# Binary Relation III

Zeyu Mi

2024-12-9

**Adapted From:**
《数理逻辑与集合论》10.5 10.8

# Recap

- **Equivalence Relation and Partitions**
  - Reflexivity, Symmetry and Transitivity.
- **Compatible Relations and Covers**
  - Reflexivity, Symmetry

# Order Relations

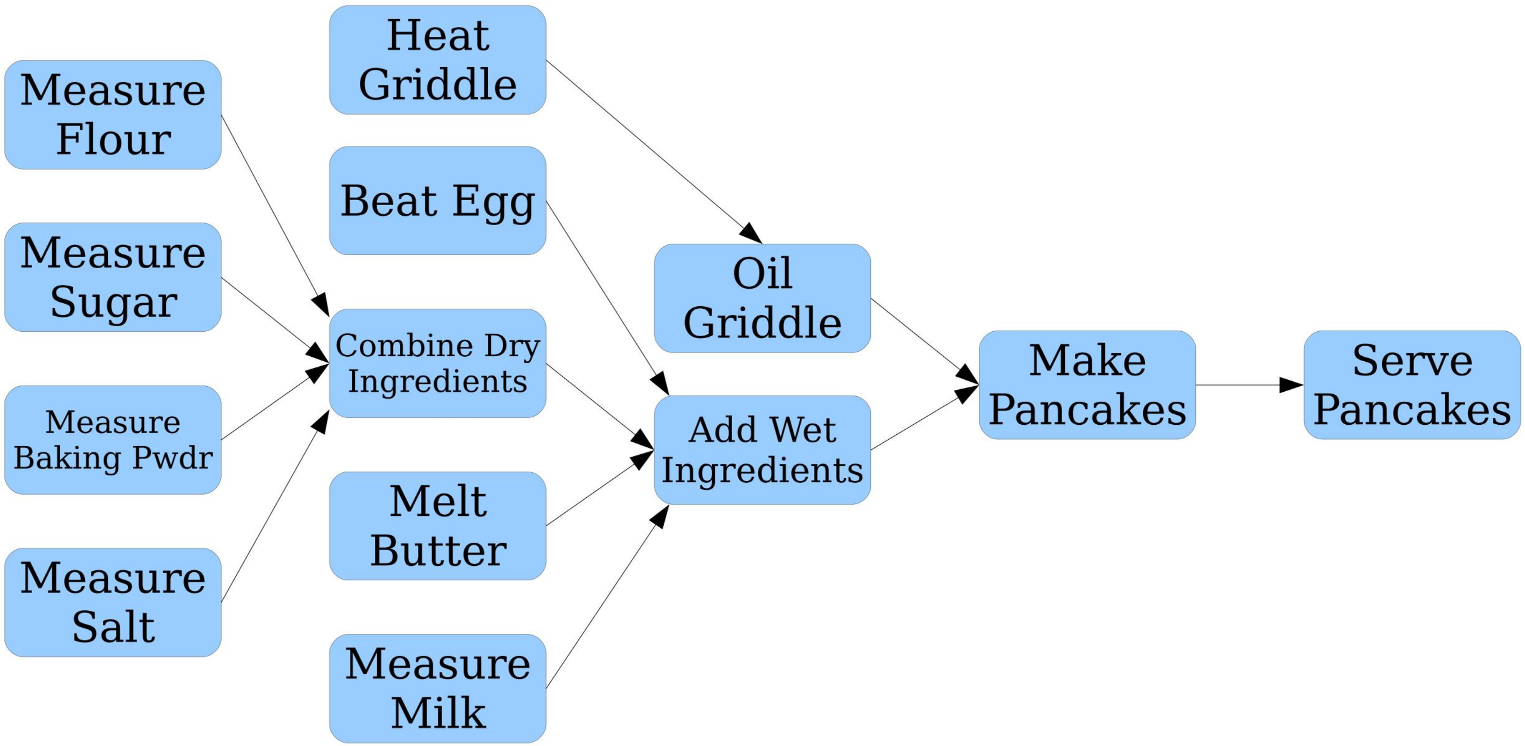
# Ordered Objects

*words in dictionary*

| | | |
|---|---|---|
| a | bread | dog |
| about | bring | far |
| after | but | fly |
| an | buy | got |
| ask | cake | Hand |
| bear | call | it |
| bed | chair | letter |

# Ordered Objects

*A recipe to make pancake*



# Ordered Objects

- Take $\leq$ on $\mathbb{R}$ as example. We know that:
  - $x \leq x$ for any $x \in \mathbb{R}$
  - If $x \leq y$ and $y \leq x$,then $x = y$
  - If $x \leq y$ and $y \leq z$,then $x \leq z$

# Ordered Objects

- Take $\leq$ on $\mathbb{R}$ as example. We know that:
  - $x \leq x$ for any $x \in \mathbb{R}$ Reflexivity
  - If $x \leq y$ and $y \leq x$,then $x = y$ ?
  - If $x \leq y$ and $y \leq z$,then $x \leq z$ Transitivity

# Anti-symmetry(反对称性)

- Some relation is symmetric only in its diagonal.
- Examples:
  - If $\boldsymbol{x} \leq \boldsymbol{y}$ and $\boldsymbol{y} \leq \boldsymbol{x}$,then $\boldsymbol{x} = \boldsymbol{y}$
  - If $\boldsymbol{A} \subseteq \boldsymbol{B}$ and $\boldsymbol{B} \subseteq \boldsymbol{A}$, then $\boldsymbol{A} = \boldsymbol{B}$
- Relations of this sort are called antisymmetric.

$\boldsymbol{R}$ on $\boldsymbol{A}$ is antisymmetric $\Leftrightarrow$

$$(\forall \boldsymbol{a})(\forall \boldsymbol{b})((\boldsymbol{a} \in \boldsymbol{A} \wedge \boldsymbol{b} \in \boldsymbol{A} \wedge \boldsymbol{aRb} \wedge \boldsymbol{bRa}) \rightarrow \boldsymbol{a} = \boldsymbol{b})$$

# Anti-symmetry : Example

- $\boldsymbol{I_A}$ and $\boldsymbol{N_A}$
- $\boldsymbol{E_A}$ when $|\boldsymbol{A}| > 1$
- $\boldsymbol{R} = \{\langle 1,1\rangle, \langle 1,2\rangle, \langle 2,1\rangle, \langle 2,3\rangle, \langle 3,3\rangle\}$ on $\{1,2,3\}$

# Anti-symmetry : Example

- $\boldsymbol{I_A}$ and $\boldsymbol{N_A}$ ☺

- $\boldsymbol{E_A}$ when $|\boldsymbol{A}| > 1$ ☹

- $\boldsymbol{R} = \{\langle 1,1\rangle, \langle 1,2\rangle, \langle 2,1\rangle, \langle 2,3\rangle, \langle 3,3\rangle\}$ on $\{1,2,3\}$ ☹
  - $1R2$ and $2R1$

# Partial-order Relation(偏序关系)

- An **partial-order relation** is a relation that is **reflexive, anti-symmetric and transitive**.
- Some example:
  - $x \leq y$
  - $x \subseteq y$
  - $x$ can be divided by $y$ (on $\mathbb{N}^+$)
- A set $\boldsymbol{A}$ together with a partial ordering $\boldsymbol{R}$ is called a partially ordered set, or poset(偏序集), and is denoted by $\langle \boldsymbol{A}, \boldsymbol{R} \rangle$.
  - $\langle \mathbb{N}, \leq \rangle$,$\langle \mathcal{P}(A), \subseteq \rangle$ are both poset.

# Ordered Objects

- Take $<$ on $\mathbb{R}$ as example. We know that:
  - $x \nless x$ for any $x \in \mathbb{R}$
  - If $x < y$ and $y < z$,then $x < z$

# Ordered Objects

- Take $<$ on $\mathbb{R}$ as example. We know that:
  - $x \nless x$ for any $x \in \mathbb{R}$ ?
  - If $x < y$ and $y < z$,then $x < z$ Transitivity

# Anti-reflexivity(非自反性)

- Some relation never holds from any element to itself
- Examples:
  - $\boldsymbol{A} \not\subset \boldsymbol{A}$ for any set $\boldsymbol{A}$
  - $\boldsymbol{x} \nless \boldsymbol{x}$ for any $\boldsymbol{x}$
- Relations of this sort are called anti-reflexive or irreflexive:

$$\boldsymbol{R} \text{ on } \boldsymbol{A} \text{ is anti-reflexive} \Leftrightarrow (\forall \boldsymbol{a})(\boldsymbol{a} \in \boldsymbol{A} \to \boldsymbol{a} \not\boldsymbol{R} \boldsymbol{a})$$

# Anti-reflexivity : Example

- $\boldsymbol{I_A}$ and $\boldsymbol{E_A}$

- $\boldsymbol{N_A}$

- $\boldsymbol{R} = \{\langle 1,2\rangle, \langle 2,1\rangle, \langle 2,3\rangle, \langle 3,3\rangle\}$ on $\{1,2,3\}$

# Anti-reflexivity : Example

- $\boldsymbol{I_A}$ and $\boldsymbol{E_A}$ ☹

- $\boldsymbol{N_A}$ ☺

- $\boldsymbol{R} = \{\langle 1,2\rangle, \langle 2,1\rangle, \langle 2,3\rangle, \langle 3,3\rangle\}$ on $\{1,2,3\}$ ☹
  - $3R3$

# Quasi-order Relation(拟序关系)

- An **quasi-order relation, or strict order relation** is a relation that is **anti-reflexive, transitive**.
- Some example:
  - $x < y$
  - $x \subset y$
  - $x$ is taller than $y$

# Quasi-order Relation

Quasi-order relation is anti-symmetric.

# Quasi-order Relation

Quasi-order relation is anti-symmetric.

Anti-reflexivity
Transitivity $\Rightarrow$ Anti-symmetric

# Quasi-order Relation

Quasi-order relation is anti-symmetric.

Anti-reflexivity
Transitivity $\Rightarrow$ Anti-symmetric

Proof:

For a quasi-order relation $\boldsymbol{R}$ on $\boldsymbol{A}$, assume that $\boldsymbol{R}$ is not anti-symmetric, then there exits $\boldsymbol{x}, \boldsymbol{y}$ such that $\boldsymbol{xRy} \wedge \boldsymbol{yRx} \wedge \boldsymbol{x} \neq \boldsymbol{y}$.

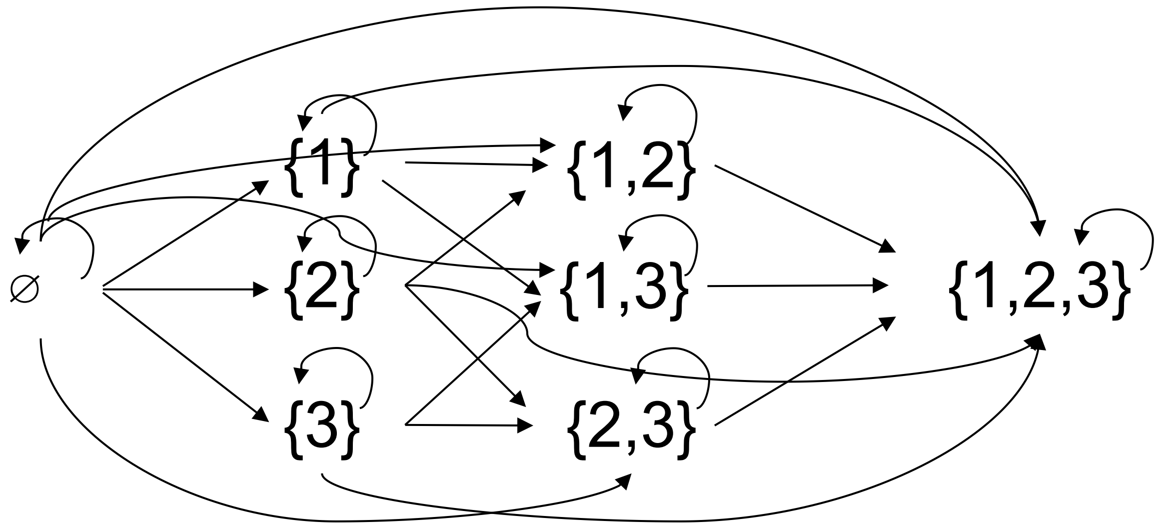
From transitivity and $\boldsymbol{xRy} \wedge \boldsymbol{yRx}$ we have $\boldsymbol{xRx}$, but $\boldsymbol{x} \not\boldsymbol{R} \boldsymbol{x}$ as $\boldsymbol{R}$ is anti-reflexivity.

Contradiction.

# Partial Order and Quasi Order

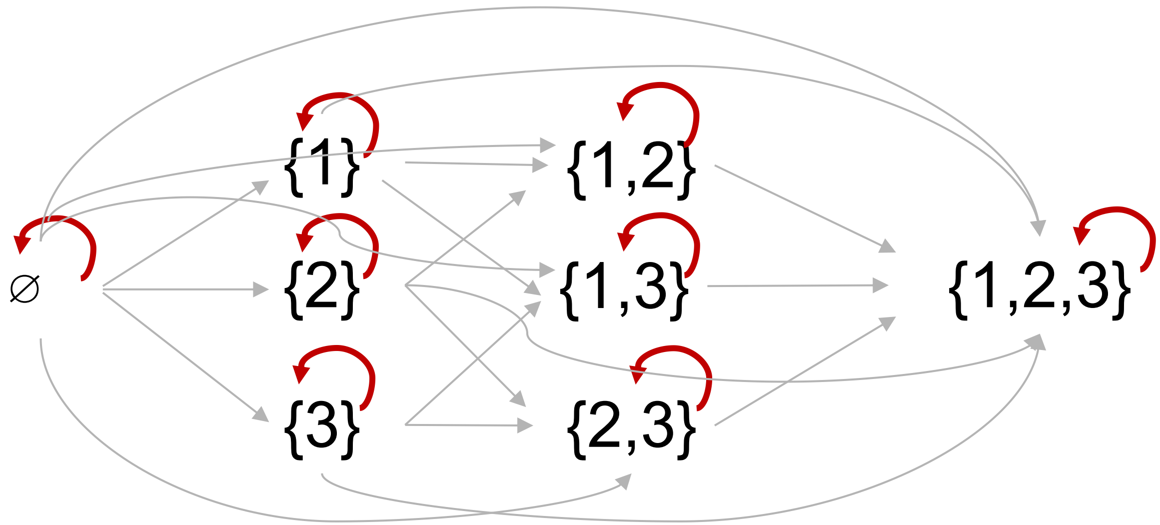
- Often donate as $\leq$ and $<$
  - Not necessarily mean numerical operation.
- Quasi order $\boldsymbol{R}$ on $\boldsymbol{A}$, $\boldsymbol{R} \cup \boldsymbol{R_0}$ is a partial order
  $"<" + "=" \rightarrow " \leq "$
- Partial order $\boldsymbol{R}$ on $\boldsymbol{A}$, $\boldsymbol{R} - \boldsymbol{R_0}$ is a quasi order
  $" \leq " - "=" \rightarrow " < "$

⟨$\mathcal{P}$ (A), ⊆⟩, A={1,2,3}




Too complicated!!!

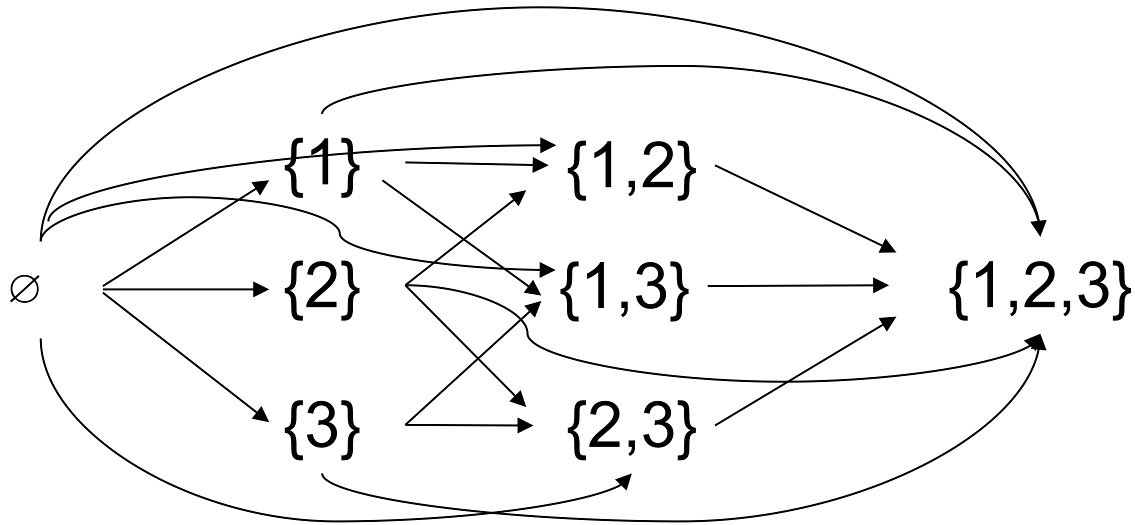
$\langle \mathcal{P}(A), \subseteq \rangle$, $A=\{1,2,3\}$



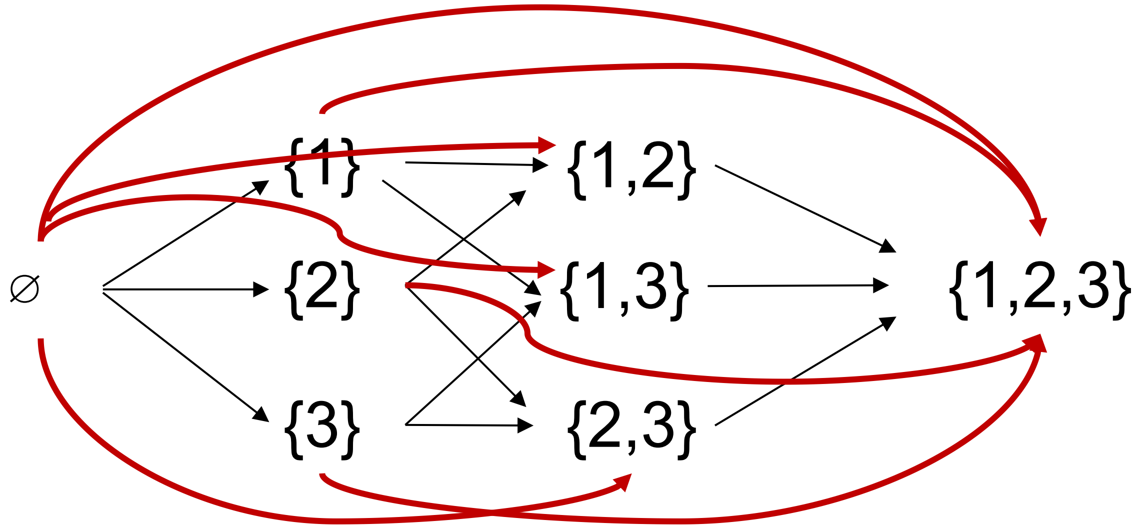

Reflexivity: Every object has a self-circle.

**Drop it.**

$\langle \mathcal{P}(A), \subseteq \rangle$, $A=\{1,2,3\}$



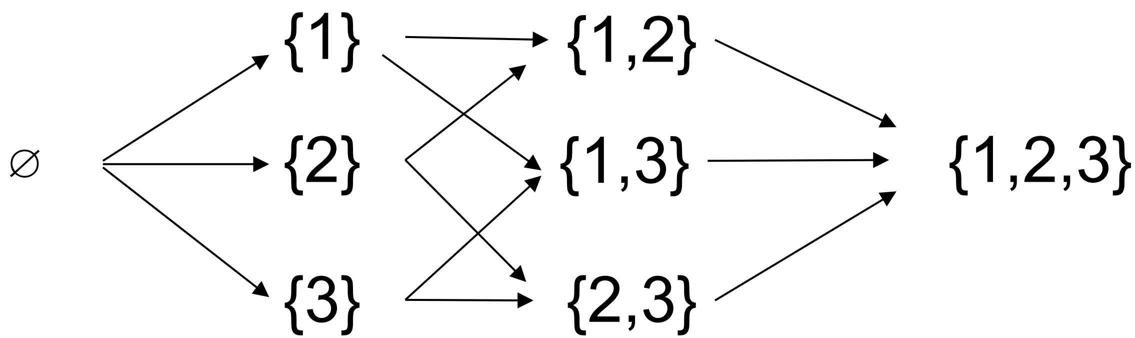
$\langle \mathcal{P}(A), \subseteq \rangle$, $A=\{1,2,3\}$




Transitivity: Some relation can be derived from "smaller" relation

**Drop it.**

$\langle \mathcal{P}(A), \subseteq \rangle$, $A=\{1,2,3\}$



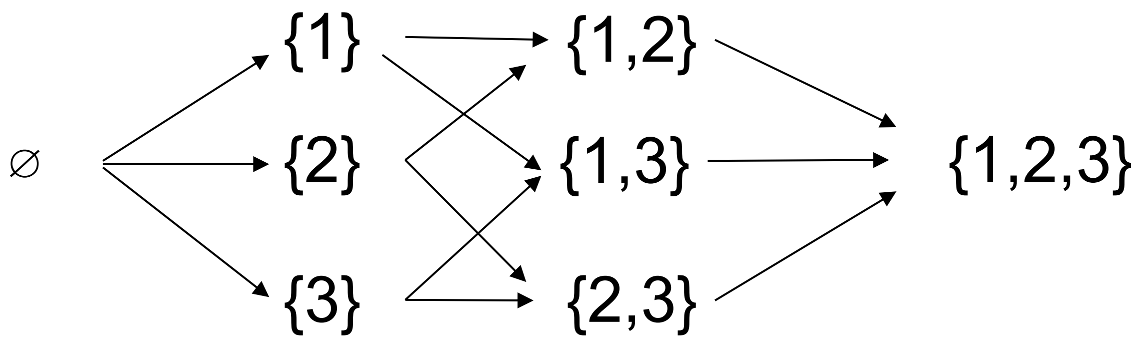
# Cover Relation

- For poset $\langle \boldsymbol{A}, \leq \rangle$, if $\boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{A}, \boldsymbol{x} \leq \boldsymbol{y}, \boldsymbol{x} \neq \boldsymbol{y}$, and there doesn't exist $\boldsymbol{z} \in \boldsymbol{A}$ such that $\boldsymbol{x} \leq \boldsymbol{z}$ and $\boldsymbol{z} \leq \boldsymbol{y}$, we call that **$\boldsymbol{y}$ covers $\boldsymbol{x}$**.
- $\boldsymbol{y}$ covers $\boldsymbol{x}$ means that $\boldsymbol{y}$ is exactly greater than $\boldsymbol{x}$ and there's no elements between them.
- We can define a **cover relation** on $\boldsymbol{A}$:

$$cov(A) = \{\langle \boldsymbol{x}, \boldsymbol{y} \rangle | \boldsymbol{x} \in \boldsymbol{A} \wedge \boldsymbol{y} \in \boldsymbol{A} \wedge (\boldsymbol{y}\ covers\ \boldsymbol{x})\}$$

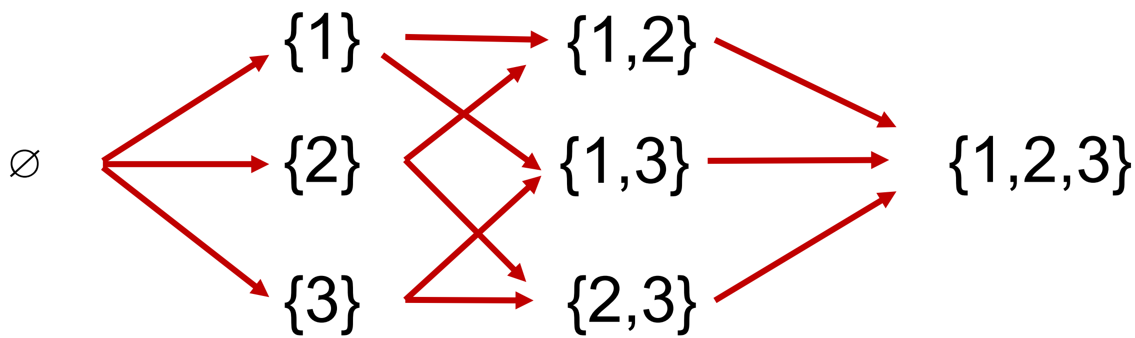
$\langle \mathcal{P}(A), \subseteq \rangle$, $A=\{1,2,3\}$

$cover(\mathcal{P}(A))$



$\langle \mathcal{P}(A), \subseteq \rangle$, $A=\{1,2,3\}$

$cover(\mathcal{P}(A))$

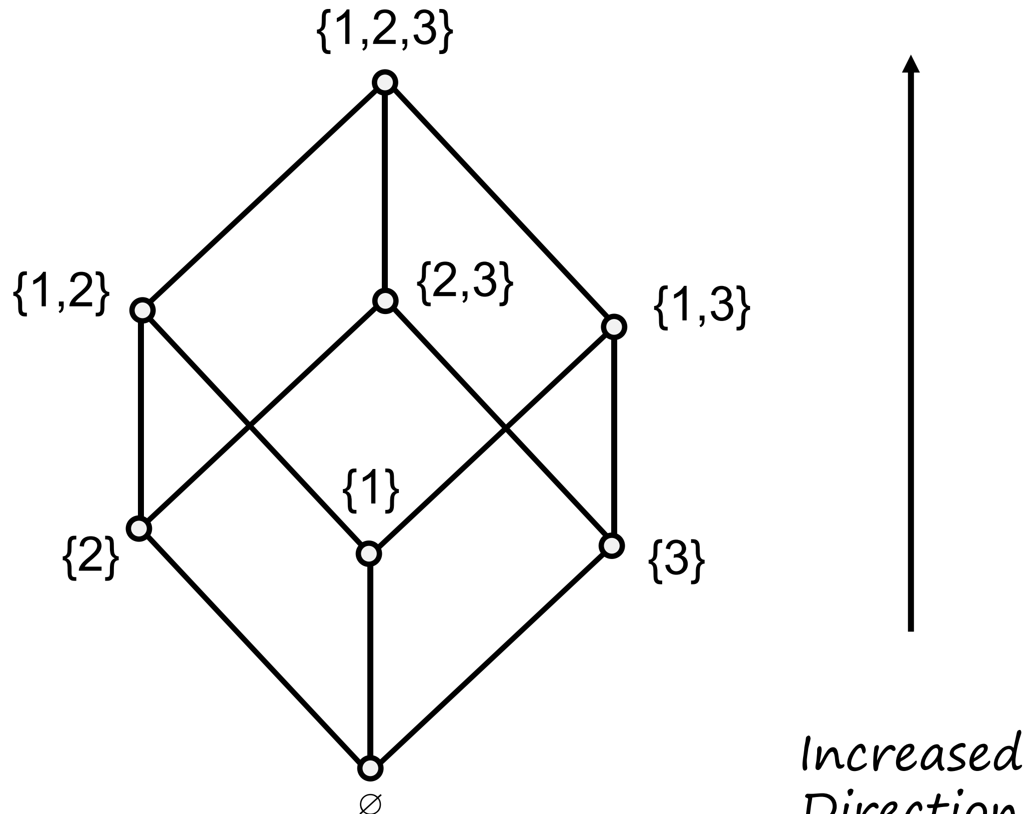



All directed edge are
towards the same way

**Choose a direction as
increased direction.**

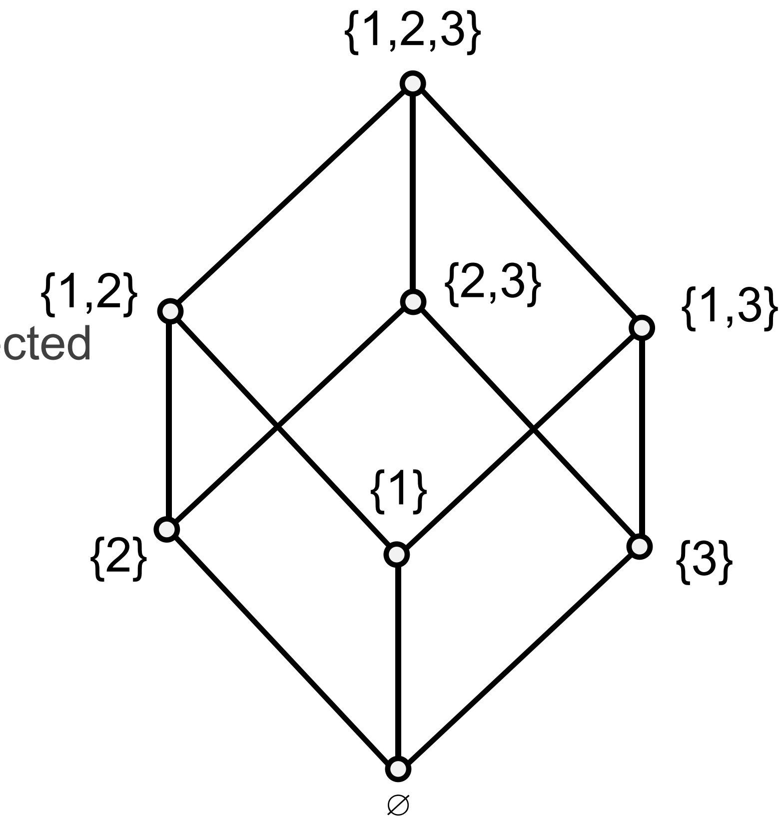
$\langle \mathcal{P}(A), \subseteq \rangle$, $A=\{1,2,3\}$

$cover(\mathcal{P}(A))$



# Hasse Diagram(哈斯图)

- For poset $\langle A, \leq \rangle$:
  - Every vertex is an element
  - If $x \leq y \wedge x \neq y$, then vertex $y$ is above vertex $x$
  - If $\langle x, y \rangle \in cov(A)$, then there's an undirected edge between $x$ and $y$



# Hasse Diagram

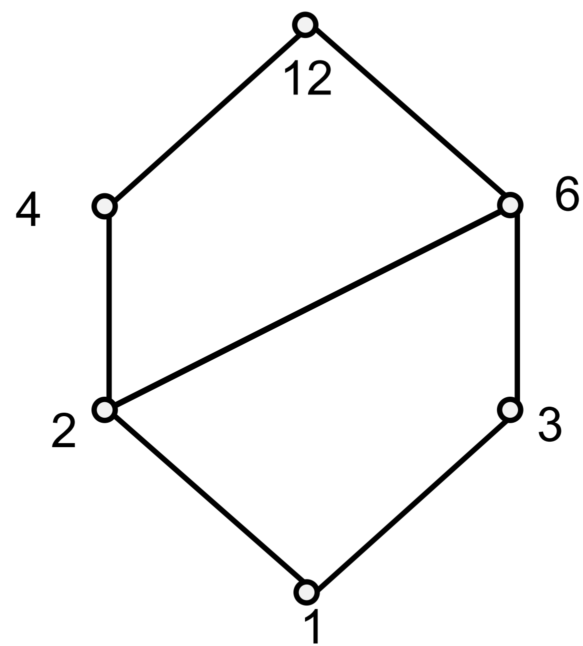
- Define the divisibility relation on $A = \{1,2,3,4,6,12\}$:

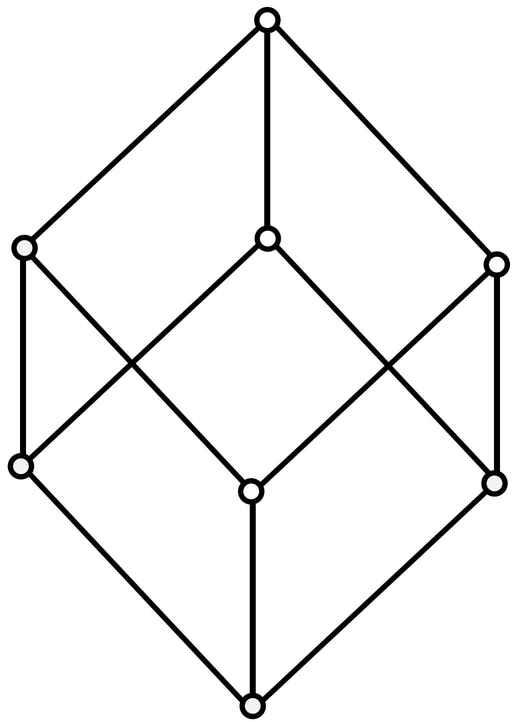
$$R = \{\langle x, y\rangle | x \in A \land y \in A \land x|y\}$$

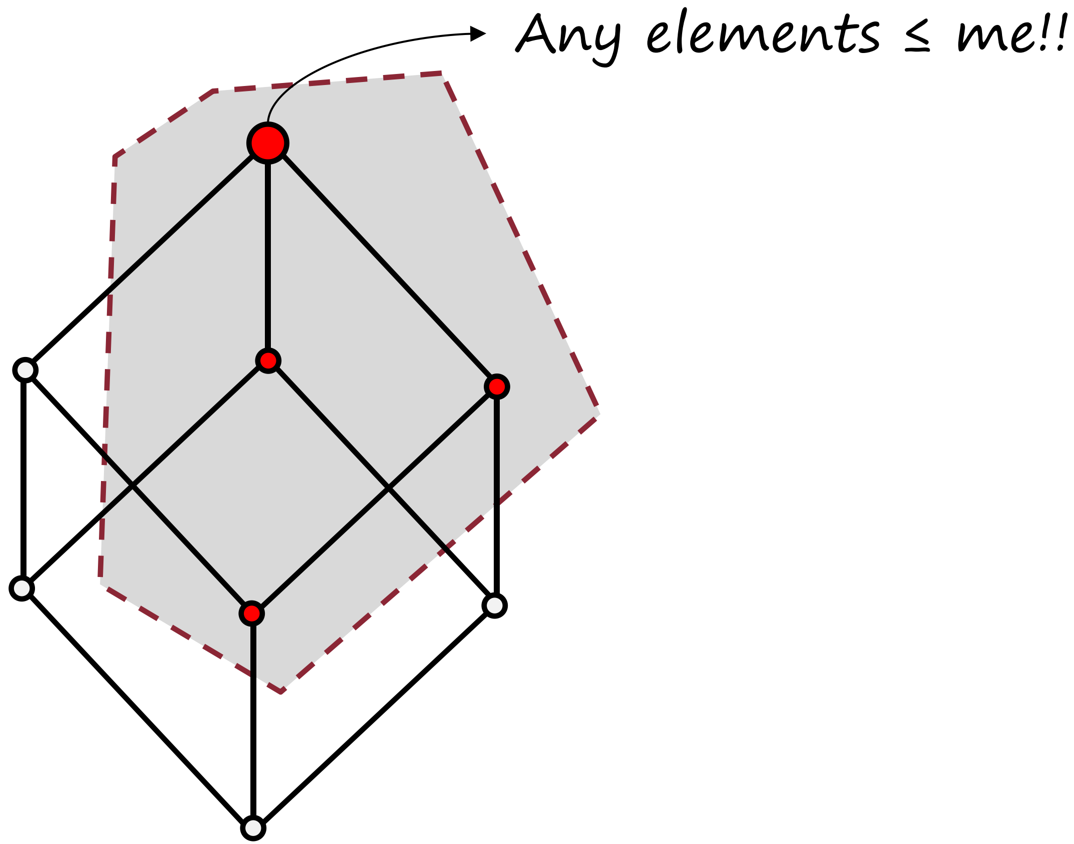
# Hasse Diagram

- Define the divisibility relation on $A = \{1,2,3,4,6,12\}$:

$$R = \{\langle x, y\rangle | x \in A \wedge y \in A \wedge x|y\}$$



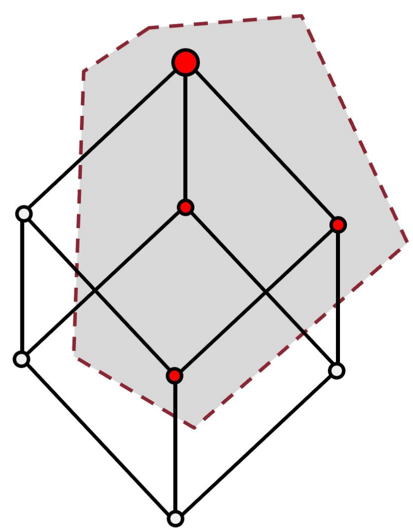


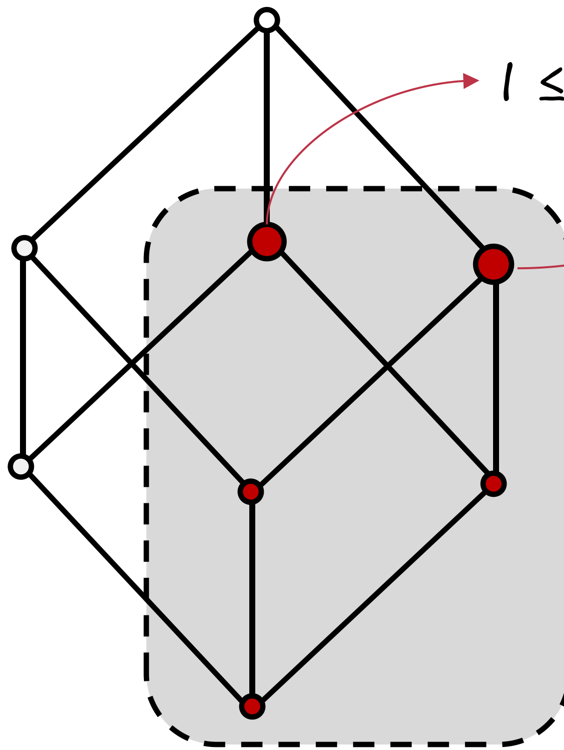

Any elements ≤ me!!!

# Maximum Elements(最大元)

- For a poset $\langle \boldsymbol{A}, \leq \rangle$, and $\boldsymbol{B} \subseteq \boldsymbol{A}$:
- If there exists $\boldsymbol{y} \in \boldsymbol{B}$ such that $\forall \boldsymbol{x} \in \boldsymbol{B}, \boldsymbol{x} \leq \boldsymbol{y}$, we call $\boldsymbol{y}$ the **maximum** elements of $\boldsymbol{B}$

$$\boldsymbol{y\ is\ maximum} \Leftrightarrow \boldsymbol{y} \in \boldsymbol{B} \wedge (\forall \boldsymbol{x})(\boldsymbol{x} \in \boldsymbol{B} \rightarrow \boldsymbol{x} \leq \boldsymbol{y})$$





I ≤ no elements except myself

# Maximal and Maximum Elements

- For a poset $\langle \boldsymbol{A}, \leq \rangle$, and $\boldsymbol{B} \subseteq \boldsymbol{A}$:
  - If there exists $\boldsymbol{y} \in \boldsymbol{B}$ such that $\forall \boldsymbol{x} \in \boldsymbol{B}$, $\boldsymbol{x} \leq \boldsymbol{y}$, we call $\boldsymbol{y}$ the **maximum** elements(最大元) of $\boldsymbol{B}$

$$\boldsymbol{y\ is\ maximum} \Leftrightarrow \boldsymbol{y} \in \boldsymbol{B} \wedge (\forall \boldsymbol{x})(\boldsymbol{x} \in \boldsymbol{B} \rightarrow \boldsymbol{x} \leq \boldsymbol{y})$$

  - If there exists $\boldsymbol{y} \in \boldsymbol{B}$ such that $\forall \boldsymbol{x} \in \boldsymbol{B}$, if $\boldsymbol{y} \leq \boldsymbol{x}$, then $\boldsymbol{x} = \boldsymbol{y}$. We call $\boldsymbol{y}$ the **maximal** elements(极大元) of $\boldsymbol{B}$

$$\boldsymbol{y\ is\ maximal} \Leftrightarrow \boldsymbol{y} \in \boldsymbol{B} \wedge (\forall \boldsymbol{x})((\boldsymbol{x} \in \boldsymbol{B} \wedge \boldsymbol{y} \leq \boldsymbol{x}) \rightarrow \boldsymbol{x} = \boldsymbol{y})$$

# Maximal and Maximum Elements

- For a poset $\langle A, \leq \rangle$, and $B \subseteq A$:

I beat every one.

  - If there exists $y \in B$ such that $\forall x \in B$, $x \leq y$, we call $y$ the **maximum** elements(最大元) of $B$

$$y \text{ is maximum} \Leftrightarrow y \in B \wedge (\forall x)(x \in B \rightarrow x \leq y)$$

  - If there exists $y \in B$ such that $\forall x \in B$, if $y \leq x$, then $x = y$. We call $y$ the **maximal** elements(极大元) of $B$

$$y \text{ is maximal} \Leftrightarrow y \in B \wedge (\forall x)((x \in B \wedge y \leq x) \rightarrow x = y)$$

No one can beat me except myself.

# Minimum(最小) and Minimal(极小) Elements

- For a poset $\langle \boldsymbol{A}, \leq \rangle$, and $\boldsymbol{B} \subseteq \boldsymbol{A}$:
  - If there exists $\boldsymbol{x} \in \boldsymbol{B}$ such that $\forall \boldsymbol{y} \in \boldsymbol{B}$, $\boldsymbol{x} \leq \boldsymbol{y}$, we call $\boldsymbol{x}$ the **minimum** elements(最小元) of $\boldsymbol{B}$

$$\boldsymbol{x\ is\ minimum} \Leftrightarrow \boldsymbol{x} \in \boldsymbol{B} \wedge (\forall \boldsymbol{y})(\boldsymbol{y} \in \boldsymbol{B} \rightarrow \boldsymbol{x} \leq \boldsymbol{y})$$

  - If there exists $\boldsymbol{x} \in \boldsymbol{B}$ such that $\forall \boldsymbol{y} \in \boldsymbol{B}$, if $\boldsymbol{y} \leq \boldsymbol{x}$, then $\boldsymbol{x} = \boldsymbol{y}$. We call $\boldsymbol{x}$ the **minimal** elements(极小元) of $\boldsymbol{B}$

$$\boldsymbol{x\ is\ minimal} \Leftrightarrow \boldsymbol{x} \in \boldsymbol{B} \wedge (\forall \boldsymbol{y})((\boldsymbol{y} \in \boldsymbol{B} \wedge \boldsymbol{y} \leq \boldsymbol{x}) \rightarrow \boldsymbol{x} = \boldsymbol{y})$$

# Minimum(最小) and Minimal(极小) Elements

- For a poset $\langle A, \leq \rangle$, and $B \subseteq A$:

Everyone beats me.

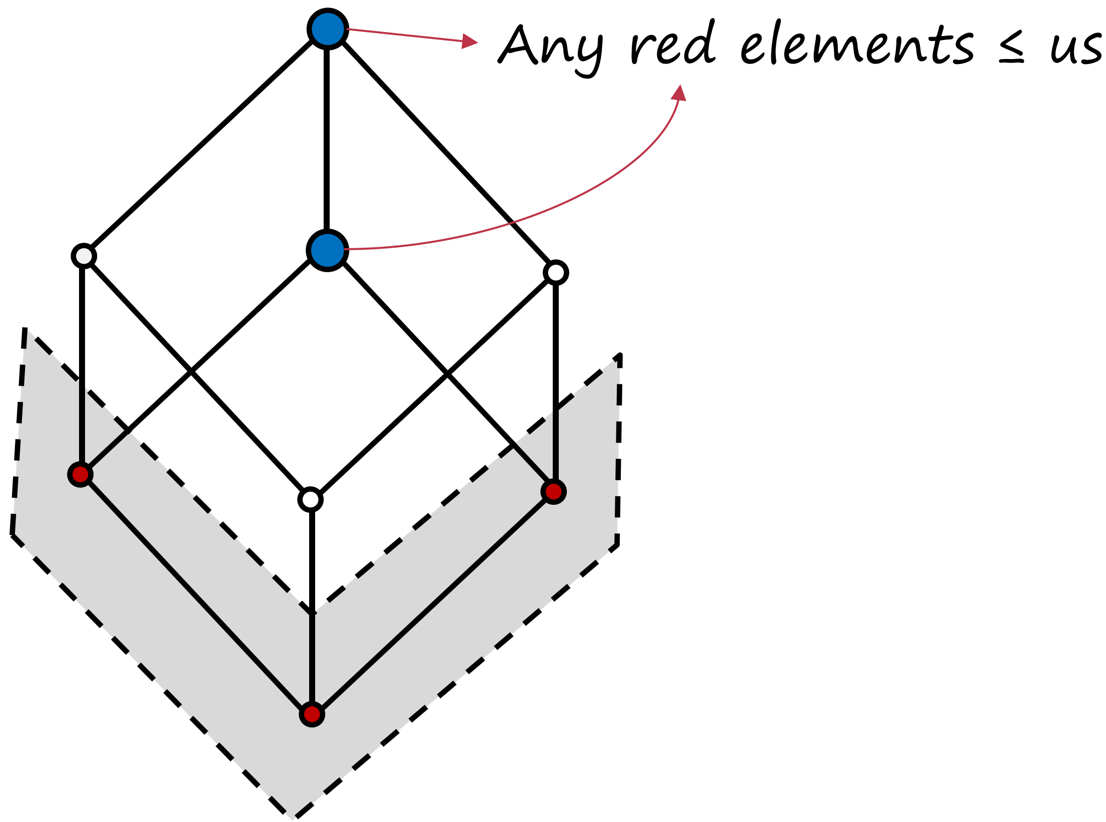
  - If there exists $x \in B$ such that $\forall y \in B$, $x \leq y$, we call $x$ the **minimum** elements(最小元) of $B$

$$x \text{ is minimum} \Leftrightarrow x \in B \land (\forall y)(y \in B \rightarrow x \leq y)$$

  - If there exists $x \in B$ such that $\forall y \in B$, if $y \leq x$, then $x = y$. We call $x$ the **minimal** elements(极小元) of $B$

$$x \text{ is minimal} \Leftrightarrow x \in B \land (\forall y)((y \in B \land y \leq x) \rightarrow x = y)$$

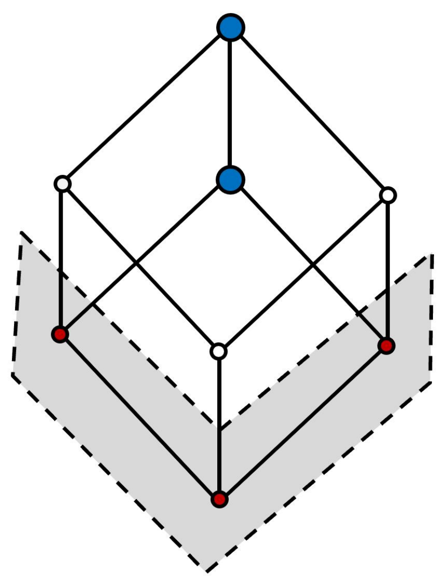
I can beats no one except myself.

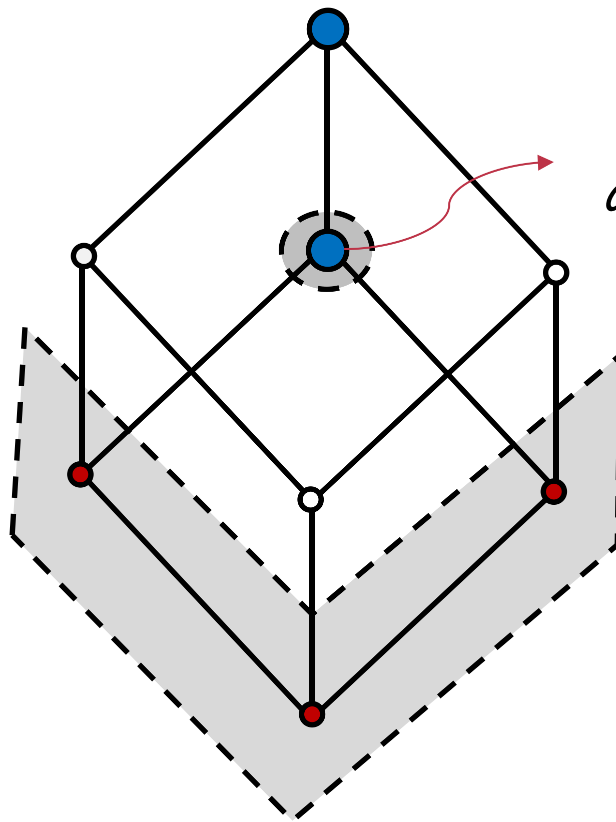

Any red elements ≤ us

# Upper Bound(上界)

- For a poset $\langle \boldsymbol{A}, \leq \rangle$, and $\boldsymbol{B} \subseteq \boldsymbol{A}$:
  - If there exists $\boldsymbol{y} \in \boldsymbol{A}$ such that $\forall \boldsymbol{x} \in \boldsymbol{B}$, $\boldsymbol{x} \leq \boldsymbol{y}$, we call $\boldsymbol{y}$ the **upper bound** of $\boldsymbol{B}$

$$\boldsymbol{y\ is\ upper\ bound} \Leftrightarrow \boldsymbol{y} \in \boldsymbol{A} \wedge (\forall \boldsymbol{x})(\boldsymbol{x} \in \boldsymbol{B} \rightarrow \boldsymbol{x} \leq \boldsymbol{y})$$

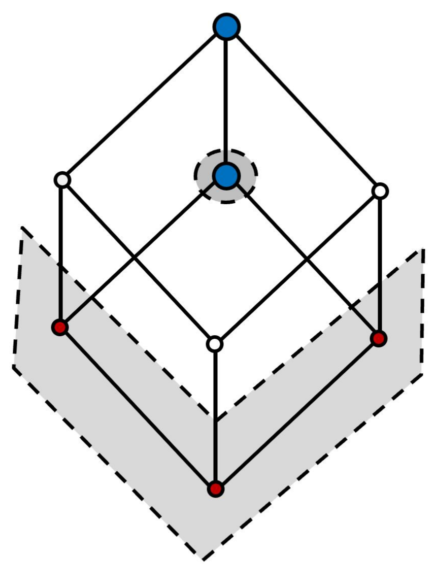




Any red elements ≤ me
and I ≤ any blue elements.

# Supremum(上确界)

- For a poset $\langle \boldsymbol{A}, \leq \rangle$, and $\boldsymbol{B} \subseteq \boldsymbol{A}$:
  - Let $\boldsymbol{C} = \{\boldsymbol{y} |\ \boldsymbol{y}\ is\ the\ upper\ bound\ of\ \boldsymbol{B}\}$, then the minimum elements in $\boldsymbol{C}$ is the **supremum** of $\boldsymbol{B}$.



# Upper Bound and Supremum

- For a poset $\langle \boldsymbol{A}, \leq \rangle$, and $\boldsymbol{B} \subseteq \boldsymbol{A}$:
  - If there exists $\boldsymbol{y} \in \boldsymbol{A}$ such that $\forall \boldsymbol{x} \in \boldsymbol{B}$, $\boldsymbol{x} \leq \boldsymbol{y}$, we call $\boldsymbol{y}$ the **upper bound（上界）** of $\boldsymbol{B}$

$$\boldsymbol{y\ is\ upper\ bound} \Leftrightarrow \boldsymbol{y} \in \boldsymbol{A} \wedge (\forall \boldsymbol{x})(\boldsymbol{x} \in \boldsymbol{B} \rightarrow \boldsymbol{x} \leq \boldsymbol{y})$$

  - Let $\boldsymbol{C} = \{\boldsymbol{y} |\ \boldsymbol{y}\ is\ the\ upper\ bound\ of\ \boldsymbol{B}\}$, then the minimum elements in $C$ is the **supremum（上确界）** of $\boldsymbol{B}$.

# Lower Bound(下界) and Infimum(下确界)

- For a poset $\langle \boldsymbol{A}, \leq \rangle$, and $\boldsymbol{B} \subseteq \boldsymbol{A}$:
  - If there exists $\boldsymbol{x} \in \boldsymbol{A}$ such that $\forall \boldsymbol{y} \in \boldsymbol{B}$, $\boldsymbol{x} \leq \boldsymbol{y}$, we call $\boldsymbol{x}$ the **lower bound（下界）** of $\boldsymbol{B}$

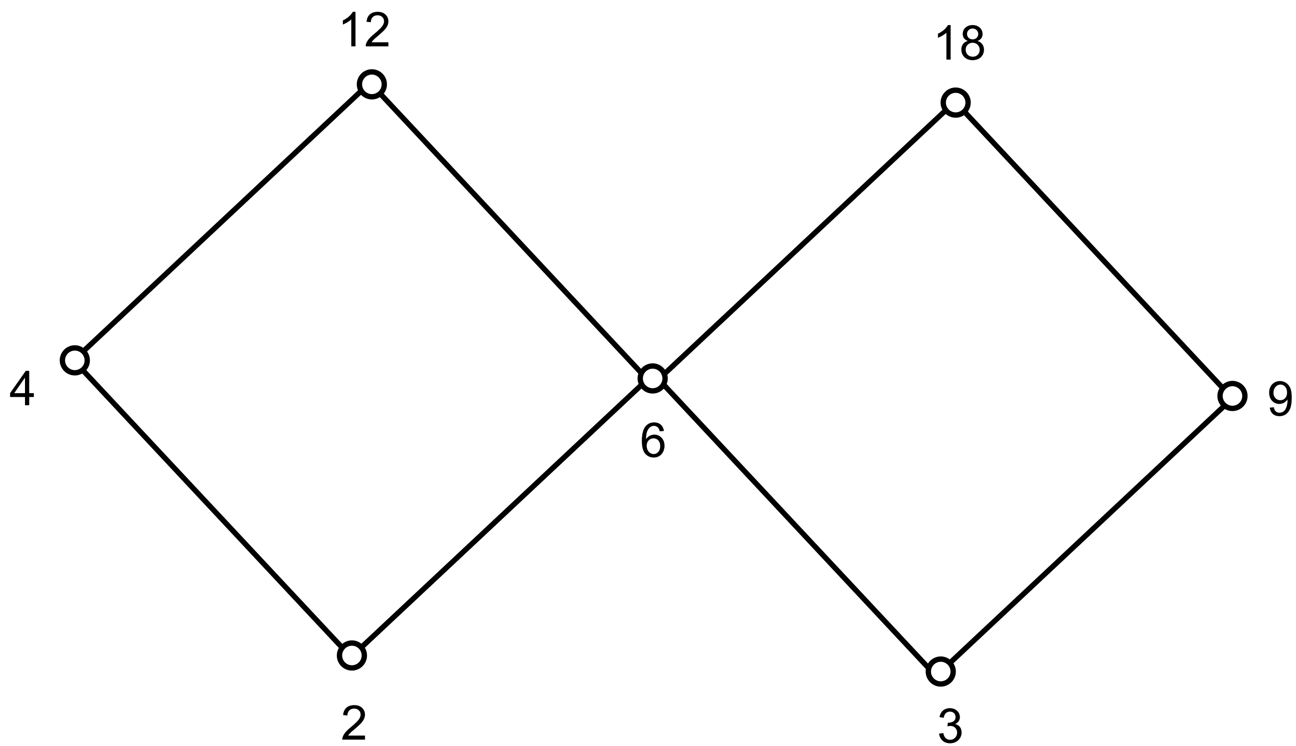
$$\boldsymbol{x\ is\ lower\ bound} \Leftrightarrow \boldsymbol{x} \in \boldsymbol{A} \wedge (\forall \boldsymbol{y})(\boldsymbol{y} \in \boldsymbol{B} \rightarrow \boldsymbol{x} \leq \boldsymbol{y})$$

  - Let $\boldsymbol{C} = \{\boldsymbol{x} |\ \boldsymbol{x}\ is\ the\ lower\ bound\ of\ \boldsymbol{B}\}$, then the maximum elements in $C$ is the **infimum（下确界）** of $\boldsymbol{B}$.

# Bound

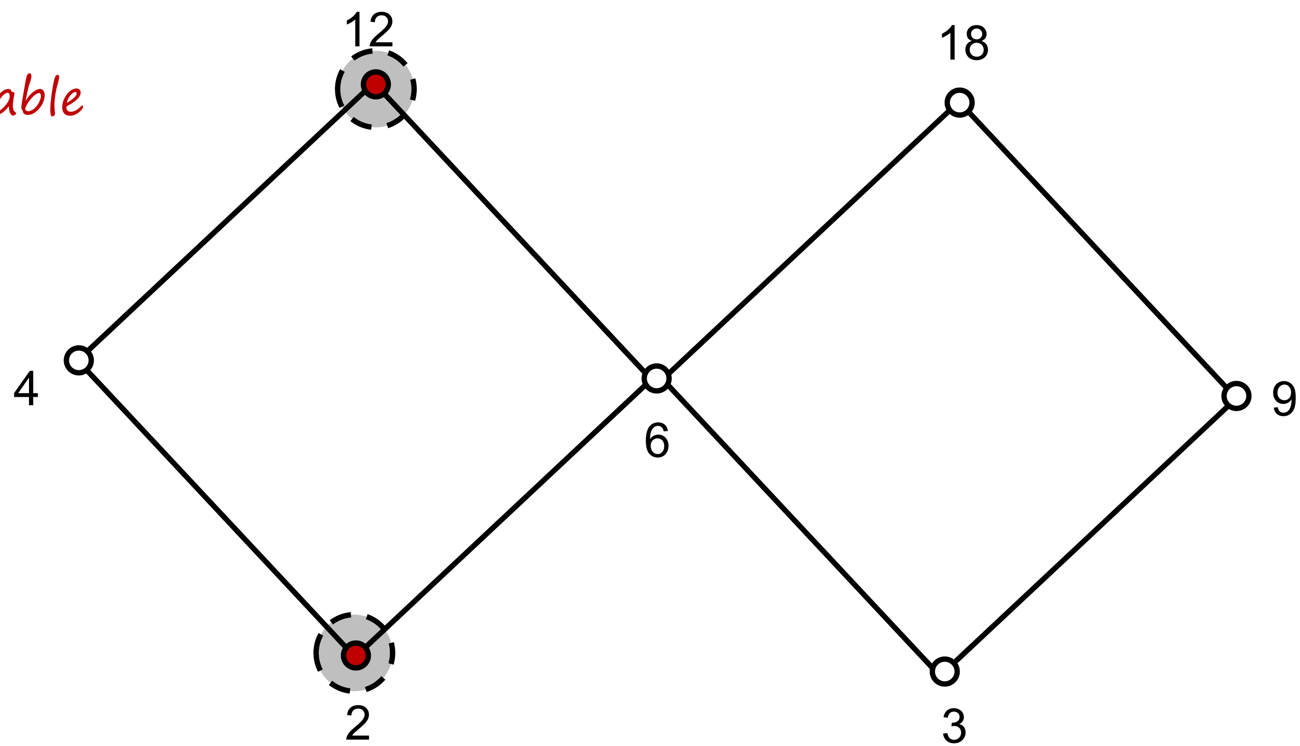
- For a poset $\langle \boldsymbol{A}, \leq \rangle$, and $\boldsymbol{B} \subseteq \boldsymbol{A}$:
  - The upper (lower) bounds or supremum(infimum) of $\boldsymbol{B}$ may in or not in $\boldsymbol{B}$.
  - The upper (lower) bounds of $\boldsymbol{B}$ may not exist, and may be not unique.
  - The supremum (infimum) of $\boldsymbol{B}$ may not exist, but it must be unique if exists.

$$A = \{2,3,4,6,9,12,18\}$$
$$R = \{\langle x, y\rangle | x \in A \land y \in A \land x|y\}$$

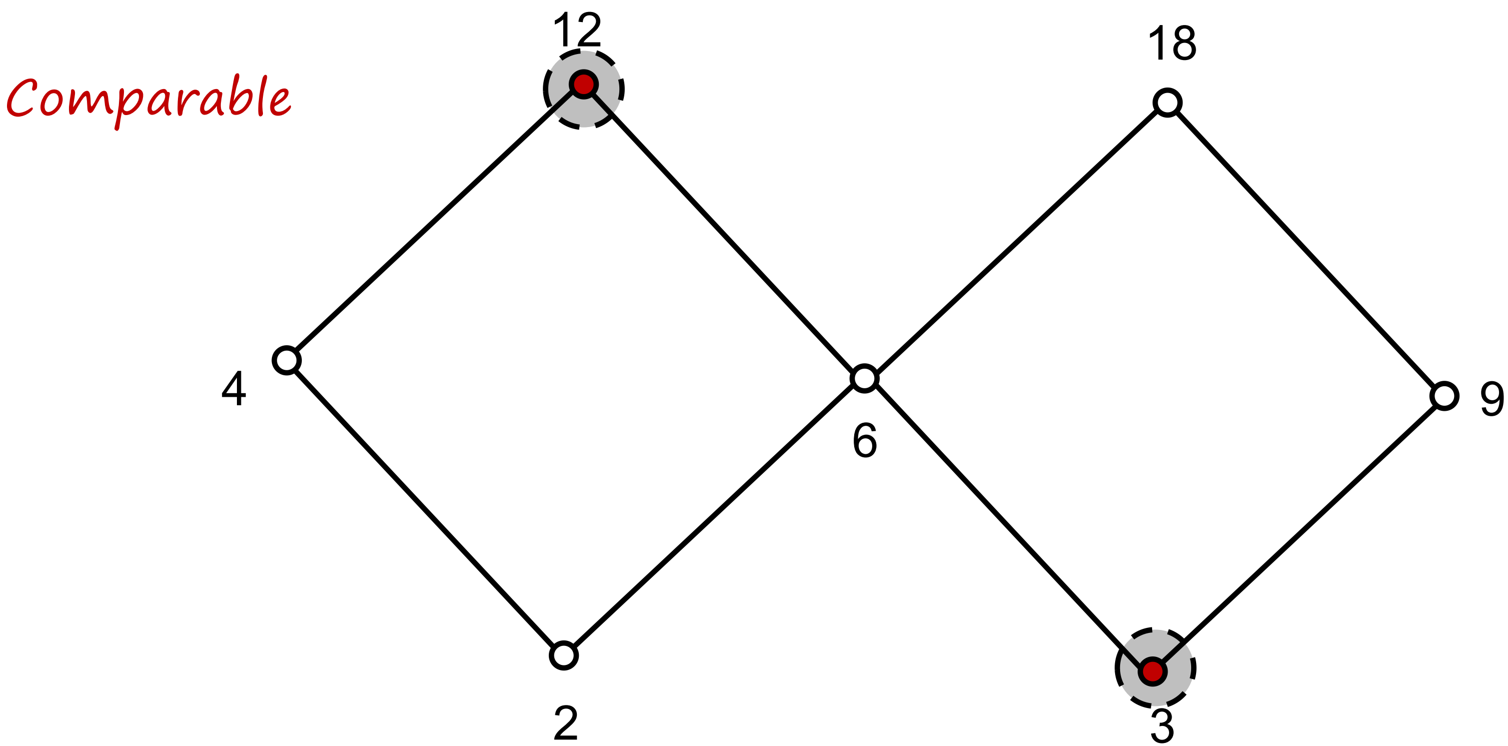


$$A = \{2,3,4,6,9,12,18\}$$
$$R = \{\langle x, y\rangle | x \in A \land y \in A \land x|y\}$$

Comparable

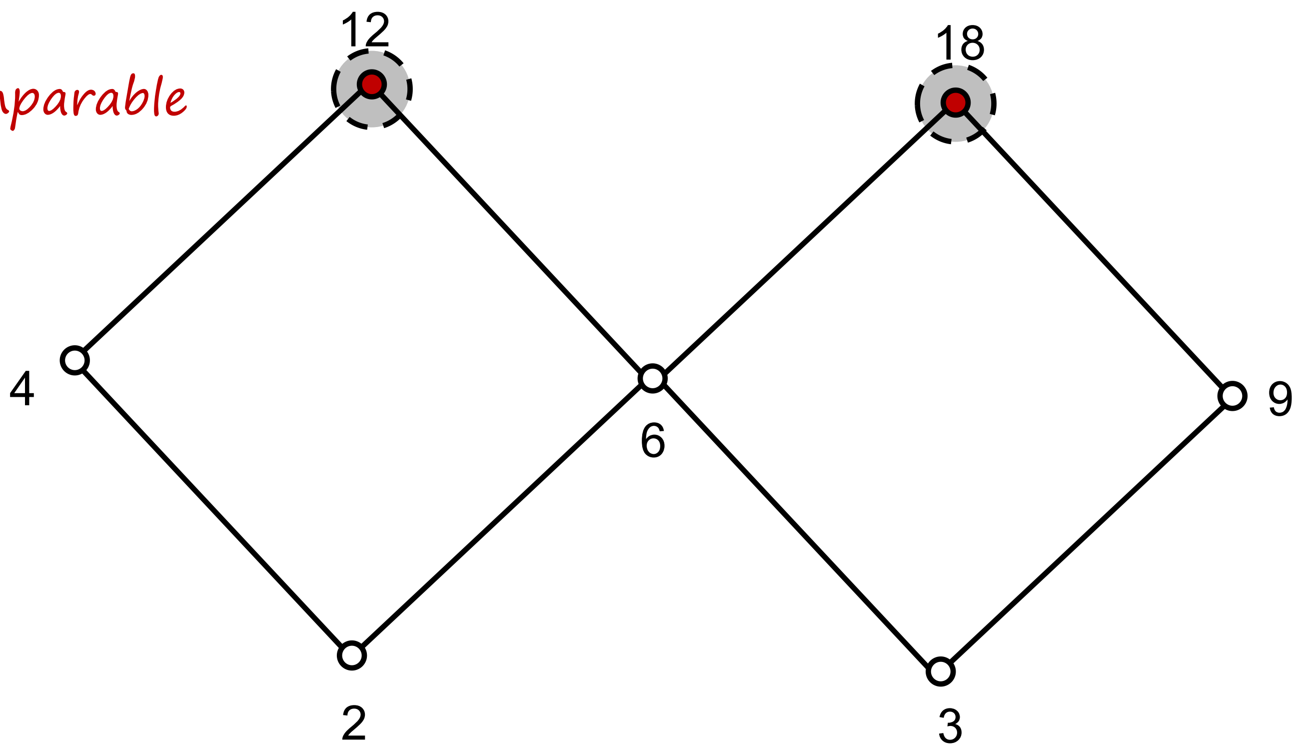


$$A = \{2,3,4,6,9,12,18\}$$
$$R = \{\langle x, y\rangle | x \in A \wedge y \in A \wedge x|y\}$$



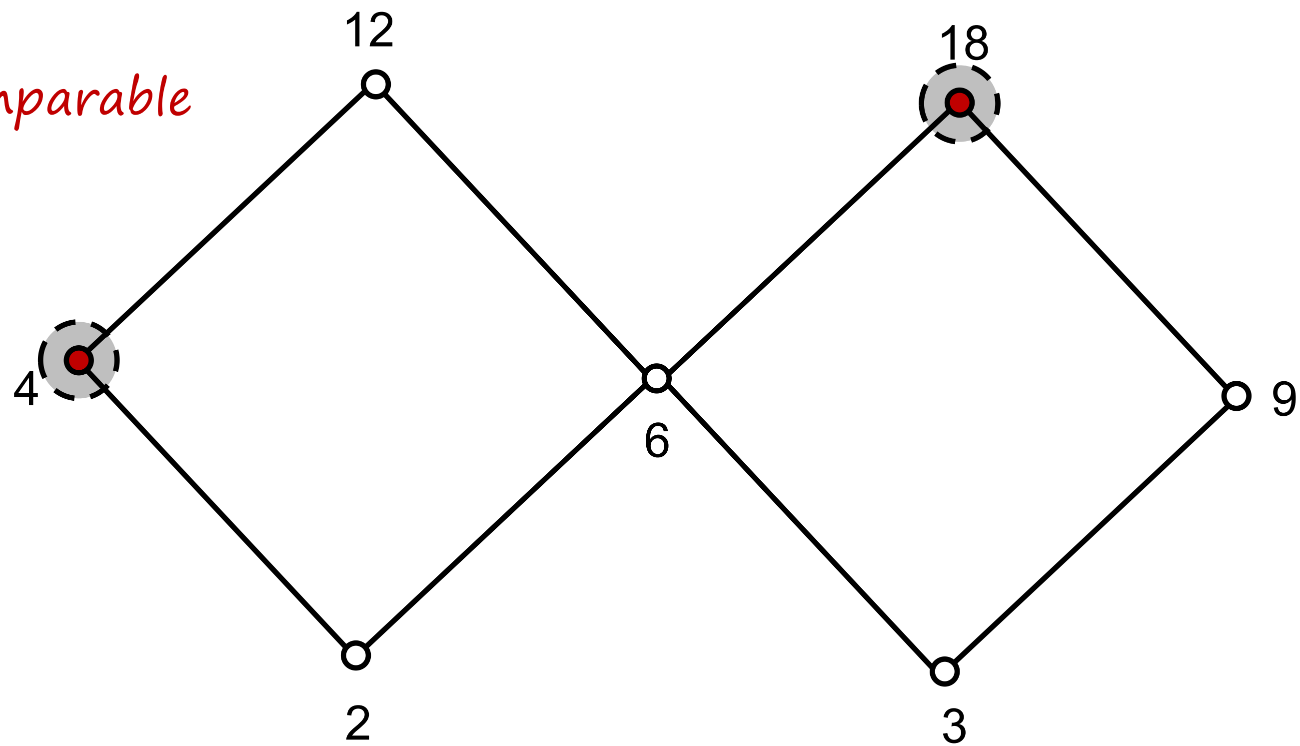
$$A = \{2,3,4,6,9,12,18\}$$
$$R = \{\langle x, y\rangle | x \in A \wedge y \in A \wedge x|y\}$$

Not Comparable



$$A = \{2,3,4,6,9,12,18\}$$
$$R = \{\langle x, y\rangle | x \in A \land y \in A \land x|y\}$$

Not Comparable



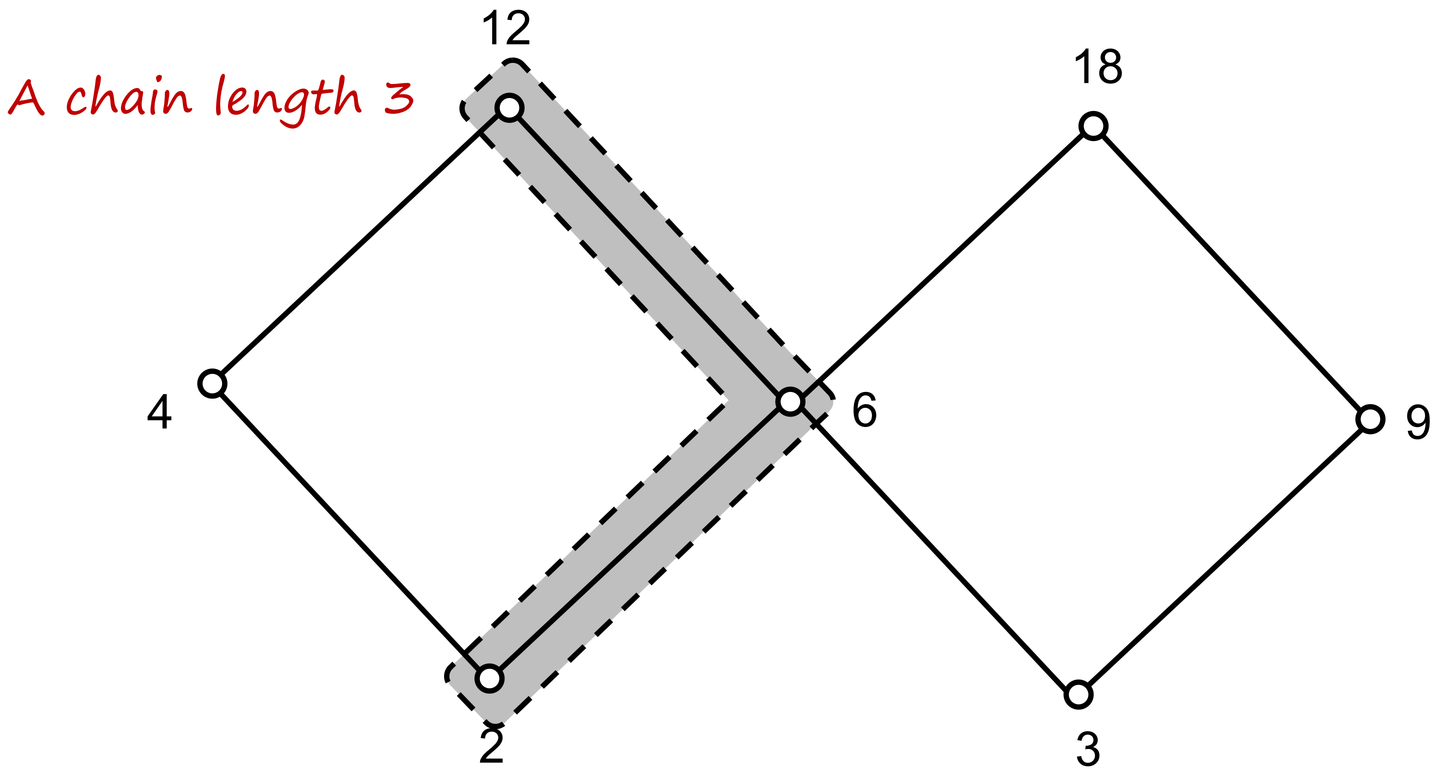
# Total Order Relation(全序关系)

- For a poset $\langle \boldsymbol{A}, \leq \rangle$:
  - For any $\boldsymbol{x} \in \boldsymbol{A}$ and $\boldsymbol{y} \in \boldsymbol{A}$, if $\boldsymbol{x} \leq \boldsymbol{y}$ **or** $\boldsymbol{y} \leq \boldsymbol{x}$ **,** then we call $\boldsymbol{x}$ and $\boldsymbol{y}$ are comparable.
  - If for any $\boldsymbol{x} \in \boldsymbol{A}$ and $\boldsymbol{y} \in \boldsymbol{A}$, we have $x$ and $y$ are comparable, then we called $\leq$ is a **total order relation** on $\boldsymbol{A}$, $\langle \boldsymbol{A}, \leq \rangle$ is a total order set.
- $\langle \mathbb{N}, \leq \rangle$ is a total order set
- For a non-empty set $\boldsymbol{A}$, $\langle \mathcal{P}(\boldsymbol{A}), \subseteq \rangle$ is not a total order set.

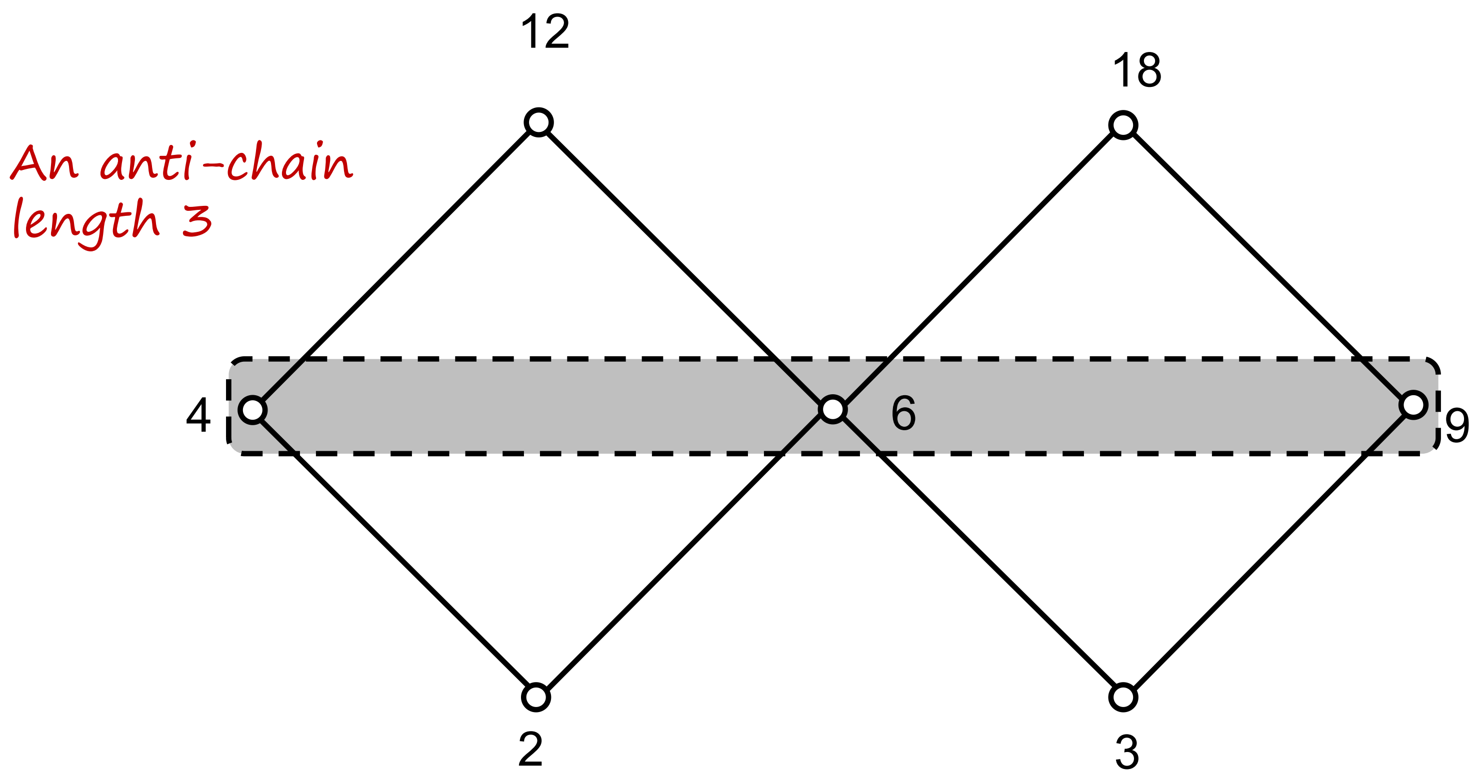
# Chain(链) and Anti-chain(反链)

- For a poset $\langle \boldsymbol{A}, \leq \rangle$, $\boldsymbol{B} \subseteq \boldsymbol{A}$:
  - For any $\boldsymbol{x} \in \boldsymbol{B}$ and $\boldsymbol{y} \in \boldsymbol{B}$, if $\boldsymbol{x}$ and $\boldsymbol{y}$ are **comparable**, we call $\boldsymbol{B}$ is a **chain** on $\boldsymbol{A}$, $|\boldsymbol{B}|$ is the length of this chain.
  - For any $\boldsymbol{x} \in \boldsymbol{B}$ and $\boldsymbol{y} \in \boldsymbol{B}$, if $\boldsymbol{x}$ and $\boldsymbol{y}$ are **not comparable**, we call $\boldsymbol{B}$ is an **anti-chain** on $\boldsymbol{A}$, $|\boldsymbol{B}|$ is the length of this anti-chain.

$$A = \{2,3,4,6,9,12,18\}$$
$$R = \{\langle x, y\rangle | x \in A \land y \in A \land x|y\}$$



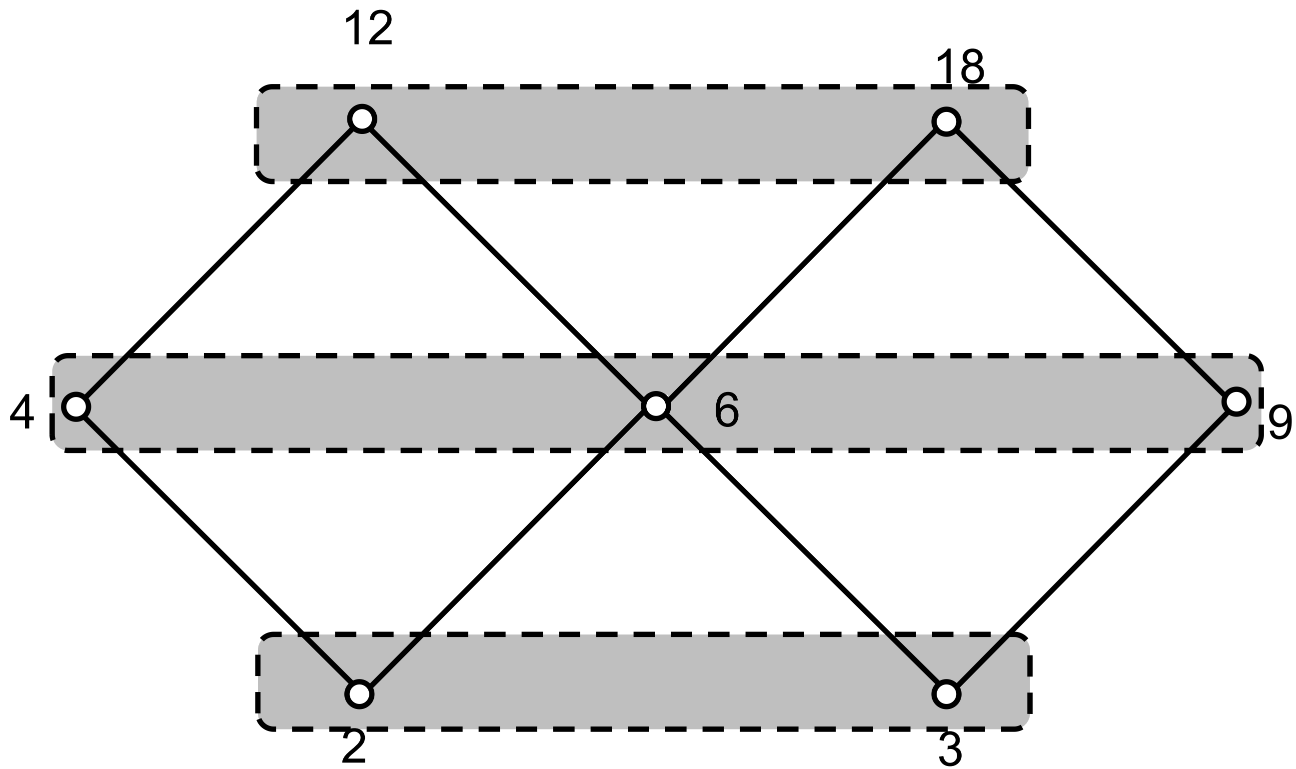
$$A = \{2,3,4,6,9,12,18\}$$
$$R = \{\langle x, y\rangle | x \in A \wedge y \in A \wedge x | y\}$$



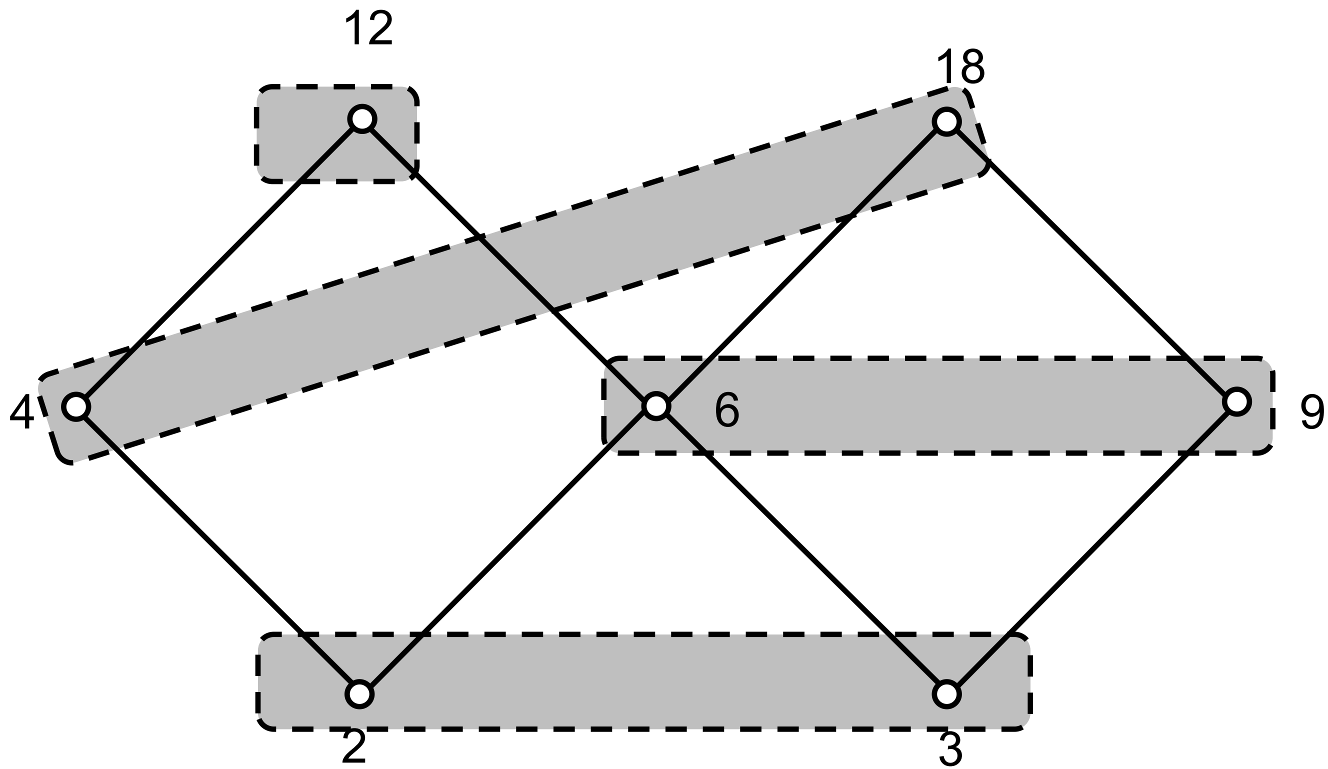
# Decompose Poset

- For a poset $\langle \boldsymbol{A}, \leq \rangle$, we can decompose $\boldsymbol{A}$ into non-disjoint anti-chain.

$$A = \{2,3,4,6,9,12,18\}$$
$$R = \{\langle x, y\rangle | x \in A \land y \in A \land x|y\}$$



$$A = \{2,3,4,6,9,12,18\}$$
$$R = \{\langle x, y\rangle | x \in A \wedge y \in A \wedge x|y\}$$



# Decompose Poset

- For a poset $\langle \boldsymbol{A}, \leq \rangle$, we can decompose $\boldsymbol{A}$ into non-disjoint anti-chain.

> For a poset $\langle \boldsymbol{A}, \leq \rangle$, if the maximal length of a chain is $n$, then the decomposition of $\boldsymbol{A}$ has at least $n$ anti-chains.

Proof:

Consider the longest chain which has length $n$, the $n$ elements of this chain must be in different anti-chains in the decomposition (as any of two are comparable, they can not in a same anti-chain). So there's at least $n$ anti-chains.

# Well Order Relation(良序关系)

- For a poset $\langle \boldsymbol{A}, \leq \rangle$, if any non-empty subset of $\boldsymbol{A}$ has the minimal elements, then we call $\leq$ is a **well order relation**, $\langle \boldsymbol{A}, \leq \rangle$ is a well order set.
- $\langle \mathbb{N}, \leq \rangle$ is a well order set. $\langle \mathbb{Z}, \leq \rangle$ is not a well order set.

# Well Order Relation

A well order set is a total order set.

- For a well-order set $\langle \boldsymbol{A}, \leq \rangle$, $\forall \boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{A}$,the subset $\{x, y\}$ has a minimum element（最小元）, so $x \leq y$ or $y \leq x$. $\langle \boldsymbol{A}, \leq \rangle$ is a total order set.

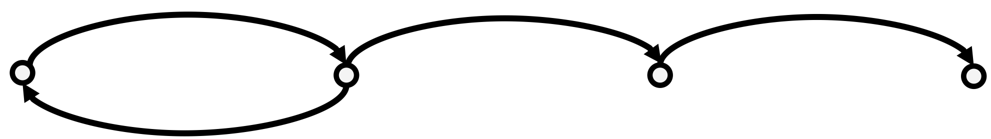
A finite total order set is a well order set.

# Properties of Relations

| | |
|---|---|
| Reflexivity | $(\forall x)(x \in A \rightarrow xRx)$ |
| Anti-reflexivity | $(\forall x)(x \in A \rightarrow xRx)$ |
| Symmetry | $(\forall x)(\forall y)((x \in A \land y \in A \land xRy) \rightarrow yRx)$ |
| Anti-symmetry | $(\forall x)(\forall y)((x \in A \land y \in A \land xRy \land yRx) \rightarrow x = y)$ |
| Transitivity | $(\forall x)(\forall y)(\forall z)$ $((x \in A \land y \in A \land z \in A \land xRy \land yRz) \rightarrow xRz)$ |

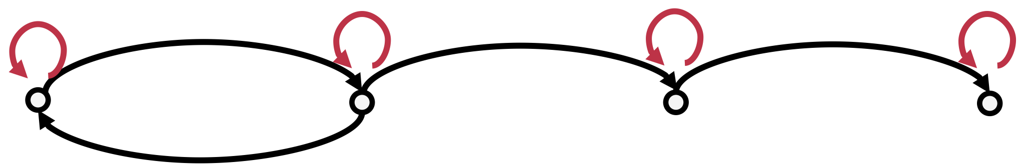
Some relation may not good enough
to have those relations

But we can help them!



Some relation may not good enough
to have those relations

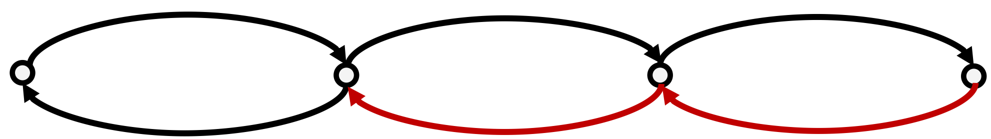
But we can help them!



Reflexivity

Some relation may not good enough
to have those relations

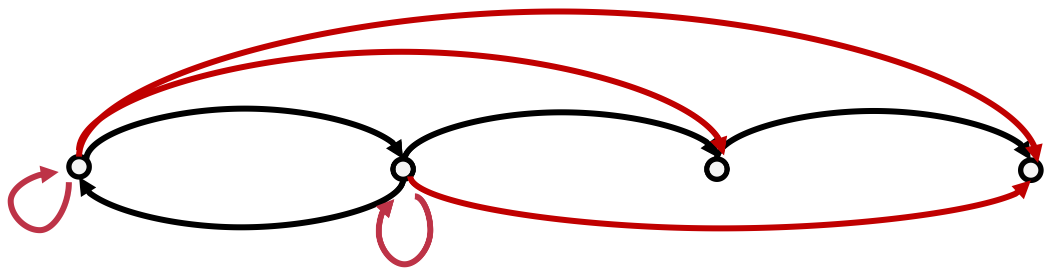
But we can help them!



Symmetry

Some relation may not good enough
to have those relations

But we can help them!



Transitivity

# Reflexive Closure(自反闭包)

- For a relation $\boldsymbol{R}$ on non-empty set $\boldsymbol{A}$, if there are another relation $R'$ such that:
  - $\boldsymbol{R'}$ is **reflexive**
  - $\boldsymbol{R} \subseteq \boldsymbol{R'}$.
  - For any **reflexive** relation $R''$ on $\boldsymbol{A}$, $\boldsymbol{R} \subseteq \boldsymbol{R''} \rightarrow \boldsymbol{R'} \subseteq \boldsymbol{R''}$.

R' is generated based on R

R' is the "smallest" relation to make R reflexive

# Reflexive Closure(自反闭包)

- For a relation $\boldsymbol{R}$ on non-empty set $\boldsymbol{A}$, if there are another relation $R'$ such that:
  - $\boldsymbol{R'}$ is **reflexive**
  - $\boldsymbol{R} \subseteq \boldsymbol{R'}$.
  - For any **reflexive** relation $R''$ on $\boldsymbol{A}$, $\boldsymbol{R} \subseteq \boldsymbol{R''} \rightarrow \boldsymbol{R'} \subseteq \boldsymbol{R''}$.
- We call that $\boldsymbol{R'}$ is the **reflexive** closure of $\boldsymbol{R}$, donated by $\boldsymbol{r(R)}$

# Symmetric Closure(对称闭包)

- For a relation $\boldsymbol{R}$ on non-empty set $\boldsymbol{A}$, if there are another relation $R'$ such that:
  - $\boldsymbol{R'}$ is **symmetric**
  - $\boldsymbol{R} \subseteq \boldsymbol{R'}$.
  - For any **symmetric** relation $R''$ on $\boldsymbol{A}$, $\boldsymbol{R} \subseteq \boldsymbol{R''} \rightarrow \boldsymbol{R'} \subseteq \boldsymbol{R''}$.
- We call that $\boldsymbol{R'}$ is the **symmetric** closure of $\boldsymbol{R}$, donated by $\boldsymbol{s(R)}$

# Transitive Closure(传递闭包)

- For a relation $\boldsymbol{R}$ on non-empty set $\boldsymbol{A}$, if there are another relation $R'$ such that:
  - $\boldsymbol{R'}$ is **transitive**
  - $\boldsymbol{R} \subseteq \boldsymbol{R'}$.
  - For any **transitive** relation $R''$ on $\boldsymbol{A}$, $\boldsymbol{R} \subseteq \boldsymbol{R''} \rightarrow \boldsymbol{R'} \subseteq \boldsymbol{R''}$.
- We call that $\boldsymbol{R'}$ is the **transitive** closure of $\boldsymbol{R}$, donated by $\boldsymbol{t(R)}$

# Closure

- For a relation $\boldsymbol{R}$ on non-empty set $\boldsymbol{A}$
  - $\boldsymbol{R}$ is reflexive $\Leftrightarrow \boldsymbol{r(R) = R}$
  - $\boldsymbol{R}$ is symmetric $\Leftrightarrow \boldsymbol{s(R) = R}$
  - $\boldsymbol{R}$ is transitive $\Leftrightarrow \boldsymbol{t(R) = R}$

# Constructing Closure

For a relation $\boldsymbol{R}$ on non-empty set $\boldsymbol{A}$, $\boldsymbol{r}(\boldsymbol{R}) = \boldsymbol{R} \cup \boldsymbol{R}^{\mathbf{0}}$

- $\boldsymbol{R} \cup \boldsymbol{R}^{\mathbf{0}}$ is reflexive as $\forall \boldsymbol{x} \in \boldsymbol{A}, \langle \boldsymbol{x}, \boldsymbol{x} \rangle \in \boldsymbol{R}^{\mathbf{0}}$
- Assume that $\boldsymbol{R}''$ is reflexive and $\boldsymbol{R} \subseteq \boldsymbol{R}''$.
  - Because $\boldsymbol{R}''$ is reflexive, $\boldsymbol{R}^{\mathbf{0}} = \boldsymbol{I}_{\boldsymbol{A}} \subseteq \boldsymbol{R}''$
  - $\boldsymbol{R} \subseteq \boldsymbol{R}'' \wedge \boldsymbol{R}^{\mathbf{0}} \subseteq \boldsymbol{R}'' \Rightarrow \boldsymbol{R} \cup \boldsymbol{R}^{\mathbf{0}} \subseteq \boldsymbol{R}''$

# Constructing Closure

For a relation $\boldsymbol{R}$ on non-empty set $\boldsymbol{A}$, $\boldsymbol{s}(\boldsymbol{R}) = \boldsymbol{R} \cup \boldsymbol{R}^{-1}$

- $\langle x, y\rangle \in R \cup R^{-1} \Leftrightarrow \langle x, y\rangle \in R \vee \langle x, y\rangle \in R^{-1}$

  $\Rightarrow \langle y, x\rangle \in R^{-1} \vee \langle y, x\rangle \in R \Leftrightarrow \langle y, x\rangle \in R \cup R^{-1}$

- Assume that $\boldsymbol{R}''$ is symmetric and $\boldsymbol{R} \subseteq \boldsymbol{R}''$.

$$\langle x, y\rangle \in R \cup R^{-1} \begin{cases} \langle x, y\rangle \in R \Rightarrow \langle x, y\rangle \in R'' \\ \langle x, y\rangle \in R^{-1} \Rightarrow \langle y, x\rangle \in R \\ \Rightarrow \langle y, x\rangle \in R'' \Rightarrow \langle x, y\rangle \in R'' \end{cases} \Rightarrow \quad \langle x, y\rangle \in R''$$

So $\boldsymbol{R} \cup \boldsymbol{R}^{-1} \subseteq \boldsymbol{R}''$

# Constructing Closure

For a relation $\boldsymbol{R}$ on non-empty set $\boldsymbol{A}$, $|\boldsymbol{A}| = n$,

$$\boldsymbol{t}(\boldsymbol{R}) = \boldsymbol{R} \cup \boldsymbol{R}^{\boldsymbol{2}} \cdots \cup \boldsymbol{R}^{\boldsymbol{n}}$$

# Lemma

Let $\boldsymbol{R}$ be a relation on a set $\boldsymbol{A}$. There is a path of length $\boldsymbol{n}(\boldsymbol{n} \in \mathbb{N}^{+})$ from $\boldsymbol{a}$ to $\boldsymbol{b}$ if and only if $\langle \boldsymbol{a}, \boldsymbol{b} \rangle \in \boldsymbol{R}^{\boldsymbol{n}}$

***Proof:***

By definition, there is a path from $\boldsymbol{a}$ to $\boldsymbol{b}$ of length 1 if and only if $\langle \boldsymbol{a}, \boldsymbol{b} \rangle \in \boldsymbol{R}$, so the lemma is true when $n = 1$.

Assume that the lemma is true when $n = k (k \in \mathbb{N}^{+})$. For $n = k + 1$

There is a path of length $k + 1$ from $\boldsymbol{a}$ to $\boldsymbol{b}$

$\Leftrightarrow$ There exists $\boldsymbol{c}$ such that there's a path of length 1 from $\boldsymbol{a}$ to $\boldsymbol{c}$ and a path of length $k$ from $\boldsymbol{c}$ to $\boldsymbol{b}$

$\Leftrightarrow (\exists \boldsymbol{c})\left( \langle \boldsymbol{a}, \boldsymbol{c} \rangle \in \boldsymbol{R} \wedge \langle \boldsymbol{c}, \boldsymbol{b} \rangle \in \boldsymbol{R}^{\boldsymbol{k}} \right) \Leftrightarrow \langle \boldsymbol{a}, \boldsymbol{b} \rangle \in \boldsymbol{R}^{\boldsymbol{k+1}}$

So the lemma is true when $n = k + 1$. Q.E.D

# Connectivity Relation(连接关系)

- Let $\boldsymbol{R}$ be a relation on a set $\boldsymbol{A}$. The **connectivity relation** $\boldsymbol{R}^+$ consists of the pairs $\langle \boldsymbol{a}, \boldsymbol{b} \rangle$ such that there is a path of length at least one from $\boldsymbol{a}$ to $\boldsymbol{b}$ in $\boldsymbol{R}$.
- By definition:

$$\boldsymbol{R}^+ = \boldsymbol{R} \cup \boldsymbol{R}^2 \cup \boldsymbol{R}^3 \cup \cdots$$

- But we only need to consider the path of length as most $|\boldsymbol{A}|$, assume that $|\boldsymbol{A}| = \boldsymbol{n}$

$$\boldsymbol{R}^+ = \boldsymbol{R} \cup \boldsymbol{R}^2 \cup \boldsymbol{R}^3 \cup \cdots \cup \boldsymbol{R}^n$$

# Constructing Closure

For a relation $\boldsymbol{R}$ on non-empty set $\boldsymbol{A}$, $|\boldsymbol{A}| = n$,
$$\boldsymbol{t(R)} = \boldsymbol{R^+} = \boldsymbol{R} \cup \boldsymbol{R^2} \cdots \cup \boldsymbol{R^n}$$

***Proof:***

- $R \subseteq R^+$ by definition.
- $\boldsymbol{R^+}$ **is transitive**: For $\langle \boldsymbol{x}, \boldsymbol{y} \rangle \in \boldsymbol{R^+}$ and $\langle \boldsymbol{y}, \boldsymbol{z} \rangle \in \boldsymbol{R^+}$, it means there are paths from $\boldsymbol{x}$ to $\boldsymbol{y}$ and $\boldsymbol{y}$ to $\boldsymbol{z}$ , so there's a path from $\boldsymbol{x}$ to $\boldsymbol{z}$. Then $\langle \boldsymbol{x}, \boldsymbol{z} \rangle \in \boldsymbol{R^+}$.
- Assume $\boldsymbol{R''}$ is transitive and $\boldsymbol{R} \subseteq \boldsymbol{R''}$.
  - For any $\langle \boldsymbol{x}, \boldsymbol{y} \rangle \in \boldsymbol{R^+}$, it means that there's a path of length $k (k \in \mathbb{N}^+)$ from $\boldsymbol{x}$ to $\boldsymbol{y}$,
  - Let the path be $\boldsymbol{x}, \boldsymbol{p_1}, .. \boldsymbol{p_{k-1}}, \boldsymbol{y}$, so $\langle \boldsymbol{x}, \boldsymbol{p_1} \rangle, \langle \boldsymbol{p_1}, \boldsymbol{p_2} \rangle ... \langle \boldsymbol{p_{k-1}}, \boldsymbol{y} \rangle \in \boldsymbol{R}$.
  - Because $\boldsymbol{R} \subseteq \boldsymbol{R''}$, $\langle \boldsymbol{x}, \boldsymbol{p_1} \rangle, \langle \boldsymbol{p_1}, \boldsymbol{p_2} \rangle ... \langle \boldsymbol{p_{k-1}}, \boldsymbol{y} \rangle \in \boldsymbol{R''}$.
  - According to the transitivity of $\boldsymbol{R''}$, we have $\langle \boldsymbol{x}, \boldsymbol{y} \rangle \in \boldsymbol{R''}$.
  - So $R^+ \subseteq R''$.

# Exercise

- $\boldsymbol{A = \{a, b, c\}}$, $\boldsymbol{R = \{\langle a, b\rangle, \langle b, c\rangle, \langle c, a\rangle\}}$, find $\boldsymbol{r(R)}$, $\boldsymbol{s(R)}$ and $\boldsymbol{t(R)}$.

# Exercise

- $A = \{a, b, c\}$, $R = \{\langle a, b\rangle, \langle b, c\rangle, \langle c, a\rangle\}$, find $r(R)$, $s(R)$ and $t(R)$.
- **Answer:**
  - $r(R) = R \cup R^0 = \{\langle a, a\rangle, \langle a, b\rangle, \langle b, b\rangle, \langle b, c\rangle, \langle c, a\rangle, \langle c, c\rangle\}$
  - $s(R) = R \cup R^{-1} = \{\langle a, b\rangle, \langle a, c\rangle, \langle b, a\rangle, \langle b, c\rangle, \langle c, a\rangle, \langle c, b\rangle\}$
  - $R^2 = \{\langle a, c\rangle, \langle b, a\rangle, \langle c, b\rangle\}$,
  - $R^3 = \{\langle a, a\rangle, \langle b, b\rangle, \langle c, c\rangle\}$
  - **So** $t(R) = R \cup R^2 \cup R^3 =$
    $\{\langle a, a\rangle, \langle a, b\rangle, \langle a, c\rangle, \langle b, a\rangle, \langle b, b\rangle, \langle b, c\rangle, \langle c, a\rangle, \langle c, b\rangle, \langle c, c\rangle\}$

# Transitive Closure

- $t(R) = R \cup R^2 \cup R^3 \cup \cdots \cup R^n$

It's hard to compute all of those when R is large!

# Transitive Closure: Warshall Algorithm

- Warshall Algorithm is an efficient algorithm to calculate transitive closure

```
Warshall_Algorithm(M[1…n,1…n]):
  M⁰ ← M
  for i ← 1 to n:
    for j ← 1 to n:
      if Mⁱ⁻¹[j,i]==1:
        for k ← 1 to n:
          Mⁱ[j,k] = Mⁱ⁻¹[j,k]∨ Mⁱ⁻¹[i,k]
  return Mⁿ
```

# Transitive Closure: Warshall Algorithm

$$\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

```
Warshall_Algorithm(M[1…n,1…n]):
  M⁰ ← M
  for i ← 1 to n:
    for j ← 1 to n:
      if Mⁱ⁻¹[j,i]==1:
        for k ← 1 to n:
          Mⁱ[j,k] = Mⁱ⁻¹[j,k]∨Mⁱ⁻¹[i,k]
  return Mⁿ
```

# Transitive Closure: Warshall Algorithm

$$\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

```
Warshall_Algorithm(M[1…n,1…n]):
  M⁰ ← M
  for i ← 1 to n:
    for j ← 1 to n:
      if Mⁱ⁻¹[j,i]==1:
        for k ← 1 to n:
          Mⁱ[j,k] = Mⁱ⁻¹[j,k]∨Mⁱ⁻¹[i,k]
  return Mⁿ
```

i=1,j=1

# Transitive Closure: Warshall Algorithm

$$\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

```
Warshall_Algorithm(M[1…n,1…n]):
  M⁰ ← M
  for i ← 1 to n:
    for j ← 1 to n:
      if Mⁱ⁻¹[j,i]==1:
        for k ← 1 to n:
          Mⁱ[j,k] = Mⁱ⁻¹[j,k]∨Mⁱ⁻¹[i,k]
  return Mⁿ
```

i=1,j=1

# Transitive Closure: Warshall Algorithm

$$\begin{pmatrix} 1 \vee 1 & 1 \vee 1 & 0 \vee 0 & 0 \vee 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

```
Warshall_Algorithm(M[1…n,1…n]):
  M⁰ ← M
  for i ← 1 to n:
    for j ← 1 to n:
      if Mⁱ⁻¹[j,i]==1:
        for k ← 1 to n:
          Mⁱ[j,k] = Mⁱ⁻¹[j,k]∨Mⁱ⁻¹[i,k]
  return Mⁿ
```

i=1,j=1

# Transitive Closure: Warshall Algorithm

$$\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

```
Warshall_Algorithm(M[1…n,1…n]):
  M⁰ ← M
  for i ← 1 to n:
    for j ← 1 to n:
      if Mⁱ⁻¹[j,i]==1:
        for k ← 1 to n:
          Mⁱ[j,k] = Mⁱ⁻¹[j,k]∨Mⁱ⁻¹[i,k]
  return Mⁿ
```

i=1,j=1

# Transitive Closure: Warshall Algorithm

$$\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

```
Warshall_Algorithm(M[1…n,1…n]):
  M⁰ ← M
  for i ← 1 to n:
    for j ← 1 to n:
      if Mⁱ⁻¹[j,i]==1:
        for k ← 1 to n:
          Mⁱ[j,k] = Mⁱ⁻¹[j,k]∨Mⁱ⁻¹[i,k]
  return Mⁿ
```

i=1,j=2

# Transitive Closure: Warshall Algorithm

$$\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

```
Warshall_Algorithm(M[1…n,1…n]):
  M⁰ ← M
  for i ← 1 to n:
    for j ← 1 to n:
      if Mⁱ⁻¹[j,i]==1:
        for k ← 1 to n:
          Mⁱ[j,k] = Mⁱ⁻¹[j,k]⋁Mⁱ⁻¹[i,k]
  return Mⁿ
```

i=1,j=3

# Transitive Closure: Warshall Algorithm

$$\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

```
Warshall_Algorithm(M[1…n,1…n]):
  M⁰ ← M
  for i ← 1 to n:
    for j ← 1 to n:
      if Mⁱ⁻¹[j,i]==1:
        for k ← 1 to n:
          Mⁱ[j,k] = Mⁱ⁻¹[j,k]∨Mⁱ⁻¹[i,k]
  return Mⁿ
```

i=1, j=4

# Transitive Closure: Warshall Algorithm

$$\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

```
Warshall_Algorithm(M[1…n,1…n]):
  M⁰ ← M
  for i ← 1 to n:
    for j ← 1 to n:
      if Mⁱ⁻¹[j,i]==1:
        for k ← 1 to n:
          Mⁱ[j,k] = Mⁱ⁻¹[j,k]∨Mⁱ⁻¹[i,k]
  return Mⁿ
```

i=2,j=1

# Transitive Closure: Warshall Algorithm

$$\begin{pmatrix} 1\vee 0 & 1\vee 0 & 0\vee 0 & 0\vee 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

i=2,j=1

```
Warshall_Algorithm(M[1…n,1…n]):
  M⁰ ← M
  for i ← 1 to n:
    for j ← 1 to n:
      if Mⁱ⁻¹[j,i]==1:
        for k ← 1 to n:
          Mⁱ[j,k] = Mⁱ⁻¹[j,k]∨Mⁱ⁻¹[i,k]
  return Mⁿ
```

# Transitive Closure: Warshall Algorithm

$$\begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

i=2, j=1

```
Warshall_Algorithm(M[1…n,1…n]):
  M⁰ ← M
  for i ← 1 to n:
    for j ← 1 to n:
      if Mⁱ⁻¹[j,i]==1:
        for k ← 1 to n:
          Mⁱ[j,k] = Mⁱ⁻¹[j,k]∨Mⁱ⁻¹[i,k]
  return Mⁿ
```

# Transitive Closure: Warshall Algorithm

$$\begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

```
Warshall_Algorithm(M[1…n,1…n]):
  M⁰ ← M
  for i ← 1 to n:
    for j ← 1 to n:
      if Mⁱ⁻¹[j,i]==1:
        for k ← 1 to n:
          Mⁱ[j,k] = Mⁱ⁻¹[j,k]∨Mⁱ⁻¹[i,k]
  return Mⁿ
```

i=2,j=2

# Transitive Closure: Warshall Algorithm

$$\begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

```
Warshall_Algorithm(M[1…n,1…n]):
  M⁰ ← M
  for i ← 1 to n:
    for j ← 1 to n:
      if Mⁱ⁻¹[j,i]==1:
        for k ← 1 to n:
          Mⁱ[j,k] = Mⁱ⁻¹[j,k]∨Mⁱ⁻¹[i,k]
  return Mⁿ
```

i=2,j=3

# Transitive Closure: Warshall Algorithm

$$\begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & \color{red}{1} & 0 & 0 \end{pmatrix}$$

```
Warshall_Algorithm(M[1…n,1…n]):
  M⁰ ← M
  for i ← 1 to n:
    for j ← 1 to n:
      if Mⁱ⁻¹[j,i]==1:
        for k ← 1 to n:
          Mⁱ[j,k] = Mⁱ⁻¹[j,k]∨Mⁱ⁻¹[i,k]
  return Mⁿ
```

i=2,j=4

# Transitive Closure: Warshall Algorithm

$$\begin{array}{c} \\ \rightarrow \\ \\ \rightarrow \end{array}\begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 \vee 0 & 1 \vee 0 & 0 \vee 0 & 0 \vee 1 \end{pmatrix}$$

i=2,j=4

```
Warshall_Algorithm(M[1…n,1…n]):
  M⁰ ← M
  for i ← 1 to n:
    for j ← 1 to n:
      if Mⁱ⁻¹[j,i]==1:
        for k ← 1 to n:
          Mⁱ[j,k] = Mⁱ⁻¹[j,k]∨Mⁱ⁻¹[i,k]
  return Mⁿ
```

# Transitive Closure: Warshall Algorithm

$$\begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

i=2,j=4

```
Warshall_Algorithm(M[1…n,1…n]):
  M⁰ ← M
  for i ← 1 to n:
    for j ← 1 to n:
      if Mⁱ⁻¹[j,i]==1:
        for k ← 1 to n:
          Mⁱ[j,k] = Mⁱ⁻¹[j,k]∨Mⁱ⁻¹[i,k]
  return Mⁿ
```

# Transitive Closure: Warshall Algorithm

$$\begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

i=4, j=1

```
Warshall_Algorithm(M[1…n,1…n]):
  M⁰ ← M
  for i ← 1 to n:
    for j ← 1 to n:
      if Mⁱ⁻¹[j,i]==1:
        for k ← 1 to n:
          Mⁱ[j,k] = Mⁱ⁻¹[j,k]∨Mⁱ⁻¹[i,k]
  return Mⁿ
```

# Transitive Closure: Warshall Algorithm

$$\begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

i=4, j=1

```
Warshall_Algorithm(M[1…n,1…n]):
  M⁰ ← M
  for i ← 1 to n:
    for j ← 1 to n:
      if Mⁱ⁻¹[j,i]==1:
        for k ← 1 to n:
          Mⁱ[j,k] = Mⁱ⁻¹[j,k]∨Mⁱ⁻¹[i,k]
  return Mⁿ
```

# Transitive Closure: Warshall Algorithm

$$\begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

```
Warshall_Algorithm(M[1…n,1…n]):
  M⁰ ← M
  for i ← 1 to n:
    for j ← 1 to n:
      if Mⁱ⁻¹[j,i]==1:
        for k ← 1 to n:
          Mⁱ[j,k] = Mⁱ⁻¹[j,k]∨Mⁱ⁻¹[i,k]
  return Mⁿ
```

i=4,j=1

# Transitive Closure: Warshall Algorithm

$$\begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

```
Warshall_Algorithm(M[1…n,1…n]):
  M⁰ ← M
  for i ← 1 to n:
    for j ← 1 to n:
      if Mⁱ⁻¹[j,i]==1:
        for k ← 1 to n:
          Mⁱ[j,k] = Mⁱ⁻¹[j,k]∨Mⁱ⁻¹[i,k]
  return Mⁿ
```

$i=4, j=1$

# Transitive Closure: Warshall Algorithm

$$\begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

i=4, j=4

```
Warshall_Algorithm(M[1…n,1…n]):
  M⁰ ← M
  for i ← 1 to n:
    for j ← 1 to n:
      if Mⁱ⁻¹[j,i]==1:
        for k ← 1 to n:
          Mⁱ[j,k] = Mⁱ⁻¹[j,k]∨Mⁱ⁻¹[i,k]
  return Mⁿ
```

# Transitive Closure: Warshall Algorithm

$$\begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

i=4,j=4

```
Warshall_Algorithm(M[1…n,1…n]):
  M⁰ ← M
  for i ← 1 to n:
    for j ← 1 to n:
      if Mⁱ⁻¹[j,i]==1:
        for k ← 1 to n:
          Mⁱ[j,k] = Mⁱ⁻¹[j,k]∨Mⁱ⁻¹[i,k]
  return Mⁿ
```

# Transitive Closure: Warshall Algorithm

$$\begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

```
Warshall_Algorithm(M[1…n,1…n]):
  M⁰ ← M
  for i ← 1 to n:
    for j ← 1 to n:
      if Mⁱ⁻¹[j,i]==1:
        for k ← 1 to n:
          Mⁱ[j,k] = Mⁱ⁻¹[j,k]∨Mⁱ⁻¹[i,k]
  return Mⁿ
```

i=4,j=4

# Closures

- For a relation $\boldsymbol{R}$ on a non-empty set $\boldsymbol{A}$
  - If $\boldsymbol{R}$ is reflexive, then $\boldsymbol{s}(\boldsymbol{R})$ and $\boldsymbol{t}(\boldsymbol{R})$ are reflexive
  - If $\boldsymbol{R}$ is symmetric, then $\boldsymbol{r}(\boldsymbol{R})$ and $\boldsymbol{t}(\boldsymbol{R})$ are symmetric
  - If $\boldsymbol{R}$ is transitive, then $\boldsymbol{r}(\boldsymbol{R})$ are transitive.

s(R) may not be transitive!!

# Closures

- For a relation $\boldsymbol{R}$ on non-empty set $\boldsymbol{A}$, $rs(R)$ means $r(s(R))$, the similar definition for other closure symbols, so we have:
  - $rs(R) = sr(R)$
  - $rt(R) = tr(R)$

  r is commutative with t and s

  - $st(R) \subseteq ts(R)$

  s and t are not commutative

# Binary Relation: Recap

- **Binary Relation Basics**
- **Equivalence Relation and Partitions**
  - Reflexivity, Symmetry and Transitivity.
- **Compatible Relation and Covers**
  - Reflexivity and Symmetry
- **Partial Order Relation**
  - Reflexivity, Anti-symmetry and Transitivity
- **Closure**

# Next

- **Function**
- **Introduce to Automata and DFA**
- **Turing Machine Basics.**