| | |
|---|---|
| **EECS 336: Introduction to Algorithms** | **Lecture 9** |
| **P vs. NP** | indep set, 3-sat, TSP |

**Reading:** 8.0-8.3
"guide to reductions"

**Last time:**

- max flow alg / ford-fulkerson
- duality: max flow = min cut

**Today:**

- reducitons (cont)
- tractibility and intractibility
- decision problems
- 3-SAT $\leq_{\mathcal{P}}$ INDEP-SET

## Summary of Reduction

**Def:** Y reduces to X in polynomial time (notation: $Y \leq_P X$ if any instance of $Y$ can be solved in a polynomial number of computational steps and a polynomial number of calls to black-box that solves instances of $X$.

**Note:** to prove correctness of general reduction, must show that correctness (e.g., optimality) of algorithm for $X$ implies correctness of algorithm for $Y$.

**Def:** one-call reduction maps instance of $Y$ to instance of $X$, solution of $Y$ to solution of $X$. (also called a Karp reduction)

**Note:** a one-call reduction gives two algorithms:

(a) from instance $y$ of $Y$, construct instance $x^y$ of $X$.

(b) from solution $OPT(x^y)$, construct solution to $y$ with value at least $OPT(x^y)$

**Note:** the proof of correctness of a one-call reduction gives one (additional) algorithm:

(c) from solution $OPT(y)$, construct solution to $x^y$ with value at least $OPT(y)$

**Theorem:** reduction from "(a) and (b)" is correct if (a), (b), and (c) are correct.

**Proof:**

- for instance $y$ of $Y$, let instance $x^y of X^Y$ be outcome of (a).
- (b) correct $\Rightarrow OPT(y) \geq OPT(x^y)$.
- (c) correct $\Rightarrow OPT(x^y) \geq OPT(y)$.

$\Rightarrow OPT(y) = OPT(x^y)$

$\Rightarrow$ output of reduction has value $OPT(y)$.

## Decision Problems

"problems with yes/no answer"

**Def:** A decision problem asks "does a feasible solution exist?"

**Example:** network flow in $(G, c, s, t)$ with value at least $k$.

**Example:** perfect matching in a bipartite graph $(A, B, E)$.

**Note:** objective value for decision problem is 1 for "yes" and 0 for "no".

**Note:** (b) and (c) only need to check "yes" instances.

**Theorem:** perfect matching reduces to network flow decision problem.

**Note:** Can convert optimization problem to decition problem

**Def:** the decision problem $X_d$ for optimization problem $X$ is has input $(x, \theta)$ = "does instance $x$ of $X$ have a feasible solution with value at most (or at least) $\theta$?"

## Tractability and Intractability

Consequences of $Y \leq_P X$:

1. if $X$ can be solved in polynomial time then so can $Y$.

   Example: $X$ = network-flow; $Y$ = bipartite matching.

2. if $Y$ cannot be solved in polynomial time then neither can $X$.

# Reductions for Intractability

"reduce known hard problem $Y$ to problem $X$ to show that $X$ is hard"

## Problem $Y$: 3-SAT

input: boolean formula $f(\mathbf{z}) = \bigwedge_j (l_{i1} \vee l_{i2} \vee l_{i3})$

- literal $l_{jk}$ is variable "$z_i$" or negation "$\bar{z}_i$"
- "and of ors"
- e.g., $f(\mathbf{z}) = (z_1 \vee \bar{z}_2 \vee x_3) \wedge (z_2 \vee \bar{z}_5 \vee z_6) \wedge \cdots$

output:

- "Yes" if assignment $\mathbf{z}$ with $f(\mathbf{z}) = T$ exists

  e.g., $\mathbf{z} = (T, T, F, T, F, \ldots)$

- "No" otherwise.

## Problem $X$: INDEP-SET

input: $G = (V, E)$, $k$

output: $S \subset V$

- satisfying $\forall v \in S,\ (u, v) \notin E$
- $|S| \geq \theta$

## Reduction

**Lemma:** 3-SAT $\leq_{\mathcal{P}}$ INDEP-SET

**Part I:** forward instance construction

convert 3-SAT instance $f$ into INDEP-SET instance $(G, \theta)$.

literal $j$ in clause $i$

- vertices $v_{ij}$ correspond to literals $l_{ij}$
- edges for:
  - clause (in triangle)

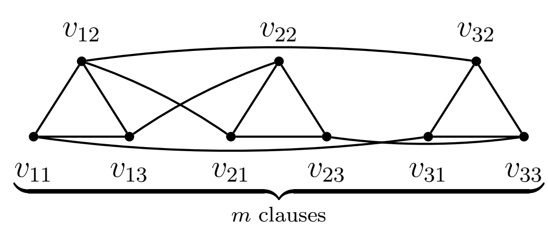
    "at most one vertex selected per clause"

  - conflicted literals.

    "vertices for conflicting literals cannot be selected"

- "vertex $v_{ij}$ is selected" $\Rightarrow$ "literal $l_{ij}$ is true".
- "indep set of size $m$ $\Leftrightarrow$ "satisfying assignment"

**Example:** $f(z_1, z_2, z_3, z_4) = (z_1 \vee z_2 \vee z_3) \wedge (\bar{z}_2 \vee \bar{z}_3 \vee \bar{z}_4) \wedge (\bar{z}_1 \vee \bar{z}_2 \vee z_4)$




Runtime Analysis: linear time (one vertex per literal).

**Part II:** reverse certificate construction

construct assignment $\mathbf{z}$ from $S$

(if $G$ has indep. set $S$ size $\geq m$ then $f$ is satisfiable.)

(a) For each $z_r$

- if exists nodes in $S$ are labeled by "$z_r$"

  $\Rightarrow$ set $z_r = 1$

- else

  $\Rightarrow$ set $z_r = 0$

**Note:** no two nodes $u, v \in S$ labeled by both $z_r$ or $\bar{z}_r$, if so, there is $(u, v)$ edge so $S$ would not be independent.

(b) $f(\mathbf{z}) = T$:

- $S$ has $|S| = m$

$\Rightarrow$ $S$ has one vertex per clause.

- for caluse $i$:

  - if $v_{ij} \in S$ is not negated, then $i$ is true.

  - if $v_{ij} \in S$ is negated, then $i$ is true.

**Part III:** forward certificate construction

construct independent set $S$ from $\mathbf{z}$

(if $f$ is satisfiable then $G$ has indep. set size $\geq m$.)

- let $S'$ be nodes in $G$ corresponding to true literals.

- if more than one node in $S'$ in same triangle drop all but one.

  $\Rightarrow$ $S$.

- $|S| = m$.

- for all $u, v \in S$,

  - $u$ & $v$ not in same triangle.

  - $l_u$ and $l_v$ both true

    $\Rightarrow$ must not conflict

    $\Rightarrow$ no $(l_u, l_v)$ edge in $G$.

  - so $S$ is independent.