Even More NP-Complete Problems

Wednesday, December 9, 2020

HW11 questions?

Announcements

- HW11 finalized
  - Due Tues 12/15 11:59pm EST
- HW8 grades returned
- Please stop posting HW questions to Chegg
  - The answers arent even correct
- Stay tuned for extra credit announcement
  - Most likely to be extra time to finish coding assignments
- Fill out Course Evaluation at end of class

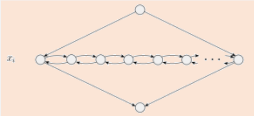
NP-Complete problems, so far

- $SAT = \{\langle \phi \rangle \mid \phi \text{ is a satisfiable Boolean formula}\}$ (Cook-Levin Theorem)
- $3SAT = \{\langle \phi \rangle \mid \phi \text{ is a satisfiable 3cnf-formula}\}$ (reduce from *SAT*)
- $CLIQUE = \{\langle G, k \rangle \mid G \text{ is an undirected graph with a } k\text{-clique}\}$ (reduce from 3*SAT*)
- $HAMPATH = \{\langle G, s, t \rangle \mid G \text{ is a directed graph with a Hamiltonian path from } s \text{ to } t\}$ (reduce from 3*SAT*)

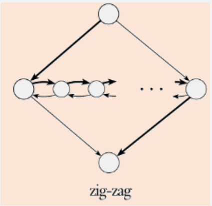
<u>General Strategy</u>: Reducing from *3SAT*

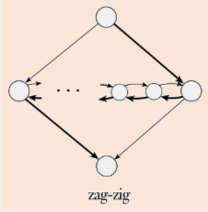
Create a **computable function** mapping formula to “gadgets”:

- Clause → some “gadget”, e.g., 

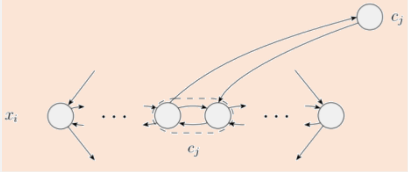
- Variable → another “gadget”, e.g., 

  Gadget is typically used in two “opposite” ways:
  - When var is assigned TRUE, or
  - When var is assigned FALSE

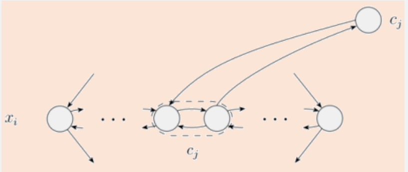




Then connect “gadgets” according to clause literals:



- Literal x_i in clause c_j → gadget x_i “detours” to c_j
- Literal $\overline{x_i}$ in clause c_j → gadget x_i “reverse detours” to c_j



<u>Theorem</u>: *SUBSET-SUM* is NP-complete

$SUBSET\text{-}SUM = \{\langle S, t\rangle \mid S = \{x_1, \ldots, x_k\}$, and for some $\{y_1, \ldots, y_l\} \subseteq \{x_1, \ldots, x_k\}$, we have $\Sigma y_i = t\}$

THEOREM 7.36

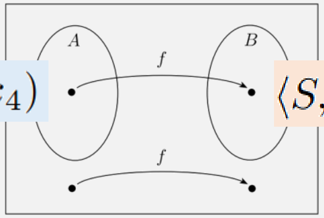
If B is NP-complete and $B \leq_P C$ for C in NP, then C is NP-complete.

<u>Strategy</u>: Use

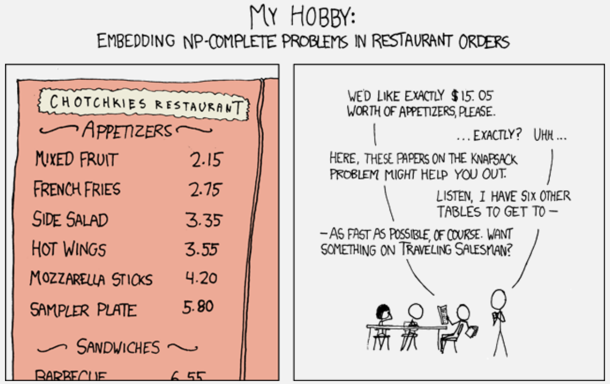
<u>Proof Parts (5)</u>:

1. Show *SUBSET-SUM* is in **NP** (done in prev class)
2. Choose **NP**-complete problem to reduce from: *3SAT*
3. Create the <u>computable function</u> *f*:

$(x_1 \vee \overline{x_2} \vee \overline{x_3}) \wedge (x_3 \vee \overline{x_5} \vee x_6) \wedge (x_3 \vee \overline{x_6} \vee x_4)$



$\langle S, t\rangle \mid S = \{x_1, \ldots, x_k\}$



4. Show it runs in poly time
5. Show Def 7.29 "iff" requirement:

ϕ is a satisfiable 3cnf-formula $\iff f(\langle \phi \rangle) = \langle S, t\rangle$ where some subset of S sums to t

Computable Fn: 3cnf → $\langle S, t \rangle$

E.g., $(x_1 \vee \overline{x_2} \vee x_3) \wedge (x_2 \vee x_3 \vee \cdots) \wedge \cdots \wedge (\overline{x_3} \vee \cdots \vee \cdots)$ ➡

- Assume formula has:
  - ***l*** variables $x_1, \ldots, x_l$
  - ***k*** clauses $c_1, \ldots, c_k$
- Computable function *f* maps:
  - Variable x_i → two numbers y_i and z_i
  - Clause c_j → two numbers g_j and h_j
- Each number has max ***l+k*** digits:
- Sum is ***l*** 1s followed by ***k*** 3s

y_i and z_i: i^{th} digit = 1

y_i: ***l***+j^{th} digit = 1 if c_j has x_i

z_i: ***l***+j^{th} digit = 1 if c_j has $\overline{x_i}$

	1	2	3	4	$\cdots$	l	c_1	c_2	$\cdots$	c_k
y_1	1	0	0	0	$\cdots$	0	1	0	$\cdots$	0
z_1	1	0	0	0	$\cdots$	0	0	0	$\cdots$	0
y_2		1	0	0	$\cdots$	0	0	1	$\cdots$	0
z_2		1	0	0	$\cdots$	0	1	0	$\cdots$	0
y_3			1	0	$\cdots$	0	1	1	$\cdots$	0
z_3			1	0	$\cdots$	0	0	0	$\cdots$	1
$\vdots$					$\ddots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
y_l						1	0	0	$\cdots$	0
z_l						1	0	0	$\cdots$	0
g_1							1	0	$\cdots$	0
h_1							1	0	$\cdots$	0
g_2								1	$\cdots$	0
h_2								1	$\cdots$	0
$\vdots$									$\ddots$	$\vdots$
g_k										1
h_k										1
t	1	1	1	1	$\cdots$	1	3	3	$\cdots$	3

g_j and h_j: ***l***+j^{th} digit = 1

The sum

Theorem: *SUBSET-SUM* is NP-complete

SUBSET-SUM $= \{\langle S, t\rangle \mid S = \{x_1, \ldots, x_k\}$, and for some $\{y_1, \ldots, y_l\} \subseteq \{x_1, \ldots, x_k\}$, we have $\Sigma y_i = t\}$

THEOREM 7.36

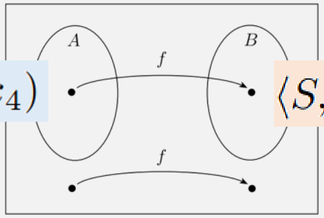
If B is NP-complete and $B \leq_{\mathrm{P}} C$ for C in NP, then C is NP-complete.

Strategy: Use

Proof Parts (5):

1. Show ~~*SUBSET-SUM* is in **NP** (done in prev class)~~
2. ~~Choose **NP**-complete problem to reduce from: *3SAT*~~
3. ~~Create the computable function *f*:~~

$(x_1 \vee \overline{x_2} \vee \overline{x_3}) \wedge (x_3 \vee \overline{x_5} \vee x_6) \wedge (x_3 \vee \overline{x_6} \vee x_4)$



$\langle S, t\rangle \mid S = \{x_1, \ldots, x_k\}$

4. Show it runs in poly time
5. Show Def 7.29 iff requirement:

ϕ is a satisfiable 3cnf-formula $\iff$ $f(\langle\phi\rangle) = \langle S, t\rangle$ where some subset of S sums to t

Polynomial Time?

E.g., $(x_1 \vee \overline{x_2} \vee x_3) \ \wedge \ (x_2 \vee x_3 \vee \cdots) \wedge \ \cdots \ \wedge (\overline{x_3} \vee \cdots \vee \cdots)$ ➡

- Assume formula has:
  - ***l*** variables $x_1, \ldots, x_l$
  - ***k*** clauses $c_1, \ldots, c_k$
- Table size: $(l + k)(2l + 2k)$
  - Creating it requires constant number of passes over the table
  - Num variables l = at most $3k$
- Total: $O(k^2)$

	1	2	3	4	$\cdots$	l	c_1	c_2	$\cdots$	c_k
y_1	1	0	0	0	$\cdots$	0	1	0	$\cdots$	0
z_1	1	0	0	0	$\cdots$	0	0	0	$\cdots$	0
y_2		1	0	0	$\cdots$	0	0	1	$\cdots$	0
z_2		1	0	0	$\cdots$	0	1	0	$\cdots$	0
y_3			1	0	$\cdots$	0	1	1	$\cdots$	0
z_3			1	0	$\cdots$	0	0	0	$\cdots$	1
$\vdots$					$\ddots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
y_l						1	0	0	$\cdots$	0
z_l						1	0	0	$\cdots$	0
g_1							1	0	$\cdots$	0
h_1							1	0	$\cdots$	0
g_2								1	$\cdots$	0
h_2								1	$\cdots$	0
$\vdots$									$\ddots$	$\vdots$
g_k										1
h_k										1
t	1	1	1	1	$\cdots$	1	3	3	$\cdots$	3

Theorem: *SUBSET-SUM* is NP-complete

SUBSET-SUM $= \{\langle S, t\rangle \mid S = \{x_1, \ldots, x_k\},$ and for some $\{y_1, \ldots, y_l\} \subseteq \{x_1, \ldots, x_k\},$ we have $\Sigma y_i = t\}$

Strategy: Use

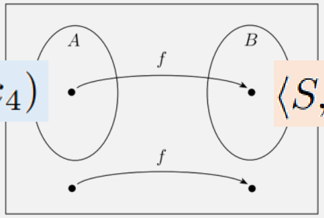
THEOREM 7.36

If B is NP-complete and $B \leq_{\mathrm{P}} C$ for C in NP, then C is NP-complete.

Proof Parts (5):

1. Show ~~*SUBSET-SUM* is in **NP** (done in prev class)~~
2. ~~Choose **NP**-complete problem to reduce from: *3SAT*~~
3. ~~Create the computable function *f*:~~

$(x_1 \vee \overline{x_2} \vee \overline{x_3}) \wedge (x_3 \vee \overline{x_5} \vee x_6) \wedge (x_3 \vee \overline{x_6} \vee x_4)$



$\langle S, t\rangle \mid S = \{x_1, \ldots, x_k\}$

4. ~~Show it runs in poly time~~
5. Show Def 7.29 iff requirement:

ϕ is a satisfiable 3cnf-formula $\iff$ $f(\langle\phi\rangle) = \langle S, t\rangle$ where some subset of S sums to t

ϕ is a satisfiable 3cnf-formula $\iff$ $f(\langle\phi\rangle) = \langle S, t\rangle$ where some subset of S sums to t

=> **If** formula is satisfiable ...

- Sum t = l 1s followed by k 3s
- Choose subset of S to include:
  - y_i if x_i = TRUE
  - z_i if x_i = FALSE
  - and some of g_i and h_i to make the sum t
- ... **Then** this subset of S must sum to t bc:
  - Left digits:
    - only one of y_i or z_i is in S
  - Right digits:
    - Top right: Each column sums to 1, 2, or 3
      - Because each clause has 3 literals
    - Bottom right:
      - Can always use g_i and/or h_i to make column sum to 3

Each column:
- At least one 1
- At most 3 1s

S only includes one

	1	2	3	4	$\cdots$	l	c_1	c_2	$\cdots$	c_k
y_1	1	0	0	0	$\cdots$	0	1	0	$\cdots$	0
z_1	1	0	0	0	$\cdots$	0	0	0	$\cdots$	0
y_2		1	0	0	$\cdots$	0	0	1	$\cdots$	0
z_2		1	0	0	$\cdots$	0	1	0	$\cdots$	0
y_3			1	0	$\cdots$	0	1	1	$\cdots$	0
z_3			1	0	$\cdots$	0	0	0	$\cdots$	1
$\vdots$					$\ddots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
y_l						1	0	0	$\cdots$	0
z_l						1	0	0	$\cdots$	0
g_1							1	0	$\cdots$	0
h_1							1	0	$\cdots$	0
g_2								1	$\cdots$	0
h_2								1	$\cdots$	0
$\vdots$									$\ddots$	$\vdots$
g_k										1
h_k										1
t	1	1	1	1	$\cdots$	1	3	3	$\cdots$	3

So each column sum (for left digits) is 1

ϕ is a satisfiable 3cnf-formula $\iff$ $f(\langle\phi\rangle) = \langle S, t\rangle$ where some sub o t

Subset must have some number with 1 in each right column

<= **If** a subset of S sums to t …

- It can only include either y_i or z_i
  - Because each left digit column must sum to
  - And no carrying is possible
- Also, since each right digit column must sum to 3:
  - And only 2 can come from g_i and h_i
  - Then for every right column, some y_i or z_i in the subset has a 1 in that column
- … **Then** a satisfying assignment is:
  - x_i = TRUE if y_i in the subset
  - x_i = FALSE if z_i in the subset
- This is satisfying because:
  - Table was constructed so 1 in column c_j for y_i or z_i means that variable x_i satisfies clause c_j
  - We already determined, for every right column, some number in the subset has a 1 in the column
  - So all clauses are satisfied

S only includes one

	1	2	3	4	$\cdots$	l	c_1	c_2	$\cdots$	c_k
y_1	1	0	0	0	$\cdots$	0	1	0	$\cdots$	0
z_1	1	0	0	0	$\cdots$	0	0	0	$\cdots$	0
y_2		1	0	0	$\cdots$	0	0	1	$\cdots$	0
z_2		1	0	0	$\cdots$	0	1	0	$\cdots$	0
y_3			1	0	$\cdots$	0	1	1	$\cdots$	0
z_3			1	0	$\cdots$	0	0	0	$\cdots$	1
$\vdots$					$\ddots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
						1	0	0	$\cdots$	0
						1	0	0	$\cdots$	0
							1	0	$\cdots$	0
							1	0	$\cdots$	0
								1	$\cdots$	0
h_2								1	$\cdots$	0
$\vdots$									$\ddots$	$\vdots$
g_k										1
h_k										1
t	1	1	1	1	$\cdots$	1	3	3	$\cdots$	3

In each right column, g_i and h_i can account for at most 2

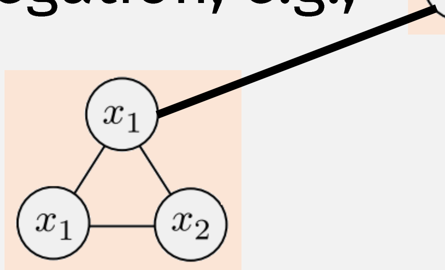
Because each column sum (for left digits) is 1

Theorem: *VERTEX-COVER* is NP-complete.

$VERTEX\text{-}COVER = \{\langle G, k\rangle \mid G$ is an undirected graph that has a k-node vertex cover$\}$

- A vertex cover of a graph is …
  - … a subset of its nodes where every edge touches one of those nodes
- Proof Sketch: Reduce *3SAT* to *VERTEX-COVER*
- The reduction maps:
- Variable x_i → 2 connected nodes
  - corresponding to the var and its negation, e.g.,

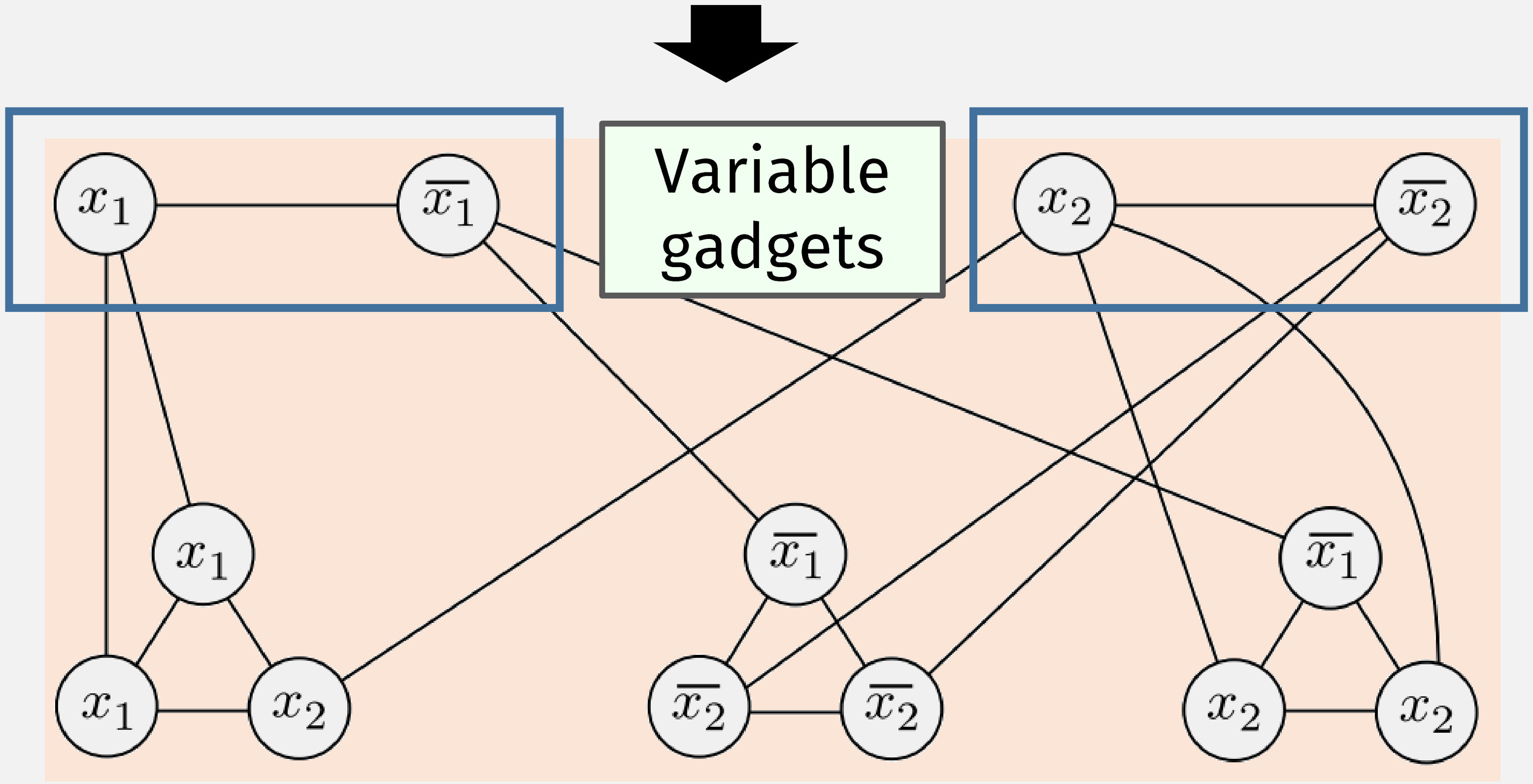
- Clause → 3 connected nodes
  - corresponding to its literals, e.g.,



- Additionally,
  - connect var and clause gadgets by …
  - … connecting nodes that correspond to the same literal

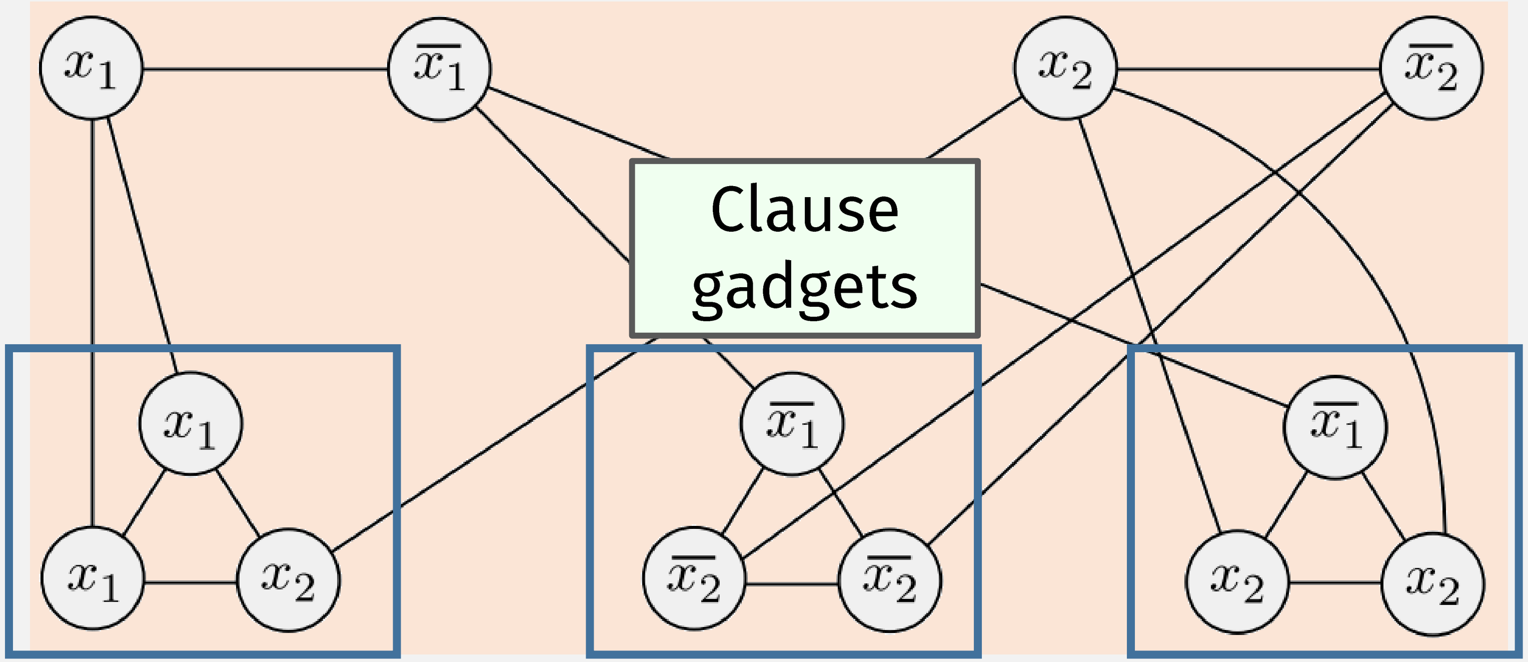
VERTEX-COVER example

$$\phi = (x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee x_2)$$



VERTEX-COVER example

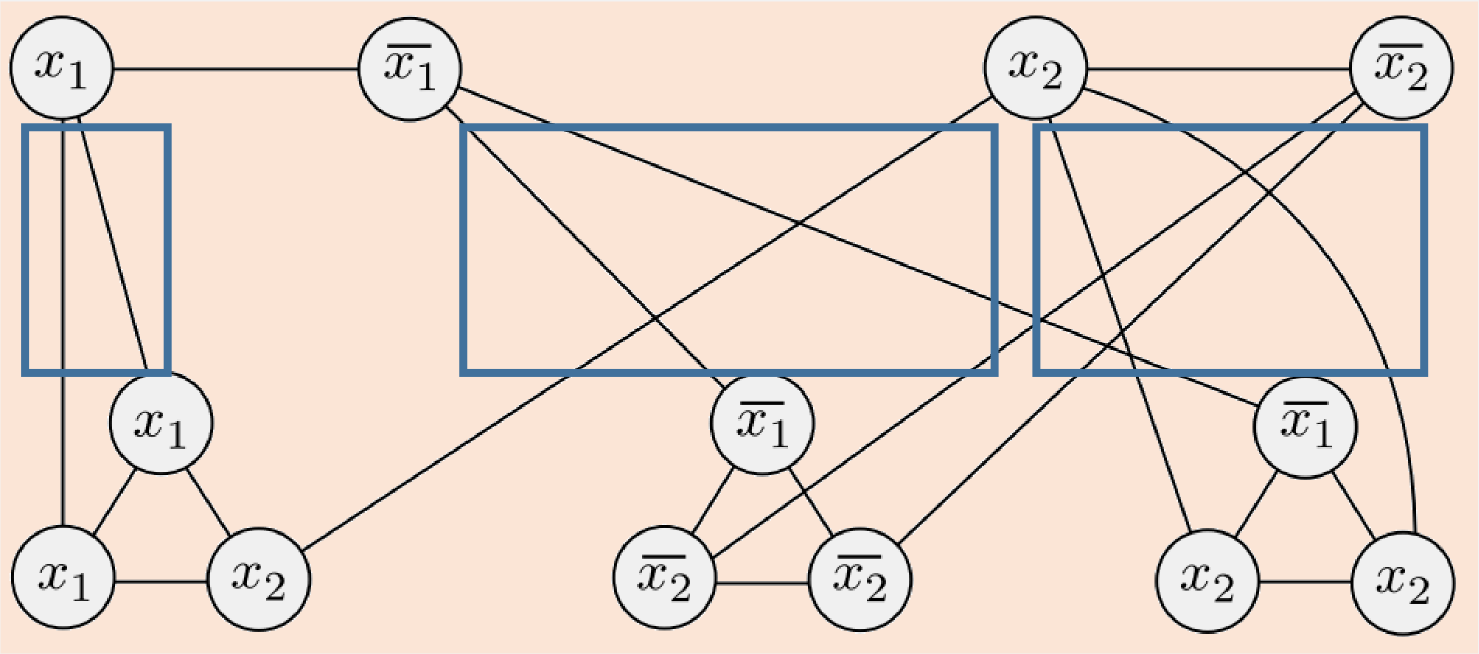
$$\phi = (x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee x_2)$$



VERTEX-COVER example

$$\phi = (x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee x_2)$$

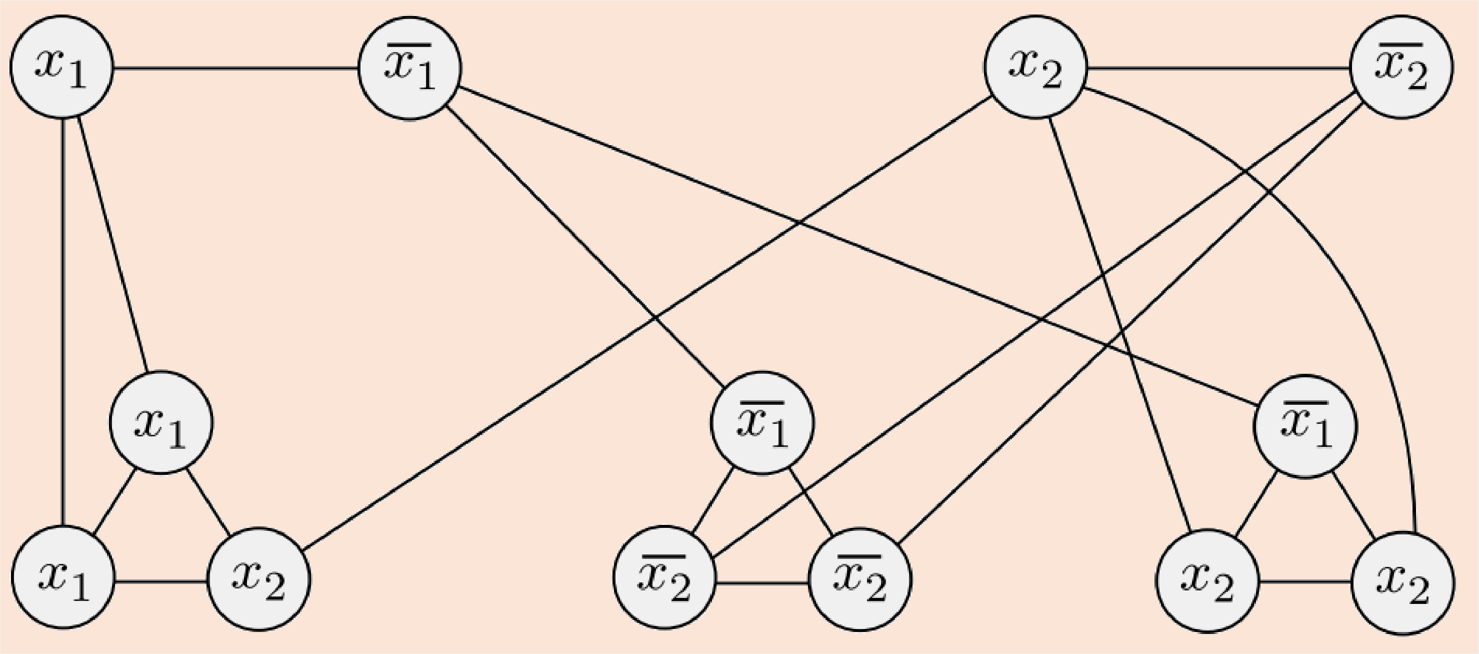
Extra edges connecting variable and clause gadgets together



VERTEX-COVER example

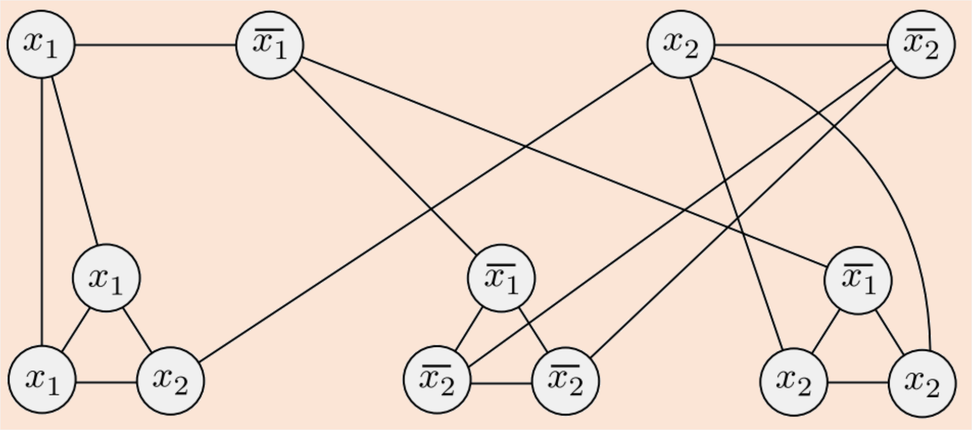
$$\phi = (x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee x_2)$$

Exercise:
Show that formula is satisfiable
⇔
Graph has a vertex cover with k nodes



$$\phi = (x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee x_2)$$

VERTEX-COVER example



- Let formula have …
  - m = # variables
  - l = # clauses
- Then graph has …
  - # nodes = $2m + 3l$

=> If satisfying assignment,

  - then show there is a k-cover where $k = m + 2l$

- Nodes in the cover:
  - In each of m var gadgets, choose 1 node corresponding to TRUE literal
  - For each of l clause gadgets, ignore 1 TRUE literal and choose other 2
  - Since there is satisfying assignment, each clause has a TRUE literal
  - Total = $m + 2l$

NO Quiz 12/9!

NO Survey 12/9!

Fill out course evaluation.