Lecture 24:

- More NP-completeness
- Space complexity (?)

Reading:
Sipser Ch 7.4-7.5, 8.1-8.2
Polynomial-time reducibility

Definition:
A function $f : \Sigma^* \rightarrow \Sigma^*$ is polynomial-time computable if there is a polynomial-time TM $M$ which, given as input any $w \in \Sigma^*$, halts with only $f(w)$ on its tape.

Definition:
Language $A$ is polynomial-time reducible to language $B$, written

$$A \leq_p B$$

if there is a polynomial-time computable function $f : \Sigma^* \rightarrow \Sigma^*$ such that for all strings $w \in \Sigma^*$, we have $w \in A \iff f(w) \in B$
NP-completeness

“The hardest languages in NP”

Definition: A language $B$ is NP-complete if

1) $B \in \text{NP}$, and

2) Every language $A \in \text{NP}$ is poly-time reducible to $B$, i.e., $A \leq_p B$ ("$B$ is NP-hard")
The usual way to prove NP-completeness

Theorem:

If

1) $C \in \text{NP}$, and
2) There is an NP-complete language $B$ such that $B \leq_p C$

then $C$ is NP-complete.
Some general reduction strategies

• Reduction by simple equivalence
  Ex. $IND - SET \leq_p VERTEX - COVER$
  $VERTEX - COVER \leq_p IND - SET$

• Reduction from special case to general case
  Ex. $VERTEX - COVER \leq_p SET - COVER$
  $3SAT \leq_p SAT$

• “Gadget” reductions
  Ex. $SAT \leq_p 3SAT$
  $3SAT \leq_p IND - SET$
**3SAT (3-CNF Satisfiability)**

**Definitions:**

- A literal either a variable of its negation \( x_5, \overline{x_7} \)
- A clause is a disjunction (OR) of literals \( \text{Ex. } x_5 \lor \overline{x_7} \lor x_2 \)
- A 3-CNF is a conjunction (AND) of clauses where each clause contains exactly 3 literals
  
  \[ C_1 \land C_2 \land \ldots \land C_m = (x_5 \lor \overline{x_7} \lor x_2) \land (\overline{x_3} \lor x_4 \lor x_1) \land \ldots \land (x_1 \lor x_1 \lor x_1) \]

\[ 3SAT = \{ \langle \varphi \rangle \mid \varphi \text{ is a satisfiable } 3 - \text{CNF} \} \]

**Last time:** 3SAT is NP-complete
Independent Set

An independent set in an undirected graph $G$ is a set of vertices that includes at most one endpoint of every edge.

$$IND - SET = \{\langle G, k \rangle | G \text{ is an undirected graph containing an independent set with } \geq k \text{ vertices} \}$$

Which of the following are independent sets in this graph?

a) $\{1\}$  
b) $\{1, 5\}$  
c) $\{2, 3, 6\}$  
d) $\{3, 4, 6\}$
Independent Set is NP-complete

1) \( IND - SET \in NP \)
2) Reduce \( 3SAT \leq_p IND - SET \)

Proof of 1) The following gives a poly-time verifier for \( IND - SET \)

Certificate: Vertices \( v_1, \ldots, v_k \)

Verifier:

“On input \( \langle G, k; v_1, \ldots, v_k \rangle \), where \( G \) is a graph, \( k \) is a natural number,
1. Check that \( v_1, \ldots v_k \) are distinct vertices in \( G \)
2. Check that there are no edges between the \( v_i \)'s.”
Independent Set is NP-complete

1) \( IND - SET \in \text{NP} \)

2) Reduce \( 3SAT \leq_p IND - SET \)

Proof of 2) The following TM computes a poly-time reduction.

“On input \( \langle \varphi \rangle \), where \( \varphi \) is a 3CNF formula,

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

2. Output \( \langle G, k \rangle \), where \( k \) is the number of clauses in \( \varphi \).”
Example of the reduction

\[ \varphi = (\overline{x_1} \lor x_2 \lor x_3) \land (x_1 \lor \overline{x_2} \lor x_3) \land (\overline{x_1} \lor x_2 \lor x_3) \]

\[ G = \]

\[ n = 3 \]

Output: \( \langle G, n \rangle \)

\[ \text{Ex. } \Rightarrow x_1 = 0, x_2 = 0, x_3 = 1 \]

is a sat. ass'nt \( \Rightarrow \) ind set of size 3

\[ \Leftrightarrow S = D's \]

\[ x_2 = 1 \]

\[ x_3 = 1 \]

is a sat. ass'nt

\[ x_1 = 0 \]
Proof of correctness for reduction

Let \( k = \# \) clauses and \( l = \# \) literals in \( \varphi \)

**Correctness:** \( \varphi \) is satisfiable iff \( G \) has an independent set of size \( k \)

\[ \Rightarrow \text{Given a satisfying assignment, select one true literal from each triangle. This is an independent set of size } k \]

\[ \Leftarrow \text{Let } S \text{ be an independent set in } G \text{ of size } k \]

- \( S \) must contain exactly one vertex in each triangle
- Set these literals to true, and set all other variables arbitrarily
- Truth assignment is consistent and all clauses are satisfied

**Runtime:** \( O(k + l^2) \) which is polynomial in input size
Some general reduction strategies

• Reduction by simple equivalence
  Ex. \( \text{IND} - \text{SET} \leq_p \text{VERTEX} - \text{COVER} \)
  \( \text{VERTEX} - \text{COVER} \leq_p \text{IND} - \text{SET} \)

• Reduction from special case to general case
  Ex. \( \text{VERTEX} - \text{COVER} \leq_p \text{SET} - \text{COVER} \)
  \( 3\text{SAT} \leq_p \text{SAT} \)

• “Gadget” reductions
  Ex. \( \text{SAT} \leq_p 3\text{SAT} \)
  \( 3\text{SAT} \leq_p \text{IND} - \text{SET} \)
Vertex Cover

Given an undirected graph \( G \), a vertex cover in \( G \) is a subset of nodes which includes at least one endpoint of every edge.

\[
\text{VERTEX } \rightarrow \text{ COVER } = \{ \langle G, k \rangle \mid \text{\( G \) is an undirected graph which has a vertex cover with } \leq k \text{ vertices} \}
\]

Which of the following are vertex covers in this graph?

a) \( \{1\} \)

b) \( \{1, 6\} \)

c) \( \{1, 2, 5\} \)

d) \( \{1, 2, 5, 6\} \)
Independent Set and Vertex Cover

Claim. $S$ is an independent set iff $V \setminus S$ is a vertex cover.

$\Rightarrow$ Let $S$ be any independent set.
- Consider an arbitrary edge $(u, v)$.
- $S$ is independent $\Rightarrow u \not\in S$ or $v \not\in S$ $\Rightarrow u \in V \setminus S$ or $v \in V \setminus S$.
- Thus, $V \setminus S$ covers $(u, v)$.

$\Leftarrow$ Let $V \setminus S$ be any vertex cover.
- Consider two nodes $u \in S$ and $v \in S$.
- Then $(u, v) \not\in E$ since $V \setminus S$ is a vertex cover.
- Thus, no two nodes in $S$ are joined by an edge $\Rightarrow S$ is an independent set.
Theorem. \( \text{IND-SET} \leq_p \text{VERTEX-COVER} \).

What do we need to do to prove this theorem?

a) Construct a poly-time nondet. TM deciding IND-SET

b) Construct a poly-time deterministic TM deciding IND-SET

c) Construct a poly-time nondet. TM mapping instances of IND-SET to instances of VERTEX-COVER

d) **Construct a poly-time deterministic TM** mapping instances of IND-SET to instances of VERTEX-COVER

e) Construct a poly-time nondet. TM mapping instances of VERTEX-COVER to instances of IND-SET

f) Construct a poly-time deterministic TM mapping instances of VERTEX-COVER to instances of IND-SET
INDEPENDENT SET reduces to VERTEX COVER

**Theorem.** IND-SET \( \leq_p \) VERTEX-COVER.

**Proof.** The following TM computes the reduction.

“On input \( \langle G, k \rangle \), where \( G \) is an undirected graph and \( k \) is an integer,

1. Output \( \langle G, n - k \rangle \), where \( n \) is the number of nodes in \( G \).

**Correctness:**

- \( G \) has an independent set of size at least \( k \) iff it has a vertex cover of size at most \( n - k \).

**Runtime:**

- Reduction runs in linear time.
Theorem. VERTEX-COVER \( \leq_p \) IND-SET

Proof. The following TM computes the reduction.

“On input \( \langle G, k \rangle \), where \( G \) is an undirected graph and \( k \) is an integer,

1. Output \( \langle G, n - k \rangle \), where \( n \) is the number of nodes in \( G \).”

Correctness:

• \( G \) has a vertex cover of size at most \( k \) iff it has an independent set of size at least \( n - k \).

Runtime:

• Reduction runs in linear time.
A Brief Tour of Space (Complexity)
**Space analysis**

**Space complexity** of a TM (algorithm) = maximum number of tape cells it uses on a worst-case input

Formally: Let $f : \mathbb{N} \rightarrow \mathbb{N}$. A TM $M$ runs in space $f(n)$ if on every input $w \in \Sigma^*$, $M$ halts on $w$ using at most $f(n)$ cells.

For nondeterministic machines: Let $f : \mathbb{N} \rightarrow \mathbb{N}$. An NTM $N$ runs in space $f(n)$ if on every input $w \in \Sigma^*$, $N$ halts on $w$ using at most $f(n)$ cells on every computational branch.
Space complexity classes

Let $f : \mathbb{N} \to \mathbb{N}$

A language $A \in \text{SPACE}(f(n))$ if there exists a basic single-tape (deterministic) TM $M$ that
1) Decides $A$, and
2) Runs in space $O(f(n))$

A language $A \in \text{NSPACE}(f(n))$ if there exists a single-tape nondeterministic TM $N$ that
1) Decides $A$, and
2) Runs in space $O(f(n))$
Space vs. Time

\[
\text{TIME}(f(n)) \subseteq \text{NTIME}(f(n)) \\
\subseteq \text{SPACE}(f(n)) \subseteq \text{NSPACE}(f(n))
\]

How about the opposite direction? Can low-space algorithms be simulated by low-time algorithms?

**Theorem:** A TM running in space \( f(n) \) also runs in time \( 2^{O(f(n))} \)

\[
\text{SPACE}(f(n)) \subseteq \text{TIME}(2^{O(f(n))})
\]
Savitch’s Theorem: Deterministic vs. Nondeterministic Space

**Theorem:** Let $f$ be a function with $f(n) \geq \log n$. Then $\text{NSPACE}(f(n)) \subseteq \text{SPACE}\left((f(n))^2\right)$.

$\text{NSPACE}(n^2) \subseteq \text{SPACE}(n^4)$
Complexity class PSPACE

**Definition:** PSPACE is the class of languages decidable in polynomial space on a basic single-tape (deterministic) TM

$$\text{PSPACE} = \bigcup_{k=1}^{\infty} \text{SPACE}(n^k)$$

**Definition:** NPSPACE is the class of languages decidable in polynomial space on a single-tape (nondeterministic) TM

$$\text{NPSPACE} = \bigcup_{k=1}^{\infty} \text{NSPACE}(n^k)$$

$$\text{PSPACE} = \text{NPSPACE} \quad \text{(via Savitch)}$$
Relationships between complexity classes

1. $P \subseteq NP \subseteq PSPACE \subseteq EXP$
   since $SPACE(f(n)) \subseteq TIME(2^{O(f(n))})$

2. $P \neq EXP$
   (via time hierarchy)

Which containments in (1) are proper?

Unknown!
Course Evaluations

bu.campuslabs.com/courseeval