Examples of problems in \( NP \)

\[
\text{SAT} = \{ \langle \varphi \rangle \mid \varphi(x_1, \ldots, x_n) \text{ is a satisfiable propositional formula} \}
\]

\[
\text{Ham Path} = \{ \langle G, s, t \rangle \mid G \text{ is a digraph with a Hamiltonian path from } s \text{ to } t \}
\]

\[
\text{Clique} = \{ \langle G, k \rangle \mid G \text{ is a graph with a clique of size } k \}
\]

\[
\text{Subset Sum} = \{ \langle a_1, a_2, \ldots, a_n, t \rangle \mid \exists S \subseteq \{1, \ldots, n\} \text{ s.t. } \sum_{i \in S} a_i = t \}
\]

Exercise: Check that all the problems above \( \in NP \).

Search-to-Decision Reductions

\( \text{SAT} \) : Given \( \varphi(x_1, \ldots, x_n) \), decide if \( \varphi \) is satisfiable.

\( \text{SAT-Search} \) : Given \( \varphi(x_1, \ldots, x_n) \), find a satisfying assignment, if it exists.

Claim: If \( \text{SAT} \in \mathbf{P} \), then there is a
Claim: If SAT is P, then there is a deterministic polytime algorithm for SAT-Search.

Proof: Given \( \varphi(x_1, \ldots, x_n) \),
\[
\text{if } \varphi \not\in \text{SAT} \text{ then return "No"}
\]
\[
\text{for } i = 1 \ldots n
\]
\[
a_i = 0
\]
\[
\text{if } \varphi(a_1, \ldots, a_i, x_{i+1}, \ldots, x_n) \not\in \text{SAT}
\]
\[
\text{then } a_i = 1
\]
\end{verbatim}
\]
\end{verbatim}
\]
\end{verbatim}
\]}
\[
\text{end for}
\]
\[
\text{return } (a_1, a_2, \ldots, a_n)
\]

Binary Search on \( \{0, 1\}^n \) using SAT-algorithm

Suppose \( \varphi(x_1, \ldots, x_n) \in \text{SAT} \)
Then \( \varphi(0, x_2, \ldots, x_n) \in \text{SAT} \) OR \( \varphi(1, x_2, \ldots, x_n) \in \text{SAT} \).

Set \( a_1 \in \{0, 1\} \) so that \( \varphi(a_1, x_2, \ldots, x_n) \in \text{SAT} \)
& recurse on \( x_2, \ldots, x_n \).

Return \( (a_1, a_2, \ldots, a_n) \) as a satisfying assignment.
Exercise: Design Search-to-Decision reductions for all the example NP problems above.

Caution: Sometimes no obvious “search-to-decision reduction is known.

Example:

Composite: Given \(<n>\) decide if \(n\) is composite (i.e., has a non-trivial factor).

Composite-Search: Given \(<n>\) find a non-trivial factor of \(n\), if it exists.

Fact: Composite \(\in\) P  

[Agrawal, Kayal, Saxena 2003]

But, Composite-Search \(\equiv\) Factoring is not known to be in PolyTime!
(Yet, Factoring is in Quantum PolyTime.)

Polytime Reductions

Reduction \(f: \Sigma^* \rightarrow \Sigma^*\)
from A to B

Yes

Yes
\[ \text{A} \rightarrow \text{B} \]

\text{\textit{d} is a polytime reduction from A to B}}

\text{if}

\begin{enumerate}
\item \text{d is computable in polytime, and}
\item \( \forall x, x \in A \Leftrightarrow \text{d}(x) \in B \).
\end{enumerate}

\text{(Same as the reductions in Computability Theory, except require "polytime-computable."})

\underline{Notation:} \quad A \leq_p B

\underline{Then:} \quad A \leq_p B \land B \in \mathbf{P} \Rightarrow A \in \mathbf{P}.

\underline{NP-Completeness}

A language B is \textit{NP-complete} if

\begin{enumerate}
\item \( B \in \mathbf{NP} \)
\item \( \forall A \in \mathbf{NP}, A \leq_p B \).
\end{enumerate}
A language $B$ is \textit{NP-hard} if \\
\forall A \in \text{NP} , \ A \leq_{p} B.

Thus, $B$ is \textit{NP-complete} if $B$ is \textit{NP-hard} AND $B \in \text{NP}$.

1. Each \textit{NP-complete} problem $B$ is a \underline{hardest} problem in \textit{NP} (every other \textit{NP} problem is reducible to $B$).

2. For every two \textit{NP-complete} problems $A, B$, we have \\
\begin{align*}
A \leq_{p} B \quad \text{and} \quad B \leq_{p} A \\
\Rightarrow \quad A \equiv_{p} B
\end{align*}
3. Let $A$ be any $NP$-complete problem. Then $A \in P \iff P = NP$.

4. \( \forall A, B \in NP, \) 
   
   \[
   A \text{ is } NP\text{-complete} \land A \leq_P B \quad \Downarrow \quad B \text{ is } NP\text{-complete}.
   \]

   (Exercise: Verify 3 & 4.)

"Trivial" $NP$-complete problem

\[
A^P_{NTM} = \{ <M, w, t> | \text{NTM } M \text{ accepts } w \text{ within } t \text{ steps} \}
\]

Claim: $A^P_{NTM}$ is $NP$-complete.

Proof: (1) $A^P_{NTM} \in NP$:
   Just simulate $M$ on $w$, non-determ.
Just simulate $M$ on $w$, non-deterministically, for $t$ steps.

(2) $\forall L \in NP$, show $L \leq_p A^p_{NTM}$:

$L$ has NTM $M$, deciding $L$, in time $n^c$ (some constant $c > 0$).

Need: a polytime-reduction $f$ s.t. $\forall x$, $x \in L \iff f(x) \in A^p_{NTM}$

$f(x) := \langle M, x, 1^{|x|c} \rangle$.

Natural NP-complete problems

Then (Cook-Levin Thm): SAT is NP-complete.

Proof: (1) SAT $\in NP$.

(2) $\forall L \in NP$ $L \leq_p SAT$. 