Today: *Some NP-Complete Problems*

- **Reductions**
  1. Circuit SAT ≤ 3SAT
  2. 3SAT ≤ Independence
  3. Independence ≤ Vertex Cover

*Reminder*

- Decision Problems - Compute Yes/No answer
- P - polynomial time problems
- NP - Yes has polynomial time verifiable proof

\[ \Pi \leq \Pi^* \text{ if } \exists \text{ poly time } f: \Pi \rightarrow \Pi^* \text{ st. } \]
\[ \Pi(x) = \text{YES } \Rightarrow \Pi^*(f(x)) = \text{YES} \]
- NP-Complete: $\forall \Pi \in \text{NP}$ if
  1. $\Pi \in \text{NP}$
  2. $\forall \Pi \in \text{NP}, \ \Pi \leq \Pi$

---

Part I: Circuit SAT

logic Circuit: Gates

<table>
<thead>
<tr>
<th>AND</th>
<th>OR</th>
<th>NOT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Circuit: Gates connected by wires with \( n \) inputs and 1 output.

Circuit Sat:

Input: Circuit on inputs \( x_1, \ldots, x_n \)

Output: Yes if circuit "satisfiable" i.e. \( \exists x_1, \ldots, x_n \) s.t. output of circuit is 1.

Example:

Output = 1

\[ x_1 = 0 \]
\[ x_2 = 1 \]
\[ x_3 = 1 \]
Theorem [Cook]: Circuit-SAT is NP-complete.

**Proof:** Model every algorithm $V_{T_1}$ carefully.

Given $V_{T_1}$, build circuit $C$

$s.t.\ V_{T_1}(x, y) = \text{Yes} \iff C(y) = 1$.

Take MIT 6.840 to do this properly
Part II: 3SAT

Problem from logic: how to verify if facts we know are consistent.

Example: 0 if it rains, I carry an umbrella.
           1 if I carry an umbrella, it doesn't rain
           3 if it doesn't rain, it is windy
           4 if it is windy, I don't carry an umbrella
           5 if I don't carry an umbrella, it rains.

Are these facts consistent, or am I just paranoid.
Formulation

1. \( \overline{X}_1 \lor X_2 \)  
   \( \overline{X}_1 = \text{not } X_1 \)

2. \( \overline{X}_2 \lor \overline{X}_1 \)

3. \( X \lor X_3 \)

4. \( \overline{X}_3 \lor \overline{X}_2 \)

5. \( X_2 \lor X_1 \)

Can they all be simultaneously true?
3SAT:

- literal: \( X_i \), \( \bar{X}_i \)
- clause: \( X_i \) or \( \bar{X}_j \) or \( \bar{X}_k \)

(in 3SAT, upto 3 literals)

- formula: \( m \) clauses \( C_1, \ldots, C_m \)

Input: formula with \( n \) variables \( X_1, \ldots, X_n \)
\( m \) clauses \( C_1, \ldots, C_m \)

Output: YES if clauses consistent.

Theorem [Cook]: 3SAT is NP-Complete

\[
\begin{cases}
3SAT \in NP \\
Circuit\ SAT \leq 3SAT
\end{cases}
\]
Proof: Circuit SAT \leq 3SAT

3SAT

Variables: \( x_1, \ldots, x_n \)
\( x_{n+1}, \ldots, x_m \)

\( (x_{n+1}, \text{represents output of } g_{n+1}) \)

Input: \( x_1, \ldots, x_n \)

Clauses: represent satisfiability of gates.

(see next page)
Examples:

1. \( x_j \rightarrow \overline{x_i} \lor 
\overline{x_j} \),
   \( x_i \lor x_j \)

2. \( x_k \rightarrow \overline{x_i} \lor \overline{x_j} \),
   \( x_i \lor x_j \)

3. \( x_k \rightarrow \overline{x_i} \lor x_k \),
   \( \overline{x_j} \lor x_k \)

4. Output = 1 \( \Rightarrow \overline{x_m} \)
Reduction of $f$

- Takes as input $x_1, .., x_n, y_m, .., y_m$

  \[ \text{eg: } \{ \text{type, input}_1, \text{input}_2 \} \]

- Outputs formula on variables $x_1, .., x_n$

  Clauses representing validity of gate $g$

  Clauses: $(x_8)$

  \[ \rightarrow (x_8 \lor \overline{x_7}) \land (\overline{x_8} \lor x_7) \]

  \[ \rightarrow (x_4 \lor x_7) \land (\overline{x_6} \lor x_7) \land (x_4 \lor x_6 \lor \overline{x}_7) \]

  \[ \rightarrow (x_5 \lor \overline{x}_6) \land (\overline{x}_5 \lor x_6) \]

  \[ \rightarrow (x_2 \lor \overline{x}_5) \land (x_3 \lor \overline{x}_5) \land (\overline{x}_2 \lor x_3 \lor x_5) \]

  \[ \rightarrow (x_1 \lor \overline{x}_4) \land (x_2 \lor \overline{x}_4) \land (\overline{x}_1 \lor \overline{x}_2 \lor x_4) \]

Variables: $x_1, .., x_8$
Claim: formula is consistent/satisfiable if circuit is satisfiable.

Proof: Assignment to $X_1, \ldots, X_m$ satisfies
formula $\iff X_{m+1}, \ldots, X_m$ are gate outputs
and $X_m = 1$. 

Part III: INDEPENDENCE

Recall

Input: graph $G$, integer $k$

Output: YES if $\exists \ I \subseteq V(G)$

1. $|I| > k$
2. $\forall i,j \in I \ (i,j) \notin E(G)$

Theorem: $[Karp] \ 3SAT \leq IND$

Proof Overview: Will create 2 vertices for each variable & 3 for each clause.

- Any ind. set has $\leq 1$ vertex/variable
- $\Rightarrow \leq 1$ vertex/clause
- ind. set of size $n+m \iff$ formula satisfiable
Variables

\[ x_1, x_2, \ldots, x_n \]

\[ \neg x_1, \neg x_2, \ldots, \neg x_n \]

Clauses: (e.g. \( x_1 \lor x_2 \lor \neg x_3 \))

Claims:

1. Every independent set has size \( \leq n + m \)
   (Since every \( \lor \), \( \land \) contribute \( \leq 1 \)).

2. Ind set of size \( n + m \) has exactly one from each \( \lor \), \( \land \).

3. Ind. set with one vertex from each \( \land \) corresponds to assignment.
4 Ind set corresponding to assignment can remove vertex from \( \square \) iff corresponding clause is satisfied.

5 \( \exists \) ind. set of size \( n \) iff formula is satisfiable.

**Conclusion**: \( \text{IND.} \) is \( \text{NP-complete} \).

- Reductions are very powerful!
- Also very useful ... e.g. \( \exists \) 3SAT solvers in public domain. Reducing your problem to 3SAT can lead to "practical" solution.
Part IV: Vertex Cover

Why? Another popular NP-complete problem.

Today: Will see what makes it hard.

Wednesday: Will see how to "solve" it.

Input: graph G, integer k;

Output: YES if G has a cover of size k

i.e. \( \exists X \subseteq V \)

1. \( |X| \leq k \)
2. \( \forall (i, j) \in E(G) \quad i \in X \) or \( j \in X \).
**Exercise:** Vertex Cover is NP-Complete

1. How do you prove it has small cover?

2. $\text{IND. SET} \leq \text{Vertex Cover}$

Hint: graph does not change!