# 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

$$\textsc{sat}(f) \quad \leq_p \quad 3\text{-}\textsc{sat}(f')$$

Given a CNF formula $f$ on variables $X = \{x_1, \ldots, x_n\}$

Construct an equivalent 3-CNF formula $f'$ on variables $X \cup \{d_1, d_2, \ldots\}$

- Initialize $f' = f$. For a long clause $C = (x_{i1} \vee x_{i2} \vee x_{i3} \vee x_{i4} \vee \ldots)$ in $f'$
- Add the clauses $(x_{i1} \vee x_{i2} \vee d_i) \wedge (\overline{d_i} \vee x_{i3} \vee x_{i4} \vee \ldots)$ to $f'$
- The new (long clause) is shorter than $C$

$$(x_{i1} \vee x_{i2} \vee \underbrace{x_{i3} \vee x_{i4} \vee \cdots}_{y}) \iff (x_{i1} \vee x_{i2} \vee d_i) \wedge (\overline{d_i} \vee \underbrace{x_{i3} \vee x_{i4} \vee \cdots}_{y})$$

**Proof:** Suppose $(x_{i1} \vee x_{i2} \vee \underbrace{x_{i3} \vee x_{i4} \vee \ldots}_{y})$ is satisfiable

- If $x_{i1} \vee x_{i2} = 1$. Set $d_i = 0$ ▷ RHS is also satisfiable
- If $x_{i1} \vee x_{i2} = 0$, then $y = 1$. Set $d_i = 1$ ▷ RHS is also satisfiable

# Reduction by encoding with gadgets

$$\text{SAT}(f) \quad \leq_p \quad 3\text{-SAT}(f')$$

Given a CNF formula $f$ on variables $X = \{x_1, \ldots, x_n\}$

Construct an equivalent 3-CNF formula $f'$ on variables $X \cup \{d_1, d_2, \ldots\}$

- Initialize $f' = f$. For a long clause $C = (x_{i1} \vee x_{i2} \vee x_{i3} \vee x_{i4} \vee \ldots)$ in $f'$
- Add the clauses $(x_{i1} \vee x_{i2} \vee d_i) \wedge (\overline{d_i} \vee x_{i3} \vee x_{i4} \vee \ldots)$ to $f'$
- The new (long clause) is shorter than $C$

$$(x_{i1} \vee x_{i2} \vee \underbrace{x_{i3} \vee x_{i4} \vee \cdots}_{y}) \iff (x_{i1} \vee x_{i2} \vee d_i) \wedge (\overline{d_i} \vee \underbrace{x_{i3} \vee x_{i4} \vee \cdots}_{y})$$

**Proof:** Suppose $(x_{i1} \vee x_{i2} \vee d_i) \wedge (\overline{d_i} \vee \underbrace{x_{i3} \vee x_{i4} \vee \ldots}_{y})$ is satisfiable

- If $d_i = 1$, then $\overline{d_i} = 0$ and $y = 1$ ▷ LHS is also satisfiable
- If $d_i = 0$, then $\overline{d_i} = 1$ and $x_{i1} \vee x_{i2} = 1$ ▷ LHS is also satisfiable

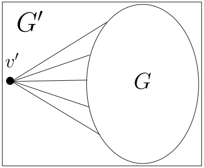
# Reduction by encoding with gadgets

HAM-PATH$(G)$ $\leq_p$ HAM-CYCLE$(G)$

Let $\mathcal{A}$ be an algorithm for HAM-CYCLE$(G)$

Given an instance $G$ of HAM-PATH$(G)$

Let $G'$ be $G$ plus a dummy vertex $v'$ adjacent to all vertices in $V(G)$




$G'$ has a Hamiltonian cycle if and only if $G$ has a Hamiltonian path

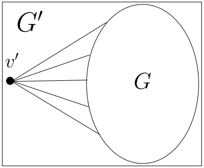
# Reduction by encoding with gadgets

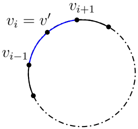
HAM-PATH$(G)$ $\leq_p$ HAM-CYCLE$(G)$

Let $\mathcal{A}$ be an algorithm for HAM-CYCLE$(G)$

Given an instance $G$ of HAM-PATH$(G)$

Let $G'$ be $G$ plus a dummy vertex $v'$ adjacent to all vertices in $V(G)$




$G'$ has a Hamiltonian cycle if and only if $G$ has a Hamiltonian path

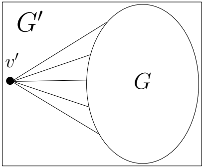
# Reduction by encoding with gadgets

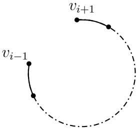
$\text{HAM-PATH}(G) \;\; \leq_p \;\; \text{HAM-CYCLE}(G)$

Let $\mathcal{A}$ be an algorithm for $\text{HAM-CYCLE}(G)$

Given an instance $G$ of $\text{HAM-PATH}(G)$

Let $G'$ be $G$ plus a dummy vertex $v'$ adjacent to all vertices in $V(G)$




$G'$ has a Hamiltonian cycle if and only if $G$ has a Hamiltonian path

1. Call $\mathcal{A}$ on $G'$
2. If $\mathcal{A}$ outputs **Yes** we will output **Yes** and vice-versa

# Polynomial Time Reduction: Cook Reducibility

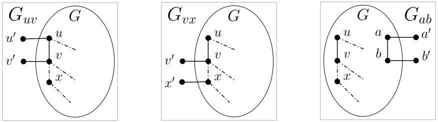
$\text{HAM-CYCLE}(G) \leq_p \text{HAM-PATH}(G)$

Let $\mathcal{A}$ be an algorithm for $\text{HAM-PATH}(G)$

Given an instance $G = (V, E)$ of $\text{HAM-CYCLE}(G)$

For each edge $e = (u, v) \in E(G)$ make the graph $G_e = (V_e, E_e)$

$V_e = V \cup \{u', v'\}$ and $E_e = E \cup \{(u, u'), (v, v')\}$




$G$ has a Hamiltonian cycle if and only if some $G_e$ has a Hamiltonian path

1. Call $\mathcal{A}$ on each of $G_{uv}$ ▷ $O(|E|)$ calls
2. If $\mathcal{A}$ outputs **Yes** on any $G_e$, we will output **Yes**
3. If $\mathcal{A}$ outputs **No** on all $G_e$, we will output **No**