18 Oct 2021

NP-complete problems

Announcements
- PSET 3 to be released Fri., due in 2 weeks after that.
- Take-home midterm 2nd wk Nov.
- Final project after that.

The class NP consists of problems that involve computing a function \( F(x) : \{0,1\}^* \rightarrow \{0,1\}^* \) such that there exists a polynomial-time algorithm ("verifier") with two inputs \( V(x,y) \) where \( |y| \leq \text{poly}(|x|) \), s.t.

\[
F(x) = 1 \iff \exists y \in \{0,1\}^{\text{poly}(|x|)} \quad V(x,y) = 1.
\]

E.g., graph 3-colorability.

Input string \( x \) is the binary encoding of an undirected graph \( G \).

(E.g., adj matrix.)

\[
F(x) = \begin{cases} 
1 & \text{if } G \text{ has a proper 3-coloring} \\
0 & \text{if not}
\end{cases}
\]
"Proper 3-coloring" means $V(G) \overset{h}{\rightarrow} \{0,1,2\}$ where $h$ takes distinct values on the endpoints of every edge.

This belongs to NP because, e.g., $V(x,y)$ could take $x = \text{binary encoding of } G$ $y = \text{binary encoding of } h; V(G) \rightarrow \{0,1,2\}$ and $V$ checks the colors on endpoints of each edge, makes sure they're different from each other.

$F(x) = 1 \iff \exists y \in \{0,1\}^{\text{poly}(\|x\|)} \quad V(x,y) = 1$

$F(x) = \bigvee_{y \in \{0,1\}^{\text{poly}(\|x\|)}} V(x,y) \quad (\text{NP})$

$F(x) = \bigwedge_{y \in \{0,1\}^{\text{poly}(\|x\|)}} V(x,y) \quad (\text{coNP})$

$F(x) = \sum_{y \in \{0,1\}^{\text{poly}(\|x\|)}} V(x,y) \quad (#P)$

$F(x) = \bigoplus_{y \in \{0,1\}^{\text{poly}(\|x\|)}} V(x,y) \quad (\oplus P)$
Reductions.
A polynomial-time Karp reduction from \( A \) to \( B \) is a function \( R : \text{inputs of } A \to \text{inputs of } B \) such that:
\[
A(x) = B(R(x)) \quad \forall x.
\]
and \( R \) runs in time \( \text{poly}(\text{size}(x)) \).

Aside. From now on a problem denoted by a letter such as \( A \) is assumed to have:
\[
\begin{align*}
\mathcal{X}_A & = \text{inputs} \to A \\
\mathcal{Y}_A & = \{ \text{potential solutions/certificates/witnesses} \}
\end{align*}
\]
\( \forall x \in \mathcal{X}_A \) is finite subset \( Y(x) \subseteq \mathcal{Y}_A \).

size: \( \mathcal{X}_A \to \mathbb{N} \)
\(\text{Graph} \to \# \text{ vertices} \)

enc: \( \mathcal{X}_A \cup \mathcal{Y}_A \to \{0,1\}^k \)
\( |\text{enc}(x)| \leq p(\text{size}(x)) \quad \forall x \)
\( |\text{enc}(y)| \leq q(\text{size}(x)) \quad \forall x \forall y \in Y(x) \)

where \( p, q \) are polynomials.
\( V_A(x,y) \): runs in time \( \text{poly}(\text{size}(x)) \).

For NP problems

\[
F(x) = \bigvee_{y \in Y(x)} V_A(x,y)
\]

Again: reduction satisfies \( A(x) = B(R(x)) \).

Notation: \( A \leq_P B \).

Note: This is a reflex trans relation.

(\( P \)-reductions can be composed.)

We write \( A \equiv_P B \) when \( A \leq_P B \) and \( B \leq_P A \).

This is an equivalence relation.

If \( C \) is a class of problem (e.g., NP)

a problem \( B \) is \( C \)-hard (e.g., \( NP \)-hard)

if every \( A \in C \) satisfies \( A \leq_P B \).

We say \( B \) is \( C \)-complete if \( B \in C \)

and \( B \) is \( C \)-hard.

(The maximal equivalent class of problems in \( C \),

ordered by \( \leq_p \).)
Def. If \( x_1, \ldots, x_n \) are Boolean vars

- a literal is one of the \( 2n \) terms formed by \( x_1, \ldots, x_n \) and their negations \( \overline{x_1}, \ldots, \overline{x_n} \).

- a clause of width \( w \) is a term formed by combining \( w \) literals using the "or" \( (\lor) \) operation.

\[ x_1 \lor \overline{x_2} \lor x_3 \lor x_4 \]

- a CNF formula is a conjunction of clauses.

A \( k \)-CNF formula is one whose clauses have width \( \leq k \).

\[ (x_1 \lor \overline{x_2} \lor x_3) \land (x_3 \lor x_4 \lor \overline{x_5}) \]

is a \( 3 \)-CNF.

The \( k \)-SAT problem asks: given a \( k \)-CNF, is there a truth assignment that satisfies it?

**Theorem (Cook-Levin):** \( k \)-SAT is NP-Complete for all \( k \geq 3 \).
Theorem (Valiant): \#PERF MATCHINGS is \#P-complete, even for bipartite graphs.

Showing \( B \) is \( \text{NP} \)-complete

1. Show \( B \in \text{NP} \) by designing a verifier \( V_B \).
2. Pick some \( A \) known to be \( \text{NP} \)-complete.
3. Design a reduction \( R \) showing that \( A \leq_p B \).