Intro to Theory of Computation

CS 332

LECTURE 24

Last time
- Class NP
- Polynomial-time reductions

Today
- Polynomial-time reductions
- NP-completeness

Sofya Raskhodnikova

3/17/2016
Basic reduction strategies

• Reduction by simple equivalence.
• Reduction from special case to general case.
• Reduction by encoding with gadgets.
Set Cover

Given a set U, called a universe, and a collection of its subsets \( S_1, S_2, \ldots, S_m \), a set cover of U is a subcollection of subsets whose union is U.

- **SET COVER** = \{ \langle U, S_1, S_2, \ldots, S_m; k \rangle \mid U \text{ has a set cover of size } k \}

Sample application.

- m available pieces of software.
- Set U of n capabilities that we would like our system to have.
- The \( i \)th piece of software provides the set \( S_i \subseteq U \) of capabilities.
- Goal: achieve all n capabilities using fewest pieces of software.

U = \{ 1, 2, 3, 4, 5, 6, 7 \}

k = 2

\[ S_1 = \{3, 7\} \]
\[ S_2 = \{3, 4, 5, 6\} \]
\[ S_3 = \{1\} \]
\[ S_4 = \{2, 4\} \]
\[ S_5 = \{5\} \]
\[ S_6 = \{1, 2, 6, 7\} \]
**Theorem.** \textsc{vertex-cover} \leq_p \textsc{set-cover}.

**Proof.** “On input \langle G, k \rangle, where \( G = (V, E) \) is an undirected graph and \( k \) is an integer,

1. Output \langle U, S_1, S_2, \ldots, S_m; k \rangle, where \( U=E \) and

\[
S_v = \{ e \in E \mid e \text{ incident to } v \}
\]

**Correctness:**

- \( G \) has a vertex cover of size \( k \) iff \( U \) has a set cover of size \( k \).
- Reduction runs in linear time.
Basic reduction strategies

- Reduction by simple equivalence.
- Reduction from special case to general case.
- Reduction by encoding with gadgets.
Boolean variables: variables that can take on values T/F (or 1/0)

Boolean operations: \( \lor, \land, \text{ and } \neg \)

Boolean formula: expression with Boolean variables and ops

\[ \text{SAT} = \{ \langle \Phi \rangle \mid \Phi \text{ is a satisfiable Boolean formula} \} \]

Literal: A Boolean variable or its negation.

Clause: OR of literals.

Conjunctive normal form (CNF): AND of clauses.

\[ \text{3SAT} = \{ \langle \Phi \rangle \mid \Phi \text{ is a satisfiable Boolean CNF formula, where each clause contains exactly 3 literals} \} \]

Each corresponds to a different variable

\[ \text{Ex: } \left( \bar{x}_1 \lor x_2 \lor x_3 \right) \land \left( x_1 \lor \bar{x}_2 \lor x_3 \right) \land \left( x_2 \lor x_3 \right) \land \left( \bar{x}_1 \lor \bar{x}_2 \lor \bar{x}_3 \right) \]

Yes: \( x_1 = \text{true, } x_2 = \text{true, } x_3 = \text{false.} \)
3SAT reduces to INDEPENDENT SET

**Theorem.** $3$-SAT $\leq_p$ INDEPENDENT-SET.

**Proof.** “On input $\langle \Phi \rangle$, where $\Phi$ is a 3CNF formula,

1. Construct graph $G$ from $\Phi$
   - $G$ contains 3 vertices for each clause, one for each literal.
   - Connect 3 literals in a clause in a triangle.
   - Connect literal to each of its negations.

2. Output $\langle G, k \rangle$, where $k$ is the number of clauses in $G$.”

$$
\Phi = \overline{x_1} \lor x_2 \lor x_3 \lor x_1 \lor \overline{x_2} \lor x_3 \lor \overline{x_1} \lor x_2 \lor x_4
$$

$k = 3$
Correctness. Let $k = \# \text{ of clauses and } \ell = \# \text{ of literals in } \Phi$. 

$\Phi$ is satisfiable iff $G$ contains an independent set of size $k$.

- $\Rightarrow$ Given satisfying assignment, select one true literal from each triangle. This is an independent set of size $k$.

- $\Leftarrow$ Let $S$ be an independent set of size $k$.
  - $S$ must contain exactly one vertex in each triangle.
  - Set these literals to true, and other literals in a consistent way.
  - Truth assignment is consistent and all clauses are satisfied.

Run time. $O(k + \ell^2)$, i.e. polynomial in the input size.
Summary

- **Basic reduction strategies.**
  - Simple equivalence: \( \text{INDEPENDENT-SET} \equiv_p \text{VERTEX-COVER} \).
  - Special case to general case: \( \text{VERTEX-COVER} \leq_p \text{SET-COVER} \).
  - Encoding with gadgets: \( 3\text{-SAT} \leq_p \text{INDEPENDENT-SET} \).

- **Transitivity.** If \( X \leq_p Y \) and \( Y \leq_p Z \), then \( X \leq_p Z \).

- **Proof idea.** Compose the two algorithms.

- **Ex:** \( 3\text{-SAT} \leq_p \text{INDEPENDENT-SET} \leq_p \text{VERTEX-COVER} \leq_p \text{SET-COVER} \).
A language $B$ is **NP-complete** if

1. $B \in \text{NP}$
2. $B$ is **NP-hard**, i.e.,

   every language in NP is poly-time reducible to $B$. 

$P$ ⊆ $NP$
Implication of poly-time reductions

**Theorem.** If

- $B$ is $\text{NP}$-complete,
- $C \in \text{NP}$ and
- $B \leq_p C$

then $C$ is $\text{NP}$-complete.
Implication of poly-time reductions

**Theorem.** If

- B is $\textbf{NP}$-complete,
- $C \in \textbf{NP}$ and
- $B \leq_p C$

then $C$ is $\textbf{NP}$-complete.

**Theorem.** If $B$ is $\textbf{NP}$-complete and $B \in \textbf{P}$ then $\textbf{P} = \textbf{NP}$.

(So, if $B$ is $\textbf{NP}$-complete and $\textbf{P} \neq \textbf{NP}$ then there is no poly-time algorithm for $B$.)

11/30/2017
\( \text{BA}_{\text{NTM}} = \{ \langle M, x, t \rangle \mid M \text{ is an NTM that accepts } x \text{ in at most } t \text{ steps} \} \)

**Theorem.** \( \text{BA}_{\text{NTM}} \) is NP-Complete.

1. \( \text{BA}_{\text{NTM}} \in \text{NP} \):
   
   The list of guesses \( M \) makes to accept \( x \) in \( t \) steps is the certificate that \( \langle M, x, t \rangle \in \text{BA}_{\text{NTM}} \).

2. For all \( A \in \text{NP} \), \( A \leq_p \text{BA}_{\text{NTM}} \).

   \( A \in \text{NP} \) iff there is an NTM \( N \) for \( A \) that runs in time \( O(n^k) \).

   Let \( f_A(w) = \langle N, w, c \mid w \mid^k \rangle \).

   \( \langle N, w, c \mid w \mid^k \rangle \in \text{BA}_{\text{NTM}} \iff N \text{ accepts } w \iff w \in A. \)
CIRCUIT-SAT is similar to 3SAT, but it is about circuits, not formulas.

**Theorem.** CIRCUIT-SAT is NP-complete. [Cook 1971, Levin 1973]

A circuit is built out of AND, OR and NOT gates.

A circuit is **satisfiable** if one can set the circuit inputs, so that the output is 1.  
CIRCUIT-SAT = \{ \langle C \rangle | C \text{ is a satisfiable circuit} \}.

\[ < \text{1 0 1} > \]
Canonical NP-Complete Problem

**Theorem.** 3SAT is NP-complete. (Book)