Reducing one (decision) problem to another

input for A → Transform to input/problem B

solve A using B : $T_R + T_B$

is L.I.S.(A) $\geq k$ ?

→ copy A $\rightarrow$ B

→ sort B

→ is L.C.S.(A,B) $\geq k$ ?

→ yes

→ yes

→ no

→ no
LOWER BOUND FOR ONE (DECISION) PROBLEM VIA ANOTHER

Input for D → Transform to input/problem C

Transform to input/problem C → Solve C

Solve C: known $\Omega(f(n)) = T_R + T_C \Rightarrow T_C = \Omega(f(n))$

Quick transformation $T_R \Rightarrow C$ is at least as difficult as D

(fast solution for C ⇒ fast solution for D)
The CONVEX HULL problem (definition by picture)

C.H. input: list of (2D) points

\[ [P_{10} \ P_5 \ P_3 \ P_{12} \ P_1 \ \ldots \ \text{etc...}] \]

\[(x_{12}, y_{12})\]

C.H. output: a graph (cycle) containing only the "extreme" points

\[ [P_3 \ P_2 \ P_{13} \ P_5 \ P_{10} \ P_6 \ P_8] \]
Example:

Transform to 2D point set: $T_R = \Theta(n) = o(n\log n)$

Solve C.H.: $T'_C = \Theta(n)$

Sorted output:

Solve sorting: known $\Omega(n\log n) = T_R + T'_R + T_{C.H.} \Rightarrow T_{C.H.} = \Omega(n\log n)$

$(x_i, x_i^2) \leftarrow (x_i)$

$\Theta(n)$
NPC REDUCTION FROM ONE (DECISION) PROBLEM TO ANOTHER

\[ D \xrightarrow{\text{output}} C \]

\( T_R = \text{polynomial} \)

\[ \text{solve } D: \text{ known NPC} = T_R + T_c = \text{poly-time} + T_c \]

\( \Rightarrow T_c \text{ is NP-hard} \)

(Or \( T_c \) is polynomial \( \Rightarrow P=NP \))
A \leq_p B : \text{ in polynomial time, } A \text{ can be reduced to } B

\Downarrow \text{ } B \text{ is at least as hard as } A.

Solving B in poly-time \Rightarrow \text{solving } A \text{ in poly-time}

\text{If } \{\text{every problem in NP}\} \leq_p B \text{ then } B \text{ is NP-hard}

(\text{and if } B \text{ is in NP, then } B \text{ is NPC})

\text{if } A \& B \text{ are NPC then } A \leq_p B \& B \leq_p A

\sim A =_p B
Thousands of NPC Problems

Reductions/transformations between them resemble a di-graph.

\[ \exists \text{ at least one problem } A \text{ s.t. } A \geq_p \{ \text{everything} \} \& \{ \text{everything} \} \geq_p \cdots \geq_p \cdots \geq_p A \]

Theorem

Must be 1 precisely.
Circuit SAT (satisfiability)
The first NPC problem.
Given a circuit, can the output ever be 1?
Circuit SAT (satisfiability)  
The first NPC problem.  
Given a circuit, can the output ever be 1?  
1 - in NP ... intuitive  
2 & every problem can be described as a circuit \{technical\}  
\{everything\} \leq_p \text{circuit-SAT}  

\[  
\begin{aligned}  
x_i : \text{input}  
\end{aligned}  
\]  
\[  
\begin{aligned}  
0 & \quad x_1 & \quad \text{OR} & \quad 0 \\
0 & \quad x_2 & \quad \text{OR} & \quad 0 \\
0 & \quad x_3 & \quad \text{NOT} & \quad 1 \\
1 & \quad x_4 & \quad \text{AND} & \quad 1 \\
1 & \quad x_5 & \quad \text{AND} & \quad 1 \\
\end{aligned}  
\]

...and we don't know how to solve this in polynomial time. (not in P)
Boolean SAT: \((x_1 \lor x_2 \lor \overline{x_3}) \land ((x_1 \leftrightarrow x_5) \lor (x_4 \rightarrow \overline{x_3})) \rightarrow ? \rightarrow 1\)

Given a circuit-SAT instance, transform (quickly) into a Boolean SAT.

\[
\begin{align*}
W & \land (W \leftrightarrow (Y_4 \land Y_2 \land Y_3)) \\
& \land (Y_4 \leftrightarrow (Y_1 \lor Y_2)) \\
& \land (Y_1 \leftrightarrow (C_1 \lor C_2)) \\
& \land (Y_2 \leftrightarrow \overline{C_3}) \\
& \land (Y_3 \leftrightarrow (C_4 \land C_5))
\end{align*}
\]
3-SAT

$(x_1 \lor x_2 \lor x_3)$

$\land (x_1 \lor x_4 \lor x_5)$

$\land (x_2 \lor \overline{x}_4 \lor x_{13})$

$\land (x_2 \lor \overline{x}_3 \lor x_3)$

$\vdots$

$k$ clauses

each w/ 3 literals
\((a \lor b \lor c) \land (b \lor \bar{c} \lor \bar{d}) \land (\bar{a} \lor c \lor d) \land (a \lor \bar{b} \lor \bar{d})\) \(\{\)k clauses\(\}

3-SAT \rightarrow \text{boolean SAT} \rightarrow \text{Circuit SAT}

\text{independent set}

Ask: does this graph have \(k\) independent vertices?

If yes, then 3-SAT is \(\checkmark\)
If no, then 3-SAT is \(\times\)

construct graph (quickly)

\(\leq 1\) per triple
CLIQUE in a graph: subset of $V$ s.t. all pairs of vertices share edges.

Size $k$ independent set in $G$? $\iff$ size $k$ clique in $G^c$?

$G$ $\xrightarrow{\text{transform}}$ \text{complement}(G) = G^c

Asking if a graph has a clique of size $k$: NPC
OTHER HARD PROBLEMS

- knapsack: given item types, w/ size & value, fill a bag w/ max value (multiples ok)
- subset sum: does a subset of given integers sum to t?
- partition: split set of integers in 2 groups with equal sums
- planar SAT: SAT without wire crossings
- set cover: given some sets, select min# sets s.t. all elements are present
- hitting set: given sets, select min# elements s.t. all sets are represented
- longest path: visiting each vertex once.
- Steiner tree: ~MST on selected subset of a graph
- traveling salesman: min-cost simple cycle on weighted complete graph
OTHER HARD PROBLEMS ... & some are even harder

Tetris  Minesweeper  Lemmings

Mario Bros  Pac-man

Prince of Persia  Portal  Doom

etc