Lecture #19

CS 170

Spring 2021

Search Problems, P and NP

- Last time: Reductions $A \to B$
  - $A \to B$ 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) = $ (instance, witness)
- Def: decide(R) = given instance x, decide if $\exists w$ such that $(x, w) \in R$ (output = yes/no)
- Def: 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 decide(R) always exist?

-

- Focus on binary relations R that are efficiently verifiable:

    -

- New question: given V_R, how hard is decide(R)?

    -

Defining P and NP

- P =

- NP =

-
    -

Defining NP-hard and NP-complete

- P = "complexity class" of all relations R such that decide(R) costs poly(|x|) (P = "polynomial")
- NP = all relations R such that given x, $\exists$ w of size |w| = poly(|x|), so $V_R(x, w)$ costs poly(|x|) when R(x, w) = 1 for some w
  - Ex: if $V_R(x, w)$ costs poly(|x|)
- Def: problem A is NP-hard if
- Def: problem A is NP-complete if

CSAT is NP-complete

- Def: CSAT is binary relation R_{CSAT} where

  $(C = \text{circuit}, w) \in R_{CSAT}$ if $C(w) = 1$

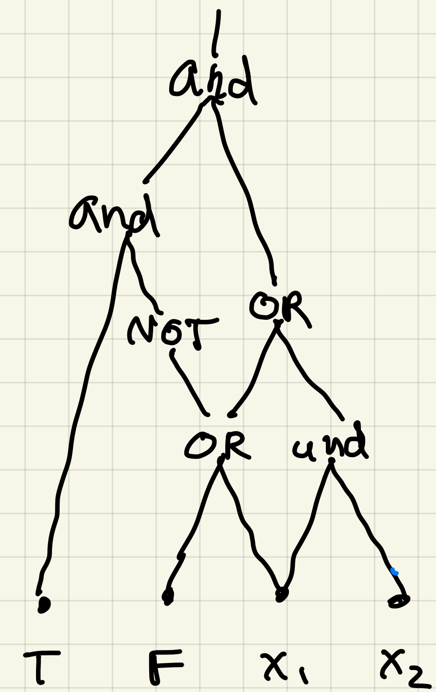
- Claim CSAT is NP-complete

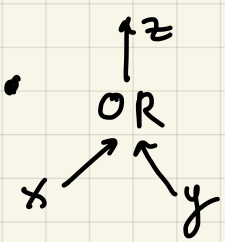
  CSAT in NP:

  CSAT NP-hard:

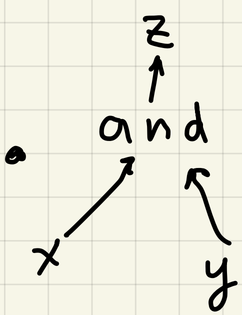
Reducing CSAT to simpler problems: SAT

- Recall what a circuit is: DAG of gates
- Convert circuit to CNF = conjunctive normal form = and of clauses like $(x_1 \vee \bar{x}_2 \vee x_3)$
  - One variable per gate in DAG:

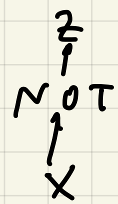


- 

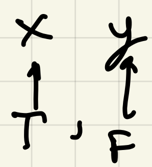
becomes

- 

becomes

- 

becomes

- 

become

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 ≤ 3 variables per clause
  - Ex: $(x_1 \vee x_2 \vee \overline{x_3}) \wedge (\overline{x}_1 \vee x_4 \vee \overline{x}_5) \wedge (x_2 \vee x_6) \wedge \cdots$
- Trick to convert $(a_1 \vee a_2 \vee a_3 \cdots \vee a_k)$ to 3SAT
  - Introduce new variables $y_1, \ldots, y_{k-3}$
  - Convert to

  - If all $a_i = F$, making above expression $= T \Rightarrow$

  - If $a_i = T$

More NP-complete problems

$$\text{All of NP} \to \text{CSAT} \to \text{SAT} \to \text{3SAT}$$

Reducing 3SAT to Independent Set (IS)

- IS: Does graph G have $\geq g$ unconnected vertices?
- Ex: $(\bar{x} \vee y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z) \wedge (x \vee y \vee z) \wedge (\bar{x} \vee \bar{y})$
- 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?