Announcements:

• 121.5: Nicole Immorlica: Econ and CS
• Sections: Polynomial time reductions, NP, etc.
• Homework 5 due today.
• Midterm 2 this Tuesday!
  • 90 minutes (70 if handwritten)
  • 2-sided cheatsheet, noncollaboratively made, plus Barak’s textbook.
  • Material through lecture 17 (Efficient Computation: P)
Where we are:

Part I: Circuits: Finite computation, quantitative study

Part II: Automata: Infinite restricted computation, quantitative study

Part III: Turing Machines: Infinite computation, qualitative study

Part IV: Efficient Computation: Infinite computation, quantitative study

Part V: Randomized computation: Extending studies to non-classical algorithms
Review of last lectures

(Poly-time)

• Reductions: $F \leq_p G \iff \exists R$ such that $\forall x \ F(x) = G(R(x))$, $R$ polytime.

• $3\text{SAT} \leq_p \text{ISET}$

• NP: problems easy to verify.

$$F : \{0,1\}^* \rightarrow \{0,1\} \text{ is in NP iff:}$$

$$\exists V_F : \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\} \ \text{s.t. } \forall x \in \{0,1\}^*, \ F(x) = 1 \iff \exists w \in \{0,1\}^* \text{ such that } V_F(x, w) = 1$$

and $V_F(x, w)$ computable in time $\text{poly}(|x|)$

• (Any problem in NP) $\leq_p \text{NANDSAT} \leq_p \text{3NAND} \leq_p \text{3SAT}$

• So 3SAT is NP-Complete!
Witness, the NP concept

Function $F$ is in NP if $\exists$ polytime $V_F$ s.t. $(F(x) = 1) \iff (\exists w: V_F(x, w) = 1)$

<table>
<thead>
<tr>
<th>Function $F$</th>
<th>Witness $w$</th>
<th>Verifier $V_F$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3SAT(formula)</td>
<td>Variable values</td>
<td>Check: formula satisfied?</td>
</tr>
<tr>
<td>Longpath($G_{x,k}$)</td>
<td>Sequence of vertices</td>
<td>Check: is path, is long</td>
</tr>
<tr>
<td>COMPOSITE($x$)</td>
<td>Factors $p, q$</td>
<td>Check: $p*q=x$</td>
</tr>
<tr>
<td>COMPOSITE($x$)</td>
<td>$y, z$</td>
<td>Check: $\frac{yz}{x} \in \mathbb{Z}, \frac{y}{x} \notin \mathbb{Z}, \frac{z}{x} \notin \mathbb{Z}$</td>
</tr>
</tbody>
</table>

$E_{6}^{2f, t}$

\[
\begin{align*}
6 & \quad 10 \quad 21 \\
210 & \quad 10 \quad 21
\end{align*}
\]
Witness, the computer game

Figure 1: A screenshot from The Witness, featuring 2D puzzles in a 3D world.
Today:

- Some NP-complete problems...
- \(3\text{SAT} \leq_p E3\text{SAT} \leq_p EU3\text{SAT} \leq_p 1\text{-in-EU3SAT} \leq_p \text{SUBSETSUM}\)
- Weak NP-hardness: hard only for big-number inputs
- Strong NP-hardness: hard even for small-number inputs.
3SAT $\leq_p$ E3SAT

Last time,
3NAND $\leq_p$ 3SAT:

\[ c = \text{NAND}(a, b) \]

\[ \overline{a} \lor \overline{b} \lor \overline{c} \]
\[ \land \]
\[ c \lor a \]
\[ \land \]
\[ c \lor b \]

3SAT: Formulas like $(x_7 \lor \overline{x}_{17} \lor x_{29}) \land (\overline{x}_7 \lor x_{15} \lor x_{22}) \land (x_{22} \lor \overline{x}_{29})$, at most 3 variables/clause.

Warning: sometimes "3SAT" $\neq$ E3SAT.

E3SAT: Formulas like $(x_7 \lor \overline{x}_{17} \lor x_{29}) \land (\overline{x}_7 \lor x_{15} \lor x_{22}) \land (x_{22} \lor \overline{x}_{29} \lor x_{22})$, exactly 3 variables/clause.
3SAT $\leq_P$ E3SAT

Reduction:

\[(x_7)
\]
\[(x_7 \lor \overline{x}_{17})
\]
\[(x_7 \lor \overline{x}_{17} \lor x_{29})
\]
\[(x_7) \land (x_7 \lor \overline{x}_{17}) \land (x_7 \lor \overline{x}_{17} \lor x_{29})
\]

Add duplicate literals to fill clauses w/ $\leq 3$ literals.

Proof:

(Sound, Complete)

\[(x_7) \land (x_7 \lor \overline{x}_{17}) \land (x_7 \lor \overline{x}_{17} \lor x_{29})\]

is satisfiable

\[(x_7 \lor x_7 \lor x_7) \land (x_7 \lor \overline{x}_{17} \lor x_7) \land (x_7 \lor \overline{x}_{17} \lor x_{29})\]

is satisfiable

\[(x_7 \lor \overline{x}_{17})\]

is satisfiable

\[(x_7 \lor \overline{x}_{17} \lor x_7)\]

is satisfiable with the same variable values
3SAT \leq_P \text{E3SAT} \quad \text{E3SAT} \leq_P \text{3SAT}

Reduction:

\begin{align*}
&(x_7) \\
&(x_7 \lor \lnot x_{17}) \\
&(x_7 \lor \lnot x_{17} \lor x_{29}) \\
&(x_7) \land (x_7 \lor \lnot x_{17}) \land (x_7 \lor \lnot x_{17} \lor x_{29})
\end{align*}

Proof:

(Sound, Complete)

\begin{align*}
&(x_7) \land (x_7 \lor \lnot x_{17}) \land (x_7 \lor \lnot x_{17} \lor x_{29}) \\
\text{is satisfiable}
\end{align*}

Q: Have we proved that E3SAT is NP-complete?
E3SAT $\leq_P$ EU3SAT

3SAT: Formulas like $(x_7 \lor \overline{x}_{17} \lor x_{29}) \land (\overline{x}_7 \lor x_{15} \lor x_{22}) \land (x_{22} \lor \overline{x}_{29})$, 
*at most 3 variables/clause*

E3SAT: Formulas like $(x_7 \lor \overline{x}_{17} \lor x_{29}) \land (\overline{x}_7 \lor x_{15} \lor x_{22}) \land (x_{22} \lor \overline{x}_{29} \lor x_{22})$, 
*Exactly 3 variables/clause.*

EU3SAT: Formulas like $(x_7 \lor \overline{x}_{17} \lor x_{29}) \land (\overline{x}_7 \lor x_{15} \lor x_{22}) \land (x_{22} \lor \overline{x}_{29} \lor x_{23})$, 
*exactly 3 unique variables/clause.*
E3SAT $\leq_p$ EU3SAT

**Reduction:**

\[
\begin{align*}
(x_7 \lor \overline{x}_{17} \lor x_7) & \Rightarrow \ (x_7 \lor \overline{y}_7 \lor \text{temp}) \land \\
(x_7 \lor \overline{y}_7 \lor \text{temp}) & \land \\
(x_7 \lor y_7 \lor \text{temp}) & \land \\
(x_7 \lor y_7 \lor \text{temp})
\end{align*}
\]

(Wherever we have $t$ copies of a variable in a clause, change $t-1$ of them and add $4(t-1)$ clauses.)

**Proof:**

(Sound, Complete)

$a$ var.

$a, \overline{a}$ literals

E3SAT formula with clauses like

\[
\begin{align*}
(x_7 \lor \overline{x}_{17} \lor y_7) & \land \\
(x_7 \lor \overline{y}_7 \lor \text{temp}) & \land \\
(x_7 \lor \overline{y}_7 \lor \text{temp}) & \land \\
(x_7 \lor y_7 \lor \text{temp}) & \land \\
(x_7 \lor y_7 \lor \overline{\text{temp}})
\end{align*}
\]

is satisfiable
EU3SAT $\leq_P$ 1-in-EU3SAT

3SAT: Formulas like $(x_7 \vee \overline{x}_{17} \vee x_{29}) \land (\overline{x}_7 \vee x_{15} \vee x_{22}) \land (x_{22} \vee \overline{x}_{29})$,
*at most* 3 variables/clause, clause is satisfied iff *at least* one literal is true.

E3SAT: Formulas like $(x_7 \vee \overline{x}_{17} \vee x_{29}) \land (\overline{x}_7 \vee x_{15} \vee x_{22}) \land (x_{22} \vee \overline{x}_{29} \vee x_{22})$,
*Exactly* 3 variables/clause, clause is satisfied iff *at least* one literal is true.

EU3SAT: Formulas like $(x_7 \vee \overline{x}_{17} \vee x_{29}) \land (\overline{x}_7 \vee x_{15} \vee x_{22}) \land (x_{22} \vee \overline{x}_{29} \vee x_{23})$,
*exactly* 3 *unique* variables/clause, clause is satisfied iff *at least* one literal is true.

1-in-EU3SAT: Formulas like $\text{ONEOF}(x_7, \overline{x}_{17}, x_{29}) \land \text{ONEOF}(\overline{x}_7, x_{15}, x_{22}) \land \text{ONEOF}(x_{22}, \overline{x}_{29}, x_{23})$,
*exactly* 3 *unique* variables/clause, clause is satisfied iff *exactly* one literal is true.
The reduction proves that EU3SAT is coNP-hard.

**Reduction:**

- (a ∨ b ∨ c) ∧ (\overline{a} ∨ \overline{b} ∨ \overline{c})
- ONEOF(\overline{a}, w, x) ∧ ONEOF(b, y, x) ∧ ONEOF(c, \overline{w}, \overline{z})

**Proof:**

(Sound, Complete)

- (a ∨ b ∨ c) is satisfiable
- ONEOF(\overline{a}, w, x) ∧ ONEOF(b, y, x) ∧ ONEOF(c, \overline{w}, \overline{z})
- y is new vars. specific to this clause

The proof shows that EU3SAT is coNP-hard.
More SAT variants...

Figure 2-1: SAT notation example.
Knapsack Problem:

Given items with costs $a_0, a_1, ..., a_{k-1}$ and values $v_0, v_1, ..., v_{k-1}$, a budget $b$, and a target value $t$, choose a subset of the items with total cost at most $b$ and value at least $t$. 

<table>
<thead>
<tr>
<th>Appetizers</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Mixed Fruit</td>
<td>2.15</td>
</tr>
<tr>
<td>French Fries</td>
<td>2.75</td>
</tr>
<tr>
<td>Side Salad</td>
<td>3.35</td>
</tr>
<tr>
<td>Hot Wings</td>
<td>3.55</td>
</tr>
<tr>
<td>Mozzarella Sticks</td>
<td>4.20</td>
</tr>
<tr>
<td>Sampler Plate</td>
<td>5.80</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sandwiches</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Barbecue</td>
<td>6.55</td>
</tr>
</tbody>
</table>
Knapsack Problem:

Given items with costs \( a_0, a_1, ..., a_{k-1} \) and values \( v_0, v_1, ..., v_{k-1} \), a budget \( b \), and a target value \( t \), choose a subset of the items with total cost at most \( b \) and value at least \( t \).

Subset Sum:

Given items with costs \( a_0, a_1, ..., a_{k-1} \) and a target value \( t \), choose a subset of the items with total cost exactly \( t \).
1-in-EU3SAT $\leq_P$ Subset Sum

Formulas like
- $\text{ONEOF}(x_7, \overline{x}_{17}, x_{29})$ ∧
- $\text{ONEOF}(\overline{x}_7, x_{15}, x_{22})$ ∧
- $\text{ONEOF}(x_{22}, \overline{x}_{29}, x_{23})$

Given items with costs $a_0, a_1, ..., a_{k-1}$ and a target value $t$, choose a subset of the items with total cost exactly $t$.

Reduction:

1-in-EU3SAT formula  
- $m$ clauses (here $m=3$)  
- $n$ variables (here $n=7$)  

\[ \text{ONEOF}(x_7, \overline{x}_{17}, x_{29}) \land \text{ONEOF}(\overline{x}_7, x_{15}, x_{22}) \land \text{ONEOF}(x_{22}, \overline{x}_{29}, x_{23}) \]

Subset Sum numbers (written in base $n + 1$)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th>0</th>
<th>1</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_7$: $T$</td>
<td>$x_7$: $F$</td>
<td>$x_{17}$: $T$</td>
<td>$x_{17}$: $F$</td>
<td>$x_{23}$: $T$</td>
<td>$x_{23}$: $F$</td>
</tr>
<tr>
<td>0 0 0 0 0 1 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 1 0 0 0 1 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 0 1 0 1 1 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 0 1 0 0 1 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Proof of Correctness?

\[ \text{Ila} \leftarrow \text{literal} \]

\[ \boxed{\text{variable}} \]

\[ \boxed{\text{clause}} \]

\[ \boxed{\text{variable}} \]

Ila uses fuori able variable.
Weak NP-hardness

Subset sum: Given items with costs $a_0, a_1, ..., a_{k-1}$ and a target value $t$, choose a subset of the items with total cost exactly $t$.

Some numbers (costs) in reduction were exponential in $n$. (Poly length!)
If all inputs were polynomial in $n$, Subset Sum isn’t NP-hard.

“Weakly NP-hard”

“Strongly NP-hard”: NP-hard even if all numerical inputs are polynomial-sized.
Traveling Salesman:

Given a (directed or undirected) graph $G$, a “distance” $d_e$ for each edge $e$, and a target $t$, is there a walk visiting all the vertices of $G$ whose total distance is at most $t$?

Strongly NP-hard (NP-hard even if $t$ and every $d_e$ is small).

Hint: Reduce from Longpath:
Given a (directed or undirected) graph $G$ and a target $t$, is there a path visiting at least $t$ vertices? (Paths can’t revisit vertices.)
Summary of Lecture:

• $3\text{SAT} \leq_p \text{E3SAT} \leq_p \text{EU3SAT} \leq_p 1\text{-in-EU3SAT} \leq_p \text{SUBSETSUM}$
• Weak NP-hardness: hard only for big-number inputs
• Strong NP-hardness: hard even for small-number inputs.