EECS 336: Lecture 13: Introduction to Algorithms

P vs. NP (cont.): Review

Reading: Chapter 8; guide to reductions

Last Time:

- NP $\leq_{\mathcal{P}}$ CIRCUIT-SAT $\leq_{\mathcal{P}}$ LE3-SAT $\leq_{\mathcal{P}}$ 3-SAT

Today:

- $\mathcal{NP}$ review
- counter examples
- requests?

$\mathcal{NP}$ hardness

"proof by contradition: solve hard problem Y with blackbox for X, so X must be hard"

One-call Reductions

1. forward instance construction: $y \implies x^y$
2. backward certificate construction: x^y is yes $\implies$ y is yes.
3. forward certificate construction: y is yes $\implies$ x^y is yes.

Conclusion: y is yes if and only if x^y is yes.

DRAW PICTURE

Compare:

- show
  - x^y is yes $\implies$ y is yes.
  - x^y is no $\implies$ y is no.
- show
  - x^y is yes $\implies$ y is yes.
  - y is yes $\implies$ x^y is yes.

Common Mistake: x^y is yes $\not\Longleftarrow$ y is yes.

Example: 3-SAT $\implies$ INDEP-SET

Part I: (erroneous)

Convert 3-SAT instance f to INDEP-SET instance $x^f = (V^f, E^f, \theta^f)$:

- Vertices $V^f = \{v_{jd} : j \in \{1, ..., m\}, d \in \{1, ..., 3\}\}$.
- Edges $E^f = \{(v_{jd}, v_{j'd'}) : l_{jd} = "z_i" \wedge l_{j'd'} = "\bar{z}_i"\}$
- Target independent set size $\theta^f = m$ (the number of clauses).

Part II: counter example

Issue: can choose multiple vertices corresponding to same clause.

Goal: simple and small counter example.

- $\mathbf{z} = (z_1, z_2, z_3)$
- $f(\mathbf{z}) = (z_1 \vee z_2 \vee z_3) \wedge (z_1 \vee z_2 \vee \bar{z}_3) \wedge (z_1 \vee \bar{z}_2 \vee z_3) \wedge (z_1 \vee \bar{z}_2 \vee \bar{z}_3) \wedge (\bar{z}_1 \vee z_2 \vee z_3) \wedge (\bar{z}_1 \vee \bar{z}_2 \vee z_3) \wedge (\bar{z}_1 \vee z_2 \vee \bar{z}_3) \wedge (\bar{z}_1 \vee \bar{z}_2 \vee \bar{z}_3)$

Note: need to show x^f is "yes" but f is "no"

Deciding is as hard as optimizing

Proof: (reduction via binary search)

- given
  - instance x of X
  - black-box $\mathcal{A}$ to solve X_d
- search(A, B) = find optimal value in $[A, B]$.
  - $D = (A + B)/2$
  - run $\mathcal{A}(x, D)$
  - if "yes," search(A, D)
  - if "no," search(D, B)

Finding solution is as hard as deciding

Example: 3-SAT

1. if f is satisfiable $\exists \mathbf{z}$ s.t. $f(\mathbf{z}) = T$
2. guess $z_n = T$
3. let $f'(z_1, ..., z_{n-1}) = f(z_1, ..., z_{n-1}, T)$
4. simply f' and convert from LE3-SAT to 3-SAT $\implies g$
5. if g is satisfiable, repeat (2) on f'
6. if f' is unsatisfiable, repeat (2) on $f''(z_1, ..., z_{n-1}) = f(z_1, ..., z_{n-1}, F)$ simplified.

Example: INDEP-SET