Polynomial Time Reduction

- Polynomial Time Reduction Definition
- Reduction by Equivalence
- Reduction from Special Cases to General Case
- Reduction by Encoding with Gadgets
- Transitivity of Reductions
- Decision, Search and Optimization Problem
- Self-Reducibility

Imdad ullah Khan

Polynomial Time Reduction

Problem A is polynomial time reducible to Problem B, $A \leq_p B$

If any instance of problem A can be solved using a polynomial amount of computation plus a polynomial number of calls to a solution of problem B

If any algorithm for problem B can be used [called (once or more) with *'clever'* legal inputs] to solve any instance of problem A

Subroutine for B takes an instance y of B and returns the solution $B(y)$

Algorithm for A transforms an instance x of A to an instance y of B. Then transforms $B(y)$ to $A(x)$

Reduction by encoding with gadgets

$\textsc{3-sat}(f) \quad \leq_p \quad \textsc{independent-set}(G, k)$

$$f = (x_{11} \vee x_{12} \vee x_{13}) \wedge (x_{21} \vee x_{22} \vee x_{23}) \wedge \dots \quad \dots \wedge (x_{m1} \vee x_{m2} \vee x_{m3})$$

We need to set each of $x_1, \dots, x_n$ to $0/1$ so as $f = 1$

Alternatively,

1. We need to pick a literal from each clause and set it to 1
2. But we cannot make conflicting settings

Reduction by encoding with gadgets

$$3\text{-SAT}(f) \quad \leq_p \quad \text{INDEPENDENT-SET}(G, k)$$

Given f on n variables and m clauses - Make a graph G as follows

- For each clause make a triangle with nodes labeled with literals
- For clauses with 2 and 1 literal make an edge or a node
- Make edges between literals appearing in different clauses as complements

$$(x_{11} \vee x_{12} \vee x_{13}) \wedge \ \ldots \ \wedge (x_{i1} \vee x_{i2} \vee x_{i3}) \wedge \ \ldots \ \wedge (x_{j1} \vee x_{j2} \vee x_{j3}) \wedge \ \ldots \ \wedge (x_{m1} \vee x_{m2} \vee x_{m3})$$

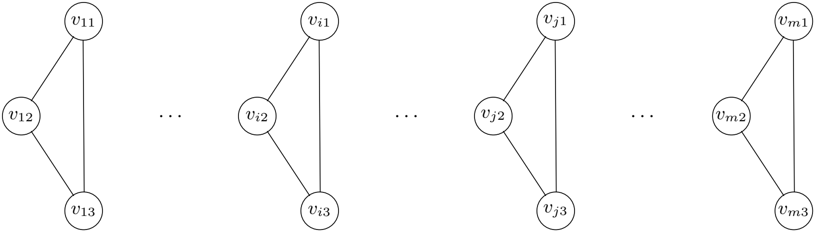
Reduction by encoding with gadgets

$$3\text{-}\textsc{sat}(f) \quad \leq_p \quad \textsc{independent-set}(G, k)$$

Given f on n variables and m clauses - Make a graph G as follows

- For each clause make a triangle with nodes labeled with literals
- For clauses with 2 and 1 literal make an edge or a node
- Make edges between literals appearing in different clauses as complements

$$(x_{11} \vee x_{12} \vee x_{13}) \wedge \ \ldots \ \wedge (x_{i1} \vee x_{i2} \vee x_{i3}) \wedge \ \ldots \ \wedge (x_{j1} \vee x_{j2} \vee x_{j3}) \wedge \ \ldots \ \wedge (x_{m1} \vee x_{m2} \vee x_{m3})$$



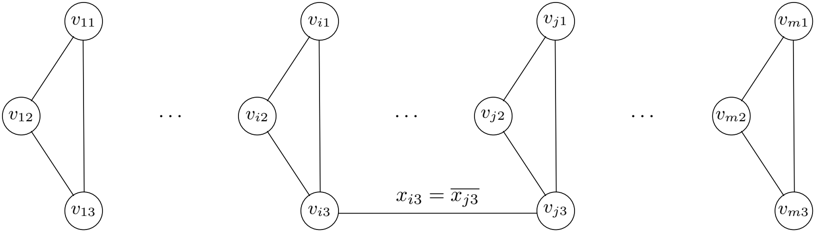
Reduction by encoding with gadgets

$3\text{-SAT}(f) \quad \leq_p \quad \text{INDEPENDENT-SET}(G, k)$

Given f on n variables and m clauses - Make a graph G as follows

- For each clause make a triangle with nodes labeled with literals
- For clauses with 2 and 1 literal make an edge or a node
- Make edges between literals appearing in different clauses as complements

$$(x_{11} \vee x_{12} \vee x_{13}) \wedge \ \ldots \ \wedge (x_{i1} \vee x_{i2} \vee x_{i3}) \wedge \ \ldots \ \wedge (x_{j1} \vee x_{j2} \vee x_{j3}) \wedge \ \ldots \ \wedge (x_{m1} \vee x_{m2} \vee x_{m3})$$



Reduction by encoding with gadgets

3-SAT(f) $\leq_p$ INDEPENDENT-SET(G, k)

- Given f on n variables and m clauses - Make a graph G
- For each clause make a triangle with nodes labeled with literals
- For clauses with 2 and 1 literal make an edge or a node
- Make edges between literals appearing in different clauses as complements

Theorem: f is satisfiable iff G has an independent set of size m

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

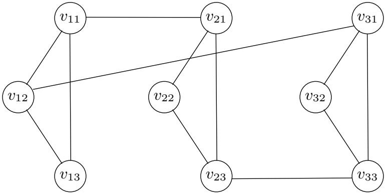
Reduction by encoding with gadgets

$$3\text{-SAT}(f) \quad \leq_p \quad \text{INDEPENDENT-SET}(G, k)$$

- Given f on n variables and m clauses - Make a graph G
- For each clause make a triangle with nodes labeled with literals
- For clauses with 2 and 1 literal make an edge or a node
- Make edges between literals appearing in different clauses as complements

Theorem: f is satisfiable iff G has an independent set of size m

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



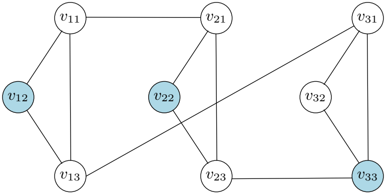
Reduction by encoding with gadgets

$3\text{-SAT}(f) \quad \leq_p \quad \text{INDEPENDENT-SET}(G,k)$

- Given f on n variables and m clauses - Make a graph G
- For each clause make a triangle with nodes labeled with literals
- For clauses with 2 and 1 literal make an edge or a node
- Make edges between literals appearing in different clauses as complements

Theorem: *f is satisfiable iff G has an independent set of size m*

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



$$x_1 = 1, \overline{x_3} = 1, \overline{x_4} = 1$$

Reduction by encoding with gadgets

$3\text{-SAT}(f) \quad \leq_p \quad \text{INDEPENDENT-SET}(G, k)$

- Given f on n variables and m clauses - Make a graph G
- For each clause make a triangle with nodes labeled with literals
- For clauses with 2 and 1 literal make an edge or a node
- Make edges between literals appearing in different clauses as complements

Theorem: f is satisfiable iff G has an independent set of size m

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

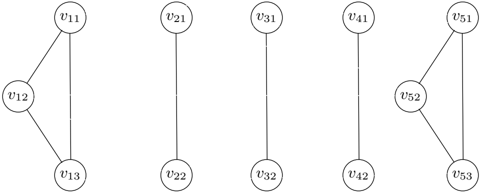
Reduction by encoding with gadgets

$3\text{-SAT}(f) \leq_p \text{INDEPENDENT-SET}(G, k)$

- Given f on n variables and m clauses - Make a graph G
- For each clause make a triangle with nodes labeled with literals
- For clauses with 2 and 1 literal make an edge or a node
- Make edges between literals appearing in different clauses as complements

Theorem: f is satisfiable iff G has an independent set of size m

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



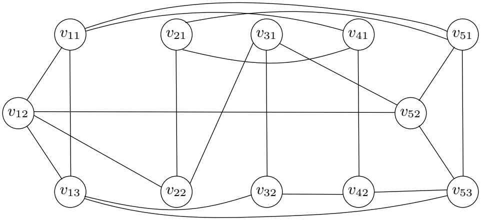
Reduction by encoding with gadgets

$3\text{-SAT}(f) \quad \leq_p \quad \text{INDEPENDENT-SET}(G, k)$

- Given f on n variables and m clauses - Make a graph G
- For each clause make a triangle with nodes labeled with literals
- For clauses with 2 and 1 literal make an edge or a node
- Make edges between literals appearing in different clauses as complements

Theorem: f is satisfiable iff G has an independent set of size m

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



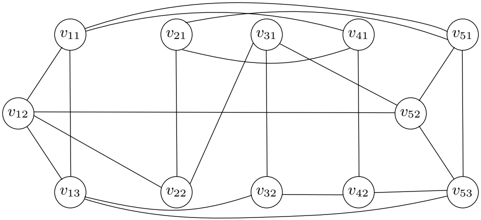
Reduction by encoding with gadgets

$$3\text{-SAT}(f) \leq_p \text{INDEPENDENT-SET}(G, k)$$

- Given f on n variables and m clauses - Make a graph G
- For each clause make a triangle with nodes labeled with literals
- For clauses with 2 and 1 literal make an edge or a node
- Make edges between literals appearing in different clauses as complements

Theorem: f is satisfiable iff G has an independent set of size m

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



No satisfying assignmnet, No independent set of size 5

Reduction by encoding with gadgets

$$3\text{-}\textsc{sat}(f) \;\; \leq_p \;\; \textsc{independent-set}(G, k)$$

Theorem: f is satisfiable iff G has an independent set of size m

Let $\mathcal{A}$ be an algorithm for the $\textsc{independent-set}(G, k)$ problem

We will use $\mathcal{A}$ to solve the $3\text{-}\textsc{sat}(f)$ problem

Given any instance f of $3\text{-}\textsc{sat}(f)$ on n variables and m clauses

- Construct the graph as outlined above
- Call $\mathcal{A}$ on $[G, m]$
- if $\mathcal{A}$ returns **Yes**, declare f satisfiable and vice-versa

$\triangleright$ G can be constructed in time polynomial in n and m