CS3100

NP-Hardness and (more) reductions
- Next HW due on paper next Wednesday
- HW after will be oral grading on Friday Nov 17

- Office hours Friday:
  10-11 am
  12-1 pm

- No class Nov. 27

  look for reading assignment
Recall: A problem is NP-Complete if it is both:

- In NP: A "yes" answer can be checked in polynomial time
- NP-Hard: via reductions
Thm: (Cook-Levine) Circuit SAT is NP-Complete

Proof: We can turn any Turing machine into a circuit.
To prove any other problem \( A \) is NP-Hard, we'll use a reduction:

Reduce a known NP-Hard problem to \( A \).

\[ \text{NP-Hard problem} \quad \text{transform} \quad \text{subroutine to solve problem A} \quad \text{poly nominal} \]
**Thm:** SAT is NP-Complete

\[ \rightarrow \text{logical sentences} \]
\[ (a \land b) \lor (c \land d) \lor (a \Rightarrow (b \lor c)) \cdots \]

**Pf:** In NP:

Given a set of boolean inputs, linear time to compute output.

**NP-Hard:** reduce CIRCUIT-SAT to SAT:

For each gate, can write a formula.

\[ (y = x_1 \land x_2) \]

\[ (y \lor z = a) \]

\[ (\overline{x_3} = z) \]

\[ \overline{x_3} \]
More carefully:

1) For any gate, can transform:
   \[ a \rightarrow c \quad \text{(c = a \lor b)} \]
   \[ \neg a \rightarrow d \quad \text{(d = \neg a)} \]

2) "And" these together, get want final output true:

\[(y_1 = x_1 \land x_4) \land (y_2 = \overline{x_4}) \land (y_3 = x_3 \land y_2) \land (y_4 = y_1 \lor x_2) \land (y_5 = \overline{x_2}) \land (y_6 = x_5) \land (y_7 = y_3 \lor y_5) \land (z = y_4 \land y_7 \land y_6) \land z\]
Is this poly-size?

Given \( n \) inputs + \( m \) gates:

Variables: \( 1 \) per input \( 3(n+m) \)

Clauses: \( 1 \) per gate

Size of SAT formula:
\[ m+n+m = O(m+n) \]

End reduction:

\[
T_{\text{CSAT}}(n) \leq O(n) + T_{\text{SAT}}(O(n)) \quad \Rightarrow \quad T_{\text{SAT}}(n) \geq T_{\text{CSAT}}(\Omega(n)) - O(n)
\]

(he wrote, "n" is total input size)
Next: 3SAT: a restricted version of SAT

Def: Conjunctive Normal Form (CNF)

\[(a \lor b \lor c \lor d) \land (b \lor \bar{c} \lor \bar{d}) \land (\bar{a} \lor c \lor d) \land (a \lor \bar{b})\]

"OR"s "and"

3SAT: SAT restricted to be CNF or exactly 3 literals per clause

\[(a \lor b \lor c) \land (\bar{a} \lor d \lor \bar{x}) \land \ldots\]

\[\lor\]

3 literals
Thm: 3SAT is NP-Hard
pf: Reduce circuitSAT to 3SAT.

Need to show any circuit can be transformed to CNF form
(so last reduction fails)

Steps:

1. Rewrite so each gate has 2 inputs:
2. Write formula, like in SAT.

3. Types:

\[
\begin{align*}
y &= a \lor b \\
y &= a \land b \\
y &= \overline{a}
\end{align*}
\]

3. Now, change to CNF: go back to truth tables.

4. Now, need 3 per clause:

\[
\begin{align*}
a &= b \land c & \rightarrow & & (a \lor \overline{b} \lor \overline{c}) \land (\overline{a} \lor b) \land (\overline{a} \lor c) \\
a &= b \lor c & \rightarrow & & (\overline{a} \lor b \lor c) \land (a \lor \overline{b}) \land (a \lor \overline{c}) \\
a &= \overline{b} & \rightarrow & & (a \lor b) \land (\overline{a} \lor \overline{b})
\end{align*}
\]

\[
\begin{align*}
a & \rightarrow & & (a \lor x \lor y) \land (a \lor \overline{x} \lor y) \land (a \lor x \lor \overline{y}) \land (a \lor \overline{x} \lor \overline{y}) \\
min{a \lor b} & \rightarrow & & (a \lor b \lor x) \land (a \lor b \lor \overline{x})
\end{align*}
\]
Note: Bigger!

How big?

If exponential, no good
Still polynomial:

For each gate:

- turned into \( \leq 3 \) clauses (wrong sites)
- \( \leq 4 \) clause
- \( \Rightarrow \leq 12 \) clauses per gate

So:

\( \Rightarrow O(m+n) \) size
Next Problem:

Independent Set:

A set of vertices in a graph with no edges between them:

\[ D \times \times \]

Decision version:

Given \( G \) and \( \#k \), is there an independent set of size \( \geq k \)?

In NP: Given \( k \) vertices, check if any edges of them...
Challenge: No booleans!

But reduction needs to take known NP-hard problem and build a graph:

We'll use 3SAT (but stop and marvel a bit first...)
Reduction:
Input is 3CNF boolean formula

\[(a \lor b \lor c) \land (b \lor \bar{c} \lor \bar{d}) \land (\bar{a} \lor c \lor d) \land (a \lor \bar{b} \lor \bar{d})\]

1. Make a vertex for each literal in each clause.
2. Connect two vertices if:
   - they are in some clause
   - they are a variable \( \lor \) its inverse
Example:

\[(a \lor b \lor c) \land (b \lor \bar{c} \lor \bar{d}) \land (\bar{a} \lor \bar{c} \lor d) \land (a \lor b \lor \bar{d})\]

A graph derived from a 3CNF formula, and an independent set of size 4.
Claim: A formula is satisfiable if and only if $G$ has an independent set of size $n$ (number of input variables).

Proof: Let $S$ be a satisfying assignment. Each clause has at least 1 true variable. Choose the corresponding vertex in $G$ to be in an independent set.

Since satisfying assignment means $X$ is true (so $\bar{X}$ won't be connected, and no 2 are connected, so no 2 are connected).
So, must have indep set of size exactly = # of clauses.

\( \leq \): Start w/ indep set in \( G \), of size \( m \).
\( G \) has \( \Delta \)'s: means indep set choose \( \leq 1 \) per triangle.
\( \Rightarrow \) exactly 1 per \( \Delta \).

Since indep set, never choose both \( x \) and \( \bar{x} \) in different clauses.

Build sat. assignment by marking all vertices in indep. set as true.
(others don't matter)
So:

3CNF formula with $k$ clauses \( \xrightarrow{O(n)} \) graph with $3k$ nodes \( \xrightarrow{\text{MaxIndSet}} \) maximum independent set size

True or False \( O(1) \) \( \geq k \)

\( T_{3SAT}(n) \leq O(n) + T_{\text{MaxIndSet}}(O(n)) \implies T_{\text{MaxIndSet}}(n) \geq T_{3SAT}(\Omega(n)) - O(n) \)

Know: 3SAT, Circuit SAT, independent set of size $k$ are NP-hard.