CSE 421

NP-Completeness

Yin Tat Lee

Cook-Levin Theorem

Theorem (Cook 71, Levin 73): 3-SAT is NP-complete, i.e., for all problems $A \in NP$, $A \leq_p$ 3-SAT.

(See CSE 431 for the proof)

- So, 3-SAT is the hardest problem in NP.

What does this say about other problems of interest? Like Independent set, Vertex Cover, …

Fact: If $A \leq_p B$ and $B \leq_p C$ then, $A \leq_p C$

Pf: Just compose the reductions from A to B and B to C

So, if we prove 3-SAT $\leq_p$ Independent set, then Independent Set, Clique, Vertex cover, Set cover are all NP-complete

3-SAT $\leq_p$ Independent Set $\leq_p$ Vertex Cover $\leq_p$ Set Cover

Steps to Proving Problem B is NP-complete

- Show **B** is **NP**-hard:
  - State: ”Reduction is from **NP**-hard Problem **A**”
  - Show what the map **f** is
  - Argue that **f** is polynomial time
  - Argue correctness: two directions Yes for **A** implies Yes for **B** and vice versa.
- Show **B** is in **NP**
  - State what hint/certificate is and why it works
  - Argue that it is polynomial-time to check.

Is NP-complete as bad as it gets?

- NO! **NP**-complete problems are frequently encountered, but there are worse:

  Some problems provably require exponential time.

  - Ex: Does **M** halt on input **x** in $2^{|x|}$ steps?

  Some require 2^n, 2^{2^n}, $2^{2^{2^n}}$, ... steps

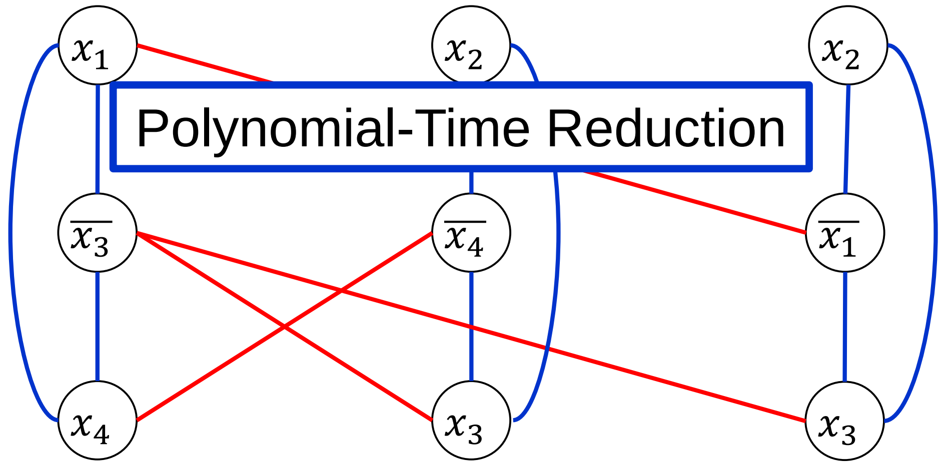
  And some are just plain uncomputable

3-SAT $\leq_p$ Independent Set

Map a 3-CNF to (G,k). Say k is number of clauses

- Create a vertex for each literal
- Joint two literals if
  - They belong to the same clause (blue edges)
  - The literals are negations, e.g., $x_i, \overline{x_i}$ (red edges)
- Set k be the # of clauses.

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



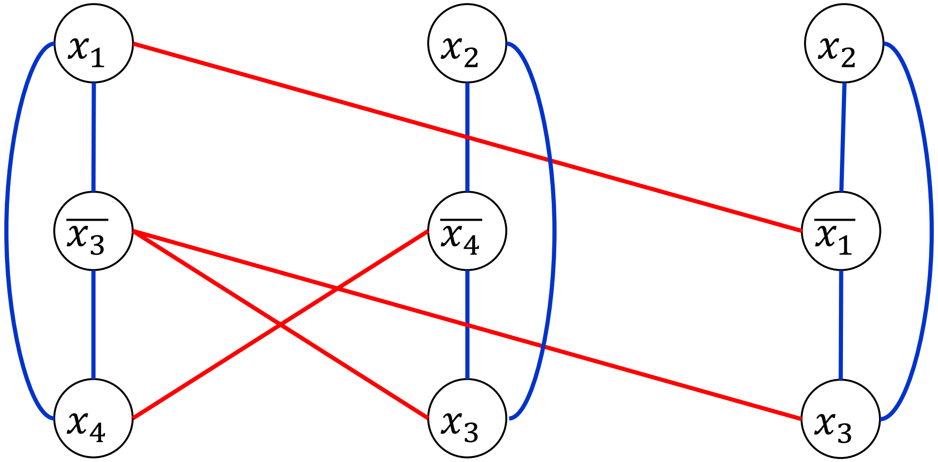
Correctness of 3-SAT $\leq_p$ Indep Set

F satisfiable => An independent of size k

Given a satisfying assignment, Choose one node from each clause where the literal is satisfied

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

Satisfying assignment: $x_1 = T, x_2 = F, x_3 = T, x_4 = F$



- S has exactly one node per clause => No blue edges between S
- S follows a truth-assignment => No red edges between S
- S has one node per clause => |S|=k

Correctness of 3-SAT $\leq_p$ Indep Set

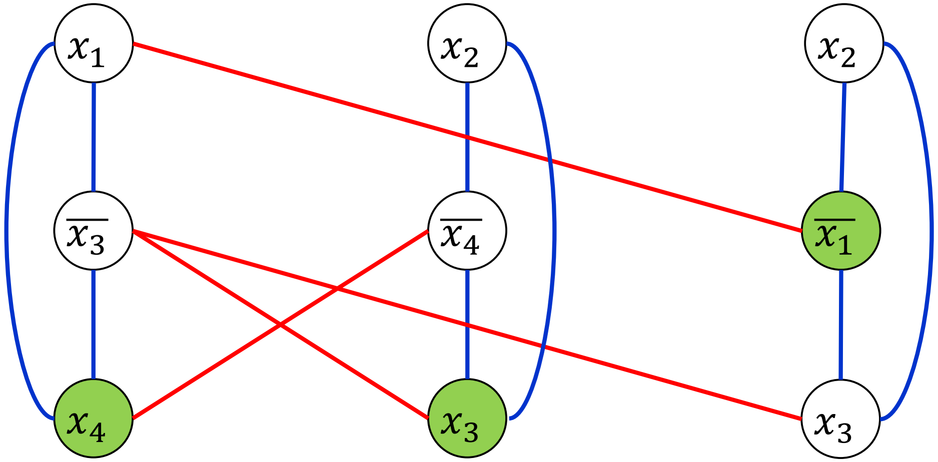
An independent set of size k => A satisfying assignment

Given an independent set S of size k.

S has exactly one vertex per clause (because of blue edges)

S does not have $x_i, \overline{x_i}$ (because of red edges)

So, S gives a satisfying assignment



Satisfying assignment: $x_1 = F, x_2 = ?, x_3 = T, x_4 = T$

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

Yet another example of NP completeness

Prove that Super Mario Bros is NP-complete.

What do we need to show?

- The problem is in NP.
- Some NP complete problem is easier than Super Mario.

Approach:

- 3SAT $\leq_P$ Super Mario


FUN 2014: Fun with Algorithms pp 40-51 | Cite as

Classic Nintendo Games Are (Computationally) Hard

Authors | Authors and affiliations

Greg Aloupis, Erik D. Demaine, Alan Guo, Giovanni Viglietta

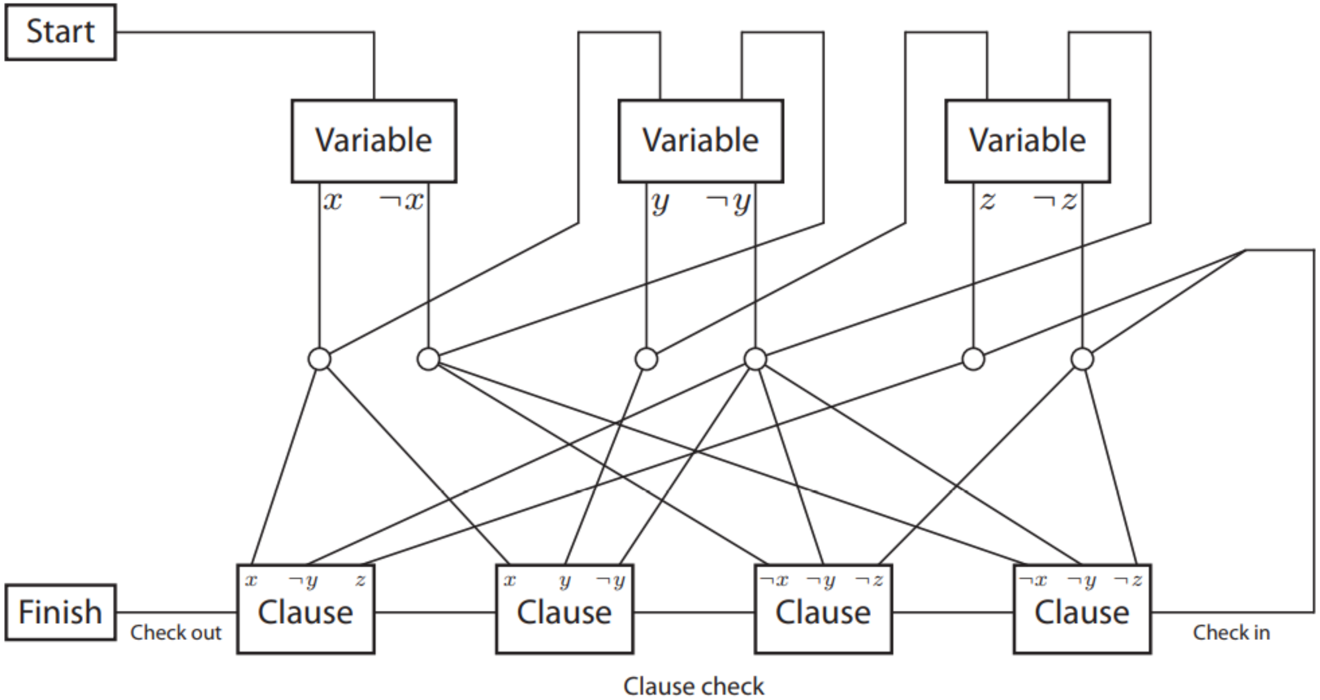
Conference paper

4 Citations | 1 Mentions | 15 Readers | 940 Downloads

Part of the Lecture Notes in Computer Science book series (LNCS, volume 8496)

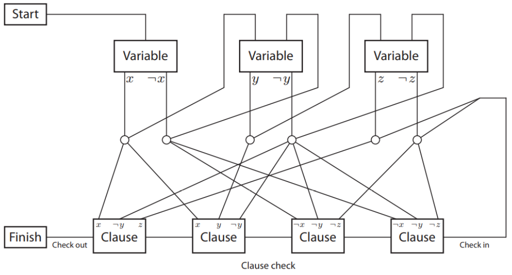
Yet another example of NP completeness

Given a 3SAT, we need to create a level.

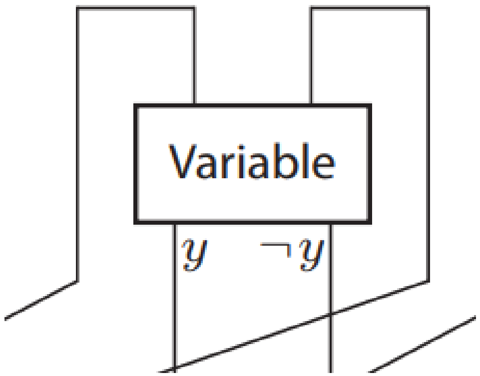


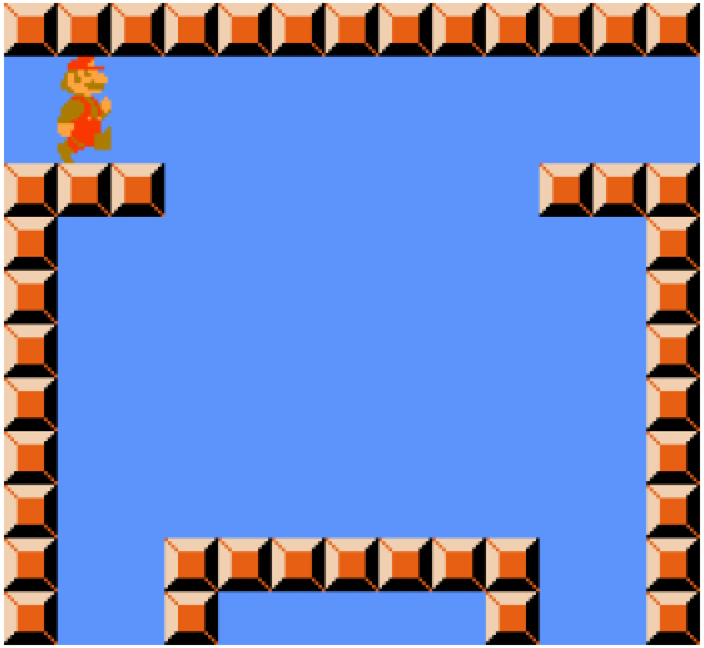
We ignore the following issues:

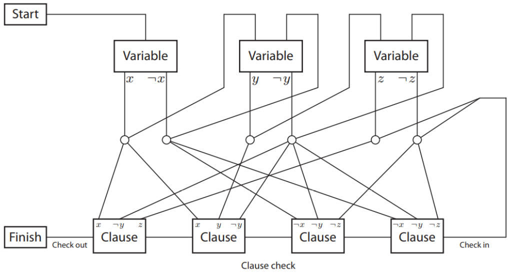
- Need to consider the “crossing” coz the level is 2-D.
- Assume Mario can go both left or right.



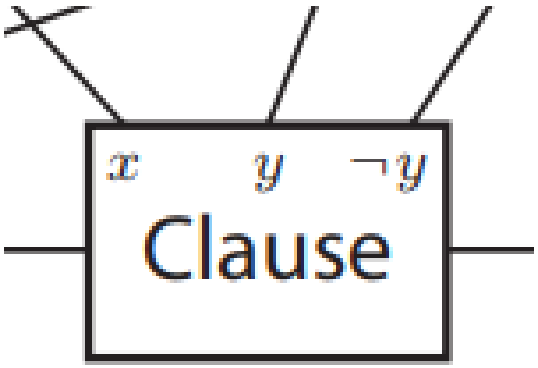
Question 1: How to create this part?







Question 2: How to create this part?



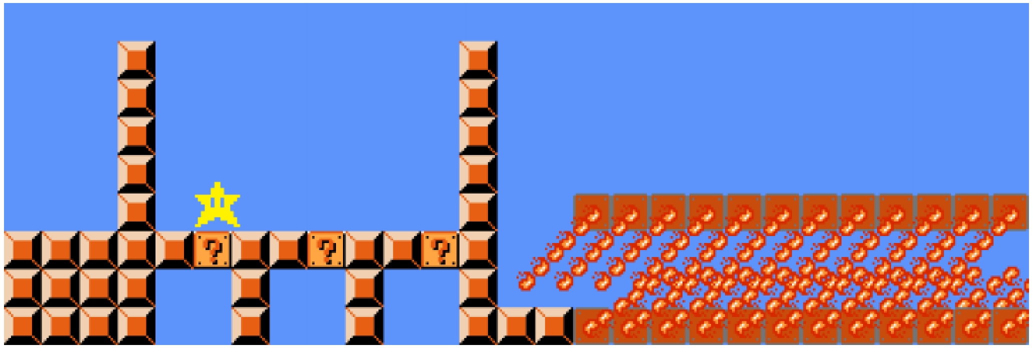


Figure 11: Clause gadget for Super Mario Bros.

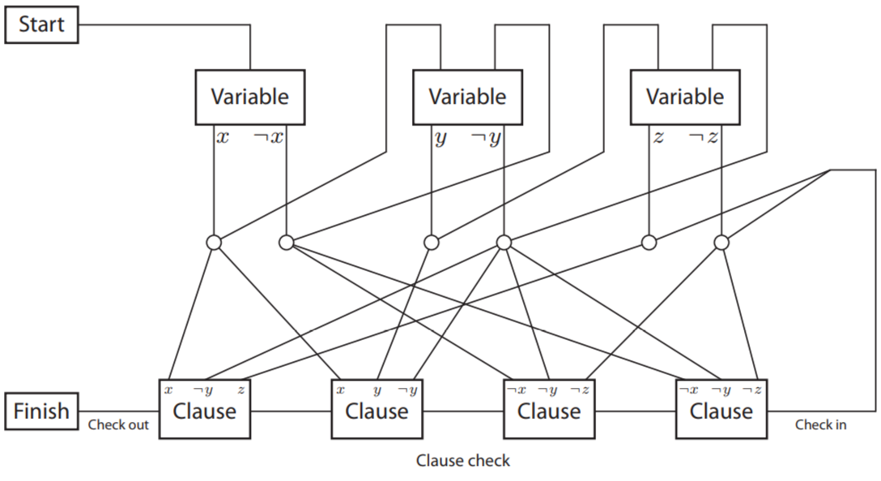
So, what you need to prove?

- If the 3SAT is satisfiable, then indeed the level is solvable.

Usually, this part is easy. This is basically due to the design of your reduction.

- If the level is solvable, then the 3SAT is satisfiable

This part usually requires more argument. Need to prove no tricky way to solve the problem without solving the 3SAT.



More NP-completeness

- Subset-Sum problem
  (Decision version of Knapsack)
  - Given **n** integers $w_1, \ldots, w_n$ and integer **W**
  - Is there a subset of the **n** input integers that adds up to exactly **W**?

- **O(nW)** solution from dynamic programming but if **W** and each w_i can be **n** bits long then this is exponential time

3-SAT $\leq_P$ Subset-Sum

- Given a 3-CNF formula with **m** clauses and **n** variables
- Will create 2**m**+2**n** numbers that are **m+n** digits long

  Two numbers for each variable x_i

  - t_i and f_i (corresponding to x_i being true or x_i being false)

  Two extra numbers for each clause

  - u_j and v_j (filler variables to handle number of false literals in clause C_j)

3-SAT $\leq_P$ Subset-Sum

$C_3 = (x_1 \lor \neg x_2 \lor x_5)$

				i						j		
	1	2	3	4	…	n	1	2	3	4	…	m
t_1	1	0	0	0	…	0	0	0	1	0	…	1
f_1	1	0	0	0	…	0	1	0	0	1	…	0
t_2	0	1	0	0	…	0	0	1	0	0	…	1
f_2	0	1	0	0	…	0	0	0	1	1	…	0
				…				….				
$u_1=v_1$	0	0	0	0	…	0	1	0	0	0	…	0
$u_2=v_2$	0	0	0	0	…	0	0	1	0	0	…	0
				…				….				
W	1	1	1	1	…	1	3	3	3	3	…	3

Graph Colorability

- Defn: Given a graph G=(V,E), and an integer k, a k-coloring of G is

  an assignment of up to k different colors to the vertices of G so that the endpoints of each edge have different colors.

- 3-Color: Given a graph G=(V,E), does G have a 3-coloring?
- Claim: 3-Color is NP-complete
- Proof: 3-Color is in NP:

  Hint is an assignment of red,green,blue to the vertices of G

  Easy to check that each edge is colored correctly

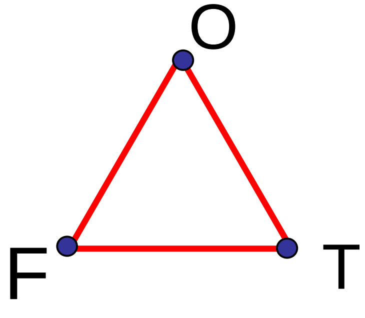
3-SAT $\leq_P$ 3-Color

- Reduction:

  We want to map a 3-CNF formula **F** to a graph **G** so that

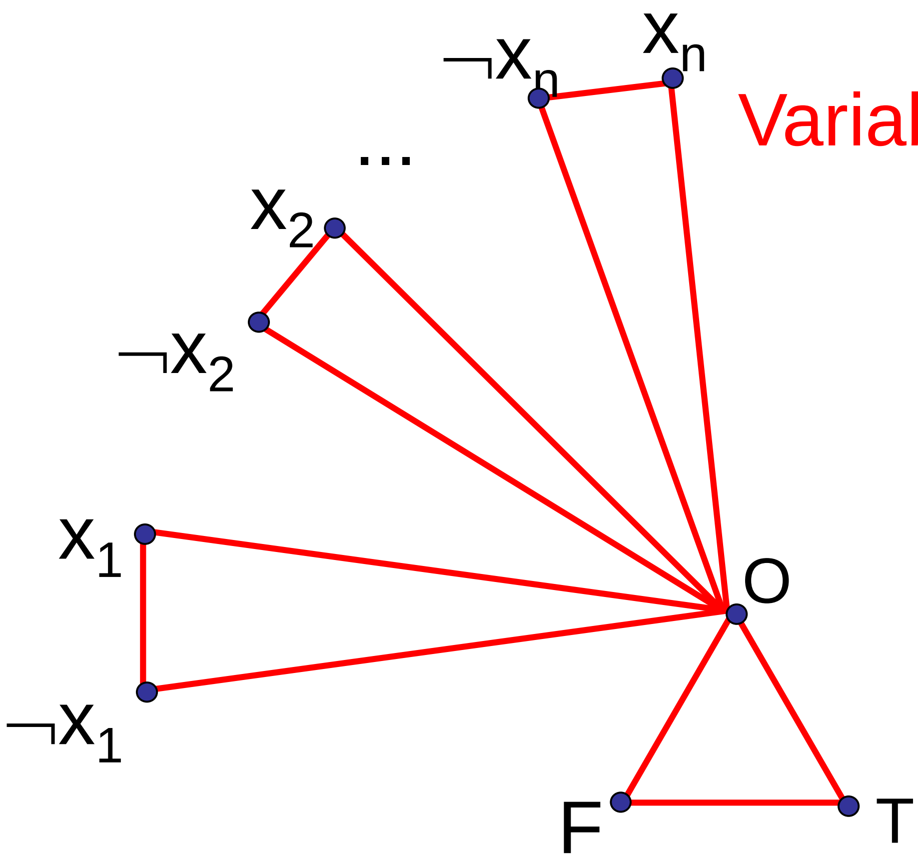
  - **G** is 3-colorable iff **F** is satisfiable

3-SAT $\leq_P$ 3-Color



Base Triangle

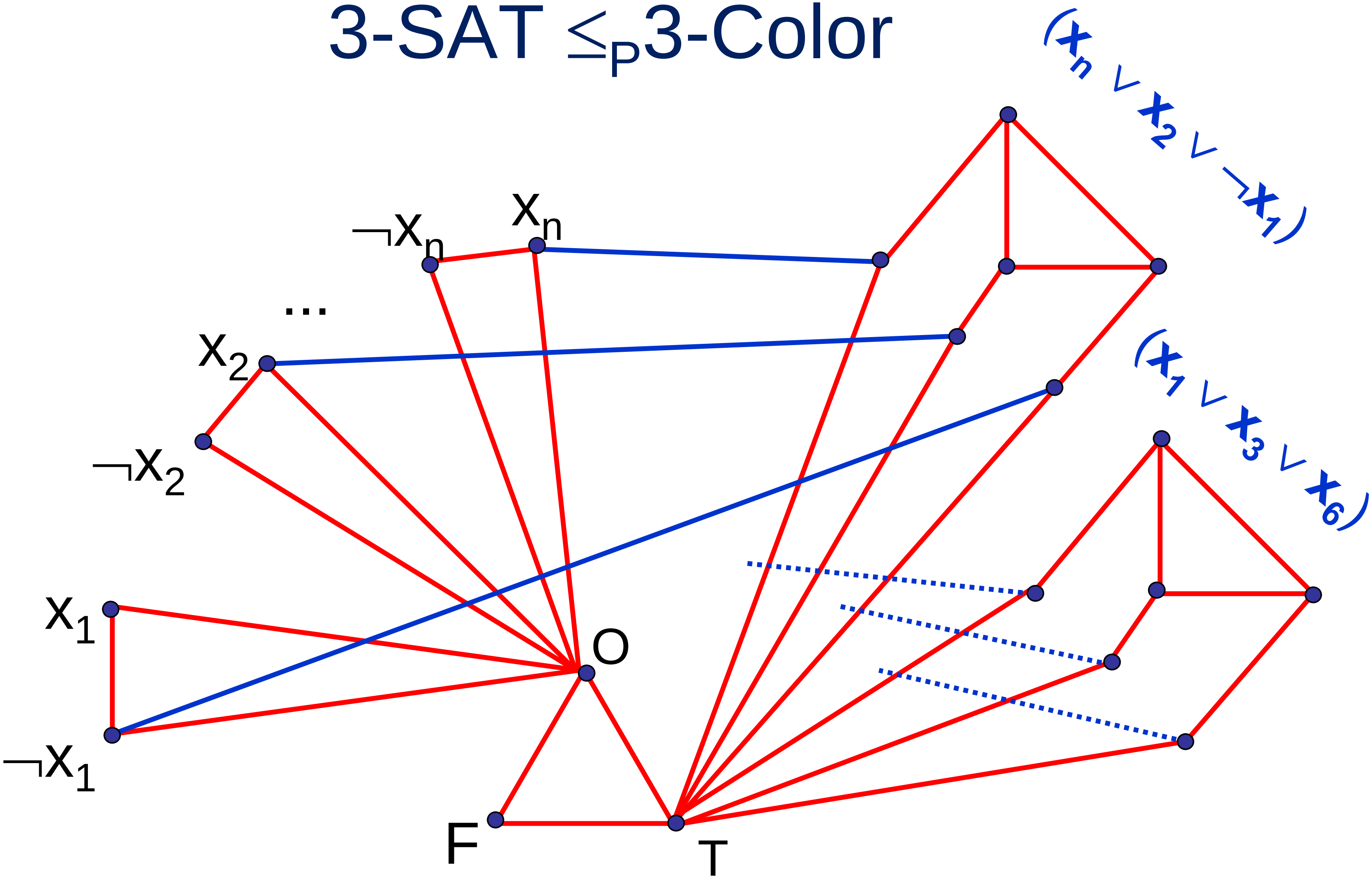
3-SAT $\leq_P$ 3-Color



Variable Part:

in 3-coloring, variable colors correspond to some truth assignment (same color as T or F)

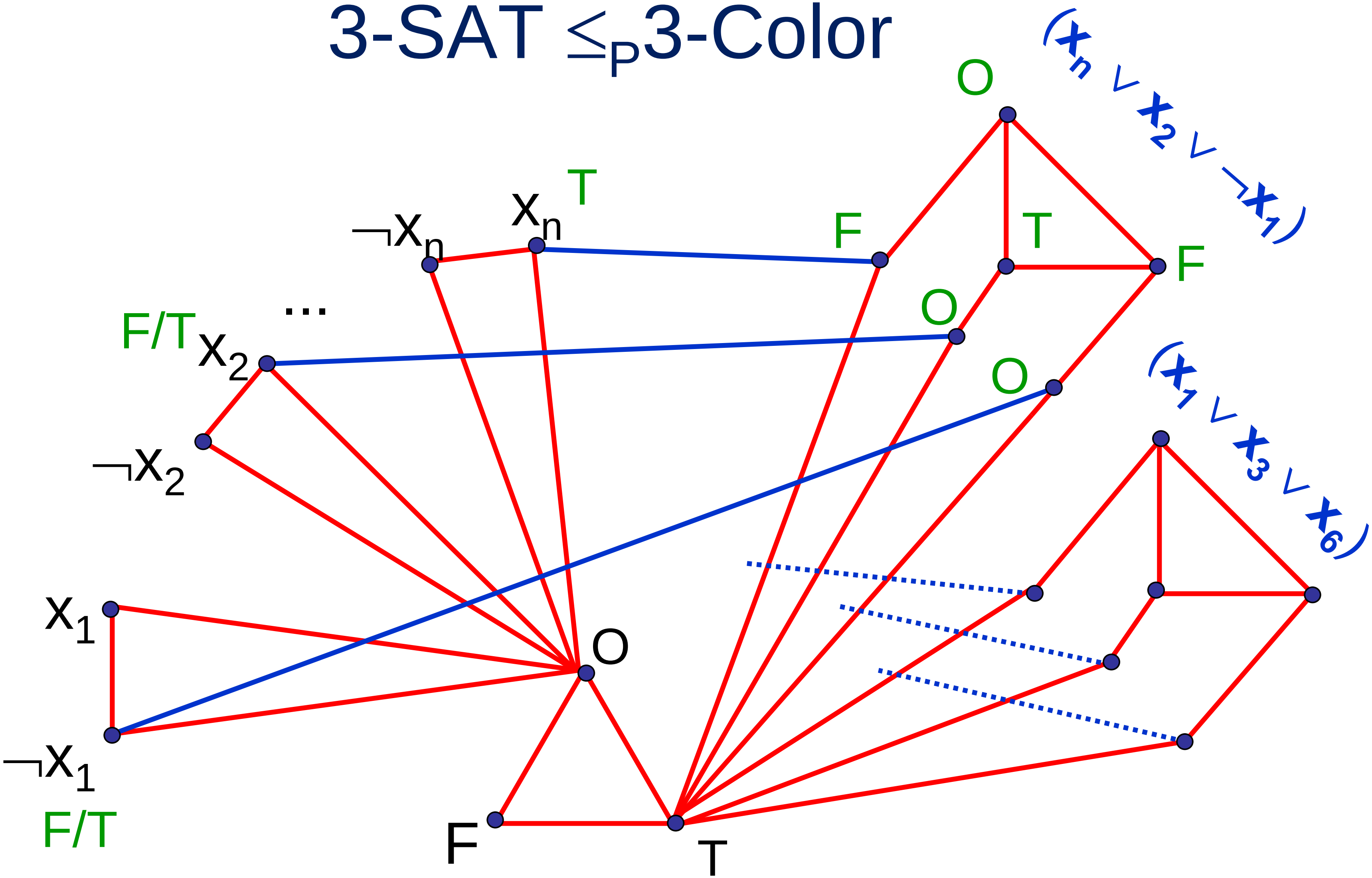
3-SAT $\leq_P$ 3-Color



Clause Part:

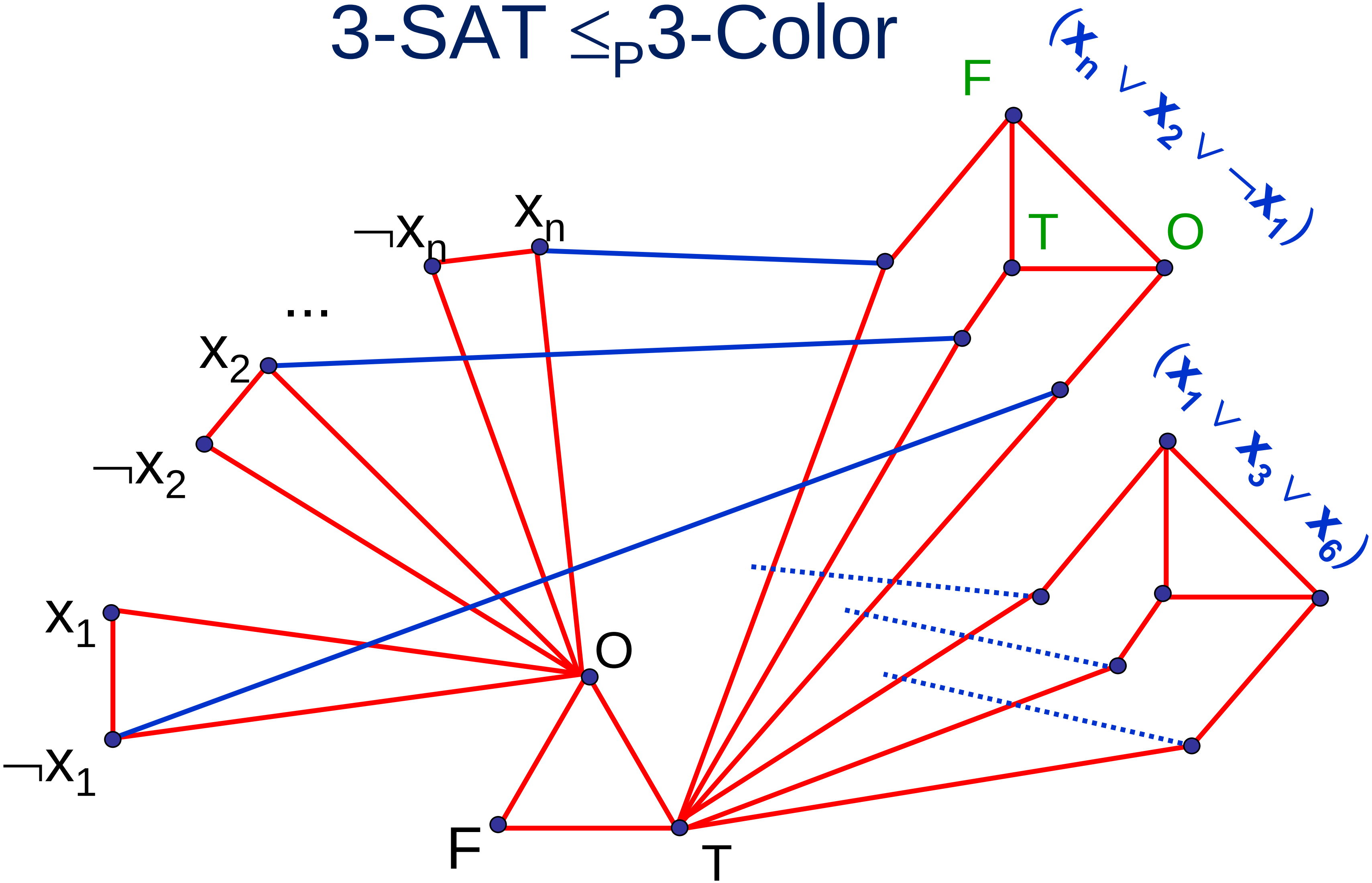
Add one 6 vertex gadget per clause connecting its ‘outer vertices’ to the literals in the clause

3-SAT $\leq_P$ 3-Color



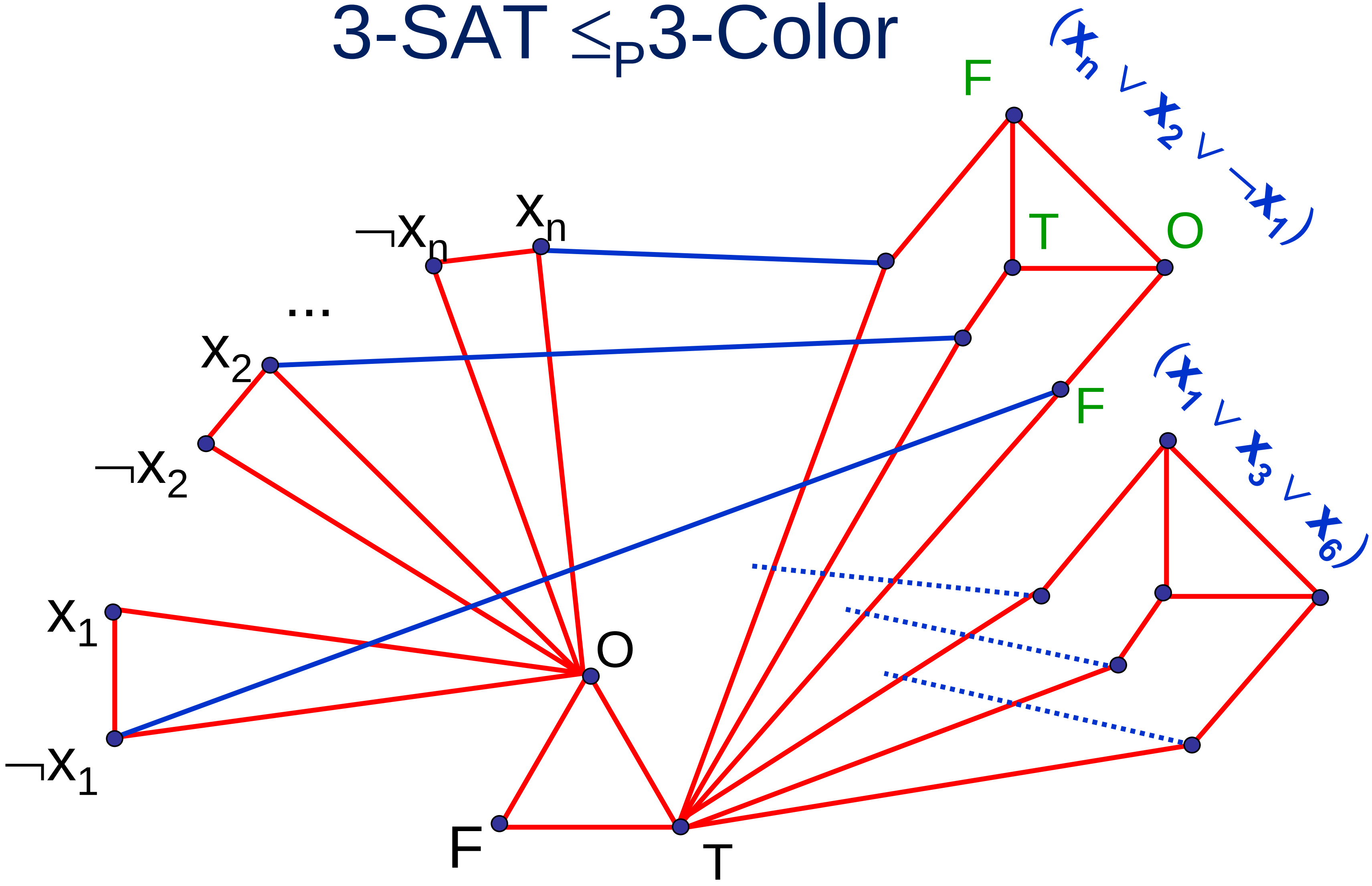
Any truth assignment satisfying the formula can be extended to a 3-coloring of the graph

3-SAT $\leq_P$ 3-Color



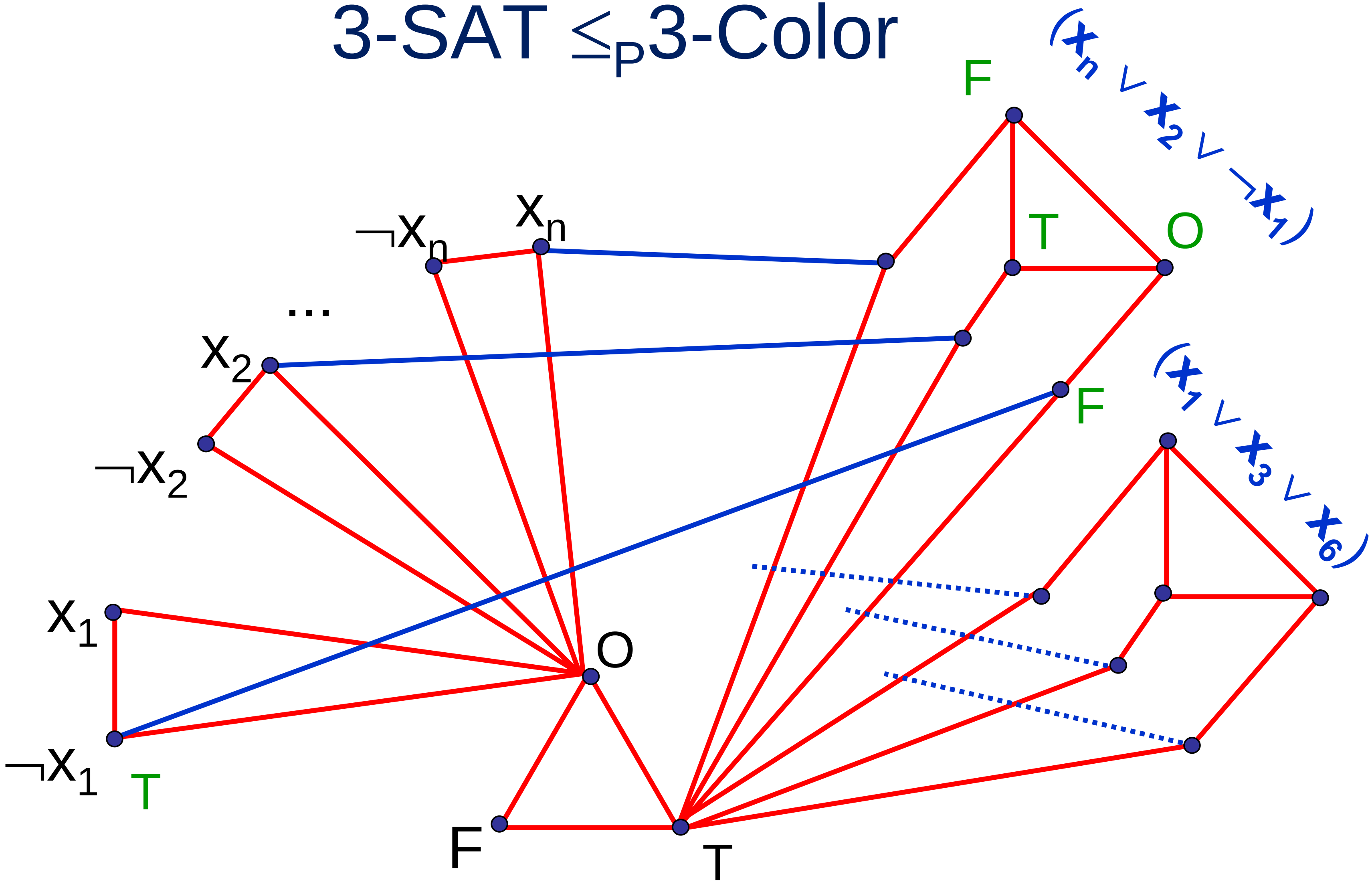
Any 3-coloring of the graph colors each gadget triangle using each color

3-SAT $\leq_P$ 3-Color



Any 3-coloring of the graph has an F opposite the O color in the triangle of each gadget

3-SAT $\leq_P$ 3-Color



Any 3-coloring of the graph has T at the other end of the blue edge connected to the F