CS 579 Computational Complexity: Lecture 4

Project 1 due 01-31
Last name: non-determinism
NP
HAMPATH

Today: reducibility. completeness.
SAT

Q: anything you need for Project II
L is an instance of SAT

the HAMPATH if P = NP

When we reduce:

A reduces solving problem A to solving problem B

I view solving B as computational resource, ask how hard to solve A B

We do this in every day life, programming decisions

let A be a TM M with output is a deterministic TM M' distinguished state Q accepting

input string

for some

minimum k as before

- meaning random errors are made
- if ever much greater, halt
- output tape contents

the time of M on input x is the transition until halting

a function f: \Sigma^* \rightarrow \Sigma^* is poly time computable if some TM M

has \forall w \in \Sigma^* output f(w), in time O(t(lw)) some polynomial

s.t., A \in \Sigma^* is poly time mapping \( f \) reducible to \( B \in \Sigma^* \\
\text{defined } A \leq_P B \text{ if } f: \Sigma^* \rightarrow \Sigma^* \text{ poly time computable } \\
\forall w, w \in A \Leftrightarrow f(w) \in B
**Theorem:** \( f \) is not necessarily injective or surjective.

**Proof:** \( A \subseteq B \), \( B \cap P = A \cap P \)

**Proof:**

1. **Inspire** \( f: \Sigma^* \rightarrow \Sigma^* \)
2. Run \( M \) on \( f(x) \) and accept if \( M \) does.

**Closure:** \( x \in A \Rightarrow f(x) \in B \Rightarrow M_{acc} \Rightarrow N_{acc} \)

**Example:** \( L \) composed of a polynomial is polynomial.

**Example:** \( A \subseteq B \), \( A \cap P = B \cap P \)

**Proof:** \( A \subseteq B \), \( B \cap NP = A \cap NP \)

**Sketch:** \( x \in A \Rightarrow k(x) \subset B \) some branch \( \Rightarrow M_{acc} \) some branch \( N_{acc} \)

**Proof:** Language \( B \) is NP-complete if

- \( B \cap NP \) \( \subseteq \) \( P \cap NP \)
- \( B \cap NP \) \( \subseteq \) \( P \cap NP \)

**Corollary:** \( B \) NP-complete. \( B \cap P \) \( \Rightarrow \) \( P = NP \)

**Corollary:** \( B \) NP-complete. \( CNGP \), \( B \cap C \) \( \subseteq \) \( C \) NP-complete

**Proof:**

- Any \( A \cap NP \), \( A \subseteq \subseteq \subseteq \subseteq C \)

**Remark:** Theorem for proof.

**Definition:** A Boolean formula \( \varphi \) is an expression involving boolean variables \( 0,1 \) and AND, OR, NOT operators. 

\( \varphi = (x \land y) \lor (x \lor z) \)

\( \varphi \) is satisfiable if there exists an assignment to the variables 

**Proof:** \( SAT \cap NP \): \( \varphi \) is satisfiable Boolean formula.
Let a boolean formula \( \phi \) be in conjunctive normal form (CNF).

\[
\phi = \bigwedge_{i=1}^{n} \bigvee_{j} c_{ij}
\]

Each clause \( C_i \) is a disjunction of literals:

\[
C_i = \bigvee_{j} x_i^j
\]

\( \phi \) is a \( k \)-CNF formula if \( \phi \) is in CNF form by each clause having \( \leq k \) literals.

3SAT = \{ \phi \mid \phi \text{ is a satisfiable boolean formula in 3CNF} \} \subseteq \text{NP} \subseteq \text{co-NP} \text{ easily checkable}

By [Cook 71, Levin 73] there exists an \( \text{NP-complete} \) problem \( G \) not-trivial, but direct and brute force SAT is \( \text{NP-complete} \), then SAT is also \( \text{NP-complete} \).

**Question:**

Let \( G = (V, E) \) be an undirected graph with clique of size \( k \), is \( G \) \( \text{NP-complete} \)?

**Proof:**

\( \text{NP-complete} \) easy.

\( \text{NP-hard} \) will give reduction \( \text{SAT} \rightarrow \text{CONJ} \text{SAT} \).

\( \phi = (x_1 \lor x_2 \lor x_3) \land (x_4 \lor x_5 \lor x_6) \land \cdots \land (\cdots) \)

\[
\phi = \bigwedge_{k=1}^{n} \bigvee_{c_{ij}} c_{ij}
\]

\( k = \# \text{ clauses} \)

\( V = \{ \text{literals appearing in clauses, or duplicates} \} \Rightarrow |V| \leq 3k \)

\( E = \text{all edges except } - \text{ self loops } \)

- between literals in same clause
- between literals that are neighbors of each other

\[
\text{Ch. of \ # \ clauses \ # (G, k) \leq \text{Clique}}
\]

**Proof:**

\( \phi \text{ satisfiable} \iff \text{Ch. (G, k)} \leq \text{Clique} \)

\( \phi \text{ satisfiable} \iff \exists \psi \text{ such that } \phi = \psi \land \text{no literal is duplicated} \)

\[ S = \{ \text{sat. literals made true by } 0, 1 \text{ w/ duplicates} \}
\]

\[ |\text{SC} | \geq 1 \text{ all } i \]

**Pick** \( T \subseteq S \text{ w/ } |\text{SC} | = 1 \text{ all } i \)
Claim: $T \subseteq V$ is a $k$-clique.

Proof:
- $|T|=k \implies$ complete
  - $t \neq t' \in T \implies$ not in same class
  - not made one by $a = 1$ not adjacent at each edge.

Claim: $G, k \in \text{CLIQUE}$, want to show $\alpha(G, k) = k$.

Case 1: $S$ is $k$-clique.

Now $|S \cap C_i| \leq 1 \implies$ no edges between vertices in same class.

Define $\alpha(x) = \begin{cases} 1 & x \in S \\ 0 & x \notin S \end{cases}$

Claim: $\chi(G, k) = k$.

Proof:
- $\alpha$ is well defined
  - $\alpha$ makes all lines in $S$ one
  - $S$ has $k$ lines per class by all elements in $G$ of same class

Rank: $\text{BSAT} \leq_P \text{HAMPATH}$ in textbook & so $\text{BSAT}$ is NP-complete

next time: Cook Levin Theorem: $\text{BSAT} \in \text{NP}$-complete