CSL 356: Analysis and Design of Algorithms

Ragesh Jaiswal

CSE, IIT Delhi

Polynomial-time reductions: Examples

Independent Set Vs Degree-3 Independent Set

Computational Intractability: Reduction

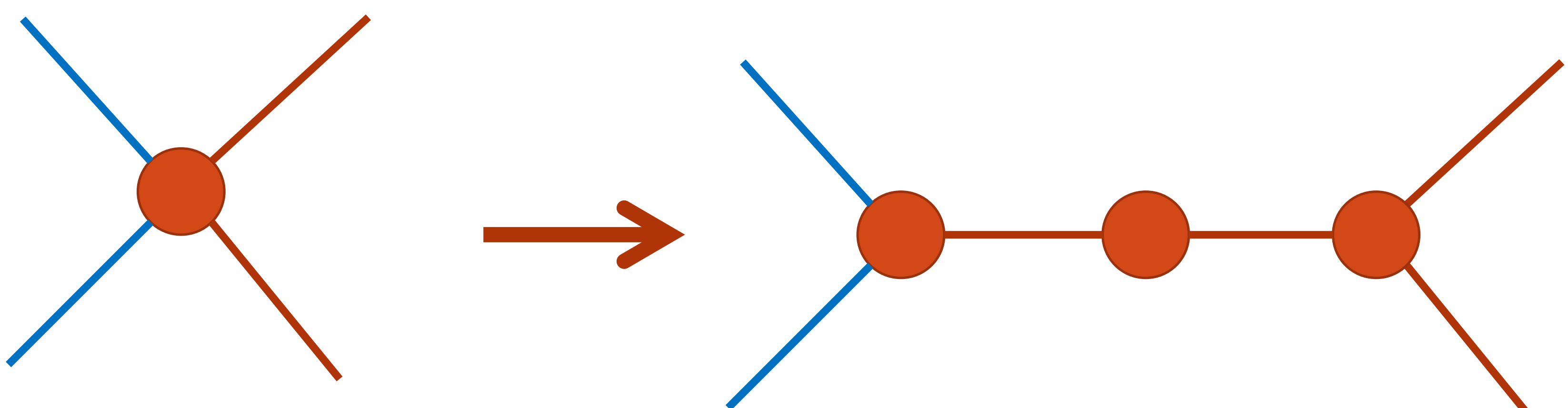
- Problem(Deg-3-Independent set): Given a graph $G = (V, E)$ of *bounded degree* 3 and an integer k, check if there is an *independent set* of size at least k in G.
  - *Graph with bounded degree 3*: A graph is said to have bounded degree 3 if the degrees of all vertices in the graph is at most 3.

- Claim: Independent-set $\leq_p$ Deg-3-Independent-set.

Computational Intractability: Reduction

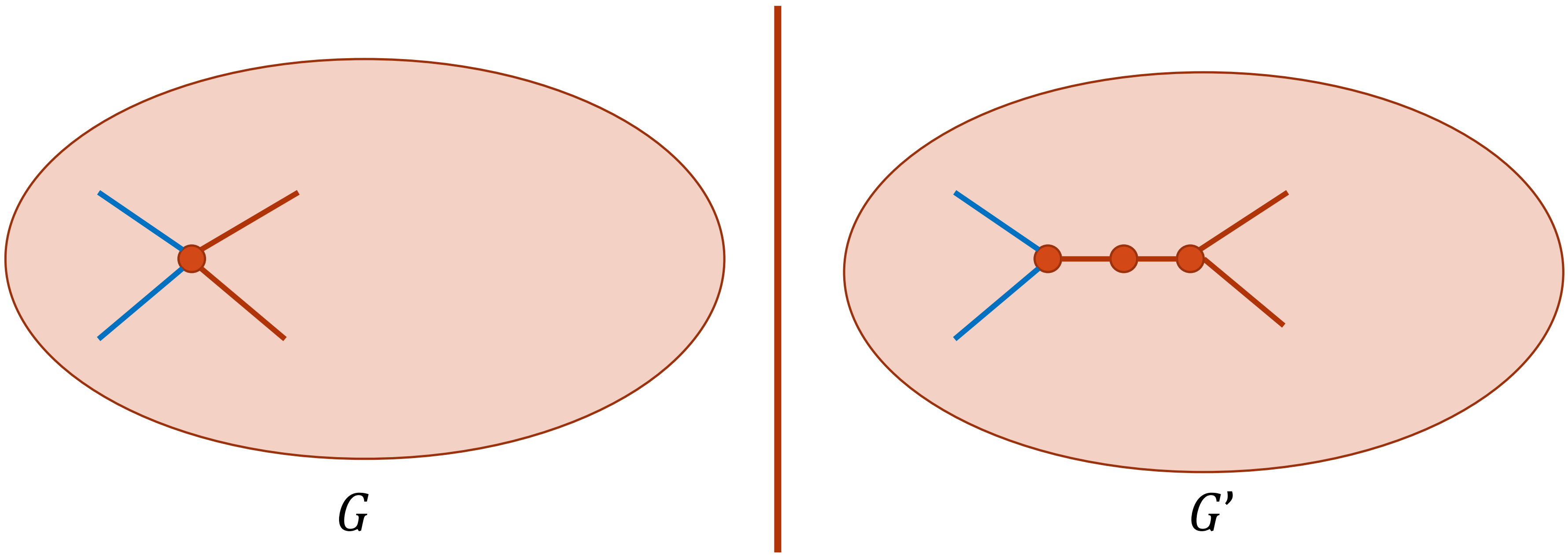
- Problem(Deg-3-Independent set): Given a graph $G = (V, E)$ of *bounded degree* 3 and an integer k, check if there is an *independent set* of size at least k in G.
  - *Graph with bounded degree 3*: A graph is said to have bounded degree 3 if the degrees of all vertices in the graph is at most 3.

- Claim: Independent-set $\leq_p$ Deg-3-Independent-set.



Computational Intractability: Reduction

- Claim: Independent-set $\leq_p$ Deg-3-Independent-set.



- Claim: G has an independent set of size at least k if and only if G' has an independent set of size at least $(k+1)$.

Polynomial-time reductions: Examples

Vertex-cover Vs Set-cover

Computational Intractability: Reduction

- Problem(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 of these sets whose union is equal to all of U?

Computational Intractability: Reduction

- Problem(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 of these sets whose union is equal to all of U?

- Claim: Vertex-cover $\leq_p$ Set-cover.

Polynomial-time reductions: Examples

Satisfiability: SAT Vs 3-SAT

Computational Intractability: Reduction

- Definitions:
  - Boolean variables: 0-1 (true/false) variables.
  - Term: A variable or its negation is called a *term*.
  - Clause: *Disjunction* of *terms* (e.g. $(x_1 \vee x_2' \vee x_3)$).
  - 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).
    - Example: $(x_1 \vee x_2'), (x_2 \vee x_3'), (x_3 \vee x_1')$
- Problem(SAT): Given a set of clauses $C_1, \ldots, C_m$ over a set of variables $x_1, \ldots, x_n$, does there exist a satisfying assignment?

Computational Intractability: Reduction

- Problem(SAT): Given a set of clauses $C_1, \ldots, C_m$ over a set of variables $x_1, \ldots, x_n$, does there exist a satisfying assignment?

- Problem(3-SAT): Given a set of clauses $C_1, \ldots, C_m$, each of length at most 3, over a set of variables $x_1, \ldots, x_n$, does there exist a satisfying assignment?

Computational Intractability: Reduction

- Problem(SAT): Given a set of clauses $C_1, \dots, C_m$ over a set of variables $x_1, \dots, x_n$, does there exist a satisfying assignment?

- Problem(3-SAT): Given a set of clauses $C_1, \dots, C_m$, each of length at most 3, over a set of variables $x_1, \dots, x_n$, does there exist a satisfying assignment?

- Claim: SAT $\leq_p$ 3-SAT

Computational Intractability: Reduction

- Problem(SAT): Given a set of clauses $C_1, \dots, C_m$ over a set of variables $x_1, \dots, x_n$, does there exist a satisfying assignment?

- Problem(3-SAT): Given a set of clauses $C_1, \dots, C_m$, each of length at most 3, over a set of variables $x_1, \dots, x_n$, does there exist a satisfying assignment?

- Claim: SAT $\leq_p$ 3-SAT
  - Proof: Main idea:
    - $(t_1 \vee t_2 \vee t_3 \vee t_4) \equiv ((t_1 \vee t_2 \vee z), (z \equiv t_3 \vee t_4))$

End