18 Oct 2021 NP-complete problems

Announcements

- P.Set 3 to be released Fri., due in 2 weeks after that
- Take-home midterm 2nd wk Nov.
- Final proj't after that.

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

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

E.g. graph 3-colorability.

Input string x is the binary encoding of an undirected graph G. (E.g. adj matrix.)

$$F(x) = \begin{cases} 1 & \text{if } G \text{ has a proper 3-coloring} \\ \emptyset & \text{if not.} \end{cases}$$

"Proper 3-coloring" means $V(G) \xrightarrow{h} \{0,1,2\}$ s.t. h takes distinct values on the endpoints of every edge.

This belongs to NP because, e.g., $V(x,y)$ could take

x = binary encoding of G

y = binary encoding of $h: V(G) \to \{0,1,2\}$

and V checks the colors on endpts of each edge, makes sure they're different from each other.

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

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

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

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

$$F(x) = \bigoplus_{y \in \{0,1\}^{poly(|x|)}} V(x,y) \qquad (\oplus P)$$

Reductions.

A polynomial-time Karp reduction from A to B is a function $R: \{\text{inputs of } A\} \to \{\text{inputs of } B\}$ s.t.

$$A(x) = B\big(R(x)\big) \quad \forall x.$$

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:

$$\mathcal{X}_A = \{\text{inputs to } A\}$$

$$\mathcal{Y}_A = \{\text{potential solutions / certificates / witnesses}\}$$

$$\forall x \in \mathcal{X}_A \ \exists \text{ finite subset } \mathcal{Y}(x) \subseteq \mathcal{Y}_A$$

$$\text{size}: \mathcal{X}_A \to \mathbb{N}$$

E.g., Graph $\mapsto$ # vertices

$$\text{enc}: \mathcal{X}_A \cup \mathcal{Y}_A \to \{0,1\}^*$$

$$|\text{enc}(x)| \le p(\text{size}(x)) \quad \forall x$$

$$|\text{enc}(y)| \le q(\text{size}(x)) \quad \forall x \ \forall y \in \mathcal{Y}(x)$$

where p, q are polynomials

$V_A(x,y)$: runs in time poly(size(x)).

For NP problems

$$F(x) = \bigvee_{y \in \mathcal{Y}_A(x)} V_A(x,y)$$

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

Notation: $A \leq_p B$.

Note: This is a refl & trans relation.
(b/c reductions can be composed.)
We write $A \equiv_p B$ when $A \leq_p B$ & $B \leq_p A$.
This is an equivalence relation.

If $\mathcal{C}$ is a class of problem (e.g. NP)
a problem B is $\mathcal{C}$-hard (e.g. NP-hard)
if every $A \in \mathcal{C}$ satisfies $A \leq_p B$.
We say B is $\mathcal{C}$-complete if $B \in \mathcal{C}$
and B is $\mathcal{C}$-hard.
(The maximal equiv. class of problems in $\mathcal{C}$,
ordered by $\leq_p$.)

Def: If $x_1, \ldots, x_n$ are Boolean vars

- a literal is one 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" ($\vee$) operation

  e.g. $x_1 \vee \overline{x_2} \vee \overline{x_3} \vee x_4$
- a CNF formula is a conjunction of clauses.

  a k-CNF formula is one whose clauses have width $\le k$.

E.g.

$$(x_1 \vee \overline{x_2} \vee \overline{x_3}) \wedge (x_3 \vee x_4 \vee \overline{x_5})$$

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 $\forall k \ge 3$.

<u>Theorem</u> (Valiant): #PERF MATCHINGS is #P-complete, even for bipartite graphs.

Showing B is NP-Complete

1. Show $B \in NP$: design verifier V_B.
2. Pick some A known to be NP-Complete.
3. Design reduction R showing that $A \leq_p B$.