Types of Problems

Easy
(Polynomial time)

- Search
- Sort
- Multiplication
- Closest Points
- Greedy Scheduling
- MWIS on a line
- Matrix Mult.

Puzzles/NP

Crossword
Sudoku
Delivery rt ≤ 100 miles
Protein Folding
Factor larger numbers
Primality Testing

Quantum

Reg.

Question: How do we identify the hardest problems in NP?

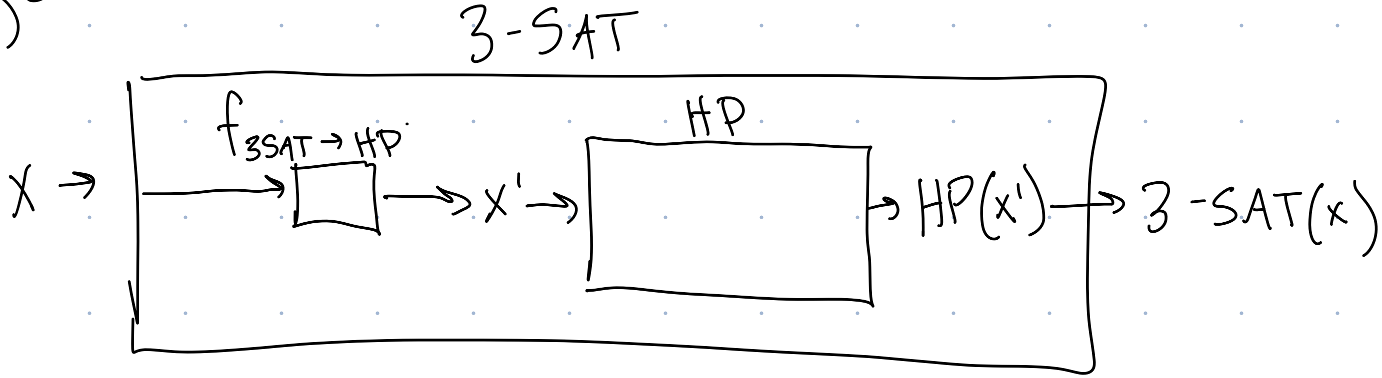
→ Empirical: If keep trying to find an alg, but can't...

→ Analytical: Can we prove a problem is hard? Yes!

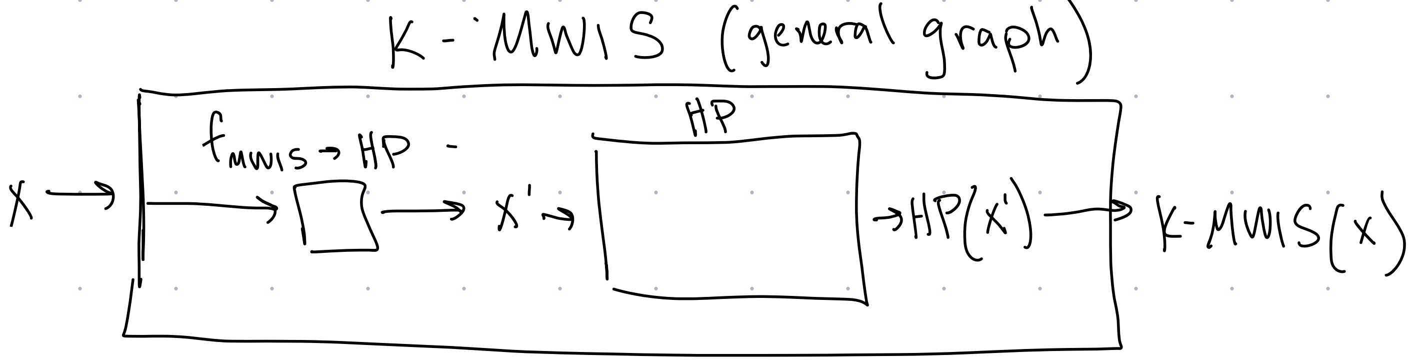
NP-Hard

A problem $Q \in$ NP-Hard if for every problem $R \in NP$, $R \leq_p Q$

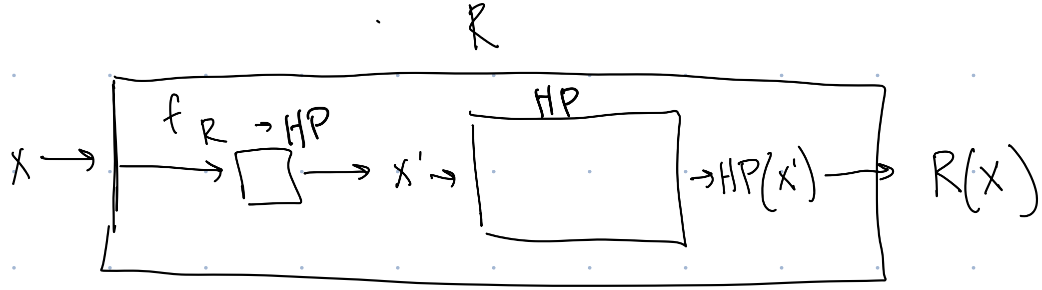
Ex: Halting Problem (HP) $\in$ NP-Hard so



Also

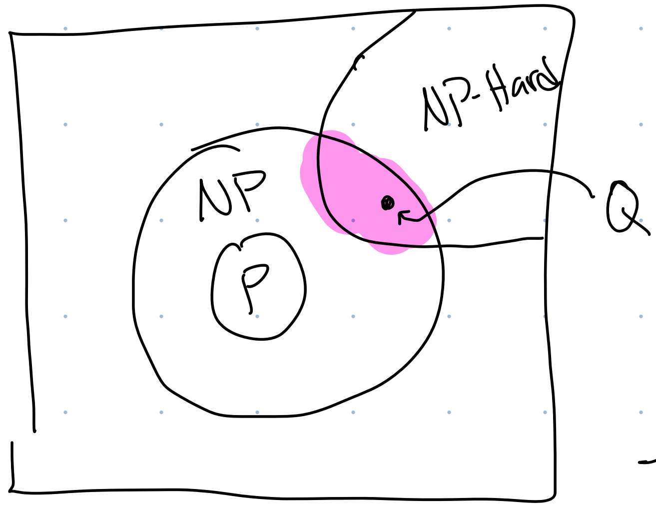


Also For any $R \in NP$, $\exists\ f_{R \to \text{HamPath}}$



→ NP-Hard problems are harder / require more resources than NP problems, b/c give power to solve all in NP

BUT



Exist problems Q s.t.

$\underbrace{Q \in NP \text{ and } Q \in NP\text{-Hard}}_{\text{NP-complete}}$!

- These must be hardest problems in NP! (Ex: MWIS, Sudoku, 3SAT, Traveling Salesperson, Ham-Path, Mario... if solve, could solve all in NP!)

def: $Q \in$ NP-Complete if $Q \in NP$ and $Q \in$ NP-Hard

$\boxed{\text{Fact 1: 3SAT} \in \text{NP-Hard}}$ (see 301)

$\boxed{\text{Lemma 1: If } Q \in \text{NP-Hard and } Q \leq_p R \text{ then } R \in \text{NP-Hard.}}$

(See pset 10)

R is harder than Q

Theorem: Ham-Path is NP-Complete

Pf
- Ham-Path $\in$ NP [insert proof here... see NP class]
- Ham-Path $\in$ NP-Hard
  - ↳ To prove, we will show 3SAT $\leq_p$ Ham-Path. Then using Fact 1 and Lemma 1, we can conclude Ham-Path $\in$ NP-Hard

Formal Definition of Polytime Reduction (Discuss)

<u>def</u>: $R \le_p Q$ (R is polytime reducible to Q) if

$\exists\, f_{R \to Q} : \{0,1\}^* \to \{0,1\}^*$, s.t.

- $\exists$ constant $c_{R \to Q}$ s.t. runtime of $f_{R \to Q}$ on input x is $O(|x|^{c_{R \to Q}})$ (Polytime)
- $\forall x \in \{0,1\}^*$, x is Yes instance for R <u>iff</u> $f_{R \to Q}(x)$ is a Yes instance for Q

(Correct input conversion, no output conversion)

Lemma: 3SAT $\le_p$ Ham-Path

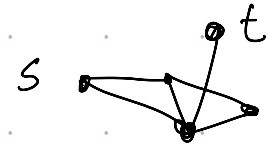
Strategy: 1. Describe $f_{3SAT\text{-}Ham\text{-}Path}$ (turn 3SAT $\rightarrow$ Ham-Path)

2. Show X is 3SAT-Yes iff $f_{3SAT \rightarrow HP}(x)$ is Ham Path-Yes

3. Show $f_{3SAT\text{-}Ham\text{-}Path}$ polytime

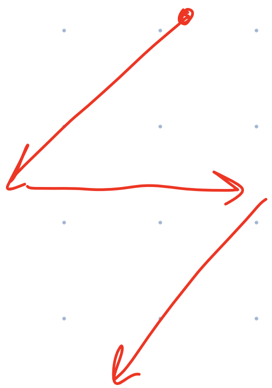
1.

$(z_1 \vee \neg z_2 \vee z_3) \wedge \cdots \Longrightarrow$

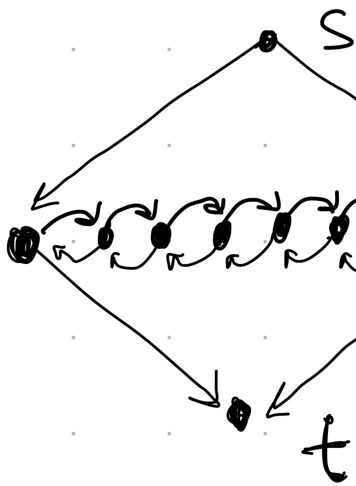


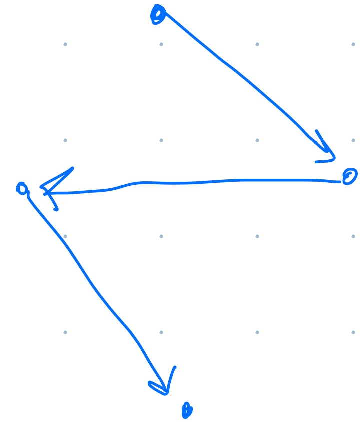
??

How many Hamiltonian Paths are in this graph?



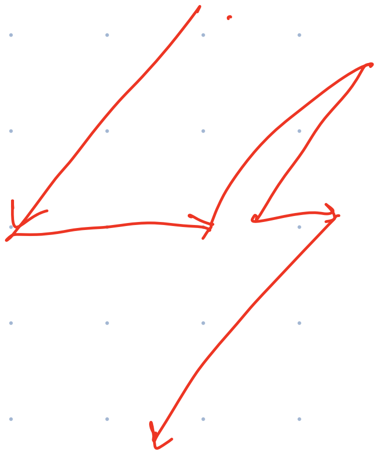
LRL





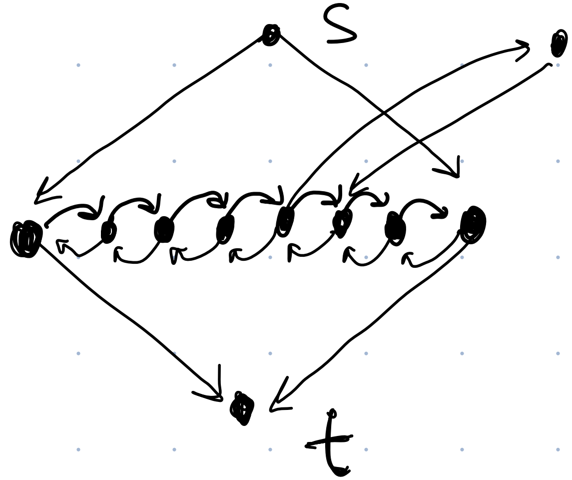
$\boxed{\text{A. } 2}$ B. 3 C. 49 D. $\binom{7}{2}$

How many Hamiltonian Paths are in this graph?



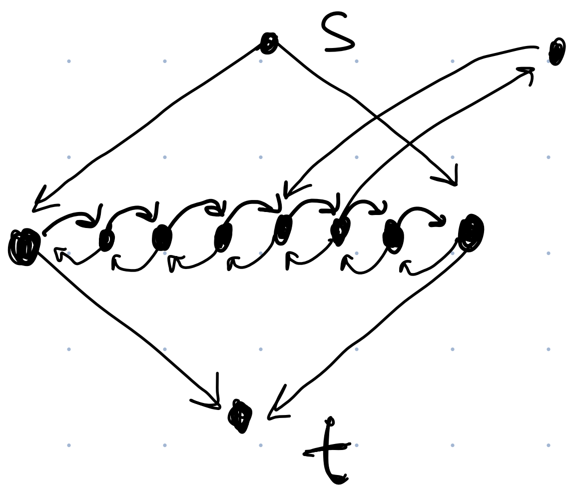
LRL

TRUE



A. 0 $\boxed{\text{B. 1}}$ C. 2 D. 3

False

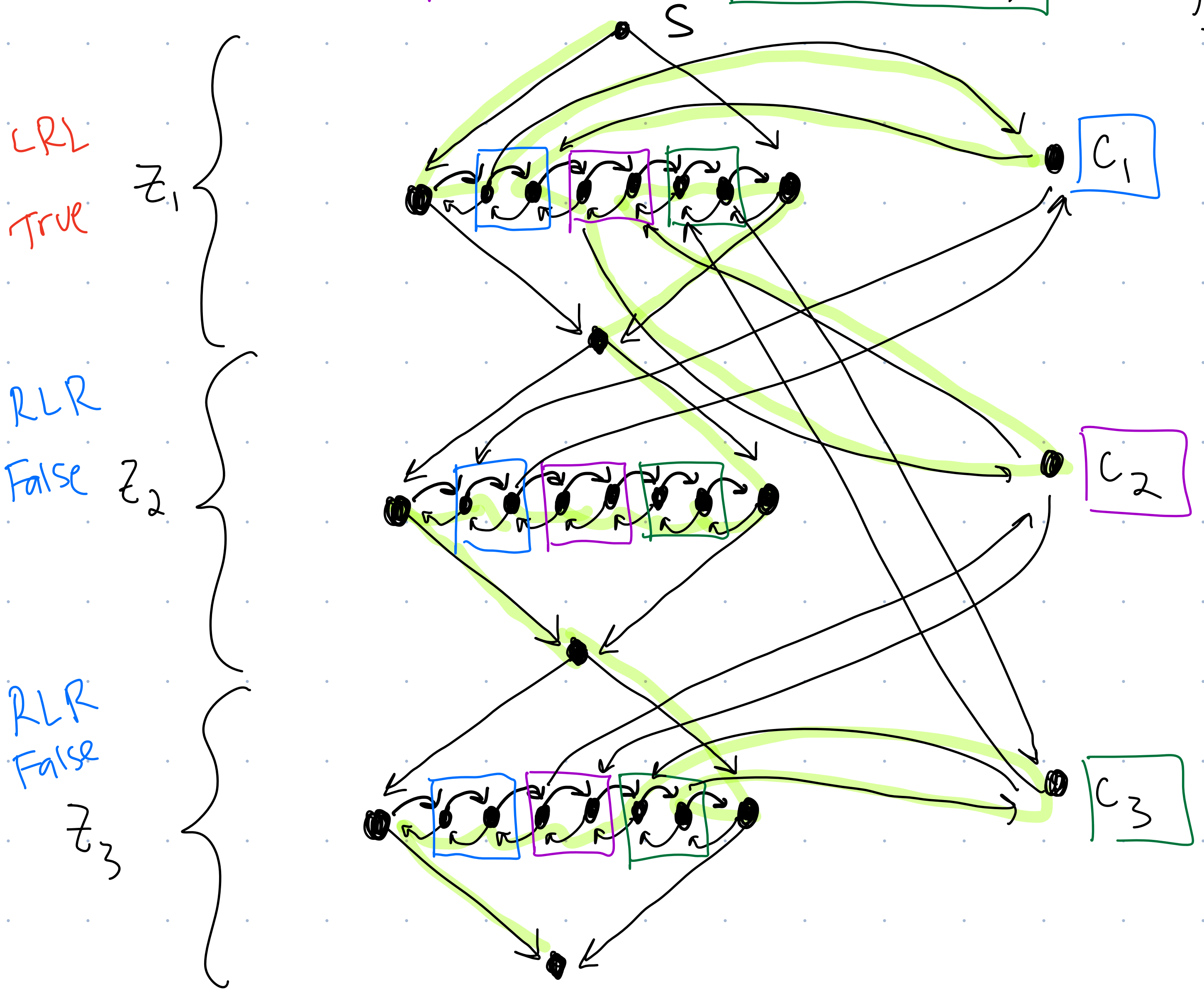


RLR

$$X = \boxed{(z_1 \lor \neg z_2)} \land \boxed{(z_1 \lor z_3)} \land \boxed{(\neg z_1 \lor \neg z_3)}$$

$\boxed{z_1 = T,\ z_2 = z_3 = F}$

Satisfying assignment!



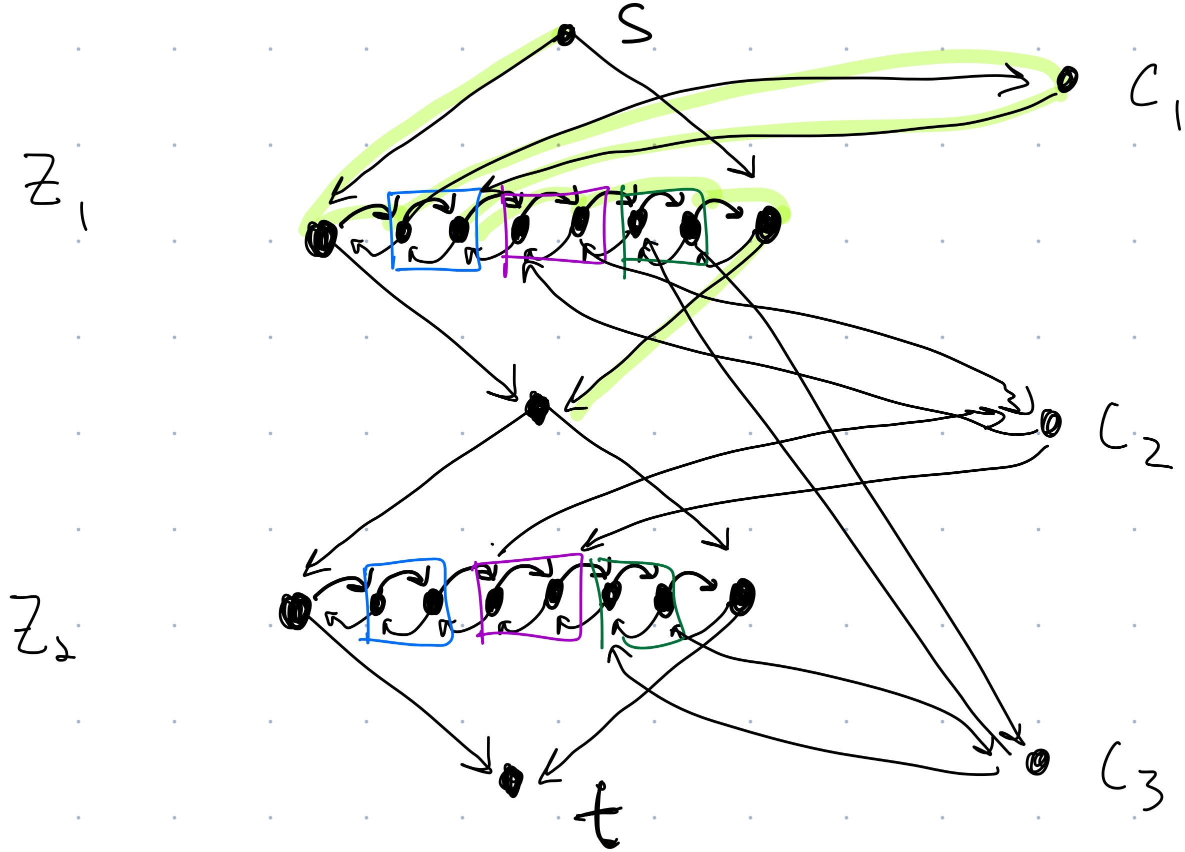
Group Work

1. Encode $(z_1) \wedge (\neg z_1 \vee z_2) \wedge (\neg z_1 \vee \neg z_2)$ into Ham-Path instance. Show get a No Instance.
2. Runtime of $f_{3SAT \to HAM\text{-}PATH}$? (Create adj matrix for graph)
3. X is a Yes-Instance of 3SAT

   iff

   $f_{3SAT-HAMPATH}(X)$ is a Yes instance of HAMPATH

1. $(z_1) \wedge (\neg z_1 \vee z_2) \wedge (\neg z_1 \vee \neg z_2)$



No Path!

2. Let m = # clauses

n = # variables

Each gadget: $2m+4$ vertices

Total gadgets: $n(2m+4)$ vertices

Clause vertices: m

$\Big\}$ $O(nm)$ vertices.

Adjacency matrix $O(n^2m^2)$ size.

Total edges: Gadget: $O(m)$

Clauses: $O(3m)$

$|x| = O(m+n)$ $\longrightarrow$ writing down graph can be done in polynomial time

3. X is a Yes-Instance of 3SAT

iff

$f_{3SAT-HAMPATH}(X)$ is a Yes instance of HAMPATH

$\Rightarrow$ If X is a Yes-Instance of 3SAT, then there is satisfying assignment:

$z_1 \to T$

$z_2 \to F$

$\vdots$

where each clause has at least one satisfying literal.
Choose one satisfying literal for each clause.
Go LRL or RLR through each gadget according to T or F, and if that variable is the chosen one for satisfying a clause, it is possible to jump out to clause variable without breaking LRL, RLR flow. In this way, we will touch each vertex once!

$\Leftarrow$ If $f_{3SAT-HAMPATH}(x)$ is a Yes instance of HAMPATH, the path must go LRL or RLR through each gadget. When the path goes out to a clause vertex, it must return to the same gadget or otherwise the path would miss vertices.

If we assign $z_i = T$ if LRL in gadget i

$z_i = F$ " RLR in gadget i

Will satisfy all clauses, since each clause will be satsisfied by variable associated with the gadget from which the clause vertex is visited.

Note: Once we prove Ham-Path $\in$ NP-Hard, we can combine with Lemma 1 to prove new problems are NP-Hard

$$\boxed{\text{Lemma 1: If } Q \in \text{NP-Hard and } Q \leq_p R \text{ then } R \in \text{NP-Hard.}}$$

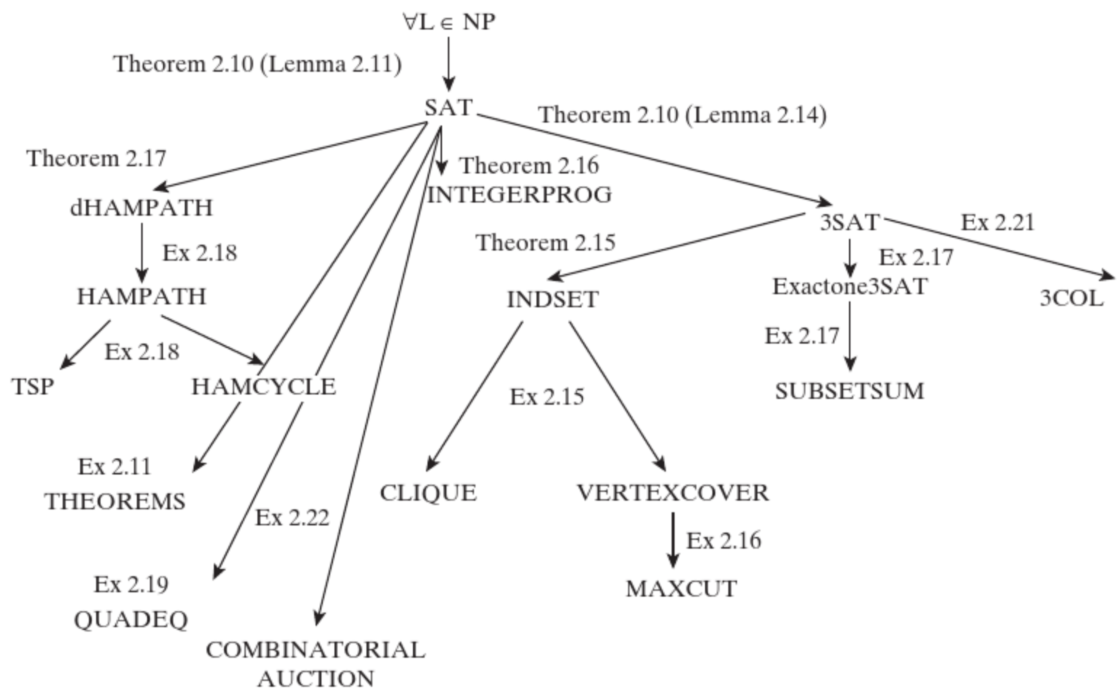


Figure 2.4. Web of reductions between the **NP**-completeness problems described in this chapter and the exercises. Thousands more are known.

(Arora + Boaz, Computational Complexity)