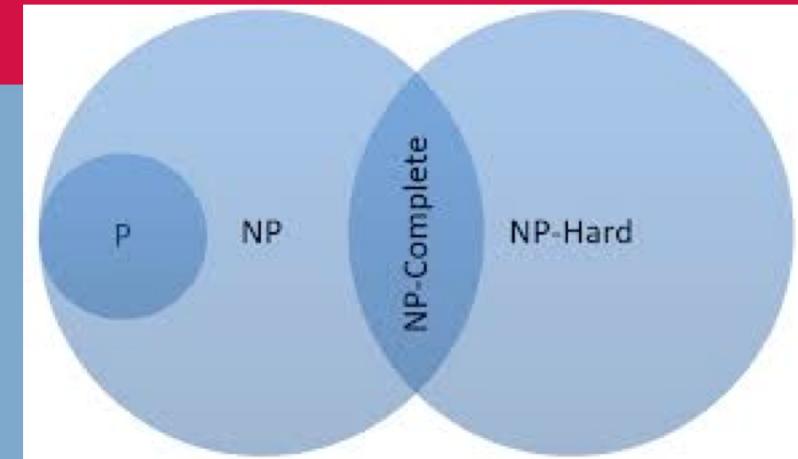




Metaheuristic Optimization

NP Complexity

Dr. Diarmuid Grimes





The First NP-Complete Problem: Satisfiability - SAT

First NP-Complete Problem



How do we get started? Need to show via brute force that some problem is NP-complete.

- Logic problem "satisfiability" (or SAT).
- Given a Boolean expression (collection of Boolean variables connected with ANDs and ORs), is it satisfiable, i.e., is there a way to assign truth values to the variables so that the expression evaluates to TRUE?

SAT is NP-Complete



- Cook's Theorem: SAT is NP-complete.
- Proof ideas 2 parts: one relatively easy, one relatively hard!
- SAT is in NP: Given a candidate solution (a truth assignment) for a CNF formula, verify in polynomial time (by plugging in the truth values and evaluating the expression) whether it satisfies the formula (makes it true).
- More complicated: show that any problem in NP can be reduced to an instance of the SAT problem, i.e. that SAT is NP-Hard

SAT is NP-Complete



- How to show that every language in NP is polynomially reducible to SAT?
- Key idea: the common thread among all the languages in NP is that each one is solved by some nondeterministic Turing machine (a formal model of computation) in polynomial time.
- Given a description of a poly time TM, construct in poly time, a CNF formula that simulates the computation of the TM.

Proving NP-Completeness By Reduction



To show L is NP-complete:

- (1) Show L is in NP.
- (2.a) Choose an appropriate known NP-complete language L' .
- (2.b) Show $L' \leq_p L$: Describe an algorithm to compute a function f such that
 - f is poly time
 - f maps inputs for L' to inputs for L s.t. x is in L' if and only if $f(x)$ is in L

Get the Direction Right!



- We want to show that L is at least as hard (time-consuming) as L' .
- So if we have an algorithm A for L , then we can solve L' with polynomial overhead
- Algorithm for L' :

input: x

compute $y = f(x)$

run algorithm A for L on y

return whatever A returns

$$L' \leq_p L$$

known unknown

A diagram illustrating a many-one reduction $L' \leq_p L$. It features two green arrows pointing upwards from the words "known" and "unknown" to the symbols "L'" and "L" respectively in the inequality expression.



3SAT is NP-Complete

Definition of 3SAT



- 3SAT is a special case of SAT: each clause contains exactly 3 literals.
- Is 3SAT in NP?
 - Yes, because SAT is in NP.
- Is 3SAT NP-complete?
 - Not obvious. It has a more regular structure, which can perhaps be exploited to get an efficient algorithm
 - In fact, 2SAT does have a polynomial time algorithm



Showing 3SAT is NP-Complete by Reduction

- (1) To show 3SAT is in NP, use same algorithm as for SAT to verify a candidate solution (truth assignment)
- (2.a) Choose SAT as known NP-complete problem.
- (2.b) Describe a reduction from SAT inputs to 3SAT inputs
 - Computable in polynomial time
 - SAT input is satisfiable iff constructed 3SAT input is satisfiable
 - Every instance of SAT can be represented as 3SAT such that
 - If the SAT instance is unsatisfiable then the 3SAT instance is unsatisfiable,
 - Else, every solution for the SAT instance must also be a solution for the 3SAT instance

Reduction from SAT to 3SAT

- We're given an arbitrary CNF formula $C = c_1 \wedge c_2 \wedge \dots \wedge c_m$ over set of variables U
 - Each c_i is a clause (disjunction of literals)
- We will replace **each clause c_i** with a **set of clauses C'_i** , and may use some **extra variables U'_i** just for this clause
- Each clause in C'_i will have **exactly 3 literals**
- Transformed input will be **conjunction** of all the clauses in all the C'_i
- New clauses are carefully chosen...

Reduction from SAT to 3SAT: Possible Clause lengths

Let $c_i = z_1 \vee z_2 \vee \dots \vee z_k$

- Case 1: $k = 1$.
 - Use extra variables y_i^1 and y_i^2 .
 - Replace c_i with 4 clauses:

$$(z_1 \vee y_i^1 \vee y_i^2)$$

$$(z_1 \vee \overline{y_i^1} \vee y_i^2)$$

$$(z_1 \vee y_i^1 \vee \overline{y_i^2})$$

$$(z_1 \vee \overline{y_i^1} \vee \overline{y_i^2})$$

Any solution for the new problem must be a solution ($k=3$) for the original problem ($k=1$)

Reduction from SAT to 3SAT

Let $c_i = z_1 \vee z_2 \vee \dots \vee z_k$

- Case 1: $k = 1$.
- Use extra variables y_i^1 and y_i^2 .
- Replace c_i with 4 clauses:

$$(z_1 \vee y_i^1 \vee y_i^2)$$

$$(z_1 \vee \overline{y_i^1} \vee y_i^2)$$

$$(z_1 \vee y_i^1 \vee \overline{y_i^2})$$

$$(z_1 \vee \overline{y_i^1} \vee \overline{y_i^2})$$

Table with all possible positions

z_1	y_1	y_2	$\neg y_1$	$\neg y_2$	C^1	C^2	C^3	C^4	F
1	1	1							
1	1	0							
1	0	1							
1	0	0							
0	1	1							
0	1	0							
0	0	1							
0	0	0							

Reduction from SAT to 3SAT

Let $c_i = z_1 \vee z_2 \vee \dots \vee z_k$

- Case 1: $k = 1$.
- Use extra variables y_i^1 and y_i^2 .
- Replace c_i with 4 clauses:

$$(z_1 \vee y_i^1 \vee y_i^2)$$

$$(z_1 \vee \overline{y_i^1} \vee y_i^2)$$

$$(z_1 \vee y_i^1 \vee \overline{y_i^2})$$

$$(z_1 \vee \overline{y_i^1} \vee \overline{y_i^2})$$

Table with all possible positions

z_1	y_1	y_2	$\neg y_1$	$\neg y_2$	C^1	C^2	C^3	C^4	F
1	1	1	0						
1	1	0	0						
1	0	1	1						
1	0	0	1						
0	1	1	0						
0	1	0	0						
0	0	1	1						
0	0	0	1						

Reduction from SAT to 3SAT

Let $c_i = z_1 \vee z_2 \vee \dots \vee z_k$

- Case 1: $k = 1$.
 - Use extra variables y_i^1 and y_i^2 .
 - Replace c_i with 4 clauses:

$$(z_1 \vee y_i^1 \vee y_i^2)$$

$$(z_1 \vee \overline{y_i^1} \vee y_i^2)$$

$$(z_1 \vee y_i^1 \vee \overline{y_i^2})$$

$$(z_1 \vee \overline{y_i^1} \vee \overline{y_i^2})$$

Table with all possible positions

z_1	y_1	y_2	$\neg y_1$	$\neg y_2$	C^1	C^2	C^3	C^4	F
1	1	1	0	0					
1	1	0	0	1					
1	0	1	1	0					
1	0	0	1	1					
0	1	1	0	0					
0	1	0	0	1					
0	0	1	1	0					
0	0	0	1	1					

Reduction from SAT to 3SAT

Let $c_i = z_1 \vee z_2 \vee \dots \vee z_k$

- Case 1: $k = 1$.
 - Use extra variables y_i^1 and y_i^2 .
 - Replace c_i with 4 clauses:

$$\begin{aligned} & (z_1 \vee y_i^1 \vee y_i^2) \\ & (z_1 \vee \overline{y_i^1} \vee y_i^2) \\ & (z_1 \vee y_i^1 \vee \overline{y_i^2}) \\ & (z_1 \vee \overline{y_i^1} \vee \overline{y_i^2}) \end{aligned}$$

Table with all possible positions

z_1	y_1^1	y_1^2	$\neg y_1^1$	$\neg y_1^2$	C_1	C_2	C_3	C_4	F
1	1	1	0	0	1	1	1	1	1
1	1	0	0	1	1	1	1	1	1
1	0	1	1	0	1	1	1	1	1
1	0	0	1	1	1	1	1	1	1
0	1	1	0	0					
0	1	0	0	1					
0	0	1	1	0					
0	0	0	1	1					

Reduction from SAT to 3SAT

Let $c_i = z_1 \vee z_2 \vee \dots \vee z_k$

- Case 1: $k = 1$.
- Use extra variables y_i^1 and y_i^2 .
- Replace c_i with 4 clauses:

$$(z_1 \vee y_i^1 \vee y_i^2)$$

$$(z_1 \vee \overline{y_i^1} \vee y_i^2)$$

$$(z_1 \vee y_i^1 \vee \overline{y_i^2})$$

$$(z_1 \vee \overline{y_i^1} \vee \overline{y_i^2})$$

Table with all possible positions

z_1	y_1	y_2	$\neg y_1$	$\neg y_2$	C^1	C^2	C^3	C^4	F
1	1	1	0	0	1	1	1	1	1
1	1	0	0	1	1	1	1	1	1
1	0	1	1	0	1	1	1	1	1
1	0	0	1	1	1	1	1	1	1
0	1	1	0	0	1				
0	1	0	0	1	1				
0	0	1	1	0	1				
0	0	0	1	1	0				

Reduction from SAT to 3SAT

Let $c_i = z_1 \vee z_2 \vee \dots \vee z_k$

- Case 1: $k = 1$.
- Use extra variables y_i^1 and y_i^2 .
- Replace c_i with 4 clauses:

$$(z_1 \vee y_i^1 \vee y_i^2)$$

$$(z_1 \vee \overline{y_i^1} \vee y_i^2)$$

$$(z_1 \vee y_i^1 \vee \overline{y_i^2})$$

$$(z_1 \vee \overline{y_i^1} \vee \overline{y_i^2})$$

Table with all possible positions

z_1	y_1	y_2	$\neg y_1$	$\neg y_2$	C^1	C^2	C^3	C^4	F
1	1	1	0	0	1	1	1	1	1
1	1	0	0	1	1	1	1	1	1
1	0	1	1	0	1	1	1	1	1
1	0	0	1	1	1	1	1	1	1
0	1	1	0	0	1	1			
0	1	0	0	1	1	0			
0	0	1	1	0	1	1			
0	0	0	1	1	0	1			

Reduction from SAT to 3SAT

Let $c_i = z_1 \vee z_2 \vee \dots \vee z_k$

- Case 1: $k = 1$.
- Use extra variables y_i^1 and y_i^2 .
- Replace c_i with 4 clauses:

$$(z_1 \vee y_i^1 \vee y_i^2)$$

$$(z_1 \vee \overline{y_i^1} \vee y_i^2)$$

$$(z_1 \vee y_i^1 \vee \overline{y_i^2})$$

$$(z_1 \vee \overline{y_i^1} \vee \overline{y_i^2})$$

Table with all possible positions

z_1	y_1	y_2	$\neg y_1$	$\neg y_2$	C^1	C^2	C^3	C^4	F
1	1	1	0	0	1	1	1	1	1
1	1	0	0	1	1	1	1	1	1
1	0	1	1	0	1	1	1	1	1
1	0	0	1	1	1	1	1	1	1
0	1	1	0	0	1	1	1		
0	1	0	0	1	1	0	1		
0	0	1	1	0	1	1	0		
0	0	0	1	1	0	1	1		

Reduction from SAT to 3SAT

Let $c_i = z_1 \vee z_2 \vee \dots \vee z_k$

- Case 1: $k = 1$.
- Use extra variables y_i^1 and y_i^2 .
- Replace c_i with 4 clauses:

$$(z_1 \vee y_i^1 \vee y_i^2)$$

$$(z_1 \vee \overline{y_i^1} \vee y_i^2)$$

$$(z_1 \vee y_i^1 \vee \overline{y_i^2})$$

$$(z_1 \vee \overline{y_i^1} \vee \overline{y_i^2})$$

Table with all possible positions

z_1	y_1	y_2	$\neg y_1$	$\neg y_2$	C^1	C^2	C^3	C^4	F
1	1	1	0	0	1	1	1	1	1
1	1	0	0	1	1	1	1	1	1
1	0	1	1	0	1	1	1	1	1
1	0	0	1	1	1	1	1	1	1
0	1	1	0	0	1	1	1	0	
0	1	0	0	1	1	0	1	1	
0	0	1	1	0	1	1	0	1	
0	0	0	1	1	0	1	1	1	

Reduction from SAT to 3SAT

Let $c_i = z_1 \vee z_2 \vee \dots \vee z_k$

- Case 1: $k = 1$.
 - Use extra variables y_i^1 and y_i^2 .
 - Replace c_i with 4 clauses:

$$(z_1 \vee y_i^1 \vee y_i^2)$$

$$(z_1 \vee \overline{y_i^1} \vee y_i^2)$$

$$(z_1 \vee y_i^1 \vee \overline{y_i^2})$$

$$(z_1 \vee \overline{y_i^1} \vee \overline{y_i^2})$$

Table with all possible positions

z_1	y_1	y_2	$\neg y_1$	$\neg y_2$	C^1	C^2	C^3	C^4	F
1	1	1	0	0	1	1	1	1	1
1	1	0	0	1	1	1	1	1	1
1	0	1	1	0	1	1	1	1	1
1	0	0	1	1	1	1	1	1	1
0	1	1	0	0	1	1	1	0	0
0	1	0	0	1	1	0	1	1	0
0	0	1	1	0	1	1	0	1	0
0	0	0	1	1	0	1	1	1	0

Reduction from SAT to 3SAT

Let $c_i = z_1 \vee z_2 \vee \dots \vee z_k$

- Case 2: $k = 2$.
 - Use extra variable y_i^1 .
 - Replace c_i with 2 clauses:

$$(z_1 \vee z_2 \vee y_i^1)$$

$$(z_1 \vee z_2 \vee \overline{y_i^1})$$

1 Extra Variable
And
2 Clauses

Proof

Truth Table with all possible positions

Reduction from SAT to 3SAT

Let $c_i = z_1 \vee z_2 \vee \dots \vee z_k$

- Case 3: $k = 3$.
 - No extra variables are needed.
 - Keep c_i :

$$(z_1 \vee z_2 \vee z_3)$$

Reduction from SAT to 3SAT

Let $c_i = z_1 \vee z_2 \vee \dots \vee z_k$

- Case 4: $k > 3$.
 - Use extra variables y_i^1, \dots, y_i^{k-3} .
 - Replace c_i with $k-2$ clauses:

$$(z_1 \vee z_2 \vee y_i^1)$$

$$(\overline{y_i^1} \vee z_3 \vee y_i^2)$$

$$(\overline{y_i^2} \vee z_4 \vee y_i^3)$$

...

...

$$(\overline{y_i^{k-5}} \vee z_{k-3} \vee y_i^{k-4})$$

$$(\overline{y_i^{k-4}} \vee z_{k-2} \vee y_i^{k-3})$$

$$(\overline{y_i^{k-3}} \vee z_{k-1} \vee z_k)$$

k-3 Extra Variable
And
K-2 Clauses

Reduction from SAT to 3SAT



Let $c_i = z_1 \vee z_2 \vee \dots \vee z_k$

- Case 4: e.g. $k = 4$.

$$\begin{aligned} & (z_1 \vee z_2 \vee y_i^1) \\ & (\overline{y_i^1} \vee z_3 \vee z_4) \end{aligned}$$

For each clause ($k=4$)

1 Extra Variable
And
2 Clauses

Correctness of Reduction



- Show that CNF formula C is satisfiable iff the 3-CNF formula C' constructed is satisfiable.
- \Rightarrow : Suppose C is satisfiable. Come up with a satisfying truth assignment for C' .
- For variables in U , use same truth assignments as for C .
- How to assign T/F to the new variables?

Correctness of Reduction



- $\leq:$ Suppose the newly constructed 3SAT formula C' is satisfiable. We must show that the original SAT formula C is also satisfiable.
- Use the same satisfying truth assignment for C as for C' (ignoring new variables).
- Show each original clause has at least one true literal in it.

Original Clause Has a True Literal

Let $c_i = z_1 \vee z_2 \vee \dots \vee z_k$

- Case 1: $k = 1$.
 - Use extra variables y_i^1 and y_i^2 .
 - Replace c_i with 4 clauses:

$$(z_1 \vee y_i^1 \vee y_i^2)$$

$$(z_1 \vee \overline{y_i^1} \vee y_i^2)$$

$$(z_1 \vee y_i^1 \vee \overline{y_i^2})$$

$$(z_1 \vee \overline{y_i^1} \vee \overline{y_i^2})$$

For every assignment of y_i^1 and y_i^2 , in order for all 4 clauses to have a true literal, z_1 must be true.

Truth Assignment for New Variables

Let $c_i = z_1 \vee z_2 \vee \dots \vee z_k$

- Case 2: $k = 2$.
 - Use extra variable y_i^1 .
 - Replace c_i with 2 clauses:

$$(z_1 \vee z_2 \vee y_i^1)$$

$$(z_1 \vee z_2 \vee \overline{y_i^1})$$

Since either z_1 or z_2 is true, it does not matter how we assign y_i^1

For either assignment of y_i^1 , in order for both clauses to have a true literal, z_1 or z_2 must be true.



Truth Assignment for New Variables

Let $c_i = z_1 \vee z_2 \vee \dots \vee z_k$

- Case 3: $k = 3$.
 - No extra variables are needed.
 - Keep c_i :

$$(z_1 \vee z_2 \vee z_3)$$

No new variables.

Truth Assignment for New Variables

CIT

Let $c_i = z_1 \vee z_2 \vee \dots \vee z_k$

- Case 4: $k > 3$.
 - Use extra variables y_i^1, \dots, y_i^{k-3} .
 - Replace c_i with $k-2$ clauses:

$$(z_1 \vee z_2 \vee y_i^1)$$

$$(\overline{y_i^1} \vee z_3 \vee y_i^2)$$

$$(\overline{y_i^2} \vee z_4 \vee y_i^3)$$

...

...

$$(\overline{y_i^{k-5}} \vee z_{k-3} \vee y_i^{k-4})$$

$$(\overline{y_i^{k-4}} \vee z_{k-2} \vee y_i^{k-3})$$

$$(\overline{y_i^{k-3}} \vee z_{k-1} \vee z_k)$$

If first true literal is z_1 or z_2 , set all y_i 's to false: then all later clauses have a true literal

Truth Assignment for New Variables

Let $c_i = z_1 \vee z_2 \vee \dots \vee z_k$

- Case 4: $k > 3$.
 - Use extra variables y_i^1, \dots, y_i^{k-3} .
 - Replace c_i with $k-2$ clauses:

$$(z_1 \vee z_2 \vee y_i^1)$$

$$(\overline{y_i^1} \vee z_3 \vee y_i^2)$$

$$(\overline{y_i^2} \vee z_4 \vee y_i^3)$$

...

...

$$(\overline{y_i^{k-5}} \vee z_{k-3} \vee y_i^{k-4})$$

$$(\overline{y_i^{k-4}} \vee z_{k-2} \vee y_i^{k-3})$$

$$(\overline{y_i^{k-3}} \vee z_{k-1} \vee z_k)$$

If first true literal is z_{k-1} or z_k , set all y_i 's to true: then all earlier clauses have a true literal

Truth Assignment for New Variables

CIT

Let $c_i = z_1 \vee z_2 \vee \dots \vee z_k$

- Case 4: $k > 3$.
 - Use extra variables y_i^1, \dots, y_i^{k-3} .
 - Replace c_i with $k-2$ clauses:

$$(z_1 \vee z_2 \vee y_i^1)$$

...

$$(\overline{y_i^1} \vee z_3 \vee y_i^2)$$

$$(\overline{y_i^{k-5}} \vee z_{k-3} \vee y_i^{k-4})$$

$$(\overline{y_i^2} \vee z_4 \vee y_i^3)$$

$$(\overline{y_i^{k-4}} \vee z_{k-2} \vee y_i^{k-3})$$

...

$$(\overline{y_i^{k-3}} \vee z_{k-1} \vee z_k)$$

If first true literal is in between, set all earlier y_i 's to true and all later y_i 's to false

Original Clause Has a True Literal

Let $c_i = z_1 \vee z_2 \vee \dots \vee z_k$

- Case 4: e.g. $k = 4$.

$$\begin{aligned} & (z_1 \vee z_2 \vee y_i^1) \\ & (\overline{y_i^1} \vee z_3 \vee z_4) \end{aligned}$$

For instance, $Z1 = \text{True}$ (the first clause will be true, regardless the value of Y^1)
And we can say that $(\neg Y^1) = \text{True}$ and the second clause will be true



Suppose in contradiction Z_1, Z_2, Z_3, Z_4 are false.

Then Y^1 must be true, but the second clause will be false $(\neg Y^1)$

If at least one $(Z1, Z2, Z3, Z4)$ is True.
Then we could say Y^1 or $\neg Y^1$ will evaluate to true

Original Clause Has a True Literal

Let $c_i = z_1 \vee z_2 \vee \dots \vee z_k$

- Case 4: e.g. $k=5$

$$(z_1 \vee z_2 \vee y_i^1)$$

$$(\overline{y_i^1} \vee z_3 \vee y_i^2)$$

$$(\overline{y_i^2} \vee z_4 \vee z_5)$$

For each clause ($k=5$)

2 Extra Variables
And
3 Clauses

Original Clause Has a True Literal

Let $c_i = z_1 \vee z_2 \vee \dots \vee z_k$

- Case 4: e.g. $k=5$

$$(z_1 \vee z_2 \vee y_i^1)$$

$$(\overline{y_i^1} \vee z_3 \vee y_i^2)$$

$$(\overline{y_i^2} \vee z_4 \vee z_5)$$

Same argument as before
($k=4$)

Suppose in contradiction $Z_1, Z_2,$
and Z_3, Z_4 are false.

Then Y^1 must be true, but the
second clause will be false ($\neg Y^1$)

If at least one (Z_1, Z_2, Z_3, Z_4, Z_5)
is True.

Then we could say Y^1 or $\neg Y^1$ will
evaluate to true

Original Clause Has a True Literal

Let $c_i = z_1 \vee z_2 \vee \dots \vee z_k$

- Case 4: $k > 3$.
 - Use extra variables y_i^1, \dots, y_i^{k-3} .
 - Replace c_i with $k-2$ clauses:

$$(z_1 \vee z_2 \vee y_i^1)$$

...

$$(\overline{y_i^1} \vee z_3 \vee y_i^2)$$

$$(\overline{y_i^{k-5}} \vee z_{k-3} \vee y_i^{k-4})$$

$$(\overline{y_i^2} \vee z_4 \vee y_i^3)$$

$$(\overline{y_i^{k-4}} \vee z_{k-2} \vee y_i^{k-3})$$

...

$$(\overline{y_i^{k-3}} \vee z_{k-1} \vee z_k)$$

Suppose in contradiction all z_i 's are false.

k-3 Extra Variable
And
K-2 Clauses

Original Clause Has a True Literal

Let $c_i = z_1 \vee z_2 \vee \dots \vee z_k$

- Case 4: $k > 3$.
 - Use extra variables y_i^1, \dots, y_i^{k-3} .
 - Replace c_i with $k-2$ clauses:

$$(z_1 \vee z_2 \vee y_i^1)$$

...

$$(\overline{y_i^1} \vee z_3 \vee y_i^2)$$

$$(\overline{y_i^{k-5}} \vee z_{k-3} \vee y_i^{k-4})$$

$$(\overline{y_i^2} \vee z_4 \vee y_i^3)$$

$$(\overline{y_i^{k-4}} \vee z_{k-2} \vee y_i^{k-3})$$

...

$$(\overline{y_i^{k-3}} \vee z_{k-1} \vee z_k)$$

Suppose in contradiction all z_i 's are false. Then y_i^1 must be true,

Original Clause Has a True Literal

Let $c_i = z_1 \vee z_2 \vee \dots \vee z_k$

- Case 4: $k > 3$.
 - Use extra variables y_i^1, \dots, y_i^{k-3} .
 - Replace c_i with $k-2$ clauses:

$$(z_1 \vee z_2 \vee y_i^1)$$

...

$$(\overline{y_i^1} \vee z_3 \vee y_i^2)$$

$$(\overline{y_i^2} \vee z_{k-3} \vee y_i^{k-4})$$

$$(\overline{y_i^3} \vee z_4 \vee y_i^3)$$

$$(\overline{y_i^{k-4}} \vee z_{k-2} \vee y_i^{k-3})$$

...

$$(\overline{y_i^{k-3}} \vee z_{k-1} \vee z_k)$$

Suppose in contradiction all z_i 's are false. Then y_i^1 must be true, so y_i^2 must be true...

Original Clause Has a True Literal

Let $c_i = z_1 \vee z_2 \vee \dots \vee z_k$

- Case 4: $k > 3$.
 - Use extra variables y_i^1, \dots, y_i^{k-3} .
 - Replace c_i with $k-2$ clauses:

$$(z_1 \vee z_2 \vee y_i^1)$$

$$(\overline{y_i^1} \vee z_3 \vee y_i^2)$$

$$(\overline{y_i^2} \vee z_4 \vee y_i^3)$$

...

...

$$(\overline{y_i^{k-5}} \vee z_{k-3} \vee y_i^{k-4})$$

$$(\overline{y_i^{k-4}} \vee z_{k-2} \vee y_i^{k-3})$$

$$(\overline{y_i^{k-3}} \vee z_{k-1} \vee z_k)$$

Suppose in contradiction all z_i 's are false. Then y_i^1 must be true, so y_i^2 must be true...

Original Clause Has a True Literal

Let $c_i = z_1 \vee z_2 \vee \dots \vee z_k$

- Case 4: $k > 3$.
 - Use extra variables y_i^1, \dots, y_i^{k-3} .
 - Replace c_i with $k-2$ clauses:

$$(z_1 \vee z_2 \vee y_i^1)$$

$$(\overline{y_i^1} \vee z_3 \vee y_i^2)$$

$$(\overline{y_i^2} \vee z_4 \vee y_i^3)$$

...

...

$$(\overline{y_i^{k-5}} \vee z_{k-3} \vee y_i^{k-4})$$

$$(\overline{y_i^{k-4}} \vee z_{k-2} \vee y_i^{k-3})$$

$$(\overline{y_i^{k-3}} \vee z_{k-1} \vee z_k)$$

Suppose in contradiction all z_i 's are false. Then y_i^1 must be true, so y_i^2 must be true...

Original Clause Has a True Literal

Let $c_i = z_1 \vee z_2 \vee \dots \vee z_k$

- Case 4: $k > 3$.
 - Use extra variables y_i^1, \dots, y_i^{k-3} .
 - Replace c_i with $k-2$ clauses:

$$(z_1 \vee z_2 \vee y_i^1)$$

$$(\overline{y_i^1} \vee z_3 \vee y_i^2)$$

$$(y_i^2 \vee z_4 \vee y_i^3)$$

...

...

$$(\overline{y_i^{k-5}} \vee z_{k-3} \vee y_i^{k-4})$$

$$(\overline{y_i^{k-4}} \vee z_{k-2} \vee y_i^{k-3})$$

$$(\overline{y_i^{k-3}} \vee z_{k-1} \vee z_k)$$

Suppose in contradiction all z_i 's are false. Then y_i^1 must be true, ... so last clause is false.



Why is Reduction Poly Time?

- The running time of the reduction (the algorithm to compute the 3SAT formula C' , given the SAT formula C) is proportional to the size of C'
- Rules for constructing C' are simple to calculate

1. original clause with 1 literal becomes 4 clauses with 3 literals each
 2. original clause with 2 literals becomes 2 clauses with 3 literals each
 3. original clause with 3 literals becomes 1 clause with 3 literals
 4. original clause with $k > 3$ literals becomes $k-2$ clauses with 3 literals each
-
- So new formula is only a constant factor larger than the original formula

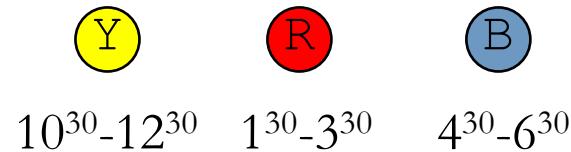


Vertex Coloring is NP-Complete

Coloring

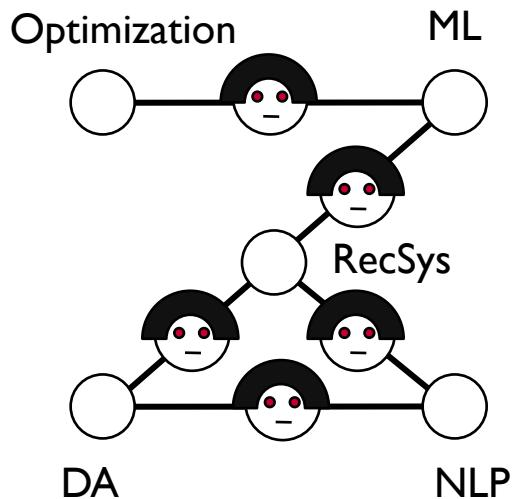


Suppose you need to schedule exams on same day in **3 time slots**



Some people are enrolled in **multiple classes**, so there may be **conflicts**

Task: Schedule the exams so there are no conflicts

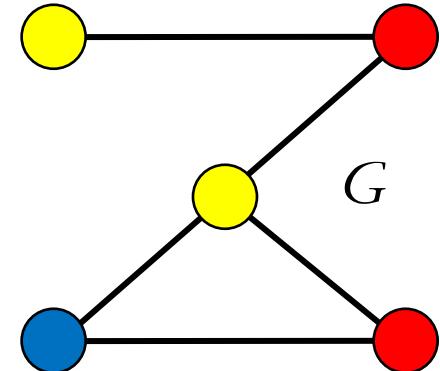


Coloring



Y R B

This is called a **valid 3-coloring**



$3\text{COL} = \{\langle G \rangle : G \text{ has a valid 3-coloring}\}$

3COL is NP-complete

Coloring

CIT

- Step 1: 3COL is in NP

What is a **solution** for 3COL?

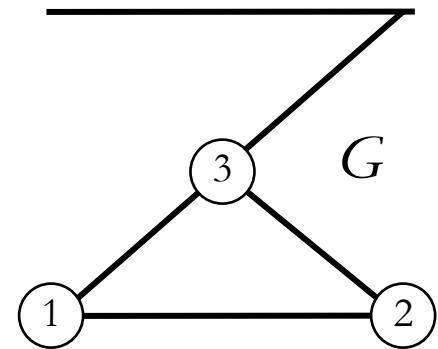
A solution s is a **coloring of vertices**
 $\{1B, 2R, 3Y, 4R, 5B\}$

$V :=$ On input $\langle G, s \rangle$

For every edge $\{u, v\}$ of G :

If u and v are assigned same color in s , *reject*.

Otherwise, *accept*.



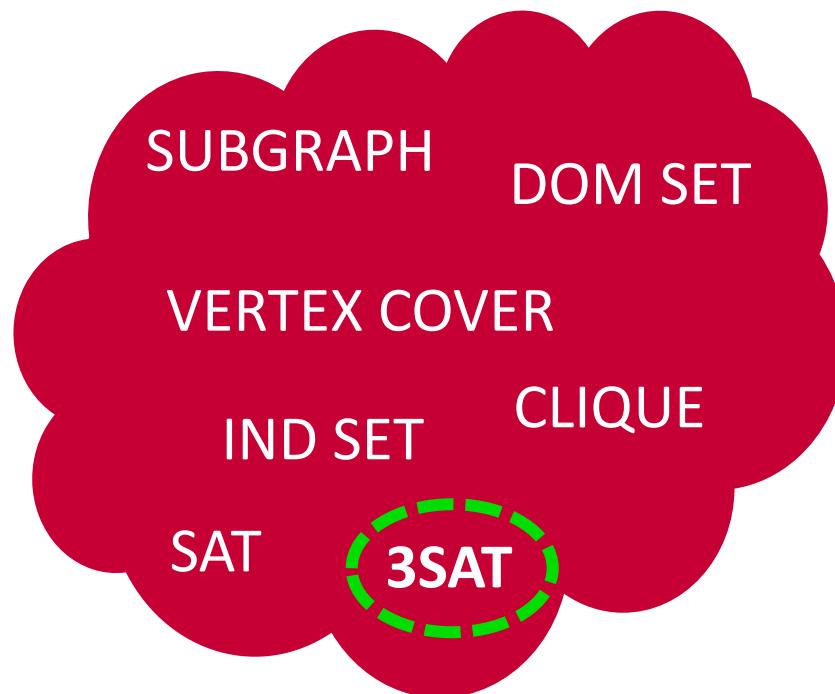
Running time = $O(m)$

Coloring



- Step 2: Some NP-complete L **reduces to** 3COL

3COL = { $\langle G \rangle$: G has a valid 3-coloring}



3COL does not look like
any of these problems

In such a case it is a good
idea to start with 3SAT

- Step 3: Reduce 3SAT to 3COL

3SAT = { $\langle f \rangle$: f is a 3CNF that has a satisfying assignment}

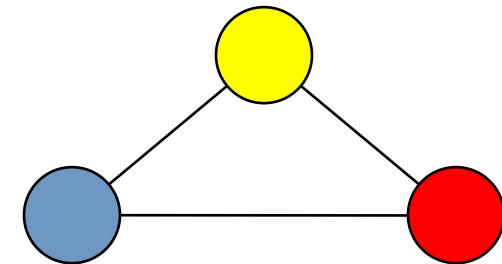
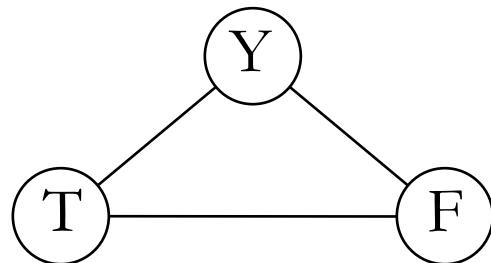
3COL = { $\langle G \rangle$: G has a valid 3-coloring}

Idea: Assignment of $\phi \leftrightarrow$ Coloring of G

Coloring: Gadgets

- Step 3: Reduce 3SAT to 3COL

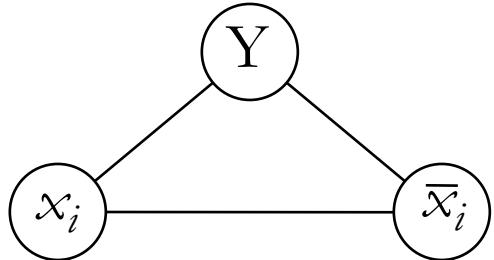
3 Colors
T represents a colour
(say Blue)
F represents a colour
(say False)
Y represents a colour
(say Yellow)



Part I: 3 special vertices T (true), F (false), and Y

Variables in my 3SAT problem must
be T (Blue) or False (Red)

Coloring: Gadgets

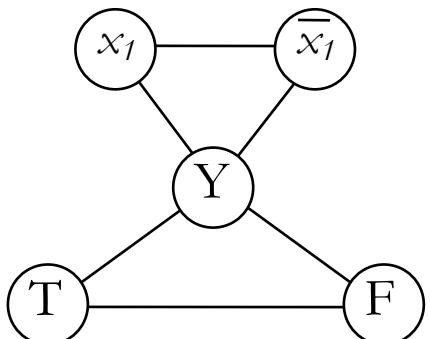


Either x_i has color of T
and \bar{x}_i has color of F

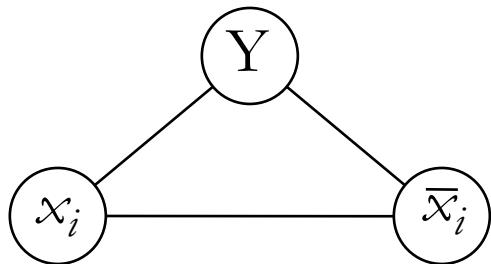
Or x_i has color of F
and \bar{x}_i has color of T

General Case → One variable:

Force a variable to take either T/F and it's negation to take the opposite



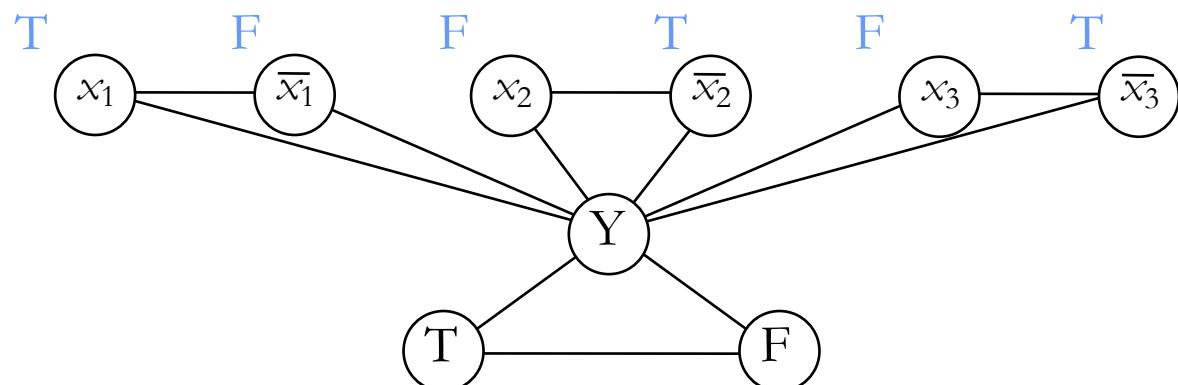
Coloring: Gadgets



Either x_i has color of T
and \bar{x}_i has color of F

Or x_i has color of F
and \bar{x}_i has color of T

Part 2: For each variable x_i



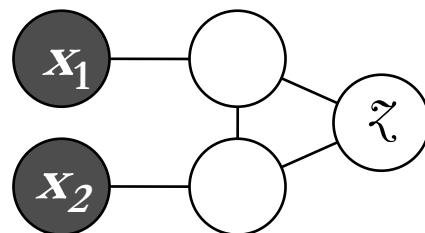
Still
incomplete

Example: $(x_1 \vee \bar{x}_1 \vee x_2) \wedge (x_1 \vee x_2 \vee \bar{x}_3)$

Coloring: OR-Gadgets

- To **encode** the clauses of ϕ , we need an **OR-gadget**

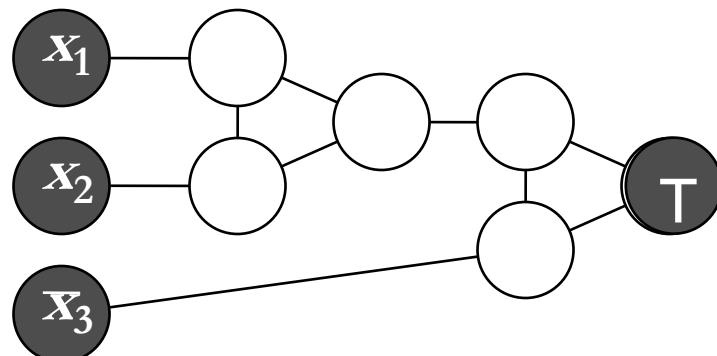
2 variables



If x_1 or x_2 is colored T, z can be colored T

If x_1 and x_2 are colored F, z must be colored F

3 variables



z can be colored T if and only if at least one literal is colored T

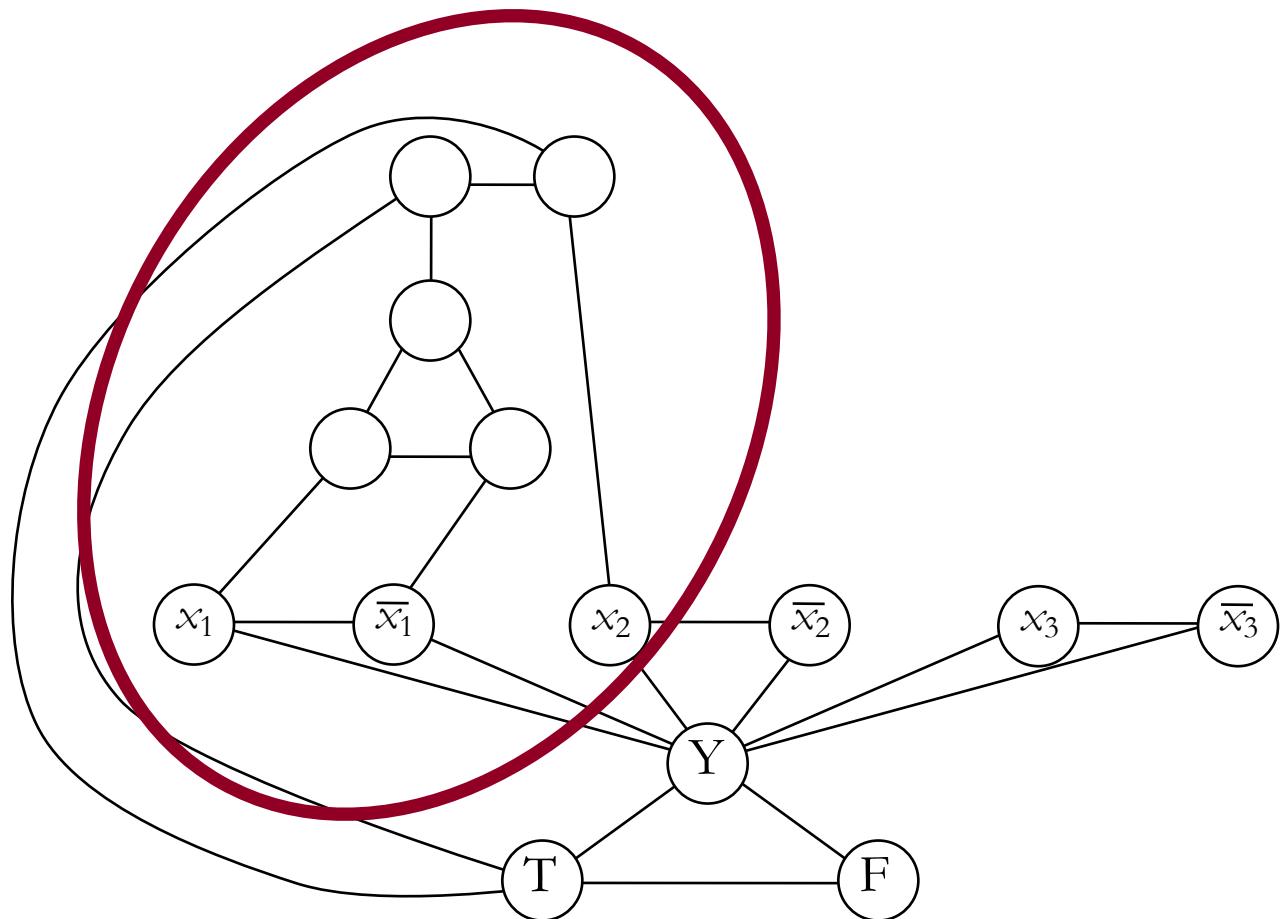
Assign color T to such z 's to force that an unsat clause in 3SAT will be unsat in 3COL

Part 3: For each clause like $(x_1 \vee x_2 \vee \bar{x}_3)$

Coloring reduction: An example

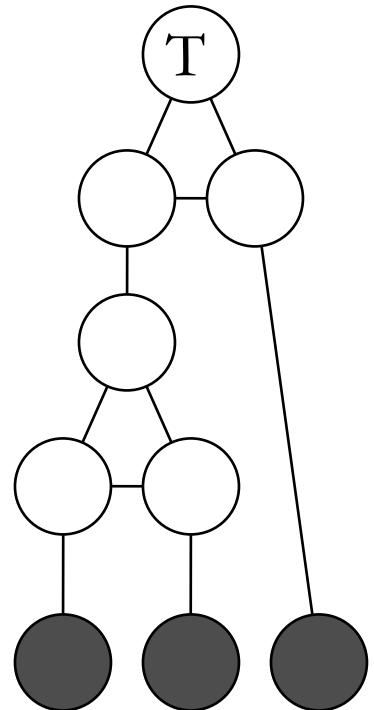


OR-gadget



Example:

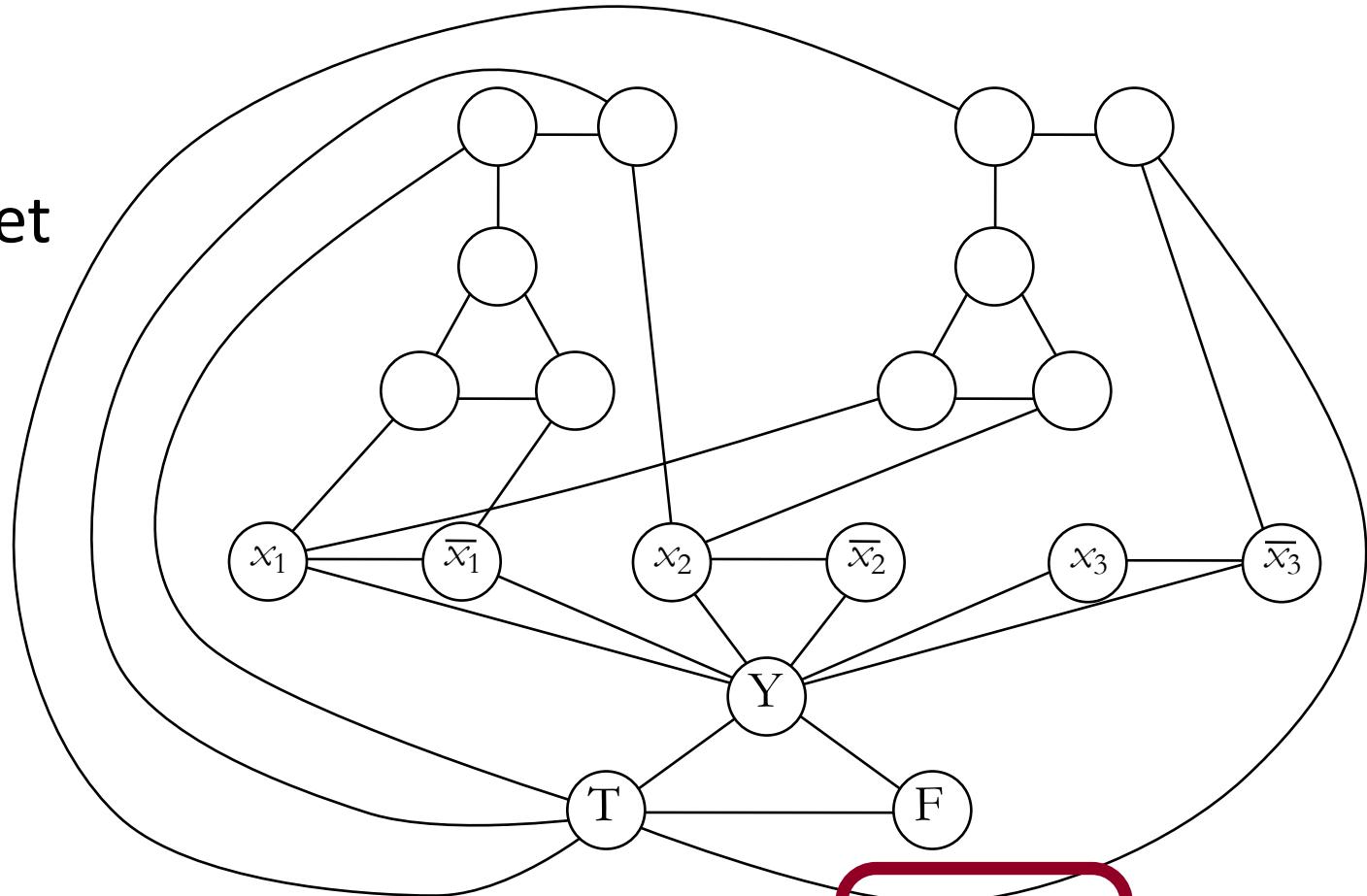
$$(x_1 \vee \bar{x}_1 \vee x_2) \wedge (x_1 \vee x_2 \vee \bar{x}_3)$$



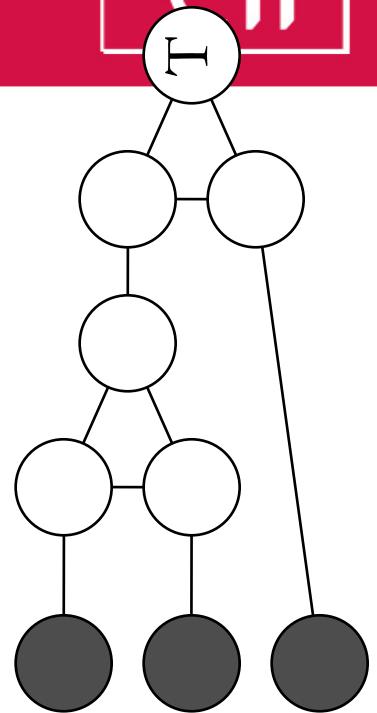
Coloring reduction: An example



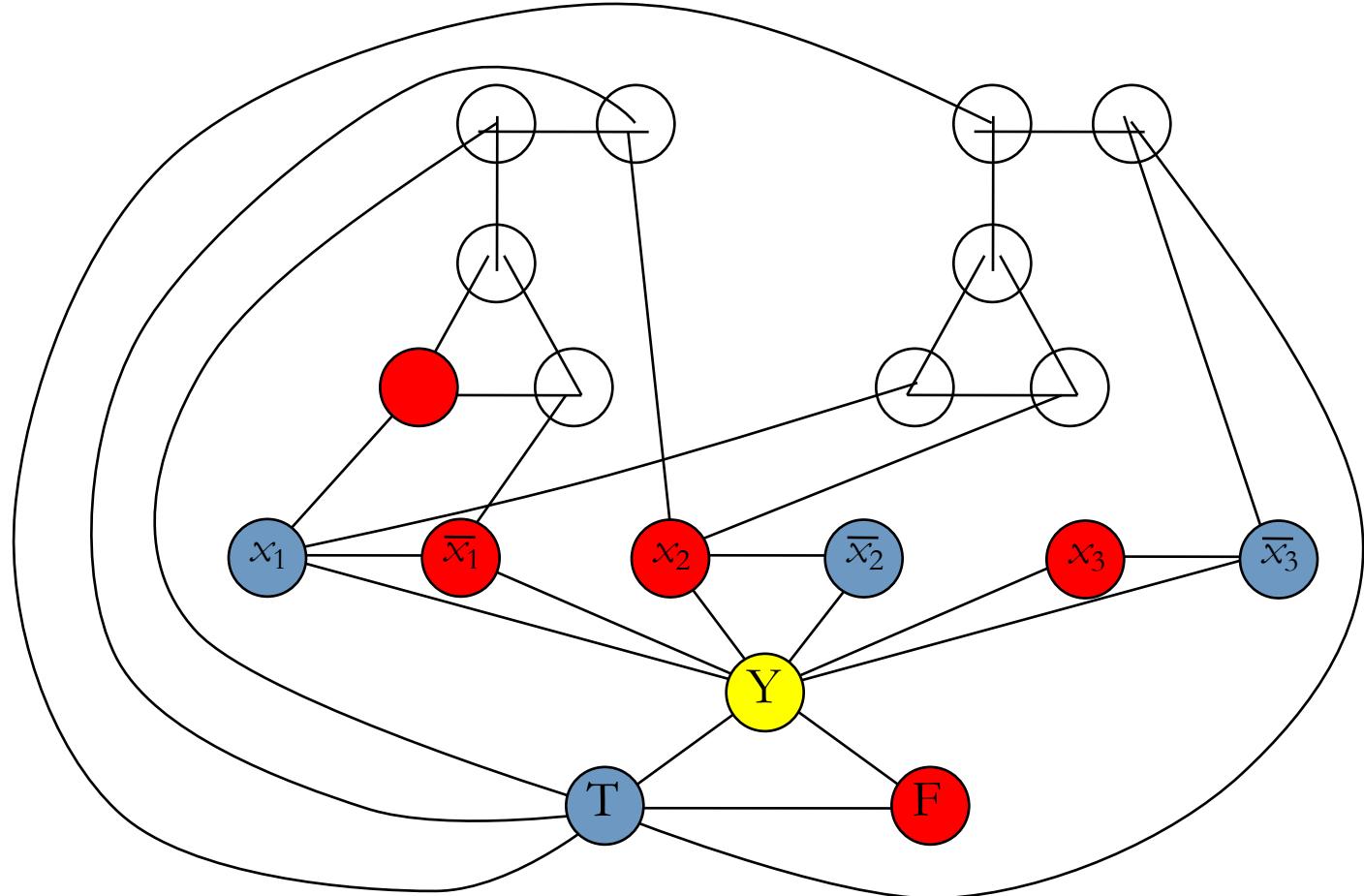
OR-gadget



Example: $(x_1 \vee \bar{x}_1 \vee x_2) \wedge (x_1 \vee x_2 \vee \bar{x}_3)$



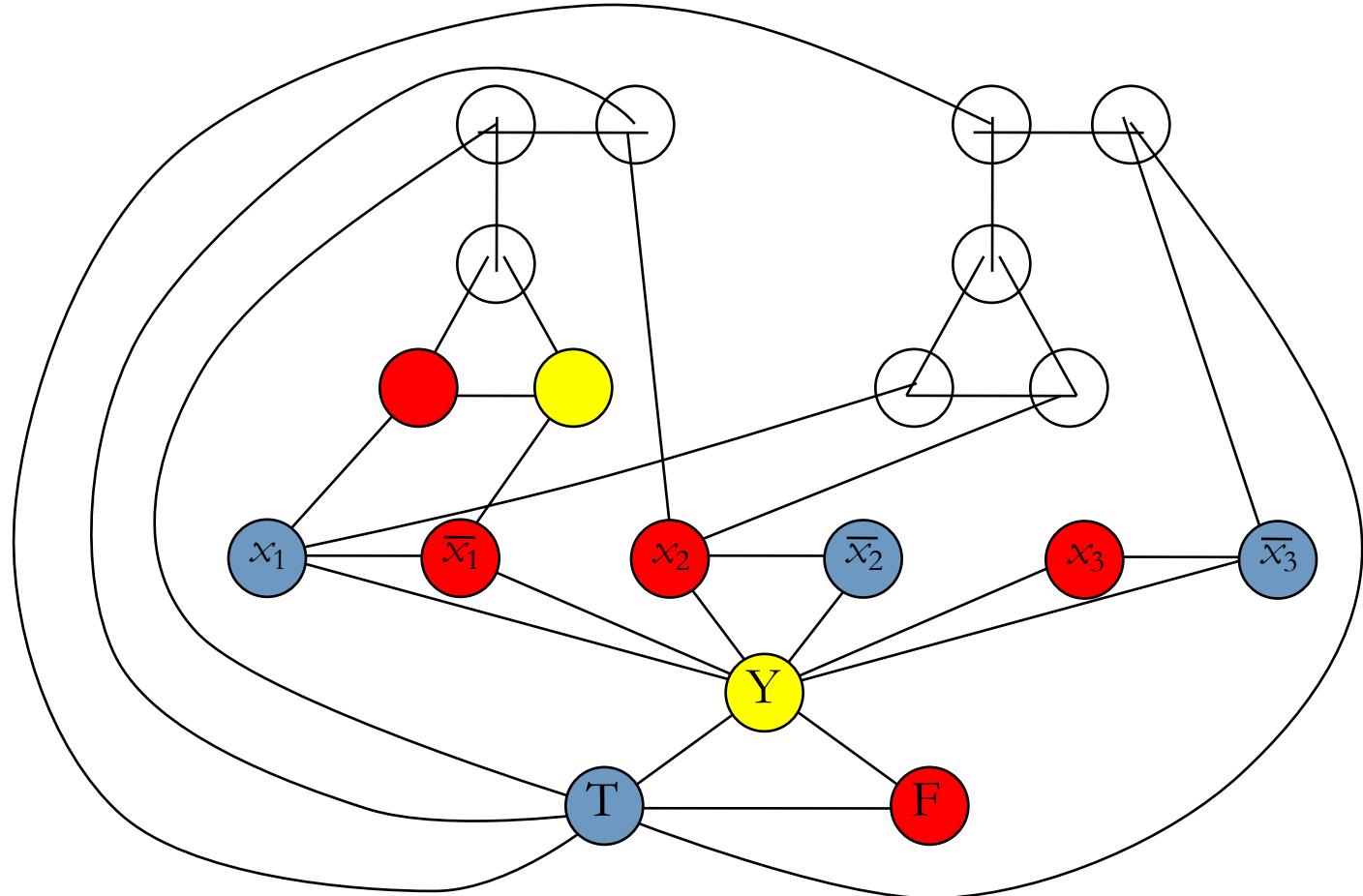
Coloring reduction: An example



Example: $(x_1 \vee \overline{x}_1 \vee x_2) \wedge (x_1 \vee x_2 \vee \overline{x}_3)$

T	F	F	T	F	T
---	---	---	---	---	---

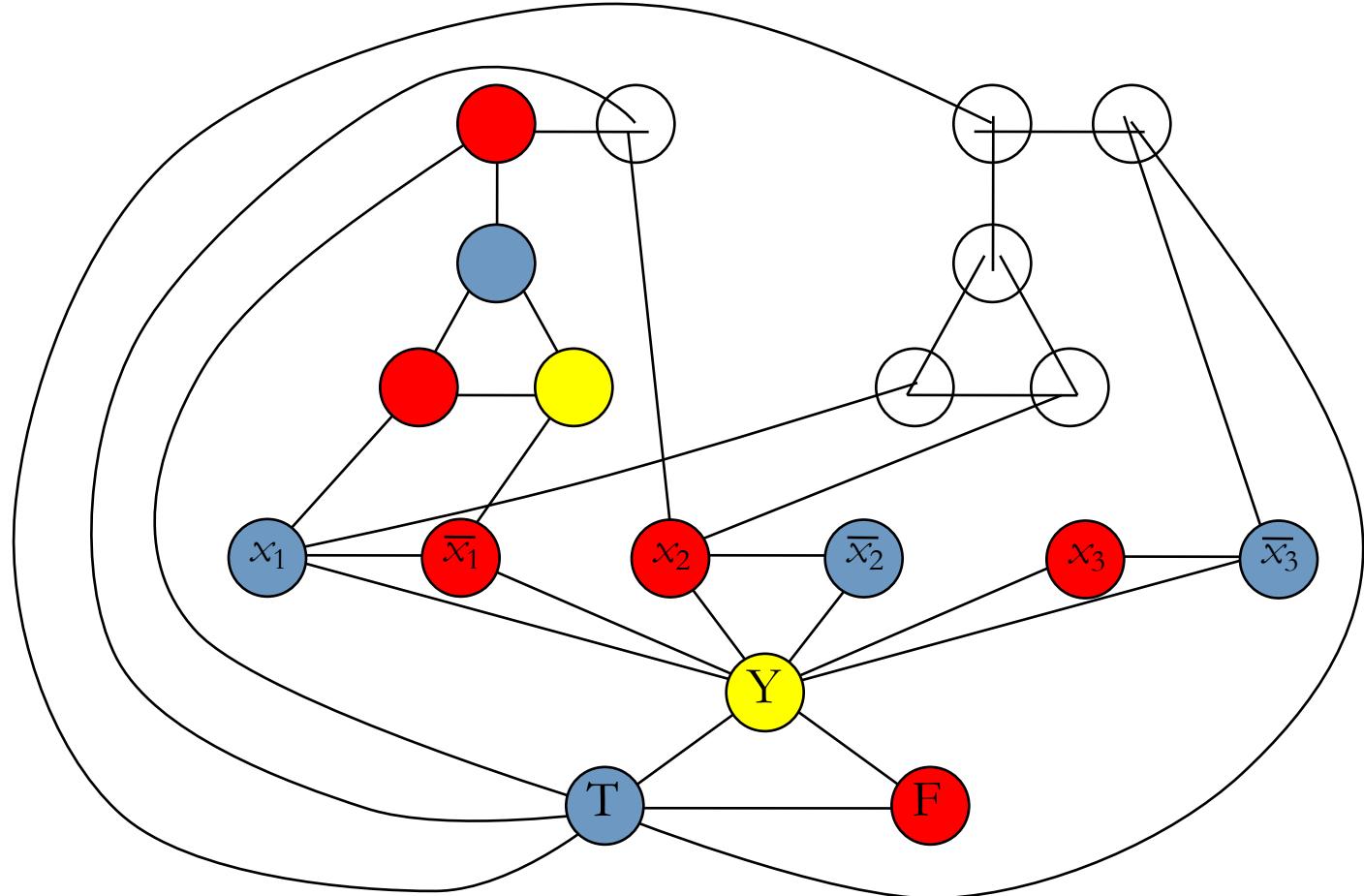
Coloring reduction: An example



Example: $(x_1 \vee \bar{x}_1 \vee x_2) \wedge (x_1 \vee x_2 \vee \bar{x}_3)$

T	F	F	T	F	T
---	---	---	---	---	---

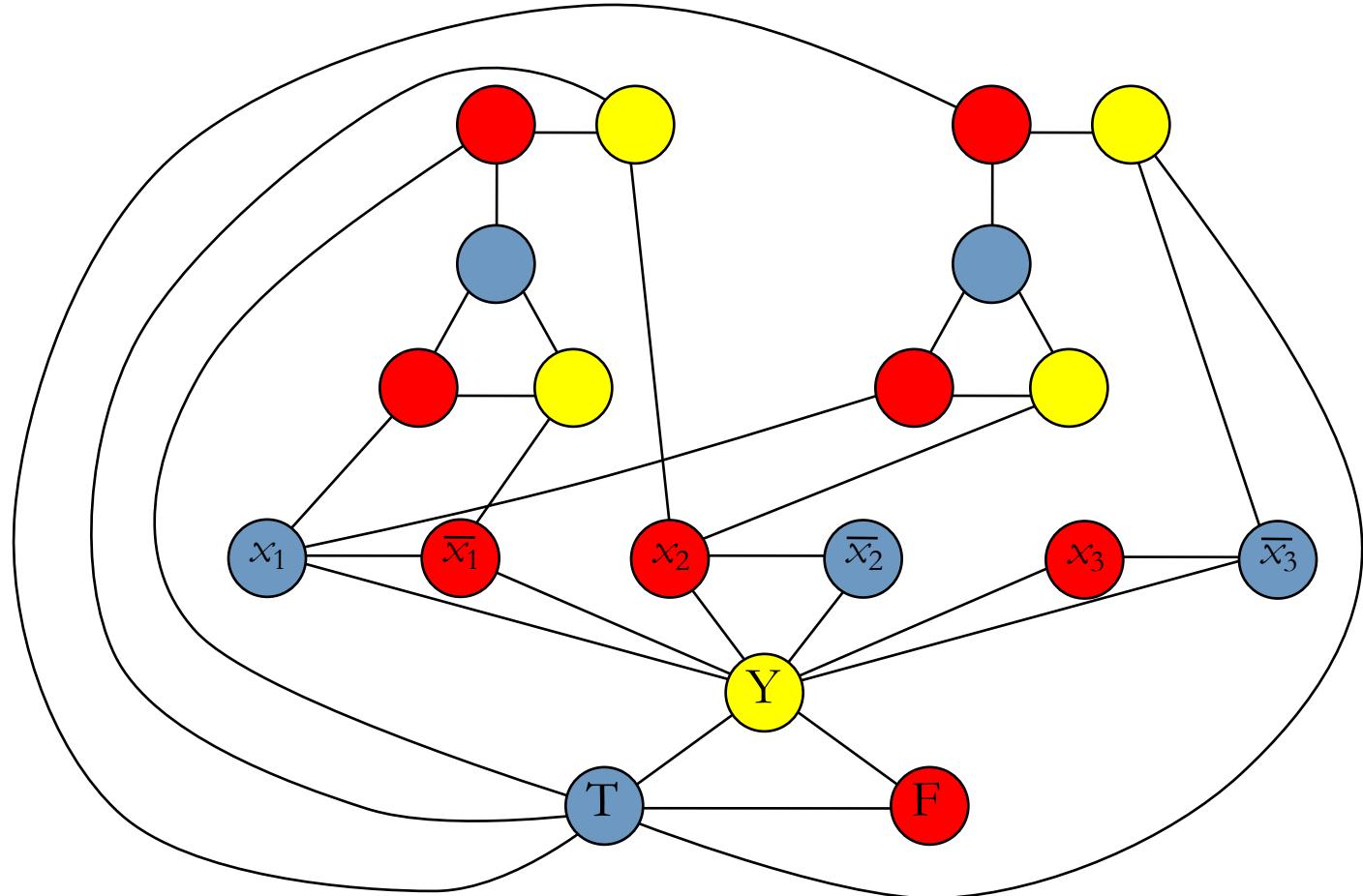
Coloring reduction: An example



Example: $(x_1 \vee \bar{x}_1 \vee x_2) \wedge (x_1 \vee x_2 \vee \bar{x}_3)$

$T \quad F \quad F \quad T \quad F \quad T$

Coloring reduction: An example



Example: $(x_1 \vee \bar{x}_1 \vee x_2) \wedge (x_1 \vee x_2 \vee \bar{x}_3)$

T	F	F	T	F	T
---	---	---	---	---	---

Coloring



- Step 3: Reduce 3SAT to 3COL

Running time = $O(m + n)$
 n = number of variables
 m = number of clauses

$R :=$ On input $\langle f \rangle$, where f is a 3CNF:

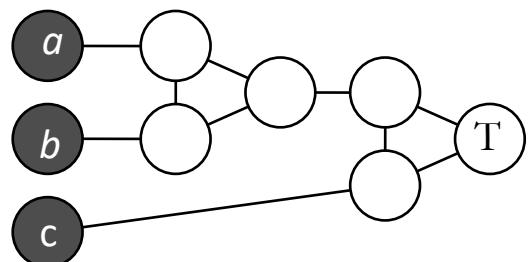
Construct the following graph G :

Add 3 special vertices T, F, Y connected by a triangle.

For every variable x_i of f , add vertices x_i and \bar{x}_i and include the triangle x_i, \bar{x}_i, Y .

For every clause $(a \vee b \vee c)$ of f , connect a, b, c , and T using the gadget below.

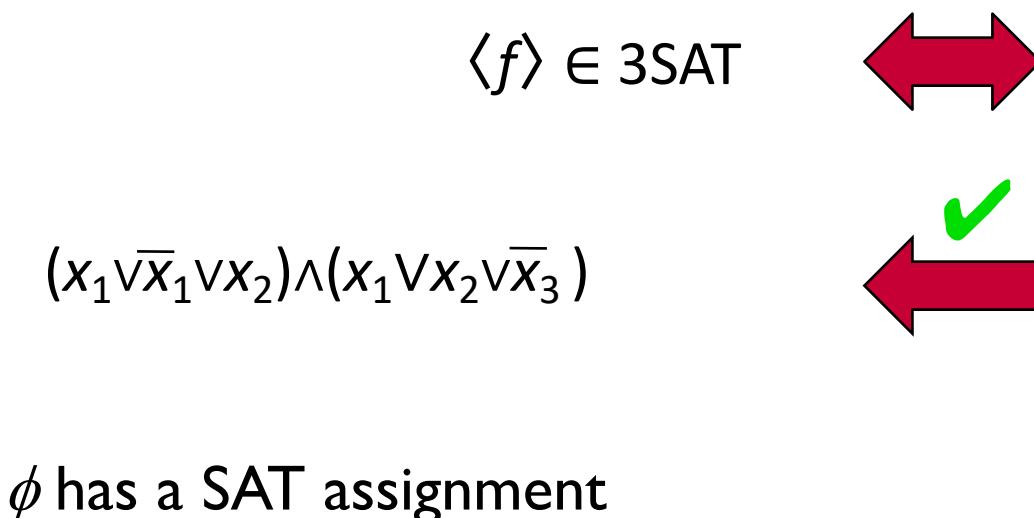
Output $\langle G \rangle$.



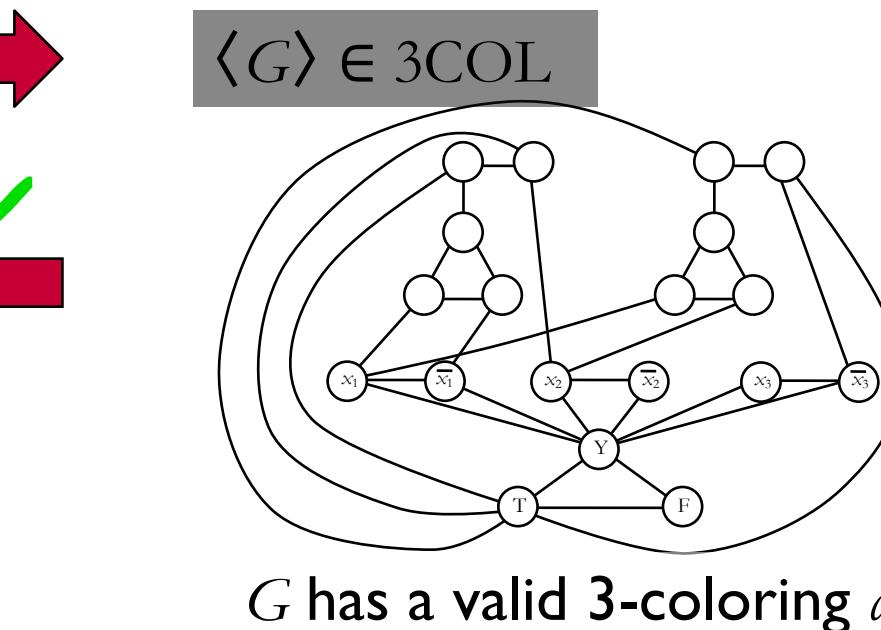
Coloring

CIT

- Step 4: Argue that the reduction is correct



$$x_i = \begin{cases} \text{true, if node } x_i \text{ has same color as T} \\ \text{false, if node } x_i \text{ has same color as F} \end{cases}$$



Since each clause gadget is colored properly, each clause must contain a true literal