CSE 431:

More NP-completeness

Paul Beame

We already know

- **3SAT $\leq_P$ CNFSAT**
- **CNFSAT $\leq_P$ CLIQUE**
- **CIRCUIT-SAT** is NP-complete
- We now show Cook-Levin Theorem that **3SAT** is NP-complete (on board)

A useful property of polynomial-time reductions

- Theorem: If **A $\leq_P$B** and **B $\leq_P$ C** then **A $\leq_P$C**
- Proof idea:
  - Compose the reduction **f** from **A** to **B** with the reduction **g** from **B** to **C** to get a new reduction **h(x)=g(f(x))** from **A** to **C**.
  - The running time bound for **h** is the running time bound for **f** plus the running time bound for **g** composed with that of **f**
  - The composition of two polynomials is also a polynomial so if **f** and **g** are polynomial-time computable then so is **h**

Cook-Levin Theorem Implications

- Theorem (Cook 1971, Levin 1973): **3-SAT** is **NP**-complete
- Corollary: **B** is **NP**-hard ⇔ **3-SAT $\leq_P$B**
  - (or **A $\leq_P$B** for any **NP**-complete problem **A**)
- Proof:
  - If **B** is **NP**-hard then every problem in **NP** polynomial-time reduces to **B**, in particular **3-SAT** does since it is in **NP**
  - For any problem **A** in **NP**, **A $\leq_P$3-SAT** and so if **3-SAT $\leq_P$B** we have **A $\leq_P$ B**.
    - therefore **B** is **NP**-hard if **3-SAT $\leq_P$B**

Reductions by Simple Equivalence

- Show: Clique $\leq_P$ Independent-Set
- Clique:
  - Given a graph **G**=(**V**,**E**) and an integer **k**, is there a subset **U** of **V** with |**U**| ≥ **k** such that every pair of vertices in **U** is joined by an edge?
- Independent-Set:
  - Given a graph **G**=(**V**,**E**) and an integer **k**, is there a subset **U** of **V** with |**U**| ≥ **k** such that no two vertices in **U** are joined by an edge?

Clique $\leq_P$ Independent-Set

- Given (**G**,**k**) as input to Independent-Set where **G**=(**V**,**E**)
- Transform to (**G'**,**k**) where **G'**=(**V**,**E'**) has the same vertices as **G** but **E'** consists of **precisely** those edges that are not edges of **G**
- **U** is an independent set in **G**

⇔ **U** is a clique in **G'**

More Reductions

- Show: Independent Set $\leq_P$ Vertex-Cover
- Vertex-Cover:
  - Given an undirected graph **G**=(**V**,**E**) and an integer **k** is there a subset **W** of **V** of size at most **k** such that every edge of **G** has at least one endpoint in **W**? (i.e. **W** covers all edges of **G**)?
- Independent-Set:
  - Given a graph **G**=(**V**,**E**) and an integer **k**, is there a subset **U** of **V** with |**U**| ≥ **k** such that no two vertices in **U** are joined by an edge?

Reduction Idea

- Claim: In a graph **G**=(**V**,**E**), **S** is an independent set iff **V-S** is a vertex cover
- Proof:
  - ⇒ Let **S** be an independent set in **G**
    - Then **S** contains at most one endpoint of each edge of **G**
    - At least one endpoint must be in **V-S**
    - **V-S** is a vertex cover
  - ⇐Let **W=V-S** be a vertex cover of **G**
    - Then **S** does not contain both endpoints of any edge (else **W** would miss that edge)
    - **S** is an independent set

Reduction

- Map (G,k) to (G,n-k)
  - Previous lemma proves correctness
- Clearly polynomial time
- We also get that
  - Vertex-Cover $\leq_P$ Independent Set

Reductions from a Special Case to a General Case

- Show: Vertex-Cover $\leq_P$ Set-Cover
- Vertex-Cover:
  - Given an undirected graph G=(V,E) and an integer k is there a subset W of V of size at most k such that every edge of G has at least one endpoint in W? (i.e. W covers all edges of G)?
- Set-Cover:
  - Given a set U of n elements, a collection $S_1,\ldots,S_m$ of subsets of U, and an integer k, does there exist a collection of at most k sets whose union is equal to U?

The Simple Reduction

- Transformation f maps (G=(V,E),k) to $(U,S_1,\ldots,S_m,k')$
  - $U \leftarrow E$
  - For each vertex $v \in V$ create a set S_v containing all edges that touch v
  - $k' \leftarrow k$
- Reduction f is clearly polynomial-time to compute
- We need to prove that the resulting algorithm gives the right answer!

Proof of Correctness

- Two directions:
  - If the answer to Vertex-Cover on (G,k) is YES then the answer for Set-Cover on f(G,k) is YES
    - If a set W of k vertices covers all edges then the collection $\{S_v \mid v \in W\}$ of k sets covers all of U
  - If the answer to Set-Cover on f(G,k) is YES then the answer for Vertex-Cover on (G,k) is YES
    - If a subcollection $S_{v_1},\ldots,S_{v_k}$ covers all of U then the set $\{v_1,\ldots,v_k\}$ is a vertex cover in G.

Problems we already know are NP-complete

- Circuit-SAT
- 3-SAT
- Independent-Set
- Clique
- Vertex-Cover
- Set-Cover

More NP-completeness

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

3-SAT $\leq_P$ Subset-Sum

- Given a 3-CNF formula with **m** clauses and **n** variables
- Will create **2m+2n** 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

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

	i: 1 2 3 4 … n	j: 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
	…	….
t	1 1 1 1 … 1	3 3 3 3 … 3

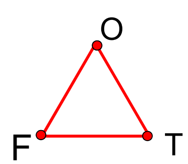
Graph Colorability

- 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

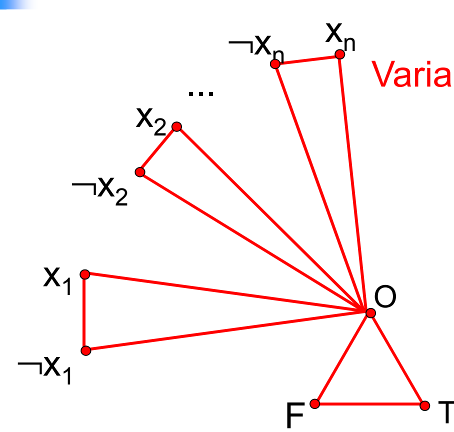
- Reduction:
  - We want to map a 3-CNF formula **F** to a graph **G** so that
    - **G** is 3-colorable iff **F** is satisfiable

3-SAT $\leq_P$ 3-Color



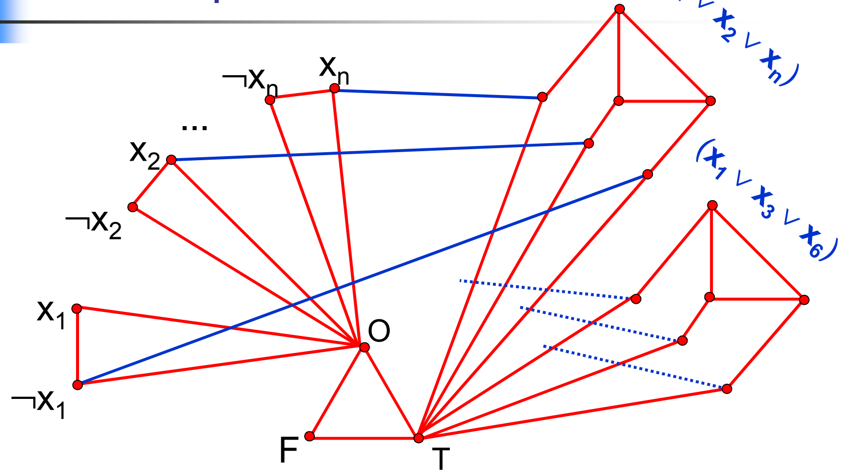
Base Triangle

3-SAT $\leq_P$ 3-Color



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

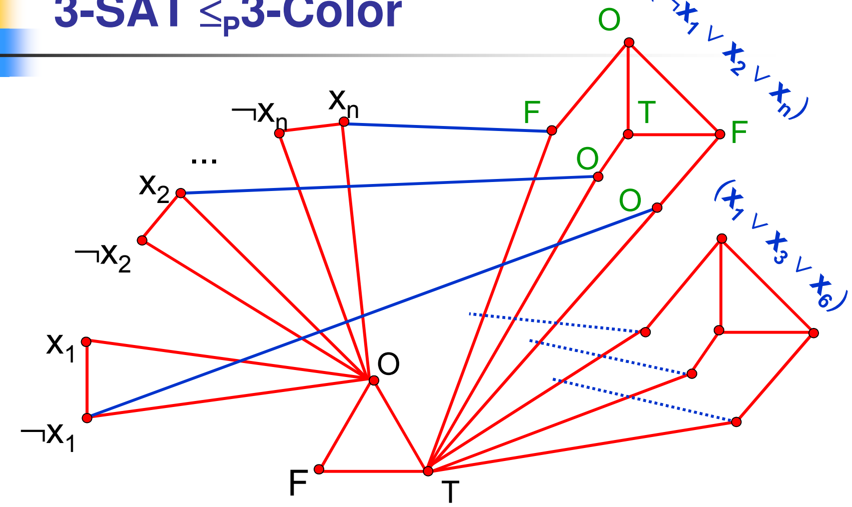
3-SAT $\leq_P$ 3-Color



Clause Part:

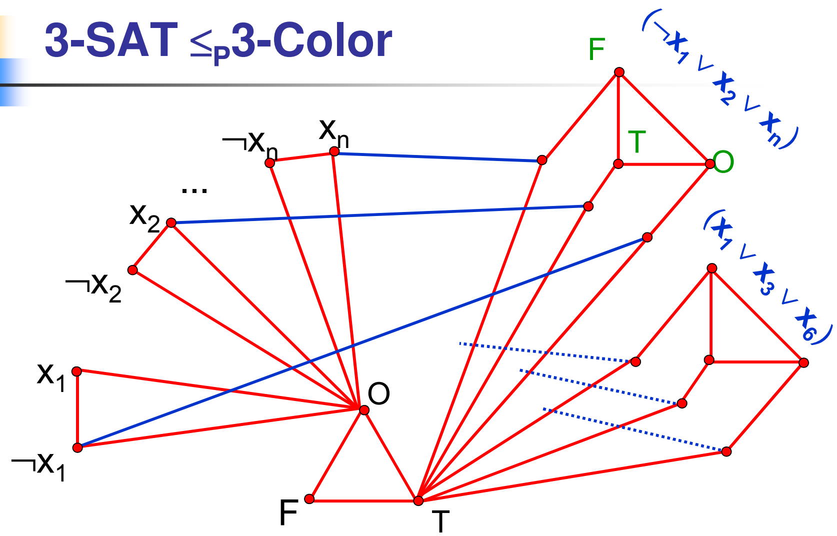
Add one 6 vertex gadget per clause connecting its 'outer vertices' to the literals in the clause

3-SAT $\leq_P$ 3-Color



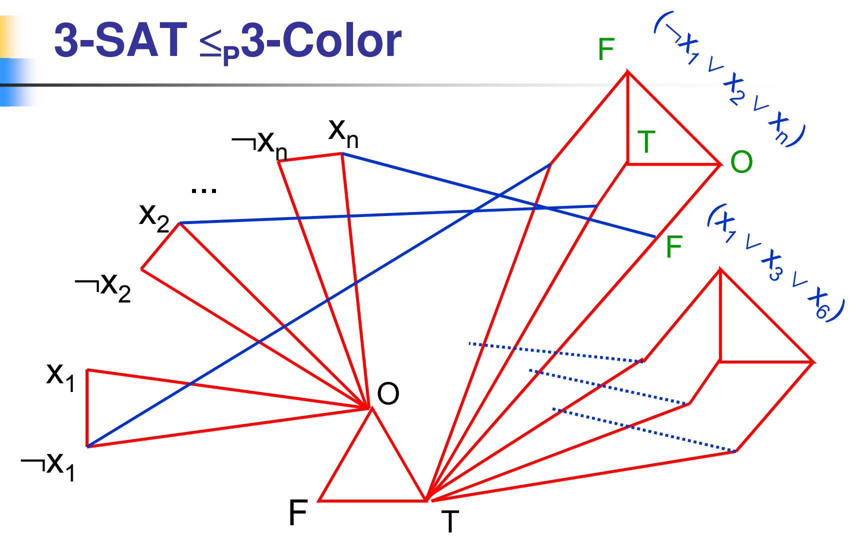
Any truth assignment satisfying the formula can be extended to a 3-coloring of the graph

3-SAT $\leq_P$ 3-Color



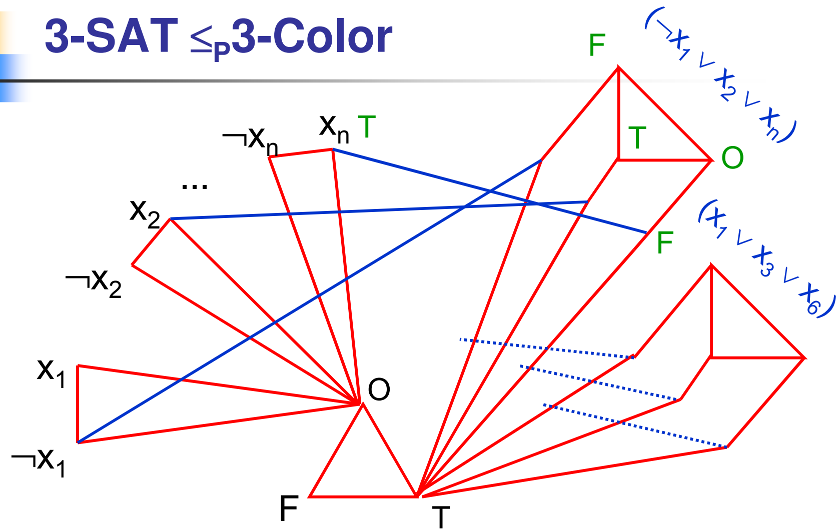
Any 3-coloring of the graph colors each gadget triangle using each color

3-SAT $\leq_P$ 3-Color



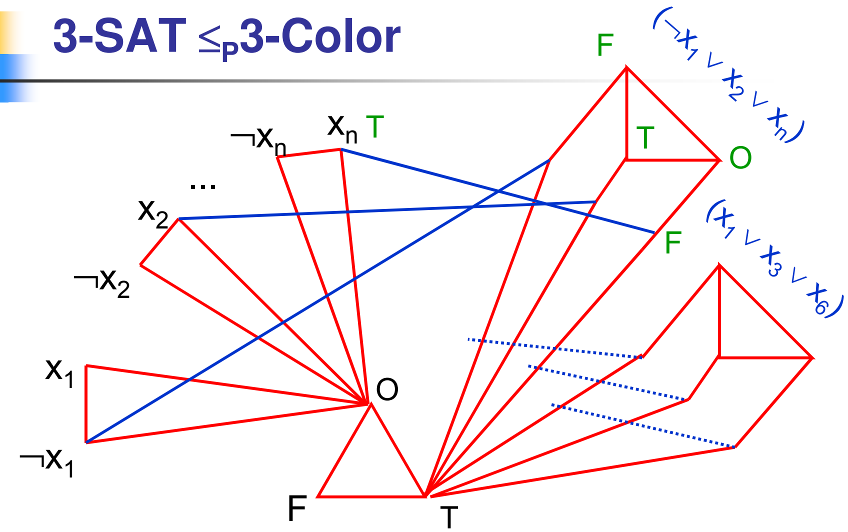
Any 3-coloring of the graph has an F opposite the O color in the triangle of each gadget

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

3-SAT $\leq_P$ 3-Color



Any 3-coloring of the graph yields a satisfying assignment to the formula

Matching Problems

- **Perfect Bipartite Matching**
  - Given a bipartite graph **G**=(**V**,**E**) where **V**=**X**∪**Y** and **E** ⊆ **X** × **Y**, is there a set **M** in **E** such that every vertex in **V** is in precisely one edge of **M** ?
- In **P**
  - Network Flow gives **O**(**nm**) algorithm where **n**=|**V**|, **m**=|**E**|.

3-Dimensional Matching

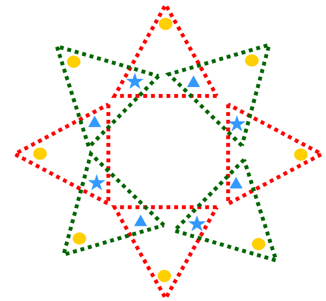
- **Perfect Bipartite Matching** is in **P**
  - Given a bipartite graph **G**=(**V**,**E**) where **V**=**X**∪**Y** and **E** ⊆ **X** × **Y**, is there a subset **M** in **E** such that every vertex in **V** is in precisely one edge of **M** ?
- **3-Dimensional Matching**
  - Given a tripartite hypergraph **G**=(**V**,**E**) where **V**=**X**∪**Y**∪**Z** and **E** ⊆ **X** × **Y**× **Z**, is there a subset **M** in **E** such that every vertex in **V** is in precisely one hyperedge of **M** ?
    - is in **NP:** Certificate is the set **M**

3-Dimensional Matching

- Theorem: **3-Dimensional Matching** is **NP**-complete
- Proof:
  - We've already seen that it is in **NP**
  - **3-Dimensional Matching** is **NP**-hard:
    - Reduction from **3-SAT**
    - Given a 3-CNF formula **F** we create a tripartite hypergraph ("hyperedges" are triangles) **G** based on **F** as follows

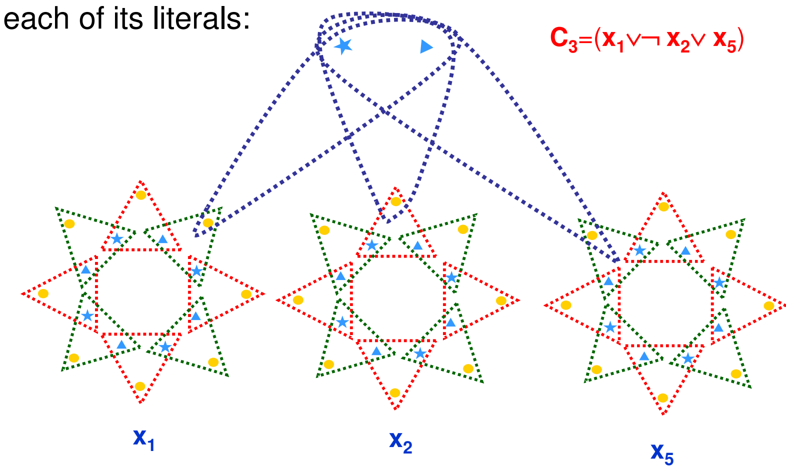
3-SAT $\leq_P$ 3-Dimensional Matching

- Variable part:
  - If variable x_i occurs r_i times in **F** create r_i red and r_i green triangles linked in a circle, one pair per occurrence
    - Perfect matching **M** must either use all the green edges leaving red tips uncovered (x_i is assigned false) or all the red edges leaving all green tips uncovered (x_i is assigned true)



3-SAT $\leq_P$ 3-Dimensional Matching

- Clause part: Two new nodes per clause joined to each of its literals:

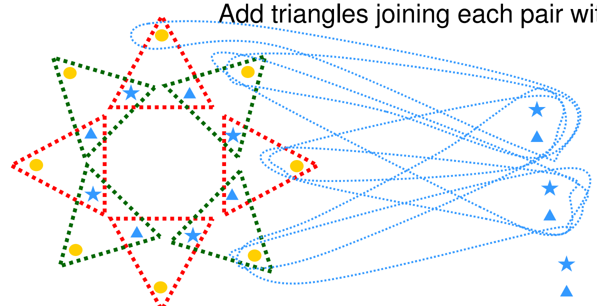


3-SAT $\leq_P$ 3-Dimensional Matching

- Slack: If there are **m** clauses then there are **3m** variable occurrences. That means **3m** total tips are not covered by whichever of red or green triangles not chosen. Of these, **m** are covered if each clause is satisfied. Need to cover the remaining **2m** tips.

Solution: Add **2m** pairs of slack vertices

Add triangles joining each pair with every tip!



3-SAT $\le_P$ 3-Dimensional Matching

- Well-formed: Each triangle has one of each type of node:

- Correctness:
  - If **F** has a satisfying assignment then choose the following triangles which form a perfect 3-dimensional matching in **G**:
    - Either the red or the green triangles in the cycle for $\mathbf{x_i}$ - the opposite of the assignment to $\mathbf{x_i}$
    - The triangle containing the first true literal for each clause and the two clause nodes
    - **2m** slack triangles one per new pair of nodes to cover all the remaining tips

3-SAT $\le_P$ 3-Dimensional Matching

- Correctness continued:
  - If **G** has a perfect 3-dimensional matching then:
    - Each blue node in the cycle for each $\mathbf{x_i}$ is contained in exactly two triangles, exactly one of which much be in **M**. If one triangle in the cycle is in **M**, the others must be the same color. We use the color not used to define the truth assignment to $\mathbf{x_i}$
    - The two nodes for any clause must be contained in an edge which must also contain a third node that corresponds to a literal made true by the truth assignment. Therefore the truth assignment satisfies **F** so it is satisfiable.