COL351: Analysis and Design of Algorithms

Ragesh Jaiswal, CSE, IITD

Computational Intractability

- Polynomial-time reduction:
  - Consider two problems X and Y.
  - Suppose there is a *black box* that solves arbitrary instances of problem X.
  - Suppose any arbitrary instance of problem Y can be solved using a polynomial number of standard computational steps and a polynomial number of calls to the black box that solves instance of problem X.
  - If the previous statement is true, then we say that Y is polynomial-time reducible to X. A short notation for this is $Y \leq_p X$.

Problem

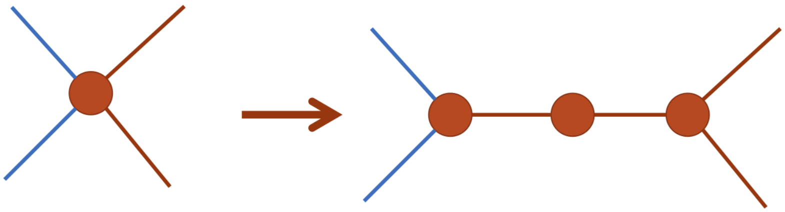
DEG-3-INDEPENDENT-SET: Given a graph $G = (V, E)$ of bounded degree 3 *(i.e., all vertices have degree* ≤ 3*)* and an integer k, check if there is an independent set of size at least k in G.

- Claim 1: INDEPENDENT-SET $\leq_p$ DEG-3-INDEPENDENT-SET

Problem

DEG-3-INDEPENDENT-SET: Given a graph $G = (V, E)$ of bounded degree 3 *(i.e., all vertices have degree* ≤ 3*)* and an integer k, check if there is an independent set of size at least k in G.

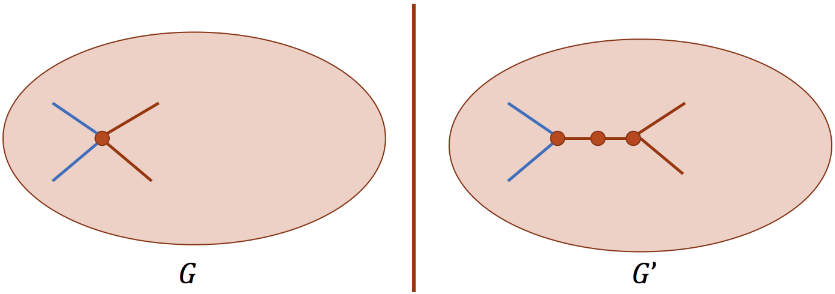
- Claim 1: INDEPENDENT-SET $\leq_p$ DEG-3-INDEPENDENT-SET
  - Idea: "Split" all vertices.



- Claim 1: INDEPENDENT-SET $\leq_p$ DEG-3-INDEPENDENT-SET

Proof of Claim 1

- Consider graph G' constructed by "splitting" a vertex of G.
- Claim 1.1: G has an independent set of size at least k if and only if G' has an independent set of size at least $(k+1)$.



Problem

SET-COVER: Given a set U of n elements, a collection $S_1, ..., S_m$ of subsets of U, and an integer k, determine if there exist a collection of at most k of these sets whose union is equal to U.

Problem

SET-COVER: Given a set U of n elements, a collection $S_1, ..., S_m$ of subsets of U, and an integer k, determine if there exist a collection of at most k of these sets whose union is equal to U.

- Claim 1: VERTEX-COVER $\leq_p$ SET-COVER.

Definition

- Boolean variables: 0-1 (true/false) variables.
- Term: A variable or its negation is called a term.
- Clause: Disjunction of terms (e.g., $(x_1 \vee \bar{x}_2 \vee x_3)$)
- Assignment: Fixing 0-1 values for each variables.
- Satisfying assignment: An assignment of variables is called a satisfying assignment for a collection of clauses if all clauses evaluate to 1 (true).
    - For example, $(x_1 \vee \bar{x}_2), (x_2 \vee \bar{x}_3), (x_3 \vee \bar{x}_1)$

Problem

SAT: Given a set of clauses $C_1, ..., C_m$ over a set of variables $x_1, ..., x_n$ determine if there exists a satisfying assignment.

Problem

SAT: Given a set of clauses $C_1, ..., C_m$ over a set of variables $x_1, ..., x_n$ determine if there exists a satisfying assignment.

Problem

3-SAT: Given a set of clauses $C_1, ..., C_m$ each of length at most 3, over a set of variables $x_1, ..., x_n$ determine if there exists a satisfying assignment.

Problem

SAT: Given a set of clauses $C_1, ..., C_m$ over a set of variables $x_1, ..., x_n$ determine if there exists a satisfying assignment.

Problem

3-SAT: Given a set of clauses $C_1, ..., C_m$ each of length at most 3, over a set of variables $x_1, ..., x_n$ determine if there exists a satisfying assignment.

- Claim 1: SAT $\leq_p$ 3-SAT

Problem

<u>SAT</u>: Given a set of clauses $C_1, ..., C_m$ over a set of variables $x_1, ..., x_n$ determine if there exists a satisfying assignment.

Problem

<u>3-SAT</u>: Given a set of clauses $C_1, ..., C_m$ each of length at most 3, over a set of variables $x_1, ..., x_n$ determine if there exists a satisfying assignment.

- <u>Claim 1</u>: SAT $\leq_p$ 3-SAT
  - <u>Main idea</u>: $(t_1 \vee t_2 \vee t_3 \vee t_4) \equiv ((t_1 \vee t_2 \vee z), (z \equiv t_3 \vee t_4))$

Problem

3-SAT: Given a set of clauses $C_1, ..., C_m$ each of length at most 3, over a set of variables $x_1, ..., x_n$ determine if there exists a satisfying assignment.

Problem

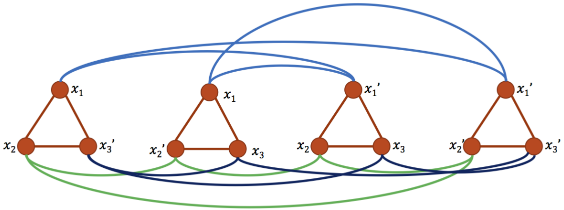
INDEPENDENT-SET: Given a graph $G = (V, E)$ and an integer k, check if there is an independent set of size at least k in G.

- Claim 1: 3-SAT $\leq_p$ INDEPENDENT-SET

- Claim 1: 3-SAT $\leq_p$ INDEPENDENT-SET

Proof sketch of Claim 1

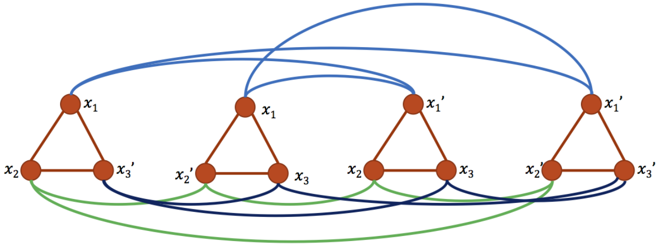
- Given an instance of the 3-SAT problem $(C_1, ..., C_m)$, we will construct an instance (G, m) of the INDEPENDENT-SET problem.
- We will then show that $(C_1, ..., C_m)$ has a satisfying assignment if and only if G has an independent set of size at least m.
- Consider an example construction:
  - 3-SAT instance: $(x_1 \vee x_2 \vee \bar{x}_3), (x_1 \vee \bar{x}_2 \vee x_3), (\bar{x}_1 \vee x_2 \vee x_3), (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3)$
  - INDEPENDENT-SET instance (G, m) for the above shown below:



- Claim 1: 3-SAT $\leq_p$ INDEPENDENT-SET

Proof sketch of Claim 1

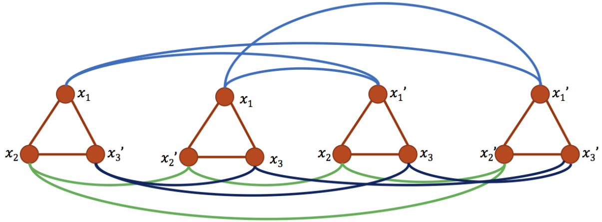
- Consider an example construction:
  - 3-SAT instance: $(x_1 \vee x_2 \vee \bar{x}_3), (x_1 \vee \bar{x}_2 \vee x_3), (\bar{x}_1 \vee x_2 \vee x_3), (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3)$
  - INDEPENDENT-SET instance (G, m) for the above shown below:
  - Claim 1.1: If (C_1, C_2, C_3, C_4) has a satisfying assignment, then G has an independent set of size 4.



- Claim 1: 3-SAT $\leq_p$ INDEPENDENT-SET

Proof sketch of Claim 1

- Consider an example construction:
  - 3-SAT instance:
    $(x_1 \vee x_2 \vee \bar{x}_3), (x_1 \vee \bar{x}_2 \vee x_3), (\bar{x}_1 \vee x_2 \vee x_3), (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3)$
  - INDEPENDENT-SET instance (G, m) for the above shown below:
  - Claim 1.1: If (C_1, C_2, C_3, C_4) has a satisfying assignment, then G has an independent set of size 4.
  - Claim 1.2: If G has an independent set of size 4, then (C_1, C_2, C_3, C_4) has a satisfying assignment.



- Claim 1: 3-SAT $\leq_p$ INDEPENDENT-SET
- Claim 2: SAT $\leq_p$ INDEPENDENT-SET

- Claim 1: 3-SAT $\leq_p$ INDEPENDENT-SET
- Claim 2: SAT $\leq_p$ INDEPENDENT-SET
  - Since SAT $\leq_p$ 3-SAT $\leq_p$ INDEPENDENT-SET

- Claim 1: 3-SAT $\leq_p$ INDEPENDENT-SET
- Claim 2: SAT $\leq_p$ INDEPENDENT-SET
  - Since SAT $\leq_p$ 3-SAT $\leq_p$ INDEPENDENT-SET
- Claim 3: SAT $\leq_p$ SET-COVER

- Claim 1: 3-SAT $\leq_p$ INDEPENDENT-SET
- Claim 2: SAT $\leq_p$ INDEPENDENT-SET
  - Since SAT $\leq_p$ 3-SAT $\leq_p$ INDEPENDENT-SET
- Claim 3: SAT $\leq_p$ SET-COVER
  - Since SAT $\leq_p$ 3-SAT $\leq_p$ INDEPENDENT-SET $\leq_p$ VERTEX-COVER $\leq_p$ SET-COVER

Computational Intractability: NP and NP-complete

- We said that the problems INDEPENDENT-SET, VERTEX-COVER, SAT seem hard.

- We said that the problems INDEPENDENT-SET, VERTEX-COVER, SAT seem hard.
- Polynomial-time reductions just give pair-wise relationships between problems.
- Is there a common theme that binds all these problems in one computational class?

- We said that the problems INDEPENDENT-SET, VERTEX-COVER, SAT seem hard.
- Polynomial-time reductions just give pair-wise relationships between problems.
- Is there a common theme that binds all these problems in one computational class?
- Let us try to extract a theme that is common to some of the problems we saw:
  - INDEPENDENT-SET: Given (G, k), determine if G has an independent set of size at least k.
  - VERTEX-COVER: Given (G, k), determine if G has a vertex cover of size at most k.
  - SAT: Given a Boolean formula Ω in CNF, determine if the formula is satisfiable.

- Let us try to extract a theme that is common to some of the problems we saw:
  - INDEPENDENT-SET: Given (G, k), determine if G has an independent set of size at least k.
    - Suppose there is an independent set of size at least k and someone gives such a subset as a certificate. Then we can verify this certificate quickly.
  - VERTEX-COVER: Given (G, k), determine if G has a vertex cover of size at most k.
    - Suppose there is a vertex cover of size at most k and someone gives such a subset as a certificate. Then we can verify this certificate quickly.
  - SAT: Given a Boolean formula Ω in CNF, determine if the formula is satisfiable.
    - Suppose the formula Ω is satisfiable and someone gives such a satisfying assignment as a certificate. Then we can verify this certificate quickly.

End