Lecture #19
Search Problems, P and NP

- Last time: Reductions $A \rightarrow B$
  - $A \rightarrow B$ means can solve $A$ using subroutine for $B$
  - $B$ "easy" (poly-time) $\Rightarrow$ $A$ easy
  - $A$ "hard" (no poly-time alg known) $\Rightarrow$ $B$ hard
- Goal - try to classify problems as easy or hard
- Def: A Binary Relation
  - Def: decide($R$)
  - Def: search($R$)
Search Problem - Example

- Def: A Binary Relation is a subset $R \subseteq \{0,1\}^* \times \{0,1\}^*$ of pairs of finite bit strings, $(x, w) = (\text{instance, witness})$

- Def: $\text{decide}(R) =$ given instance $x$, decide if $\exists w$ such that $(x, w) \in R$ (output = yes/no)

- Def: $\text{search}(R) =$ given instance $x$, find a witness $w$ such that $(x, w) \in R$ if it exists, else "no"

- Ex: Max Flow
  - Instance:
  - Witness:
  - Decide $(R)$
  - Search $(R)$
Does \text{decide}(R) always exist?

• Focus on binary relations $R$ that are efficiently verifiable:

• New question: given $V_R$, how hard is $\text{decide}(R)$?
Defining $P$ and $NP$

- $P =$
- $NP =$
Defining $NP$-hard and $NP$-complete

- $P$ = "complexity class" of all relations $R$ such that $\text{decide}(R)$ costs $\text{poly}(1 \times 1)$ ($P"polynomial")$
- $NP = \text{all relations } R \text{ such that given } x, \exists w \text{ of size } |w| = \text{poly}(1 \times 1), \text{so } V_R(x, w) \text{ costs } \text{poly}(1 \times 1)$ when $R(x, w) = 1$ for some $w$
- $Ex: \text{ if } V_R(x, w) \text{ costs } \text{poly}(1 \times 1)$

- Def: problem $A$ is $NP$-hard if
- Def: problem $A$ is $NP$-complete if
CSAT is NP-complete

- Def: CSAT is binary relation $R_{\text{CSAT}}$ where 
  $\left(C = \text{circuit}, w\right) \in R_{\text{CSAT}}$ if $C(w) = 1$

- Claim CSAT is NP-complete

CSAT in NP:

CSAT NP-hard:
Reducing CSAT to simpler problems: SAT

- Recall what a circuit is: DAG of gates
- Convert circuit to CNF = conjunctive normal form = and of clauses like \((x_1 \lor \overline{x_2} \lor \overline{x_3})\)
- One variable per gate in DAG:

  - \( f \) or becomes
    \[
    \begin{array}{c}
    \text{OR} \\
    x \\
    \downarrow \\
    y \\
    \end{array}
    \]
  - \( f \) and becomes
    \[
    \begin{array}{c}
    \text{AND} \\
    x \\
    \downarrow \\
    y \\
    \end{array}
    \]
  - \( f \) NOT becomes
    \[
    \begin{array}{c}
    \text{NOT} \\
    x \\
    \downarrow \\
    x \\
    \end{array}
    \]
  - \( f \) becomes
    \[
    \begin{array}{c}
    1 \\
    \downarrow \\
    0 \\
    \end{array}
    \]
Reducing SAT to simpler case: 3SAT

• Want to show "simple" problems are NP-complete, to make them easier to use to show others are

• 3SAT: SAT with \( \leq 3 \) variables per clause

  • Example: \((x_1 \lor x_2 \lor \overline{x_3}) \land (\overline{x_1} \lor x_4 \lor x_5) \land (x_2 \lor x_6) \land \ldots\)

• Trick to convert \((a_1, v_{a_2}, v_{a_3}, \ldots, v_{a_k})\) to 3SAT

  • Introduce new variables \( y_{i_1}, y_{i_2}, \ldots, y_{k-3} \)

  • Convert to

    • If all \( a_i = F \), making above expression = T \( \Rightarrow \)

      • If \( a_i = T \)
More NP-complete problems

All of NP

CSAT

SAT

3SAT
Reducing 3SAT to Independent Set (IS)

- **IS**: Does graph $G$ have $\geq g$ unconnected vertices?

- **Ex**: $(\overline{x} \lor y \lor \overline{z}) \land (x \lor \overline{y} \lor v \lor z) \land (x \lor y \lor v \lor z) \land (\overline{x} \lor y \lor \overline{z})$

- Transform to graph where
  - each variable is
  - each clause is
    $\Rightarrow$
  - add edge between every
    $\Rightarrow$
  - Is there an IS of size

- Is expression satisfiable?
Reducing Independent Set (IS) to...

- Vertex Cover (VC): Subset $S \subseteq V$ that touch every edge
  - Fact:

- Clique (Cl): Subset $S \subseteq V$ that is fully connected
  - Fact:
Did I forget to prove anything?