Announcements

- Homework 5 solutions online
- Exam 3 on Friday
  - TAs tell me that pens scan better than pencils, so their use is recommended

Last Time

- NP Problems and NP completeness

NP

Such problems are said to be in <u>Nondeterministic Polynomial</u> time (NP).

<u>NP-Decision</u> problems ask if there is some object that satisfies a polynomial time-checkable property.

<u>NP-Optimization</u> problems ask for the object that maximizes (or minimizes) some polynomial time-computable objective.

Examples of NP Problems

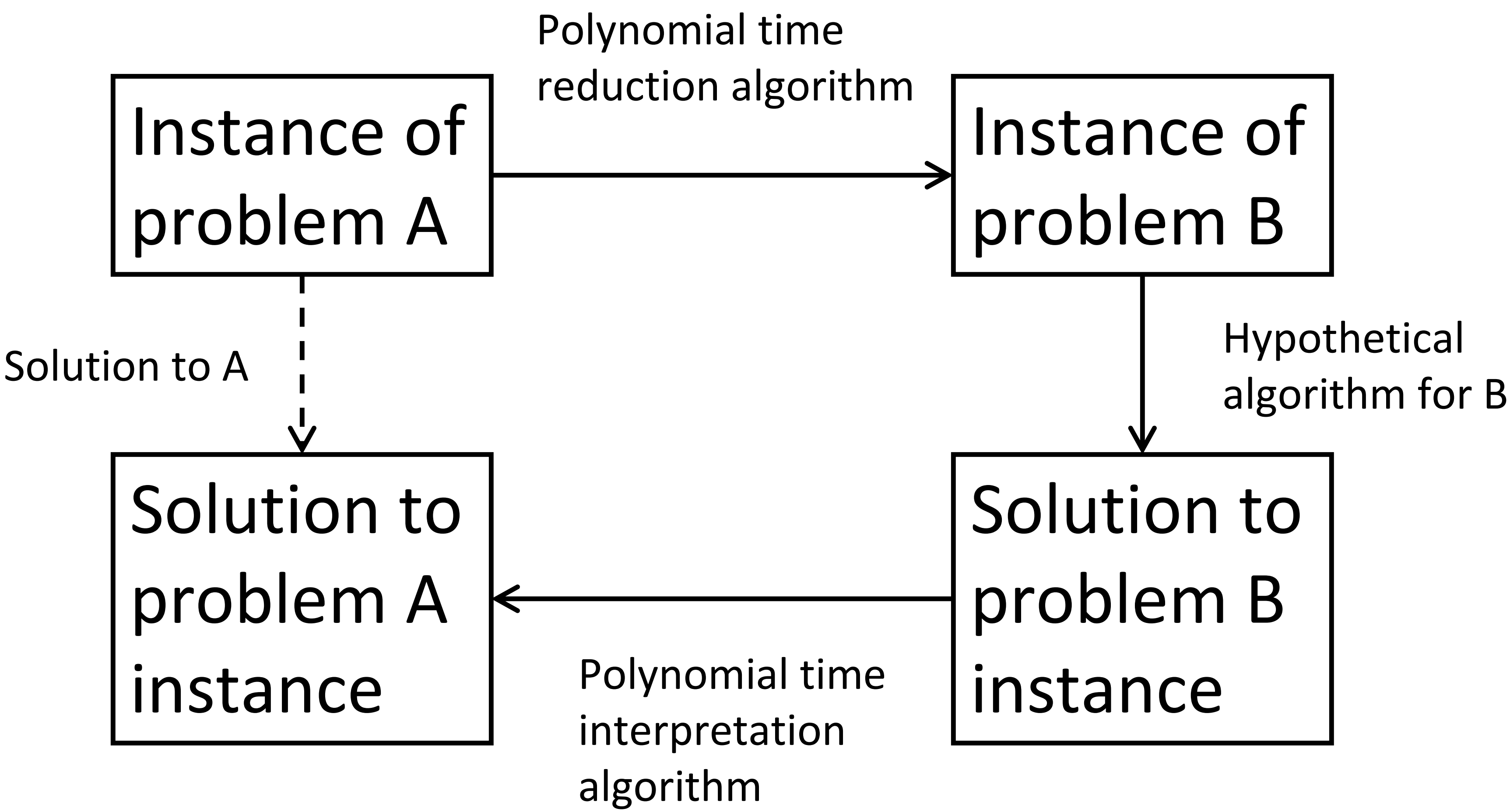
- SAT
- TSP
- Hamiltonian Cycle
- Knapsack
- Maximum Independent Set

Reductions

Reductions are a method for proving that one problem is <u>at least as hard</u> as another.

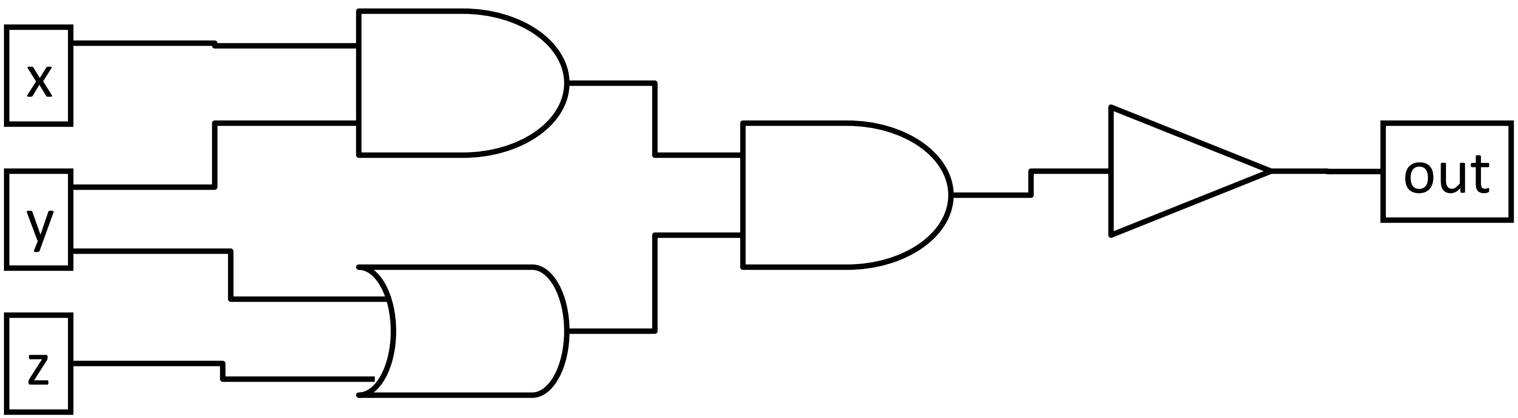
We show that <u>if</u> there is an algorithm for solving A, then we can use this algorithm to solve B. Therefore, B is no harder than A.

Reduction A → B



Circuit SAT

Problem: Given a circuit C with several Boolean inputs and one Boolean output, determine if there is a set of inputs that give output 1.



Important Reduction:

Any NP decision problem → Circuit SAT

NP-Complete

Circuit-SAT is our first example of an NP-Complete problem. That is a problem in NP that is at least as hard as any other problem in NP.

Note: Decision problems can be NP-Complete. For optimization problems, it is called NP-Hard.

Today

- Other NP-Complete Problems
  - 3SAT
  - Maximum Independent Set
  - Zero-One Equations

3-SAT

3-SAT is a special case of formula-SAT where the formula is an AND of clauses and each clause is an OR of at most 3 variables or their negations.

3-SAT

3-SAT is a special case of formula-SAT where the formula is an AND of clauses and each clause is an OR of at most 3 variables or their negations.

Example:

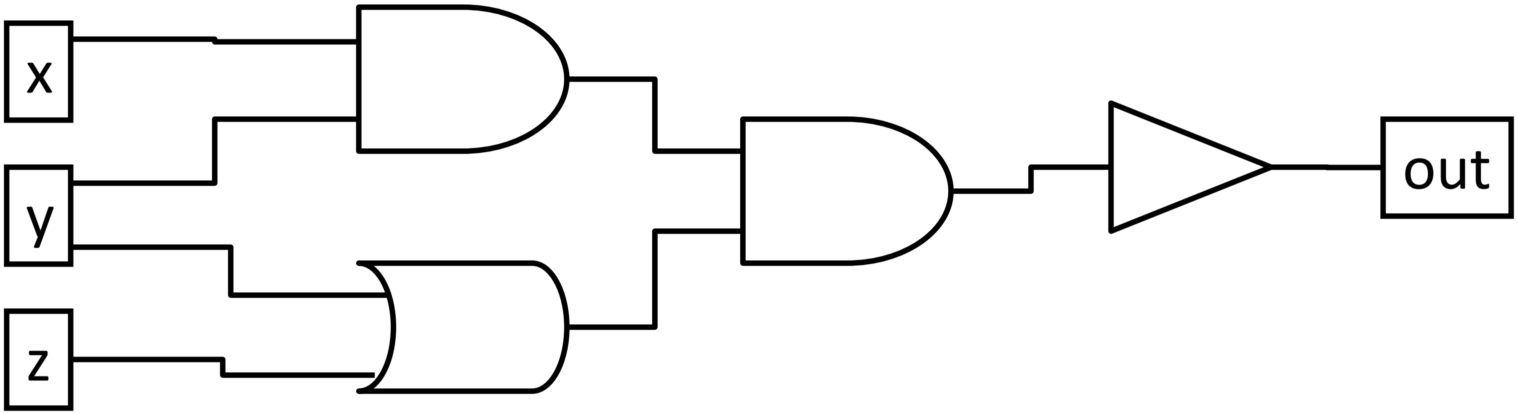
$$(x \vee y \vee z) \wedge (\bar{x} \vee u) \wedge (w \vee \bar{z} \vee u) \wedge (\bar{u} \vee w \vee \bar{z}) \wedge (\bar{y})$$

Circuit-SAT → 3-SAT

- Start with circuit

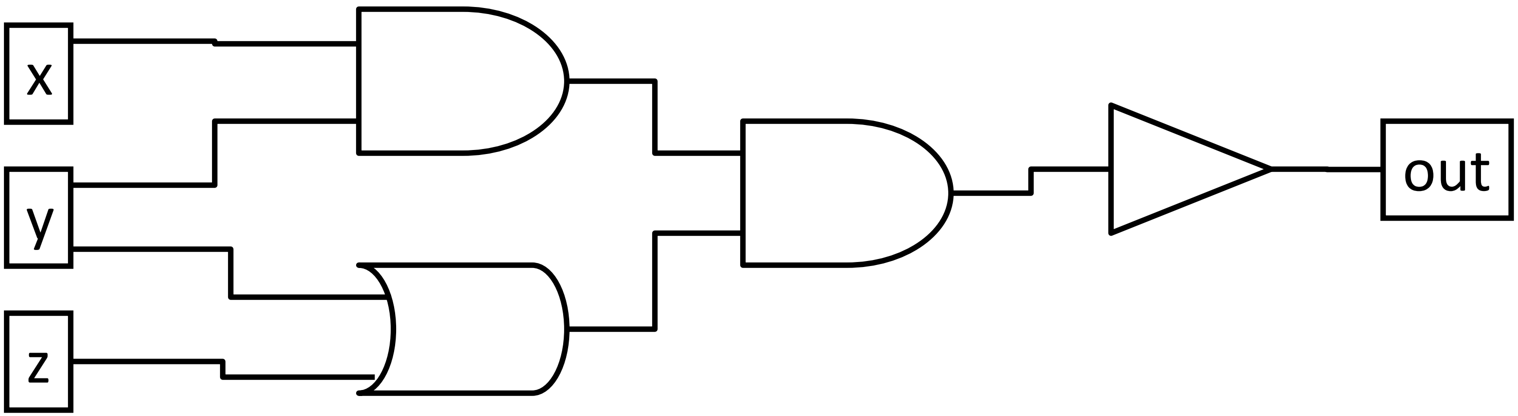
Circuit-SAT → 3-SAT

- Start with circuit



Circuit-SAT → 3-SAT

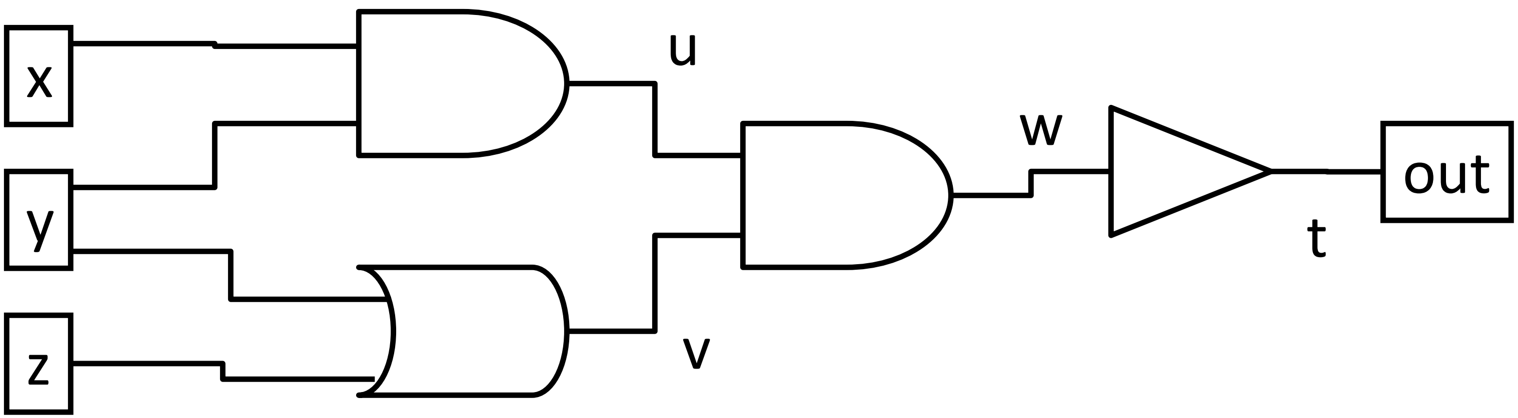
- Start with circuit



- Create variable for each wire

Circuit-SAT → 3-SAT

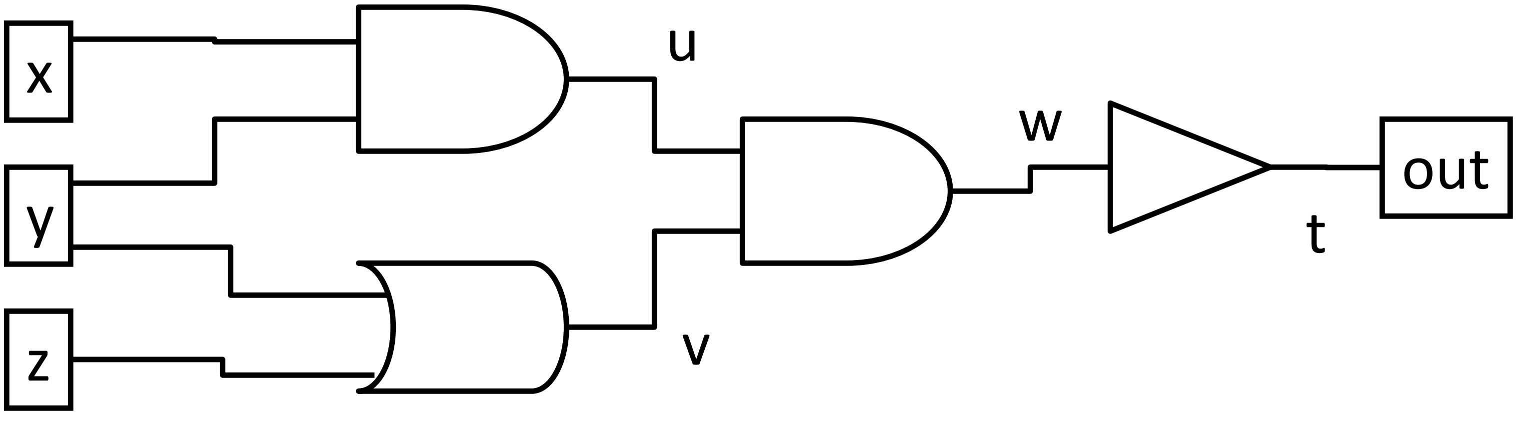
- Start with circuit



- Create variable for each wire

Circuit-SAT → 3-SAT

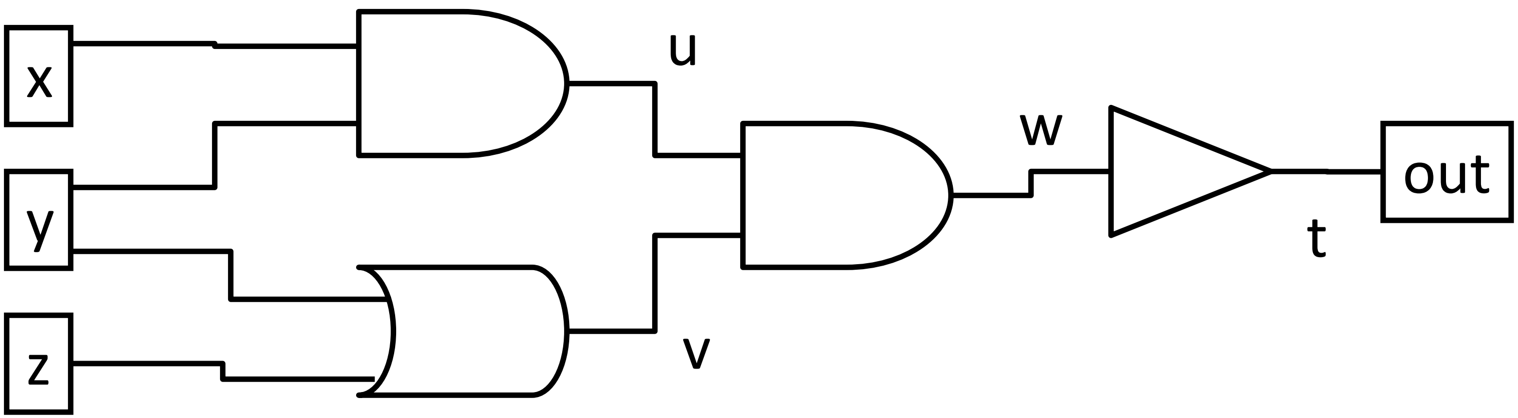
- Start with circuit



- Create variable for each wire
- Create formula with clause for each gate and output

Circuit-SAT → 3-SAT

- Start with circuit



- Create variable for each wire
- Create formula with clause for each gate and output

$$(v \iff y \vee z) \wedge (u \iff x \wedge y) \wedge (w \iff u \wedge v) \wedge (t \iff \bar{w}) \wedge (t)$$

These Aren’t 3-SAT Clauses

We have 3-variable clauses that aren’t 3-SAT clauses. Write it in terms of them.

These Aren't 3-SAT Clauses

We have 3-variable clauses that aren't 3-SAT clauses. Write it in terms of them.

- Write truth table

x	y	z	$z \iff x \vee y$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

These Aren't 3-SAT Clauses

We have 3-variable clauses that aren't 3-SAT clauses. Write it in terms of them.

- Write truth table
- Each 3-SAT clause sets

one output to false.

x	y	z	$z \iff x \vee y$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

These Aren't 3-SAT Clauses

We have 3-variable clauses that aren't 3-SAT clauses. Write it in terms of them.

- Write truth table
- Each 3-SAT clause sets one output to false.

$(x \vee y \vee \bar{z})$

x	y	z	$z \iff x \vee y$
0	0	0	1
0	0	1	0 X
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

These Aren't 3-SAT Clauses

We have 3-variable clauses that aren't 3-SAT clauses. Write it in terms of them.

- Write truth table
- Each 3-SAT clause sets one output to false.

$$(x \vee y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z)$$

x	y	z	$z \iff x \vee y$
0	0	0	1
0	0	1	~~0~~
0	1	0	~~0~~
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

These Aren't 3-SAT Clauses

We have 3-variable clauses that aren't 3-SAT clauses. Write it in terms of them.

- Write truth table
- Each 3-SAT clause sets one output to false.

$$(x \vee y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z) \wedge (\bar{x} \vee y \vee z)$$

x	y	z	$z \iff x \vee y$
0	0	0	1
0	0	1	0 X
0	1	0	0 X
0	1	1	1
1	0	0	0 X
1	0	1	1
1	1	0	0
1	1	1	1

These Aren’t 3-SAT Clauses

We have 3-variable clauses that aren’t 3-SAT clauses. Write it in terms of them.

- Write truth table
- Each 3-SAT clause sets one output to false.

$$(x \vee y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z)$$
$$\wedge (\bar{x} \vee y \vee z) \wedge (\bar{x} \vee \bar{y} \vee z)$$

x	y	z	$z \iff x \vee y$
0	0	0	1
0	0	1	~~0~~
0	1	0	~~0~~
0	1	1	1
1	0	0	~~0~~
1	0	1	1
1	1	0	~~0~~
1	1	1	1

These Aren’t 3-SAT Clauses

We have 3-variable clauses that aren’t 3-SAT clauses. Write it in terms of them.

- Write truth table
- Each 3-SAT clause sets one output to false.

$$(x \vee y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z)$$
$$\wedge (\bar{x} \vee y \vee z) \wedge (\bar{x} \vee \bar{y} \vee z)$$

$$= (z \iff x \vee y)$$

x	y	z	$z \iff x \vee y$
0	0	0	1
0	0	1	~~0~~
0	1	0	~~0~~
0	1	1	1
1	0	0	~~0~~
1	0	1	1
1	1	0	~~0~~
1	1	1	1

Note

This means that 3-SAT is <u>also</u> NP-Complete since we have:

Any problem in NP → Circuit SAT → 3-SAT

Note

This means that 3-SAT is <u>also</u> NP-Complete since we have:

Any problem in NP → Circuit SAT → 3-SAT

What other problems can we show to be NP-Complete/NP-Hard this way?

Another Look at 3-SAT

Lemma: A 3-SAT instance is satisfiable if and only if it is possible to select one term from each clause without selecting both a variable and its negation.

Another Look at 3-SAT

Lemma: A 3-SAT instance is satisfiable if and only if it is possible to select one term from each clause without selecting both a variable and its negation.

Example:

$$(x \vee y \vee z) \wedge (\bar{x} \vee y) \wedge (\bar{y} \vee \bar{z}) \wedge (\bar{x} \vee z)$$

Another Look at 3-SAT

Lemma: A 3-SAT instance is satisfiable if and only if it is possible to select one term from each clause without selecting both a variable and its negation.

Example:

$$(x \vee y \vee z) \wedge (\bar{x} \vee y) \wedge (\bar{y} \vee \bar{z}) \wedge (\bar{x} \vee z)$$

Proof

<u>If satisfiable:</u>

- Satisfying assignment causes at least one term in each clause to be true.

Proof

If satisfiable:

- Satisfying assignment causes at least one term in each clause to be true.
- Select one such term from each clause.
- Cannot contradict each other.

Proof

<u>If satisfiable:</u>

- Satisfying assignment causes at least one term in each clause to be true.
- Select one such term from each clause.
- Cannot contradict each other.

<u>**Example:**</u>

$$(x \vee y \vee z) \wedge (\bar{x} \vee y) \wedge (\bar{y} \vee \bar{z}) \wedge (\bar{x} \vee z)$$

Proof

<u>If satisfiable:</u>

- Satisfying assignment causes at least one term in each clause to be true.
- Select one such term from each clause.
- Cannot contradict each other.

<u>**Example:**</u>

$$(x \vee y \vee z) \wedge (\bar{x} \vee y) \wedge (\bar{y} \vee \bar{z}) \wedge (\bar{x} \vee z)$$

$$x = \text{False}, y = \text{True}, z = \text{False}$$

Proof

<u>If satisfiable:</u>

- Satisfying assignment causes at least one term in each clause to be true.
- Select one such term from each clause.
- Cannot contradict each other.

<u>Example:</u>

$$(x \vee y \vee z) \wedge (\bar{x} \vee y) \wedge (\bar{y} \vee \bar{z}) \wedge (\bar{x} \vee z)$$

$$x = \text{False}, y = \text{True}, z = \text{False}$$

Proof

<u>If satisfiable:</u>

- Satisfying assignment causes at least one term in each clause to be true.
- Select one such term from each clause.
- Cannot contradict each other.

<u>Example:</u>

$$(x \vee y \vee z) \wedge (\bar{x} \vee y) \wedge (\bar{y} \vee \bar{z}) \wedge (\bar{x} \vee z)$$

$$x = \text{False}, y = \text{True}, z = \text{False}$$

Proof

If there is a way to select terms:

- Set those variables to be true
  - Can do this without contradiction

Proof

<u>If there is a way to select terms:</u>

- Set those variables to be true
  - Can do this without contradiction
- Set other variables arbitrarily

Proof

If there is a way to select terms:

- Set those variables to be true
  - Can do this without contradiction
- Set other variables arbitrarily
- Causes whole statement to be true

Proof

<u>If there is a way to select terms:</u>

- Set those variables to be true
  - Can do this without contradiction
- Set other variables arbitrarily
- Causes whole statement to be true

<u>**Example:**</u> $(x \vee y \vee z) \wedge (\bar{x} \vee y)$

Proof

<u>If there is a way to select terms:</u>

- Set those variables to be true
  - Can do this without contradiction
- Set other variables arbitrarily
- Causes whole statement to be true

<u>Example:</u> $(x \vee y \vee z) \wedge (\bar{x} \vee y)$

Proof

<u>If there is a way to select terms:</u>

- Set those variables to be true
  - Can do this without contradiction
- Set other variables arbitrarily
- Causes whole statement to be true

Example: $(x \vee y \vee z) \wedge (\bar{x} \vee y)$

$$x = \text{True}, y = \text{True}$$

Proof

If there is a way to select terms:

- Set those variables to be true
  - Can do this without contradiction
- Set other variables arbitrarily
- Causes whole statement to be true

Example: $(x \vee y \vee z) \wedge (\bar{x} \vee y)$

$$x = \text{True}, y = \text{True}, z = \text{False}$$

3-SAT → Maximum Independent Set

Want to encode this select one term from each clause as a graph.

3-SAT → Maximum Independent Set

Want to encode this select one term from each clause as a graph.

Example:

$$(x \vee y \vee z) \wedge (\bar{x} \vee y) \wedge (\bar{y} \vee \bar{x})$$

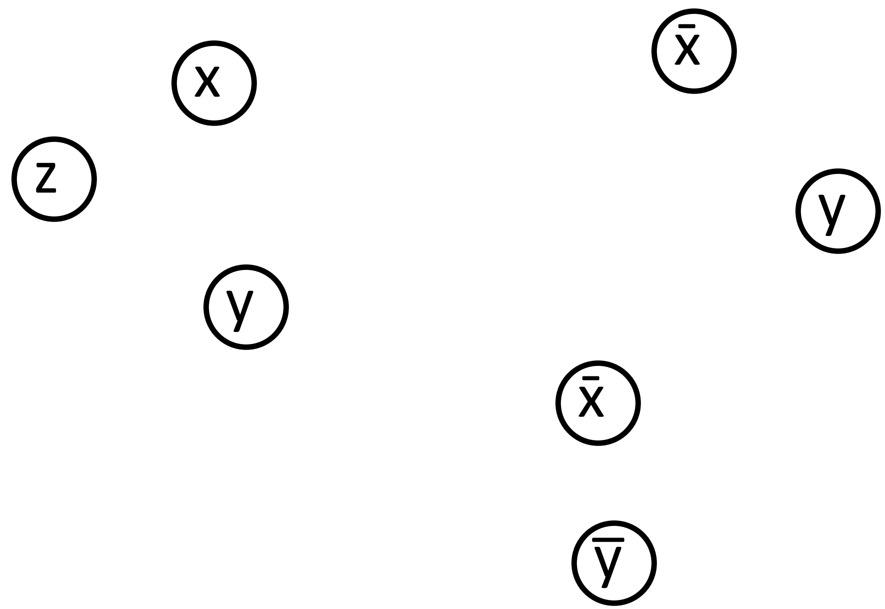
3-SAT → Maximum Independent Set

Want to encode this select one term from each clause as a graph.

- Create one vertex for each term in each clause.

Example:

$$(x \vee y \vee z) \wedge (\bar{x} \vee y) \wedge (\bar{y} \vee \bar{x})$$



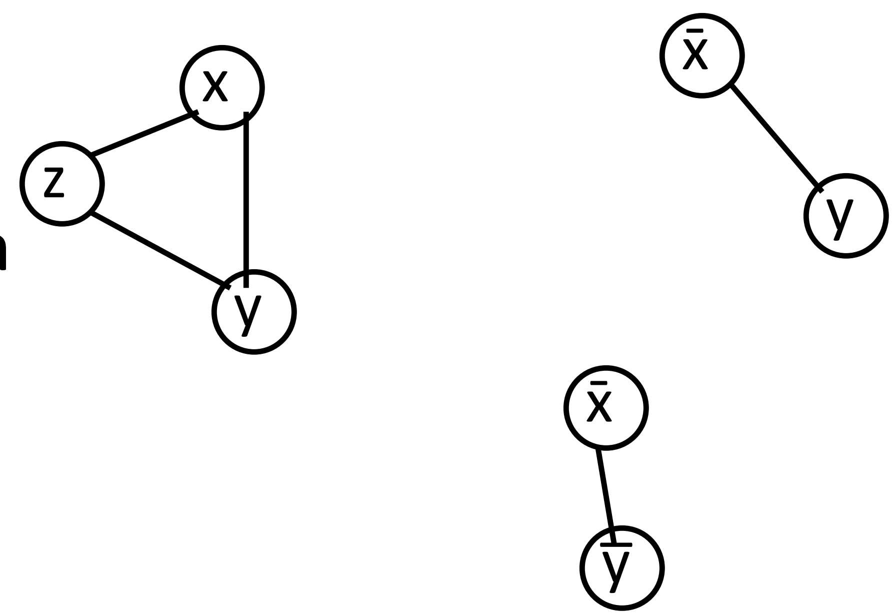
3-SAT → Maximum Independent Set

Want to encode this select one term from each clause as a graph.

- Create one vertex for each term in each clause.
- Edges between terms in same clause.

Example:

$$(x \vee y \vee z) \wedge (\bar{x} \vee y) \wedge (\bar{y} \vee \bar{x})$$



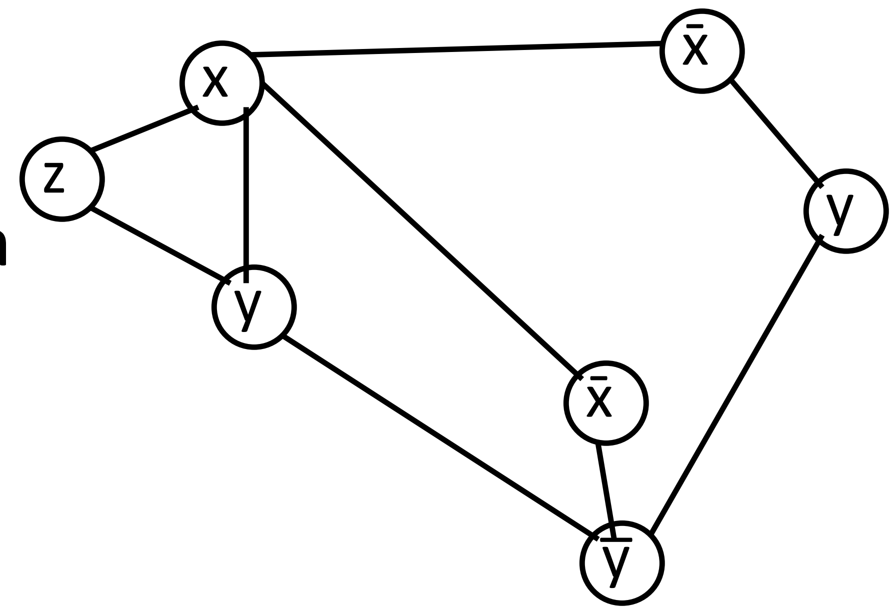
3-SAT → Maximum Independent Set

Want to encode this select one term from each clause as a graph.

- Create one vertex for each term in each clause.
- Edges between terms in same clause.
- Edges between contradictory terms.

Example:

$$(x \vee y \vee z) \wedge (\bar{x} \vee y) \wedge (\bar{y} \vee \bar{x})$$



Analysis

An independent set in this graph has:

- At most one vertex from each clause.
- No vertices representing contradictory terms.

Analysis

An independent set in this graph has:

- At most one vertex from each clause.
- No vertices representing contradictory terms.

It has an independent set of size #Clauses if and only if, you can select one term form each clause without a contradiction.

Analysis

An independent set in this graph has:

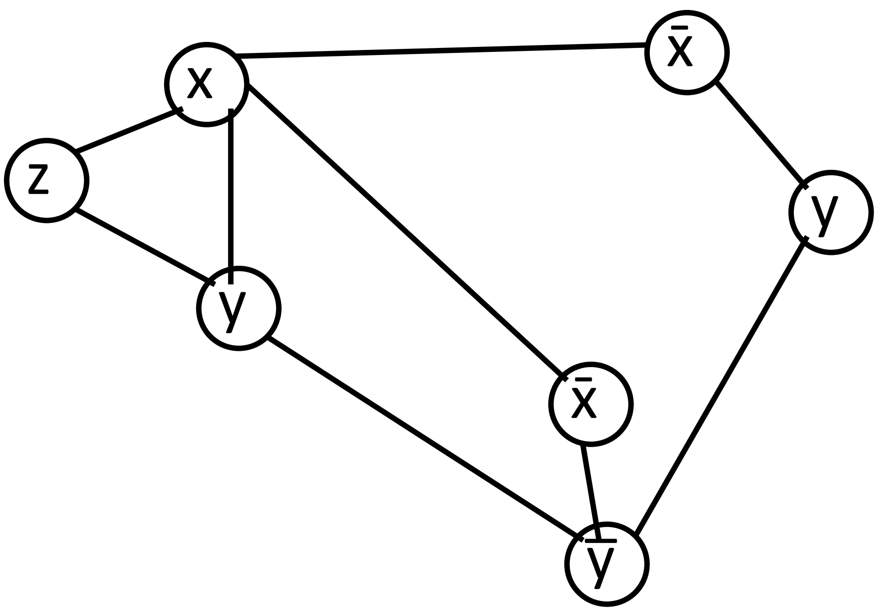
- At most one vertex from each clause.
- No vertices representing contradictory terms.

It has an independent set of size #Clauses if and only if, you can select one term form each clause without a contradiction.

Therefore, |MIS| = #Clauses if and only if the 3-SAT has a solution.

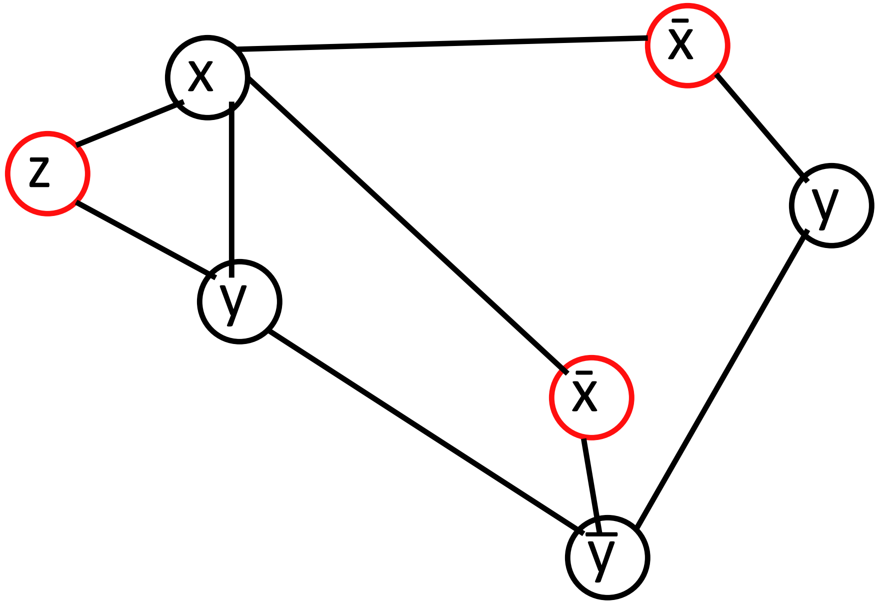
Example

$$(x \vee y \vee z) \wedge (\bar{x} \vee y) \wedge (\bar{y} \vee \bar{x})$$



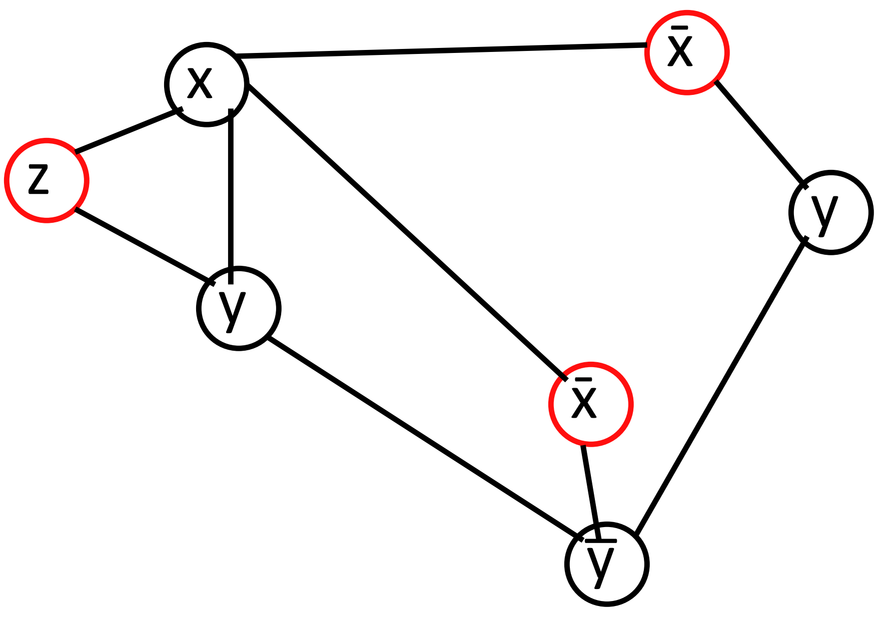
Example

$$(x \vee y \vee z) \wedge (\bar{x} \vee y) \wedge (\bar{y} \vee \bar{x})$$



Example

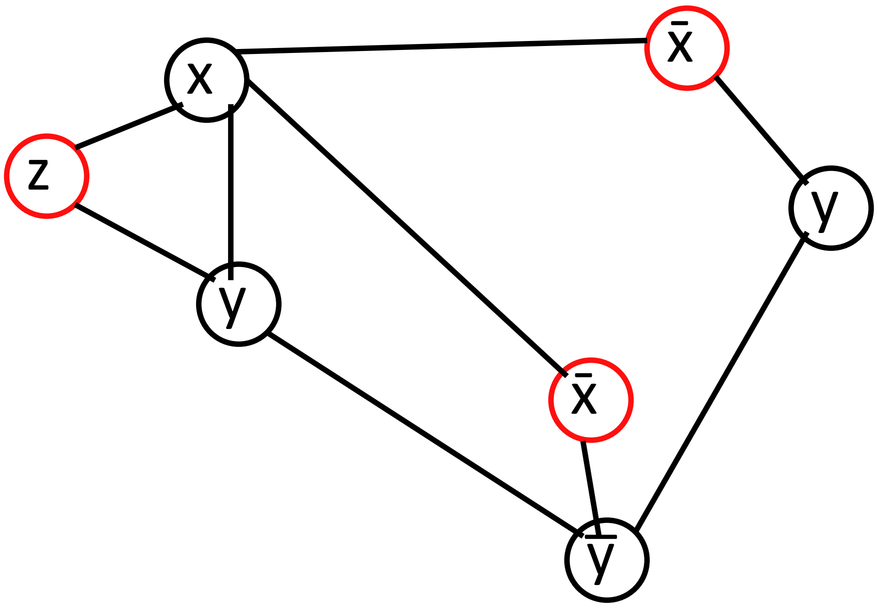
$$(x \vee y \vee z) \wedge (\bar{x} \vee y) \wedge (\bar{y} \vee \bar{x})$$



Example

$$(x \vee y \vee z) \wedge (\bar{x} \vee y) \wedge (\bar{y} \vee \bar{x})$$

$$x = \text{False}, z = \text{True}, y = \text{Whatever}$$



Intermediate Problems

To prove our more complicated reductions, it will help to have the correct problem to prove reductions from.

Intermediate Problems

To prove our more complicated reductions, it will help to have the correct problem to prove reductions from.

A convenient problem is the one the book calls Zero-One Equations.

Zero-One Equations

Problem: Given a matrix A with only 0 and 1 as entries and b a vector of 1s, determine whether or not there is an x with 0 and 1 entries so that

$$Ax = b.$$

Zero-One Equations

Problem: Given a matrix A with only 0 and 1 as entries and b a vector of 1s, determine whether or not there is an x with 0 and 1 entries so that

$$Ax = b.$$

This problem is clearly in NP. We will show that it is NP-Complete.