CSL 356: Analysis and Design of Algorithms

Ragesh Jaiswal

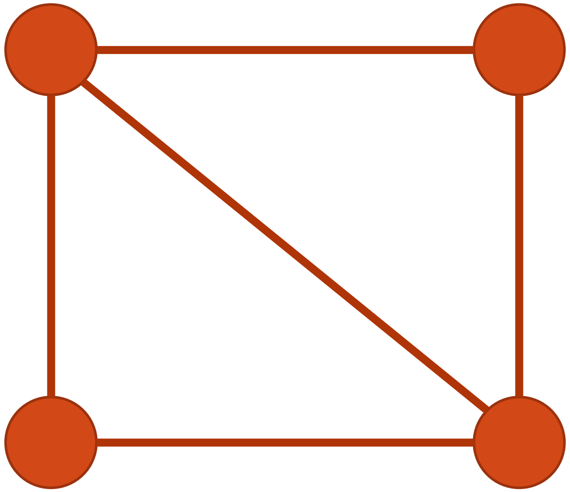
CSE, IIT Delhi

Computational Intractability

NP-complete problems: k-COLORING

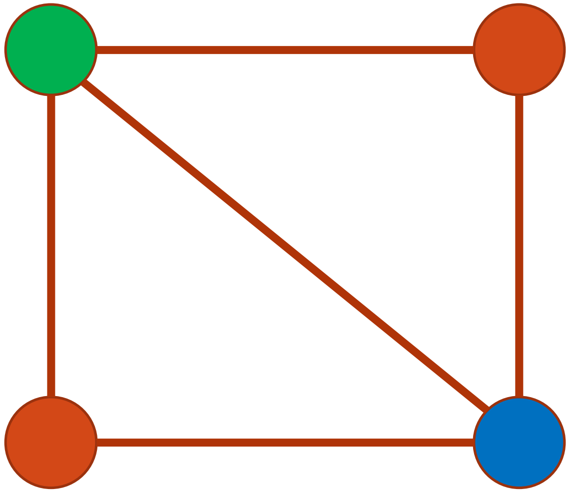
Computational Intractability: NP-complete problems

- Problem (k-coloring): Given a graph G, determine if G is k-colorable.
  - k-colorable: A graph is said to be k colorable if it is possible to assign one of k colors to each node such that for every edge (u, v), u and v are assigned different colors.



Computational Intractability: NP-complete problems

- Problem (k-coloring): Given a graph G, determine if G is k-colorable.
  - k-colorable: A graph is said to be k colorable if it is possible to assign one of k colors to each node such that for every edge (u, v), u and v are assigned different colors.



Computational Intractability: NP-complete problems

- Problem (2-coloring): Given a graph G, determine if G is 2-colorable.
- How hard is this problem?

Computational Intractability: NP-complete problems

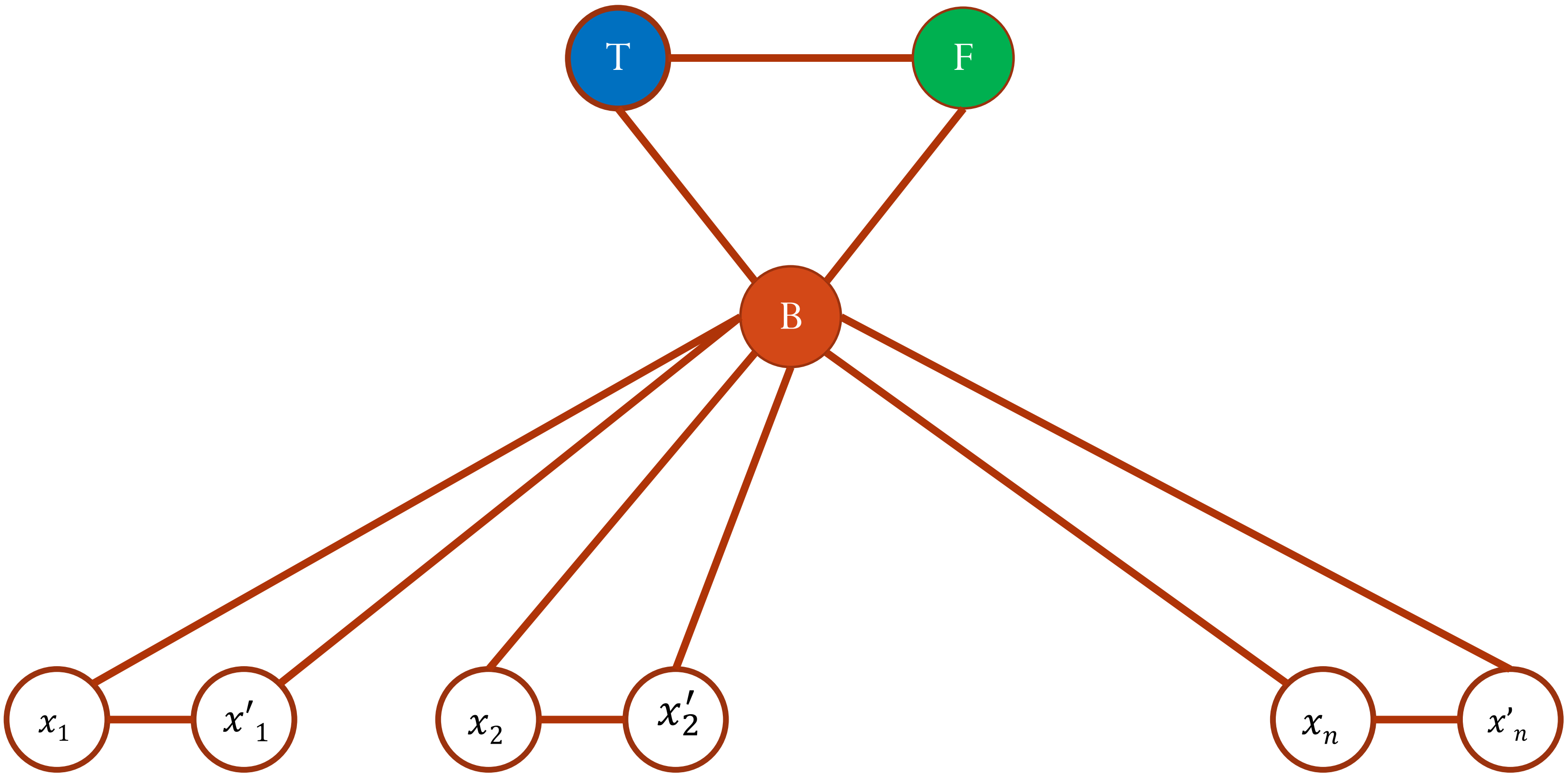
- Problem (2-coloring): Given a graph G, determine if G is 2-colorable.
- How hard is this problem?
- Claim: G is 2-colorable if and only if G is bipartite.

Computational Intractability: NP-complete problems

- Problem (3-coloring): Given a graph G, determine if G is 3-colorable.
- Claim 1: 3-coloring is **NP**-complete.
  - Claim 1.1: 3-coloring is in **NP**.
  - Claim 1.2: 3-SAT $\leq_p$ 3-coloring.

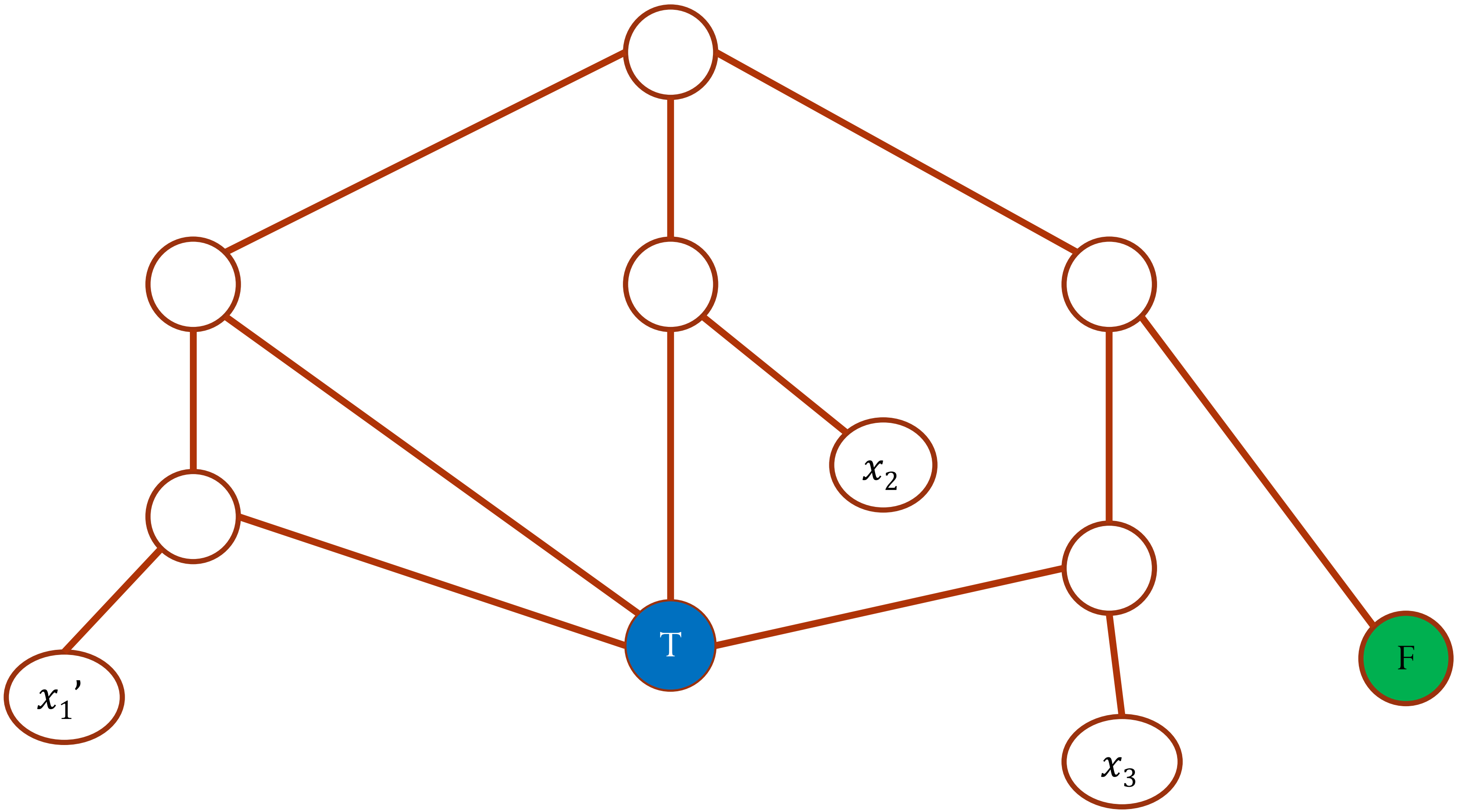
Computational Intractability: NP-complete problems

- Claim 1.2: 3-SAT $\leq_p$ 3-coloring.



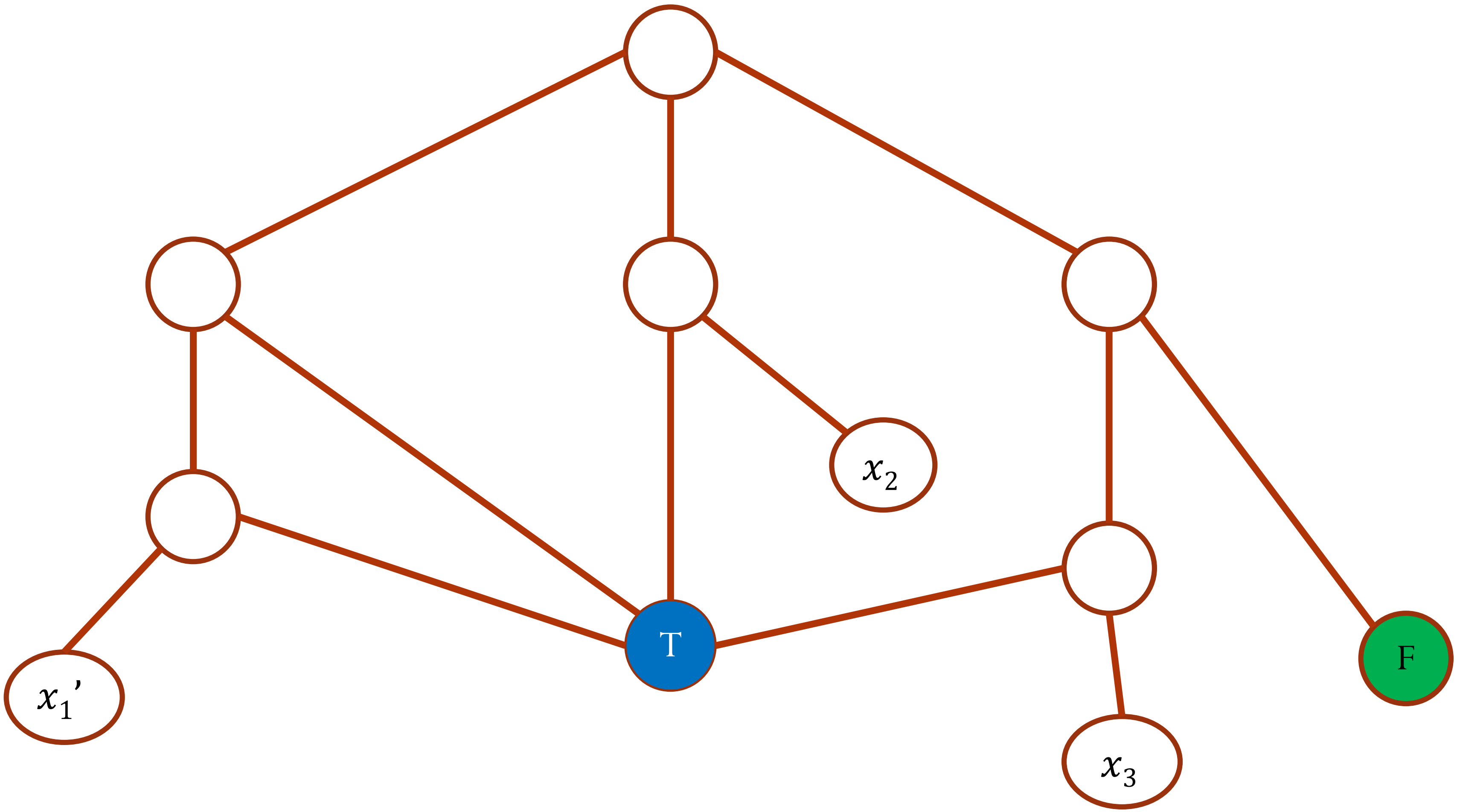
Computational Intractability: NP-complete problems

- Encoding $(x_1' \lor x_2 \lor x_3)$



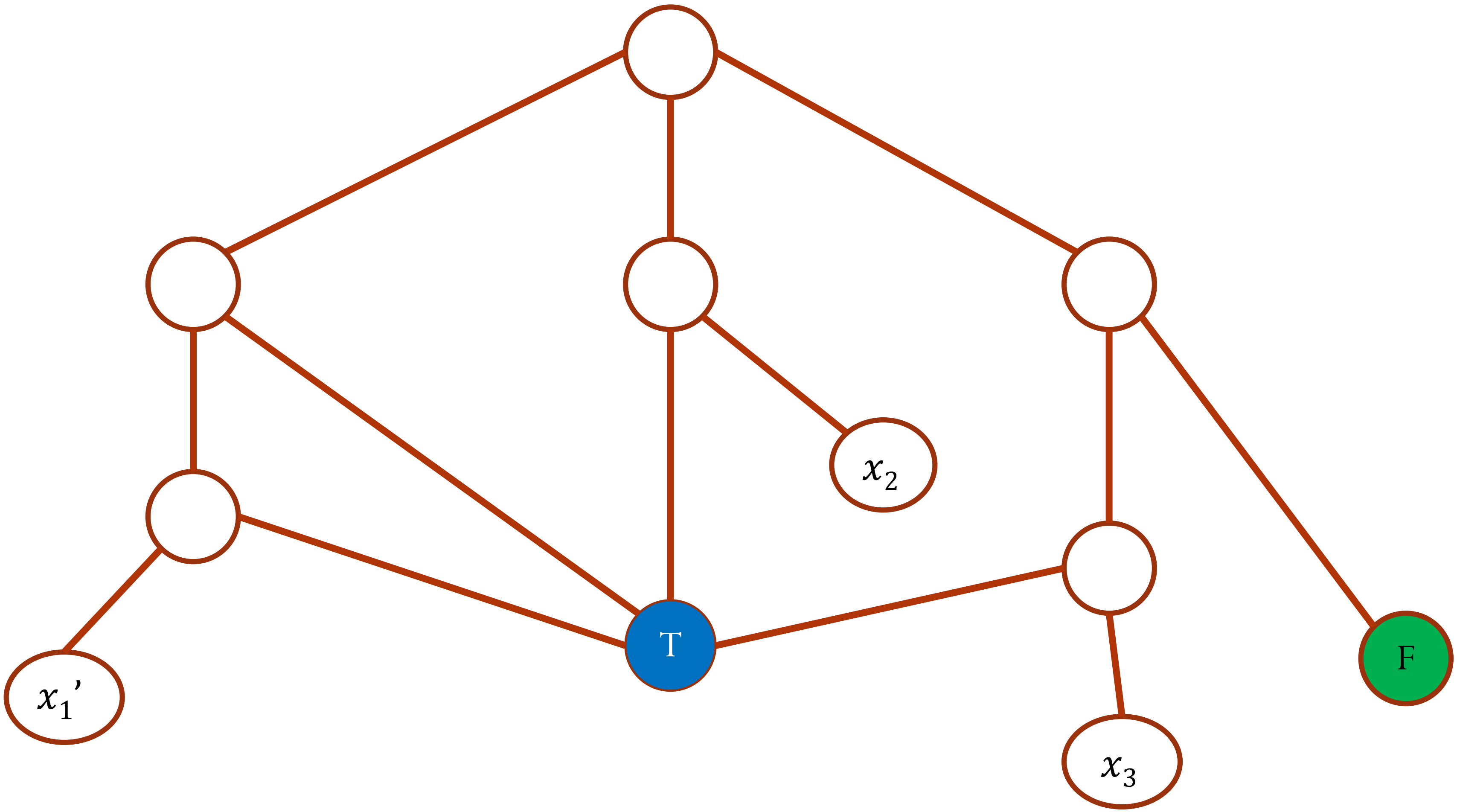
Computational Intractability: NP-complete problems

- Claim: There is no 3-coloring of the graph below with nodes x_1', x_2 and x_3 assigned F color.



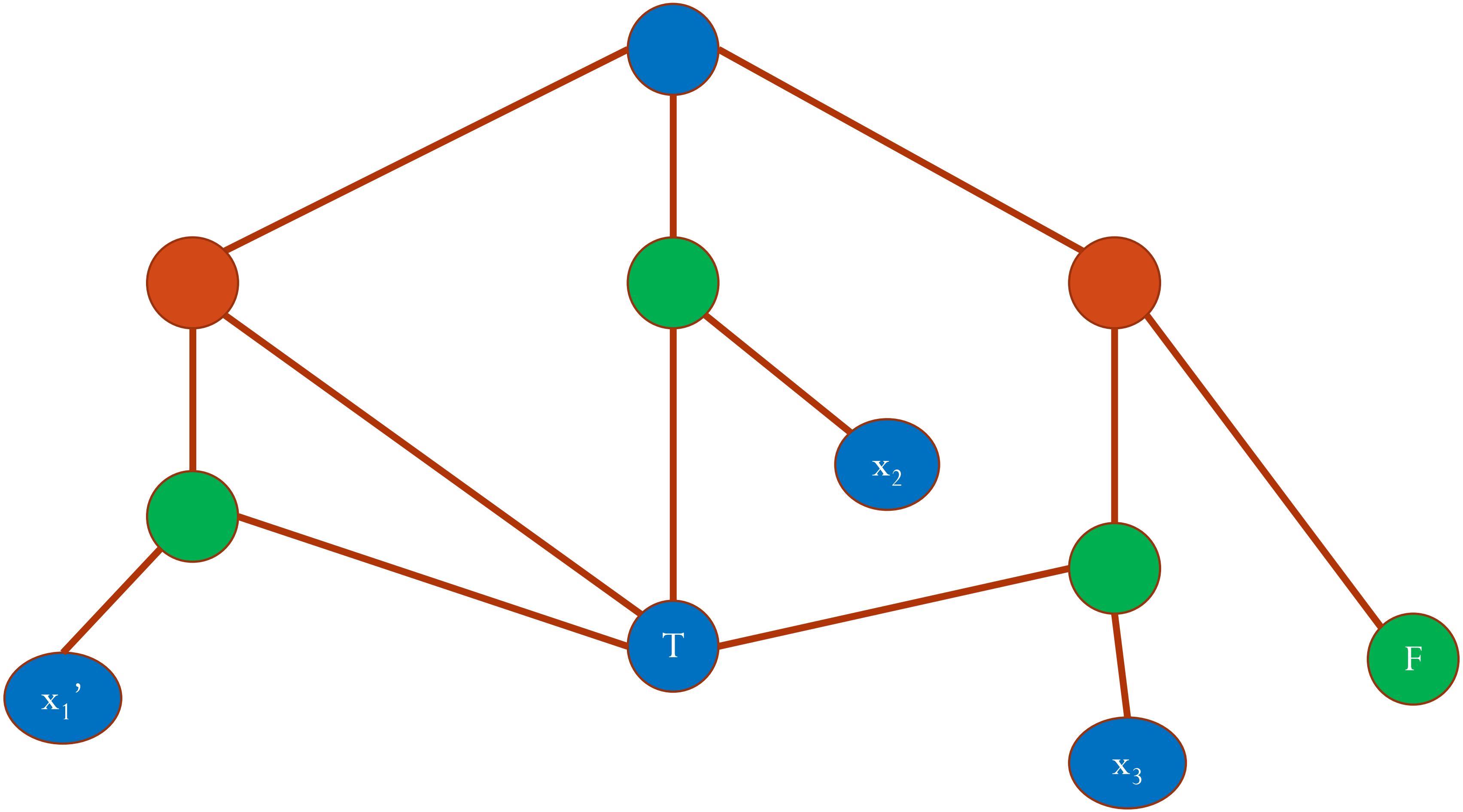
Computational Intractability: NP-complete problems

- Claim: There is a 3-coloring of the graph below with at least one of the nodes x_1', x_2 and x_3 assigned T color.



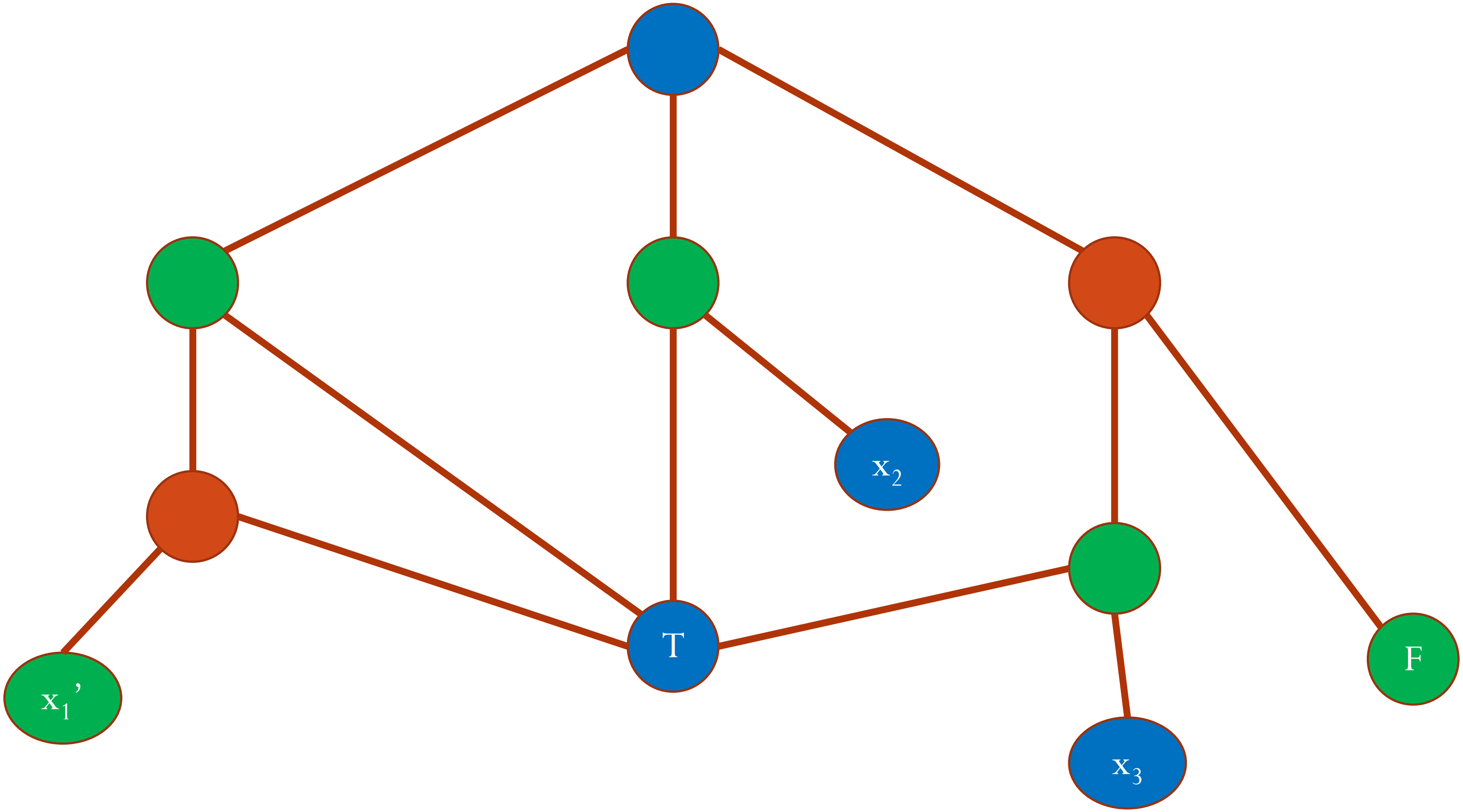
Computational Intractability: NP-complete problems

- x_1': T, x_2: T, x_3: T.



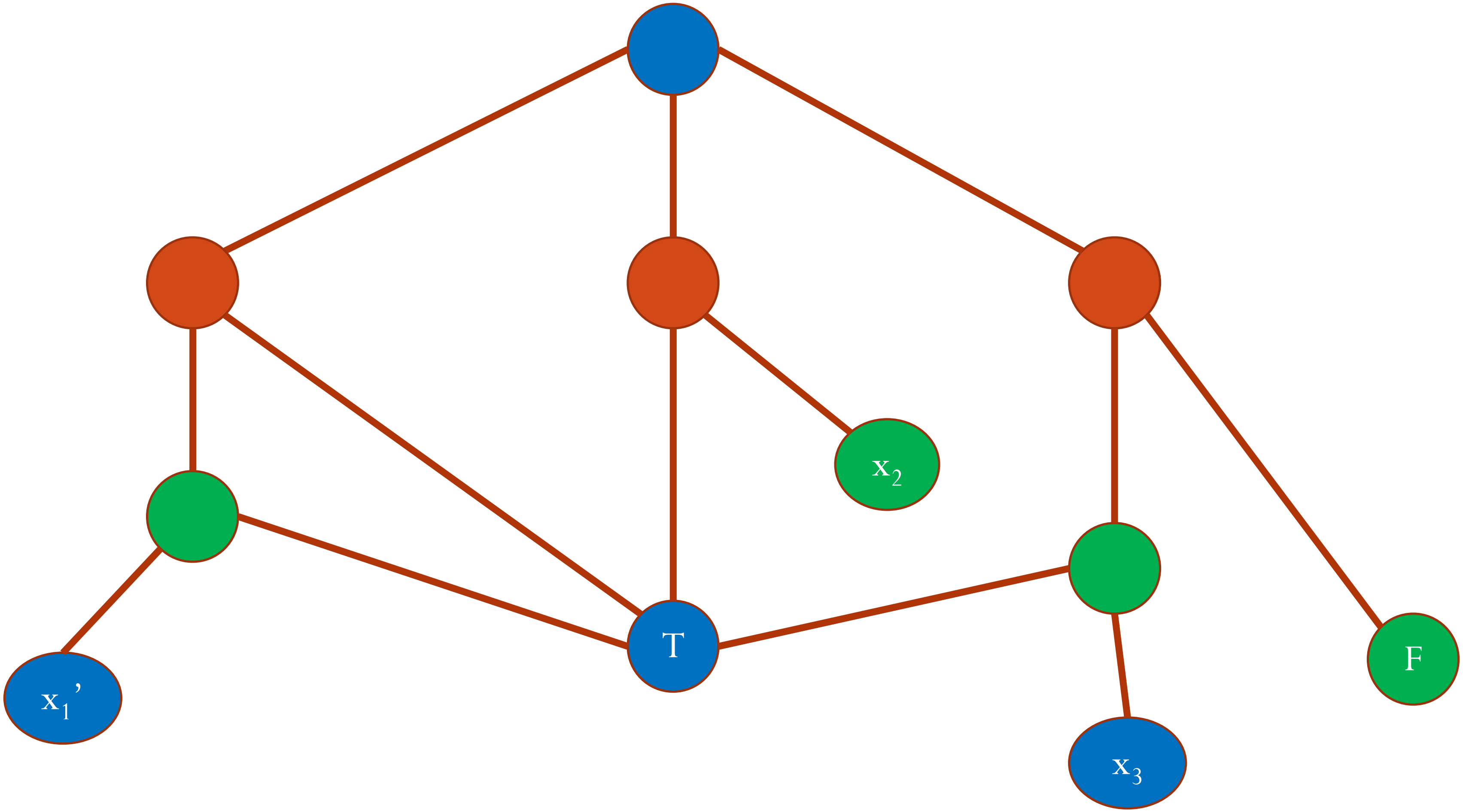
Computational Intractability: NP-complete problems

- $x_1{}'$: F, x_2: T, x_3: T.



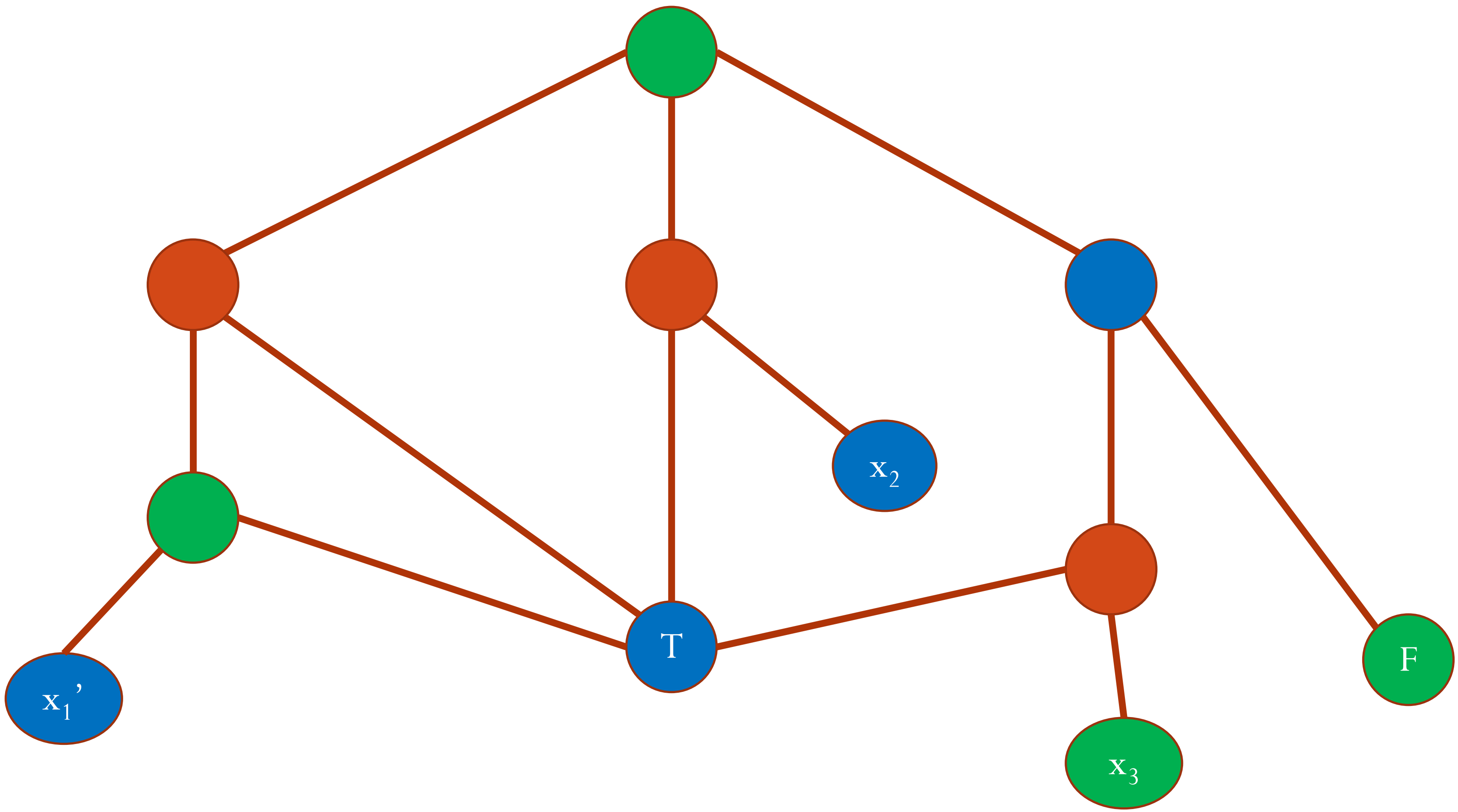
Computational Intractability: NP-complete problems

- x_1': T, x_2: F, x_3: T.



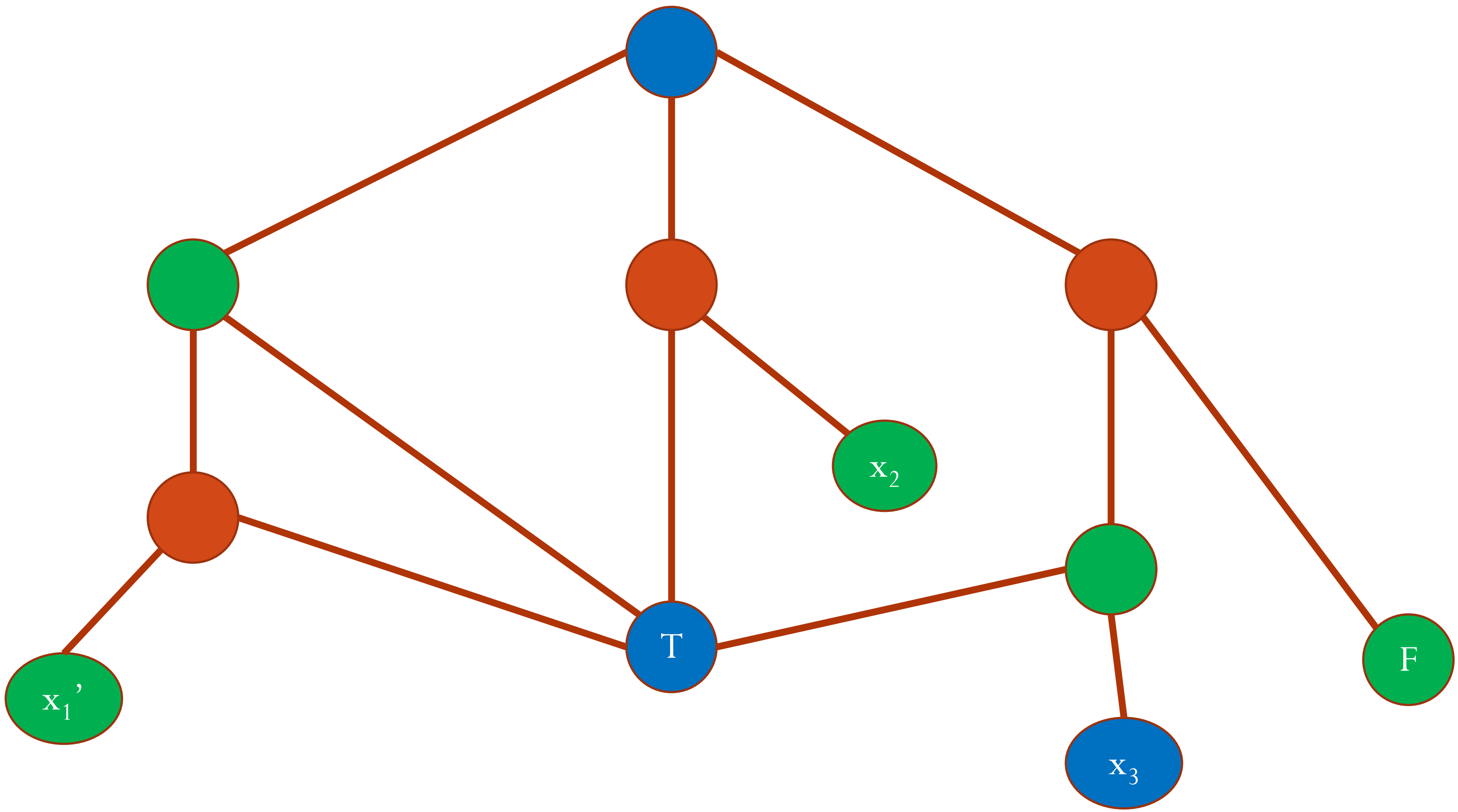
Computational Intractability: NP-complete problems

- x_1': T, x_2: T, x_3: F.



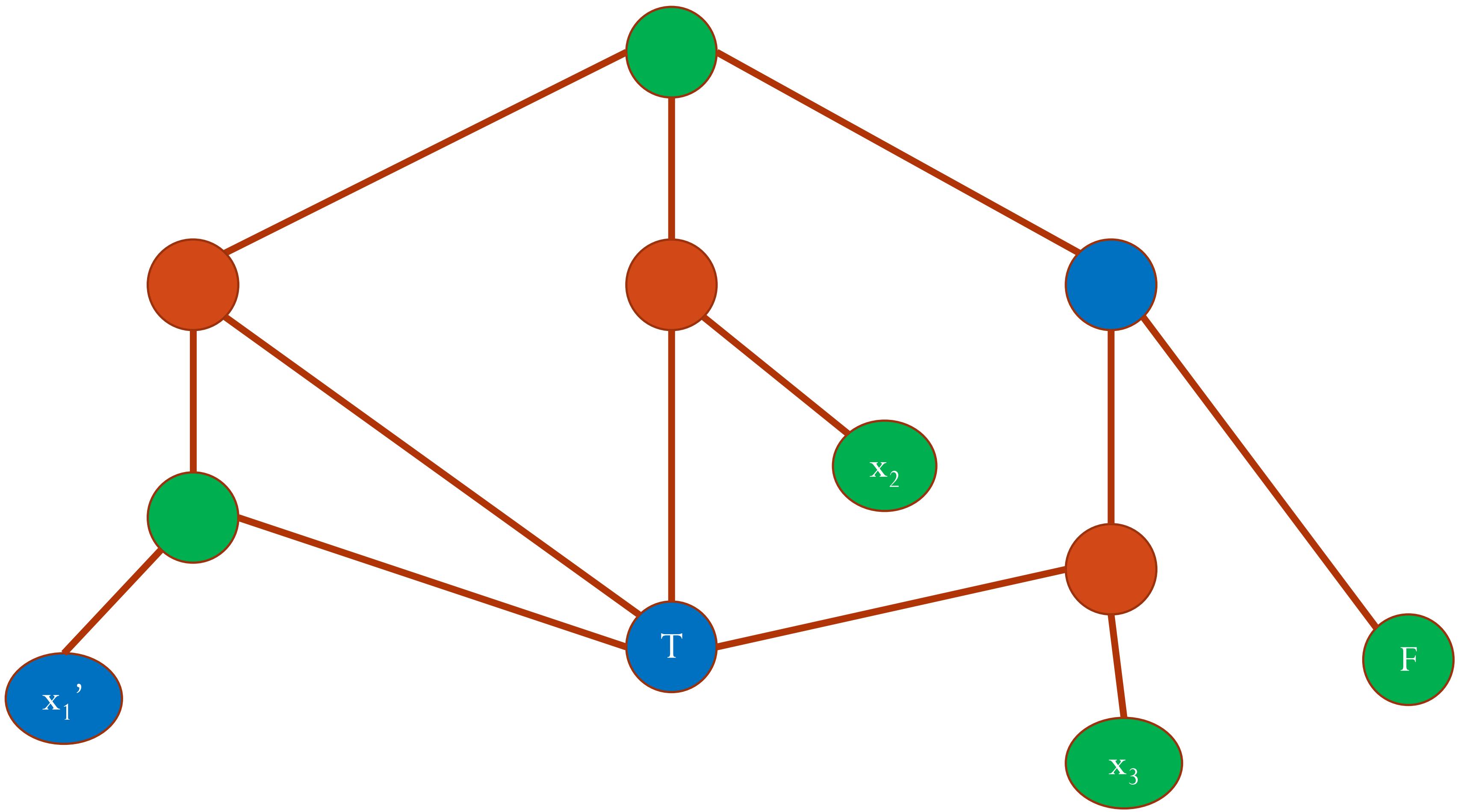
Computational Intractability: NP-complete problems

- x_1': F, x_2: F, x_3: T.



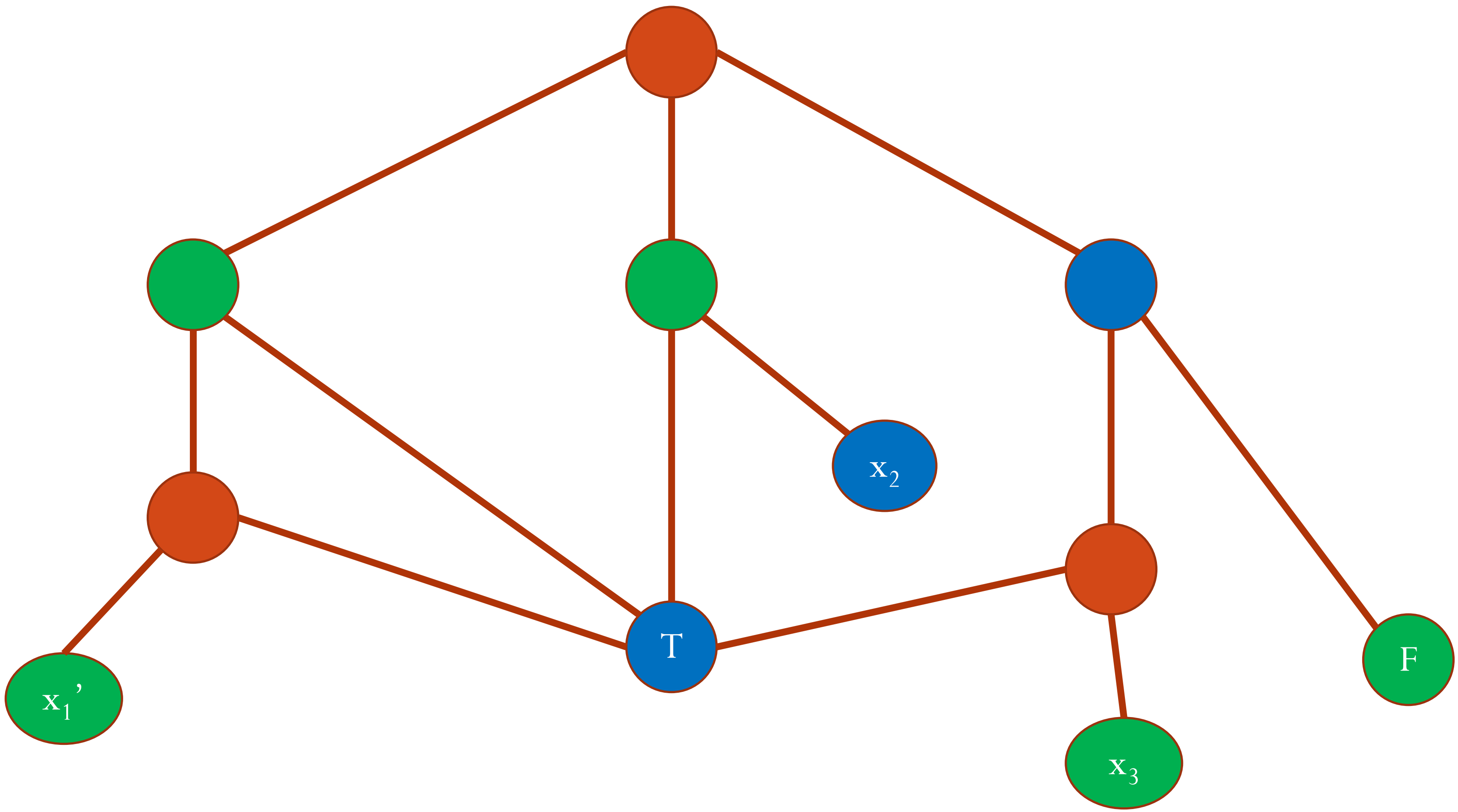
Computational Intractability: NP-complete problems

- x_1': T, x_2: F, x_3: F.



Computational Intractability: NP-complete problems

- x_1': F, x_2: T, x_3: F.



Computational Intractability: NP-complete problems

- Claim: The given formula is satisfiable if and only if the constructed graph has a 3 coloring.

Computational Intractability

NP-complete problems: SCHEDULING

Computational Intractability: NP-complete problems

- Problem(Subset-Sum): Given natural numbers $w_1, \ldots, w_n$ and a target number W, determine if there is a subset of $\{w_1, \ldots, w_n\}$ that adds up to precisely W.
- Problem(Scheduling): Given n jobs with start time s_i and duration t_i and deadline d_i, determine if all the jobs can be scheduled on a single machine such that no deadlines are missed.
- Claim 1: Subset-sum and Scheduling are in **NP**.
- Claim 2: Subset-sum $\leq_p$ Scheduling.

Computational Intractability: NP-complete problems

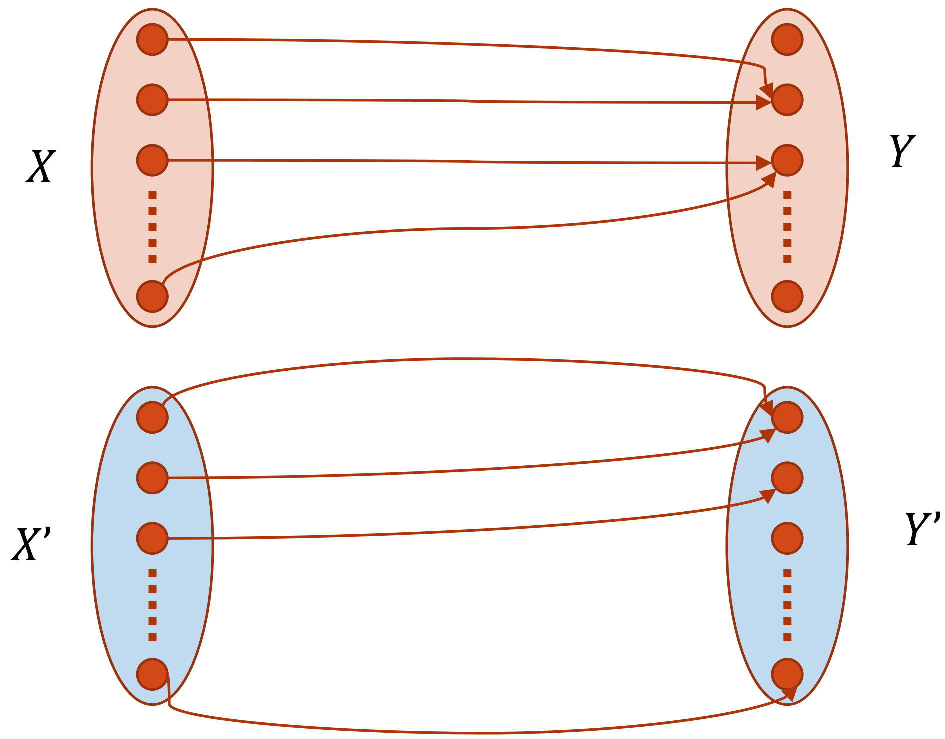
- Problem(Subset-Sum): Given natural numbers $w_1, \ldots, w_n$ and a target number W, determine if there is a subset of $\{w_1, \ldots, w_n\}$ that adds up to precisely W.
- Problem(Scheduling): Given n jobs with start time s_i and duration t_i and deadline d_i, determine if all the jobs can be scheduled on a single machine such that no deadlines are missed.
- Claim 1: Subset-sum and Scheduling are in **NP**.
- Claim 2: Subset-sum $\leq_p$ Scheduling.
  - Proof idea: Given an instance of the subset sum problem $(\{w_1, \ldots, w_n\}, W)$, we construct the following instance of the Scheduling problem: $((0, w_1, S+1), \ldots, (0, w_n, S+1), (W, 1, W+1))$. We then argue that there is a subset that sums to W if and only if the $(n+1)$ jobs can be scheduled. Here $S = w_1 + \cdots + w_n$.

Computational Intractability

NP and NP-completeness

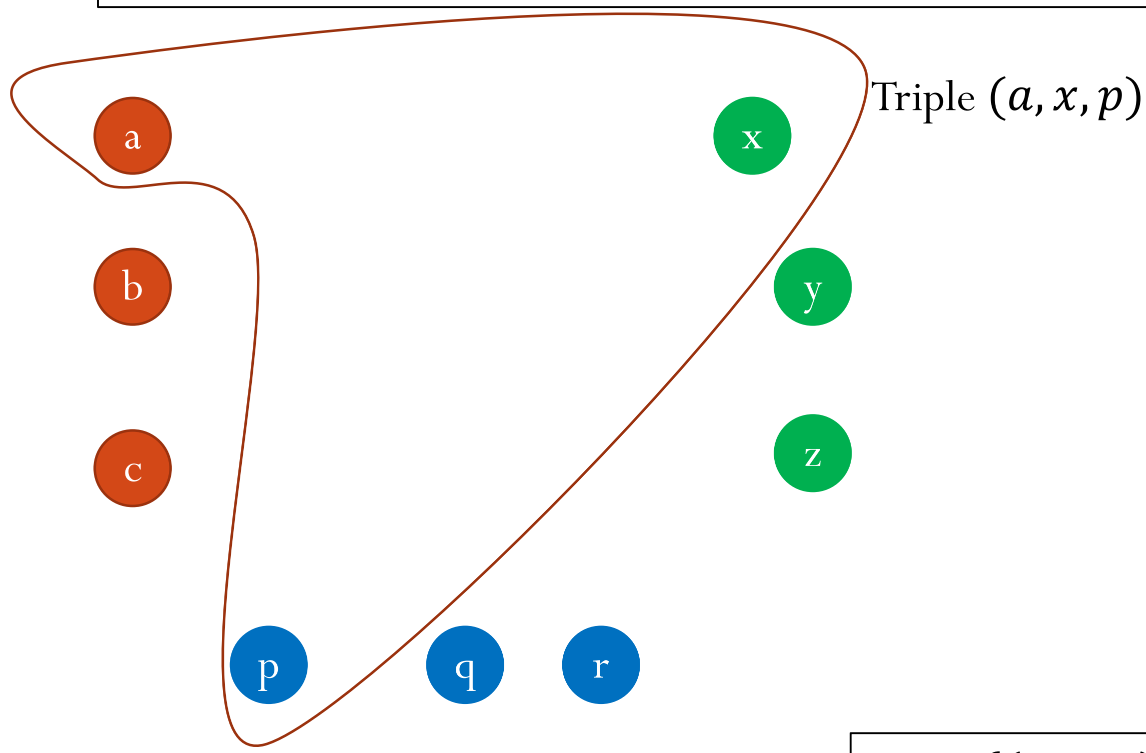
Computational intractability: Many-one reduction

- Suppose we want to show $X \leq_p Y$.
  - <u>Many-one reduction</u>: Design an efficient mapping f from the set of instances of X to set of instances of Y such that s is in X if and only if $f(s)$ is in Y.



Computational Intractability: NP-complete problems

- Problem(3-D matching): Given disjoint sets X, Y, and Z each of size n, and given a set T of triples (x, y, z), determine if there exist a subset of n triples in T such that each element of $X \cup Y \cup Z$ is contained in exactly one of these triples.



$$T = \{(a, x, p), (a, y, p), (b, y, q), (c, z, r)\}$$

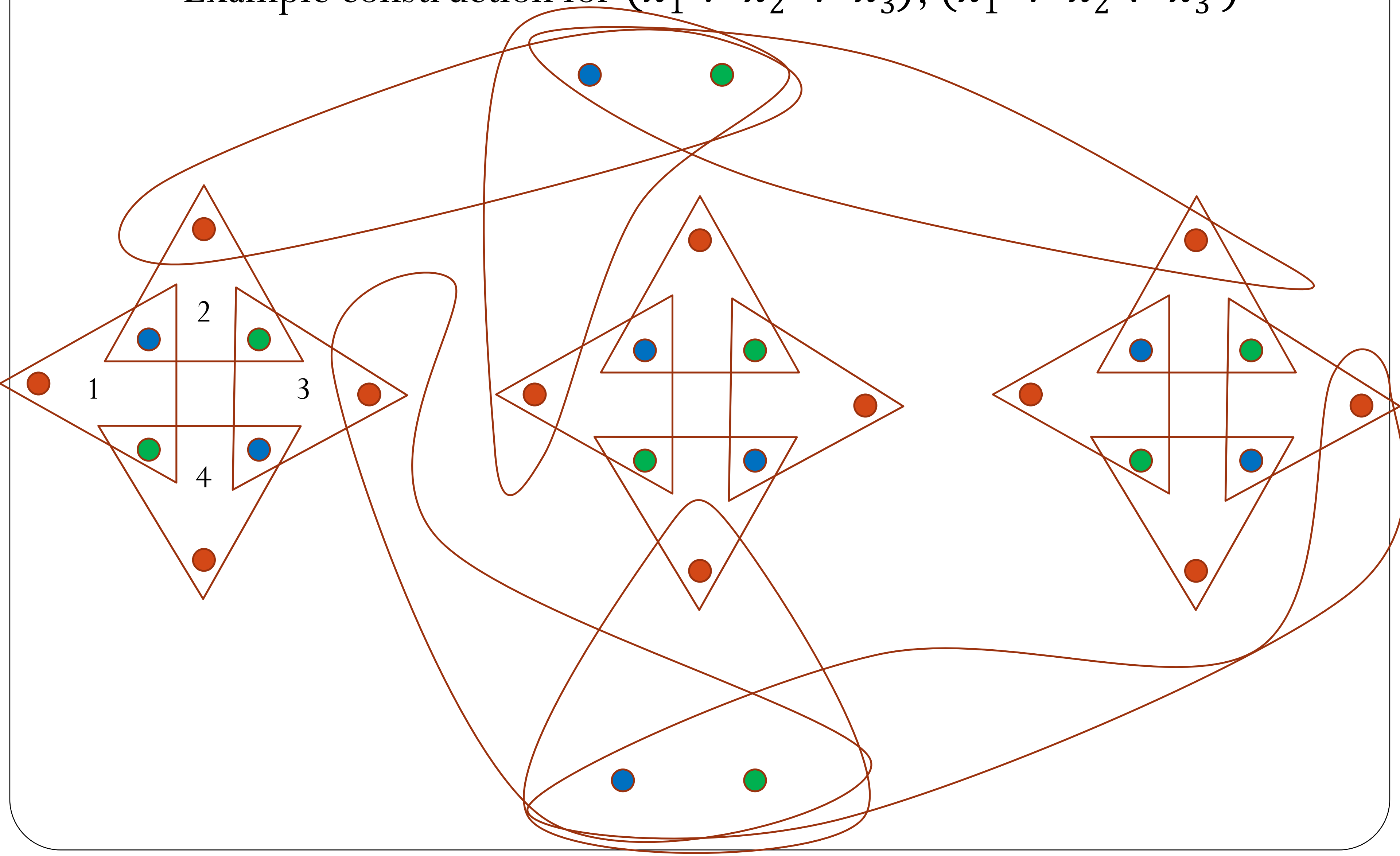
Computational Intractability: NP-complete problems

- Problem(3-D matching): Given disjoint sets X, Y, and Z each of size n, and given a set T of triples (x, y, z), determine if there exist a subset of n triples in T such that each element of $X \cup Y \cup Z$ is contained in exactly one of these triples.

- Claim 1: 3-D matching is in **NP**.
- Claim 2: 3-D matching is **NP**-complete.
  - Claim 2.1: 3-SAT $\leq_p$ 3-D matching.
  - Proof: We will show an efficient many-one reduction.

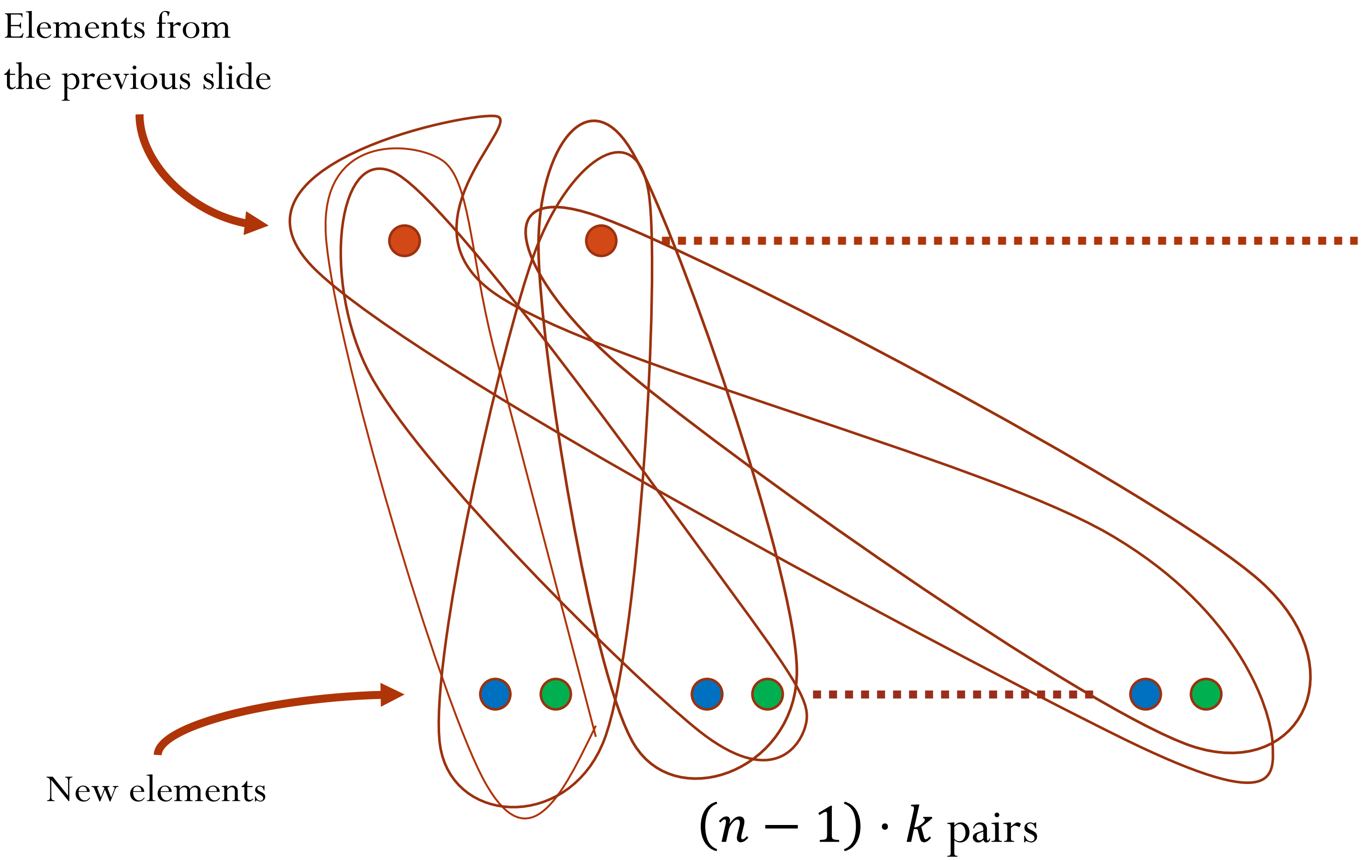
Computational Intractability: NP-complete problems

- Example construction for $(x_1 \lor x_2' \lor x_3), (x_1' \lor x_2 \lor x_3')$



Computational Intractability: NP-complete problems

- Example construction for $(x_1 \vee x_2' \vee x_3), (x_1' \vee x_2 \vee x_3')$



k denotes the number of Clauses

End