$.

Theorem. The following are equivalent:

1. $S$ is an independent set in $G$.
2. $S$ is a clique in $G^c := (V, E^c)$, where $E^c$ is the complement of $E$.
3. $V \setminus S$ is a vertex cover in $G$. 
Problems:

\textit{CLIQUE}: given \((G, k)\), does \(G\) have a clique set of size (at least) \(k\)?

\textit{VC}: given \((G, k)\), does \(G\) have a vertex cover of size (at most) \(k\)?

Reductions:

\(IS \leq_p CLIQUE: (G, k) \leftrightarrow (G^c, k)\)

\(IS \leq_p VC: (G, k) \leftrightarrow (G, |V| - k)\)
**1in3SAT**: a specialized version of 3SAT

Let $\phi$ be a 3-CNF formula

We say that $\phi$ is 1in3-satisfiable if there is an assignment that makes *exactly one* literal in each clause of $\phi$ true

The 1in3SAT problem: given a 3-CNF formula $\phi$, is $\phi$ 1in3-satisfiable?

Example:

$$\phi = (x_1 \lor x_2 \lor x_3) \land (x_1 \lor \overline{x}_2 \lor x_3) \land (\overline{x}_1 \lor \overline{x}_2 \lor x_3)$$

this is *not* 1in3-satisfiable . . . why?
Proof that 1in3SAT is \textbf{NP}-complete

Reduction: 3SAT \leq_{P} 1in3SAT

Let \( \phi \) be a 3-CNF formula:

\[
\phi = (a_1 \lor b_1 \lor c_1) \land \cdots \land (a_m \lor b_m \lor c_m)
\]

Goal: construct another 3-CNF formula \( \phi' \) such that

\( \phi \) is satisfiable \iff \( \phi' \) is 1in3-satisfiable \quad (\ast)

Map \( j \)th clause in \( \phi \) to four clauses in \( \phi' \):

\[
\begin{align*}
\overline{a}_j \lor y_{1j} \lor z_{1j} & \land \overline{b}_j \lor y_{2j} \lor z_{2j} \land \overline{c}_j \lor y_{3j} \lor z_{3j} \land \\
( y_{1j} \lor y_{2j} \lor y_{3j} ) &
\end{align*}
\]

where \( y_{1j}, z_{1j}, y_{2j}, z_{2j}, y_{3j}, z_{3j} \) are new variables

Verify (\ast): exercise
**NAE-3SAT**: yet another version of **3SAT**

Let \( \phi \) be a 3-CNF formula

We say that \( \phi \) is NAE ("not all equal") satisfiable if there is an assignment that makes *some but not all* literals in each clause true

The **NAE-3SAT** problem: given a 3-CNF formula \( \phi \), is \( \phi \) NAE-satisfiable?

Example:

\[
\phi = (x_1 \lor x_2 \lor x_3) \land (x_1 \lor \overline{x}_2 \lor x_3) \land (\overline{x}_1 \lor \overline{x}_2 \lor x_3)
\]

\( x_1 = 1, x_2 = 1, x_3 = 1 \) is not an NAE-satisfying assignment

\( x_1 = 1, x_2 = 0, x_3 = 0 \) is an NAE-satisfying assignment
We want to show that \( \text{NAE-3SAT} \) is \( \textbf{NP} \)-complete

We first prove \( \text{NAE-4SAT} \) is \( \textbf{NP} \)-complete

Reduction: \( 3\text{SAT} \leq_P \text{NAE-4SAT} \)

Let \( \phi \) be a 3-CNF formula:

\[
\phi = (a_1 \lor b_1 \lor c_1) \land \cdots \land (a_m \lor b_m \lor c_m)
\]

Goal: construct a 4CNF formula \( \phi' \) such that

\( \phi \) is satisfiable \( \iff \) \( \phi' \) is NAE-satisfiable \( (\ast) \)

Map \( j \)th clause in \( \phi \) to a clause in \( \phi' \):

\[
(a_j \lor b_j \lor c_j \lor z)
\]

where \( z \) is a new variable (shared among all clauses)

Verify \( (\ast) \): exercise
Proof that $\text{NAE-3SAT}$ is $\textbf{NP}$-complete

Reduction: $\text{NAE-4SAT} \leq_p \text{NAE-3SAT}$

Let $\phi$ be a 4-CNF formula:

$$\phi = (a_1 \lor b_1 \lor c_1 \lor d_1) \land \cdots \land (a_m \lor b_m \lor c_m \lor d_m)$$

Goal: construct a 3-CNF formula $\phi'$ such that

$$\phi \text{ is NAE-satisfiable } \iff \phi' \text{ is NAE-satisfiable } \ (\ast)$$

Map $j$th clause in $\phi$ to two clauses in $\phi'$:

$$(a_j \lor b_j \lor w_j) \land (c_j \lor d_j \lor \overline{w_j})$$

where $w_j$ is a new variable

Verify $(\ast)$: exercise
Graph Coloring

Let $G = (V, E)$ be an undirected graph

A $k$-coloring is an assignment to each vertex $v \in V$ a “color” $c(v) \in \{1, \ldots, k\}$ such that no two adjacent vertices have the same color.

Example: 3-coloring

Can it be 2-colored?
An application to compilers: *register allocation*

Suppose we want to compile a subroutine for a machine with $k$ available registers

Suppose there are $n$ temporary variables $t_1, \ldots, t_n$ used in the subroutine

Build register interference graph: one vertex $v_i$ for each temporary $t_i$, and an edge connects $v_i$ and $v_j$ iff $t_i$ and $t_j$ are ever “live” at the same time

A $k$-coloring lets us assign all the temporaries to registers, so that an assignment to one temporary never clobbers another
**3COLOR**: given an undirected graph $G$, is it 3-colorable?

Proof that $3COLOR$ is **NP**-complete:

Reduction: $NAE$-$3SAT \leq_P 3COLOR$

Let $\phi$ be a 3-CNF formula:

$$\phi = (a_1 \lor b_1 \lor c_1) \land \cdots \land (a_m \lor b_m \lor c_m)$$

Goal: construct a graph $G$ such that

$\phi$ is NAE-satisfiable $\iff$ $G$ is 3-colorable
Proof (cont’d):

Suppose there are $n$ variables in $\phi$
We define the “variable gadget”:

Looking forward: assume the colors are 0, 1, 2, and that $\nu$ gets colored 2
Then each pair $x_i/\overline{x}_i$ gets colored 0/1 or 1/0 — a valid assignment!

Each clause maps to a “clause gadget”:

Note: each node in clause gadget must get a different color

We draw an edge between corresponding nodes in the clause and variable gadgets
Example:

$$\phi = (x_1 \lor x_2 \lor x_3) \land (x_1 \lor \overline{x}_2 \lor x_3) \land (\overline{x}_1 \lor \overline{x}_2 \lor x_3)$$

The graph $G$:

3-coloring corresponding to $x_1 = 1, x_2 = 0, x_3 = 0$:
Proof (cont’d):

Need to show:

\[ \phi \text{ is NAE-satisfiable} \iff G \text{ is 3-colorable} \]

\[ \implies \text{: Suppose } \phi \text{ has an NAE-satisfying assignment:} \]

Assign \( \lor \) the color 2, and the variable gadget
nodes the colors 0/1 as in the assignment

Consider the gadget corresponding to
\( (a_j \lor b_j \lor c_j) \)

The given assignment makes two of these literals
equal to \( b \in \{0, 1\} \) and the third equal to \( \overline{b} \)

Assign the nodes for the first two literals the
colors 2 and \( \overline{b} \), and the third node the color \( b \)
Proof (cont’d):

⇐: Suppose $G$ has a 3-coloring

Assume $v$ is colored with 2

The colors of the variable gadget nodes determine an assignment

The connections between the variable and clause gadgets ensure that this is an NAE-satisfying assignment

Why? If it were not, some clause gadget would be connected up to three variable gadget nodes of the same color

This would imply the three nodes of the clause gadget were colored with the two remaining colors