

Function

Zeyu Mi

2021-12-10



Adapted From:

《数理逻辑与集合论》 10.8,11.1,11.2

Review

- **Partial Order Relation**

- Reflexivity, Anti-symmetry and Transitivity
- Total order relation

- **Closure**

- $r(R) = R \cup R^0$
- $s(R) = R \cup R^{-1}$
- $t(R) = R^* = R \cup R^2 \cup \dots \cup R^n$

How we investigate computability?

2.1 Discussion on Problems

Part1. Intro & Set theory(I) : Basics & Formal Language.

Part2. Set Theory(II): Axiom system & Cardinality.

Part3. Capture Structures: Binary Relation & Function

2.2 Discussion on Computation

Part1. Turing Machine Basics.

Part2. Variants of Turing Machine. Church-Turing Thesis.

2.3 Discussion on computability

Part1. The Language of Turing Machine. R & RE

Part2. Undecidability.

Transitive Closure

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



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($M[1..n, 1..n]$):

$M^0 \leftarrow M$

for $i \leftarrow 1$ to n :

for $j \leftarrow 1$ to n :

if $M^{i-1}[j, i] = 1$:

for $k \leftarrow 1$ to n :

$M^i[j, k] = M^{i-1}[j, k] \vee M^{i-1}[i, k]$

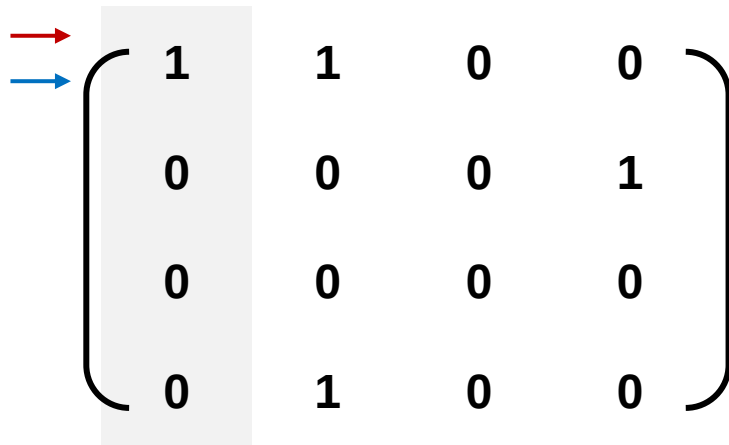
return M^n

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



1	1	0	0
0	0	0	1
0	0	0	0
0	1	0	0

$i=1, j=1$

```
Warshall_Algorithm(M[1...n,1...n]):  
   $M^0 \leftarrow M$   
  for  $i \leftarrow 1$  to  $n$ :  
    for  $j \leftarrow 1$  to  $n$ :  
      if  $M^{i-1}[j,i]=1$ :  
        for  $k \leftarrow 1$  to  $n$ :  
           $M^i[j,k] = M^{i-1}[j,k] \vee M^{i-1}[i,k]$   
  return  $M^n$ 
```

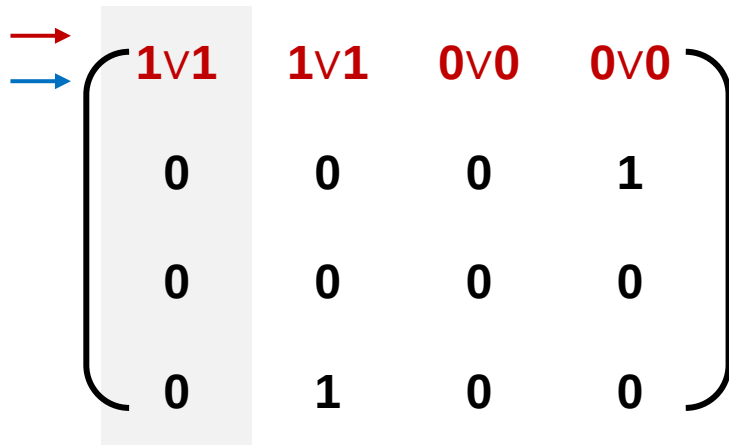
Transitive Closure: Warshall Algorithm

1	1	0	0
0	0	0	1
0	0	0	0
0	1	0	0

$i=1, j=1$

```
Warshall_Algorithm(M[1...n,1...n]):  
   $M^0 \leftarrow M$   
  for  $i \leftarrow 1$  to  $n$ :  
    for  $j \leftarrow 1$  to  $n$ :  
      if  $M^{i-1}[j,i]==1$ :  
        for  $k \leftarrow 1$  to  $n$ :  
           $M^i[j,k] = M^{i-1}[j,k] \vee M^{i-1}[i,k]$   
  return  $M^n$ 
```

Transitive Closure: Warshall Algorithm



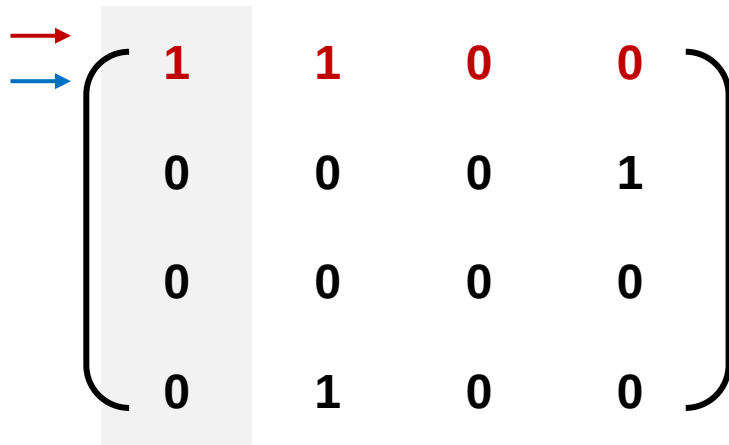
The diagram shows a 4x4 matrix with columns labeled 1v1, 1v1, 0v0, and 0v0. The first column is highlighted with a grey background. A red arrow points to the top row, and a blue arrow points to the first column. The matrix is enclosed in large square brackets.

1v1	1v1	0v0	0v0
0	0	0	1
0	0	0	0
0	1	0	0

$i=1, j=1$

```
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

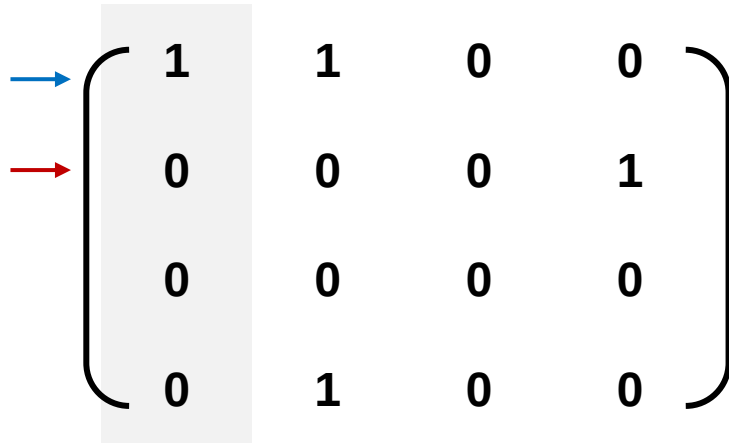


1	1	0	0
0	0	0	1
0	0	0	0
0	1	0	0

$i=1, j=1$

```
Warshall_Algorithm(M[1...n,1...n]):  
   $M^0 \leftarrow M$   
  for  $i \leftarrow 1$  to  $n$ :  
    for  $j \leftarrow 1$  to  $n$ :  
      if  $M^{i-1}[j,i]=1$ :  
        for  $k \leftarrow 1$  to  $n$ :  
           $M^i[j,k] = M^{i-1}[j,k] \vee M^{i-1}[i,k]$   
  return  $M^n$ 
```

Transitive Closure: Warshall Algorithm

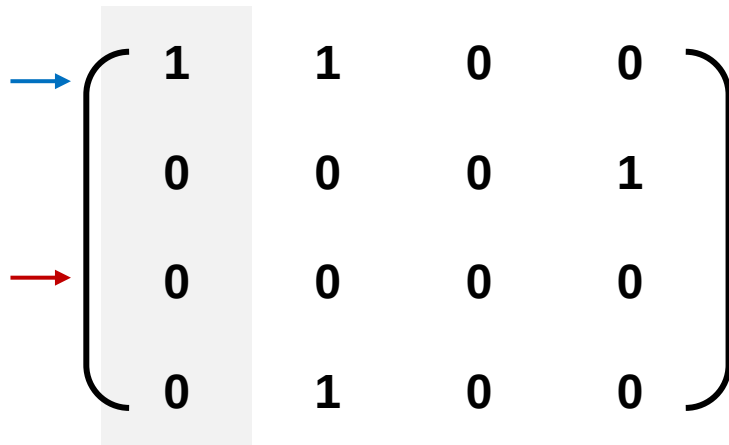


→	1	1	0	0
→	0	0	0	1
	0	0	0	0
	0	1	0	0

$i=1, j=2$

```
Warshall_Algorithm(M[1...n,1...n]):  
   $M^0 \leftarrow M$   
  for  $i \leftarrow 1$  to  $n$ :  
    for  $j \leftarrow 1$  to  $n$ :  
      if  $M^{i-1}[j,i] == 1$ :  
        for  $k \leftarrow 1$  to  $n$ :  
           $M^i[j,k] = M^{i-1}[j,k] \vee M^{i-1}[i,k]$   
  return  $M^n$ 
```

Transitive Closure: Warshall Algorithm

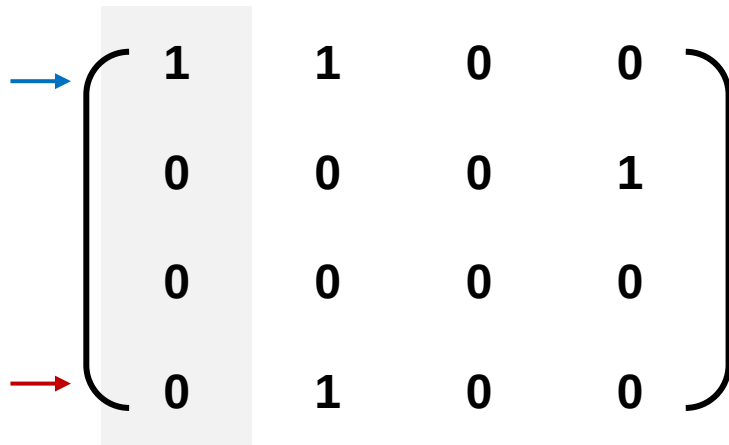


1	1	0	0
0	0	0	1
0	0	0	0
0	1	0	0

$i=1, j=3$

```
Warshall_Algorithm(M[1...n,1...n]):  
   $M^0 \leftarrow M$   
  for  $i \leftarrow 1$  to  $n$ :  
    for  $j \leftarrow 1$  to  $n$ :  
      if  $M^{i-1}[j,i]=1$ :  
        for  $k \leftarrow 1$  to  $n$ :  
           $M^i[j,k] = M^{i-1}[j,k] \vee M^{i-1}[i,k]$   
  return  $M^n$ 
```

Transitive Closure: Warshall Algorithm

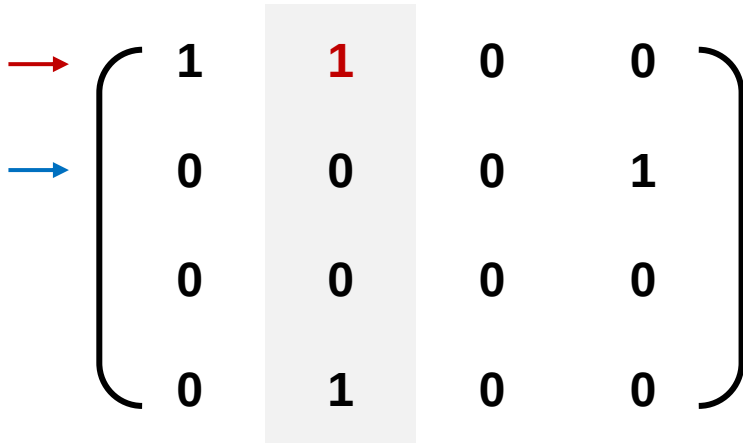


1	1	0	0
0	0	0	1
0	0	0	0
0	1	0	0

$i=1, j=4$

```
Warshall_Algorithm(M[1...n,1...n]):  
   $M^0 \leftarrow M$   
  for  $i \leftarrow 1$  to  $n$ :  
    for  $j \leftarrow 1$  to  $n$ :  
      if  $M^{i-1}[j,i] == 1$ :  
        for  $k \leftarrow 1$  to  $n$ :  
           $M^i[j,k] = M^{i-1}[j,k] \vee M^{i-1}[i,k]$   
  return  $M^n$ 
```

Transitive Closure: Warshall Algorithm


$$\begin{matrix} \rightarrow \\ \rightarrow \end{matrix} \begin{pmatrix} 1 & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

$i=2, j=1$

```
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

$$\begin{array}{c} \rightarrow \\ \rightarrow \end{array} \left(\begin{array}{cccc} 1 \vee 0 & 1 \vee 0 & 0 \vee 0 & 0 \vee 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{array} \right)$$

$i=2, j=1$

```
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
```

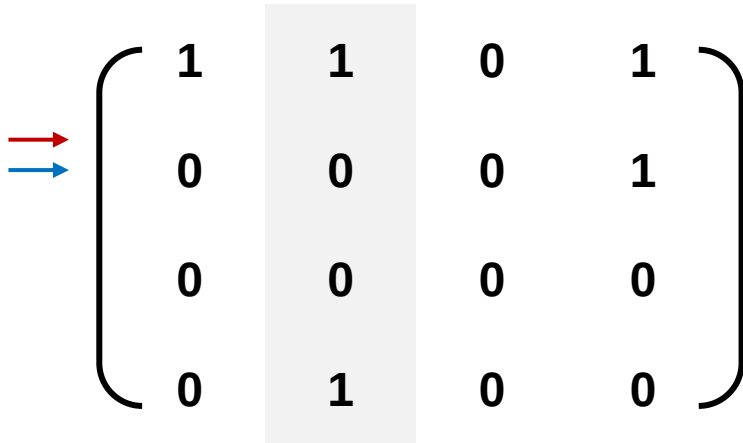
Transitive Closure: Warshall Algorithm

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

$i=2, j=1$

```
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


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

$i=2, j=2$

```
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

$$\begin{matrix} \xrightarrow{\text{blue}} \\ \xrightarrow{\text{red}} \end{matrix} \begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

$i=2, j=3$

```
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

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

$i=2, j=4$

```
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

$$\begin{array}{c} \rightarrow \\ \rightarrow \end{array} \left(\begin{array}{cccc} 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 \vee 0 & 1 \vee 0 & 0 \vee 0 & 0 \vee 1 \end{array} \right)$$

$i=2, j=4$

```
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

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

$i=2, j=4$

```
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

$$\begin{matrix} \rightarrow & \begin{pmatrix} 1 & 1 & 0 & \mathbf{1} \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ \leftarrow 0 & 1 & 0 & 1 \end{pmatrix} \end{matrix}$$

$i=4, j=1$

```
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

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

$i=4, j=1$

```
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

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

$i=4, j=1$

```
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
```

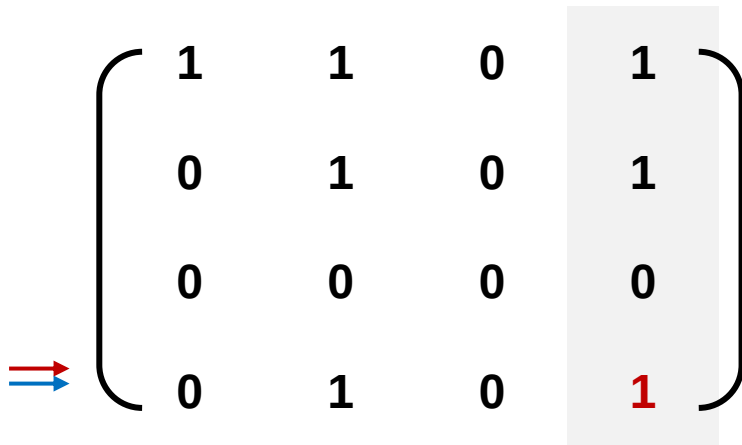
Transitive Closure: Warshall Algorithm

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

$i=4, j=1$

```
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


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

$i=4, j=4$

```
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
```

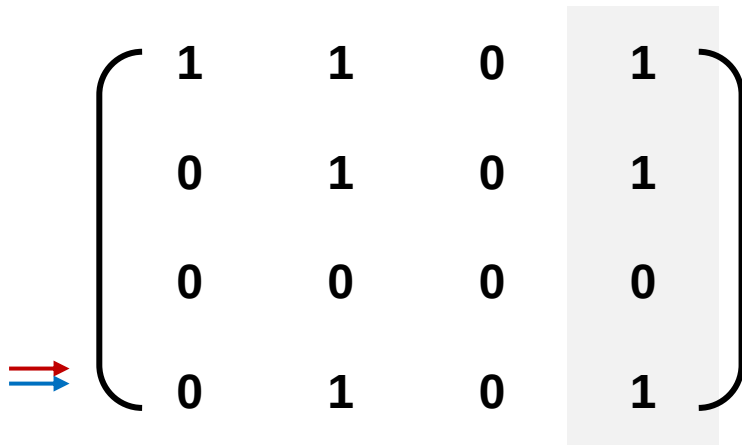
Transitive Closure: Warshall Algorithm

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

$i=4, j=4$

```
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



A 4x4 matrix is shown with the following values:

1	1	0	1
0	1	0	1
0	0	0	0
0	1	0	1

A red and blue arrow points to the first column. The fourth column is highlighted in grey.

$i=4, j=4$

```
Warshall_Algorithm(M[1...n,1...n]):  
   $M^0 \leftarrow M$   
  for  $i \leftarrow 1$  to  $n$ :  
    for  $j \leftarrow 1$  to  $n$ :  
      if  $M^{i-1}[j,i] == 1$ :  
        for  $k \leftarrow 1$  to  $n$ :  
           $M^i[j,k] = M^{i-1}[j,k] \vee M^{i-1}[i,k]$   
  return  $M^n$ 
```

Closures

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

$s(R)$ may not be transitive!!

Closures

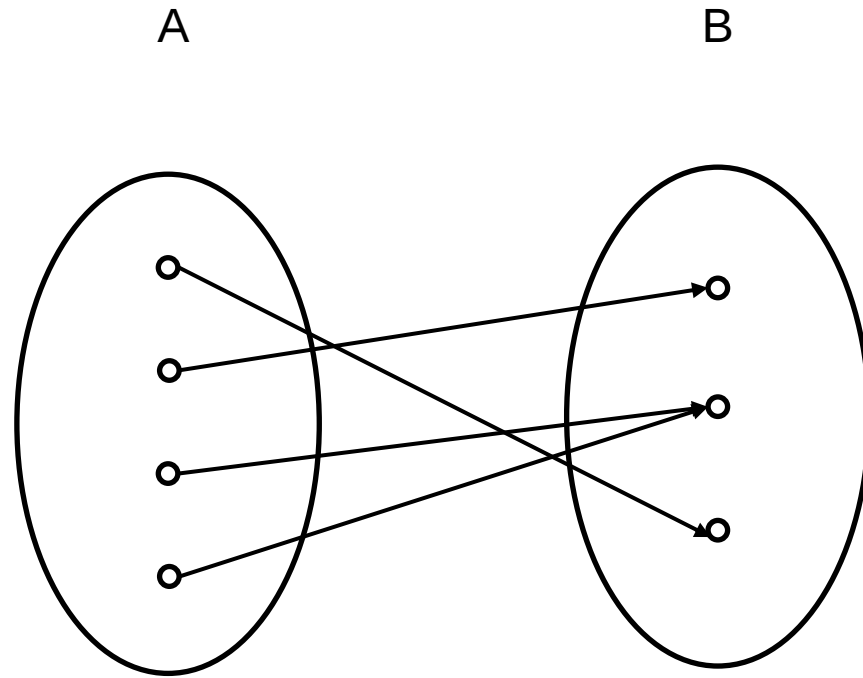
- For a relation R on non-empty set A , $rs(R)$ means $r(s(R))$, the similar definition for other closure symbols, so we have:

$$\begin{array}{l} - rs(R) = sr(R) \\ - rt(R) = tr(R) \end{array} \quad \left. \begin{array}{l} \diagup \\ \diagdown \end{array} \right\} \quad \text{\textit{r is commutative with t and s}}$$

$$- st(R) \subseteq ts(R) \quad \text{---} \quad \text{\textit{s and t are not commutative}}$$

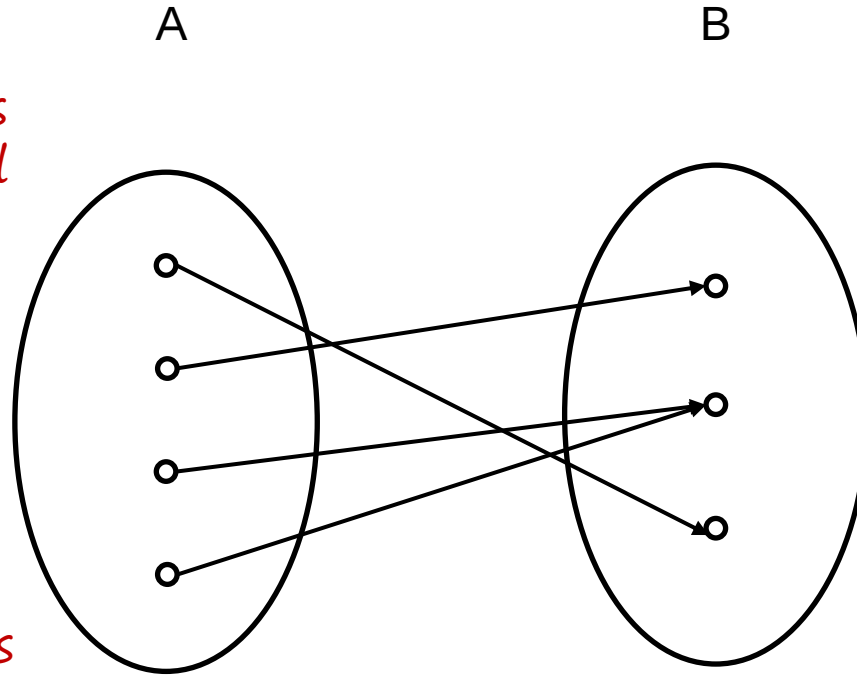
Binary Relation: Recap

- **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**



A special relation from A to B

Every elements
in A are paired



Every elements
in A are paired
only once

A special relation from A to B

Function

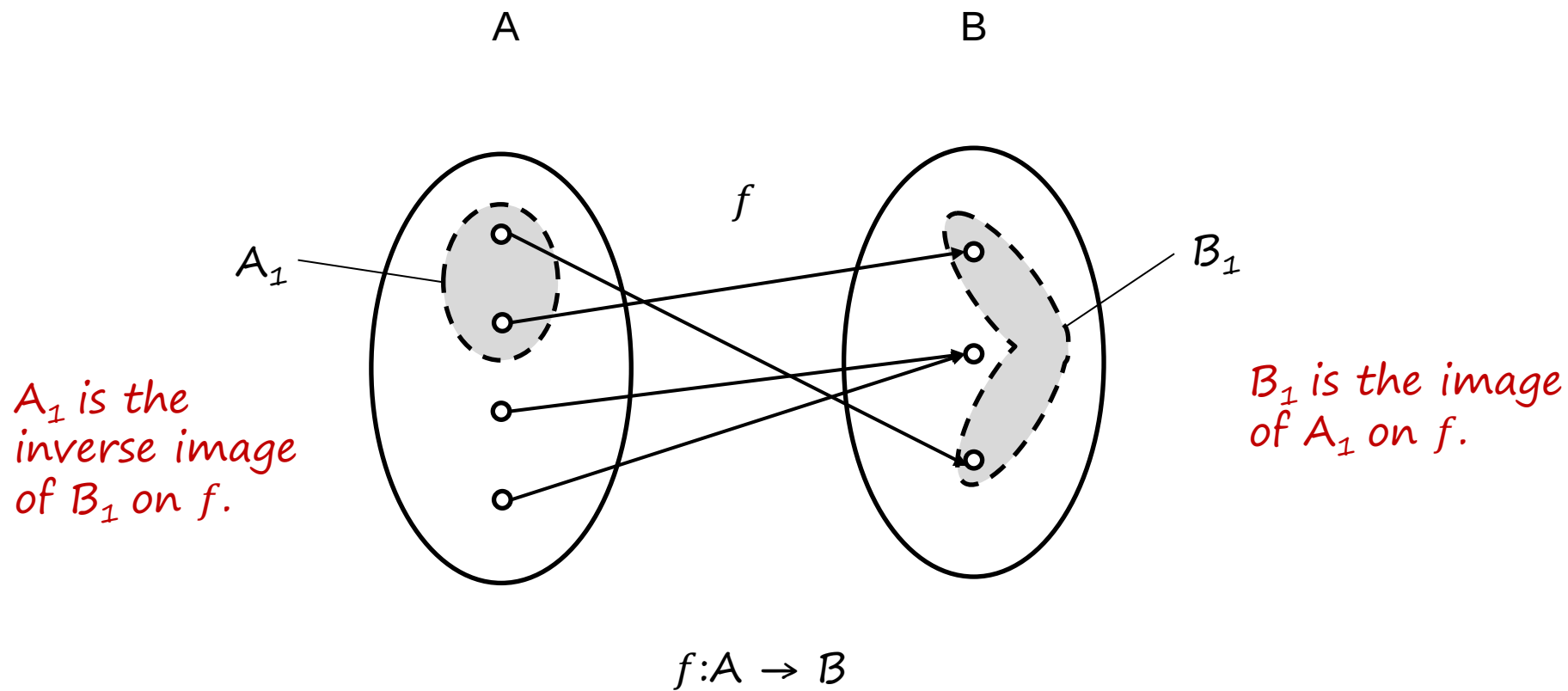
- A relation f from set A to set B is called a **function** if:
 - For any $x \in \text{dom}(f)$, there exists **unique** y such that xfy
$$(\forall x)(\forall y_1)(\forall y_2)((xfy_1 \wedge xfy_2) \rightarrow y_1 = y_2)$$
 - $\text{dom}(f) = A$
$$(\forall x)(x \in A \rightarrow (\exists y)(y \in B \wedge xfy))$$
- A function f from set A to set B is denoted as $f: A \rightarrow B$.
- xfy writing as $f: x \mapsto y$ or $f(x) = y$

Function

- All the function from set A to set B is A_B
$$A_B = \{f | f: A \rightarrow B\}$$
- If $|A| = n$, $|B| = m$
 - then $|A_B| = m^n$

Function

- For relation R on A , for a set B
 - the **restriction**(限制) of R on B , denote as $R \upharpoonright B$ is:
$$R \upharpoonright B = \{\langle x, y \rangle \mid \langle x, y \rangle \in R \wedge x \in B\}$$
 - The **image**(象) of R on B , denote as $R[B]$ is:
$$R[B] = \{y \mid \exists x (\langle x, y \rangle \in R \wedge x \in B)\}$$
- Similarly for a function $f: A \rightarrow B$, $\forall A_1 \subseteq A$ we can define $f[A_1]$ to be the **image**(象) of A_1 on function f .
- $\forall B_1 \subseteq B$, $f^{-1}[B_1]$ is the **inverse image or preimage**(原象) of B_1 on f .



Common Functions

- $f: A \rightarrow B$, then f is called **constant function** (常函数) if:
$$(\exists y)(\forall x)(x \in A \rightarrow f(x) = y)$$
- $f: A \rightarrow A$, then f is called **identity function** (恒等函数) if:
$$(\forall x)(x \in A \rightarrow f(x) = x) \text{ or } f = I_A$$
- $f: A^n \rightarrow A$ ($n \in \mathbb{N}$), then f is called a **n-ary operator** (n元运算)
- $f: \mathbb{R} \rightarrow \mathbb{R}$
 - If $(x \leq y) \rightarrow (f(x) \leq f(y))$, we call f is **monotonically increasing** (单调递增)
 - If $(x < y) \rightarrow (f(x) < f(y))$, we call f is **strictly monotonically increasing** (严格单调递增)
 - If $(x \leq y) \rightarrow (f(x) \geq f(y))$, we call f is **monotonically decreasing** (单调递减)
 - If $(x < y) \rightarrow (f(x) > f(y))$, we call f is **strictly monotonically decreasing** (严格单调递减)

Common Functions

- $F: A \rightarrow B_C$ is called a **functional**(noun, 泛函)
 - We define that $F(y) = (f(x) = x + y)$, where $f(x)$ is a certain function.
 - Then $F(1)$ is the function $f(x) = x + 1$
- Let E be the universal set, for any $A \subseteq E$, the **indicator function, or characteristic function**(特征函数) $\chi_A: E \rightarrow \{0, 1\}$ is defined by:

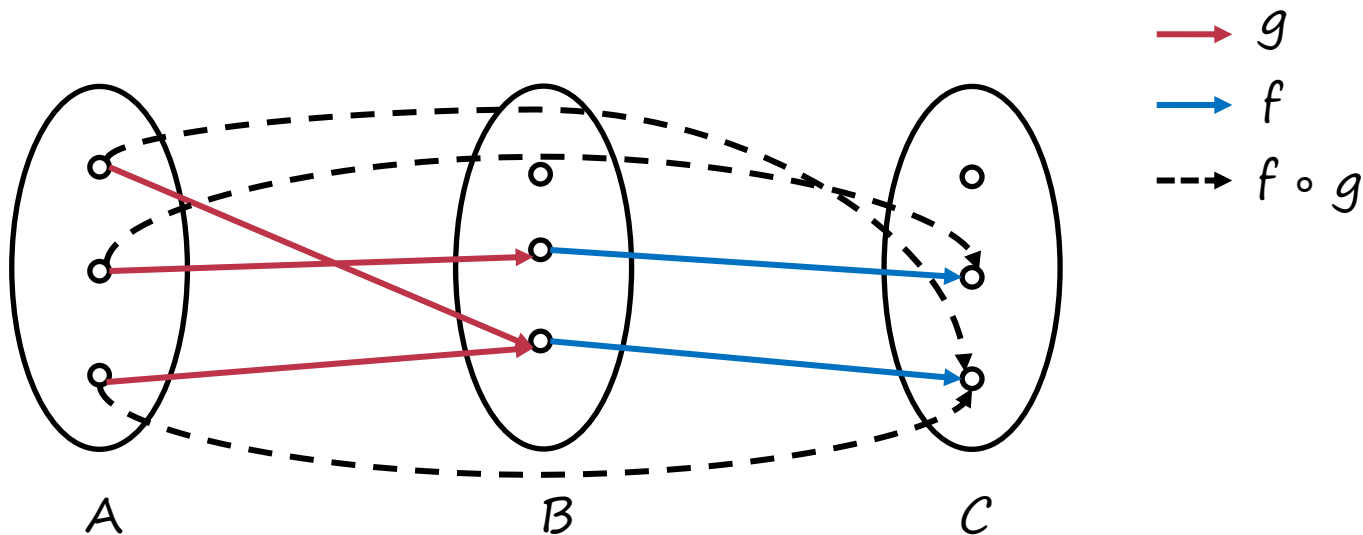
$$\chi_A(x) = \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases}$$

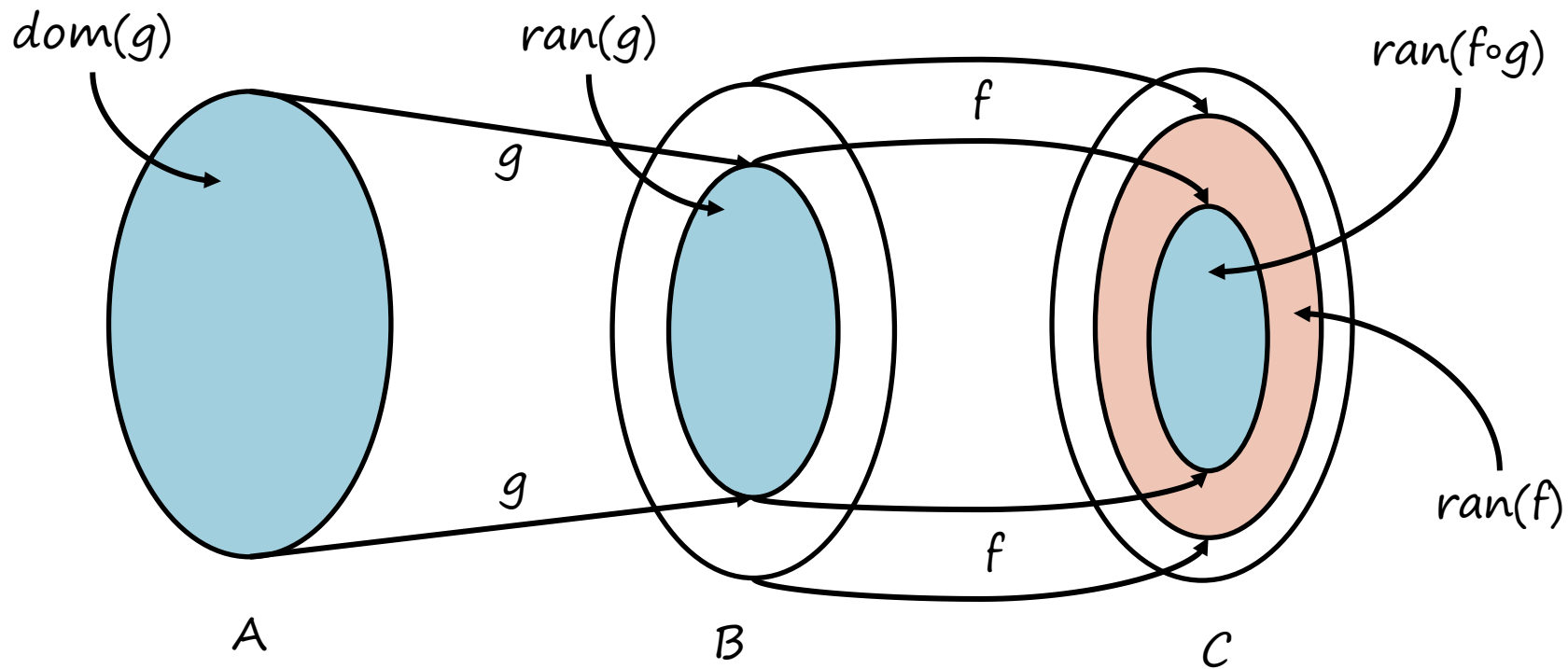
- Let R be an equivalence relation on A , g is a **canonical map or natural map**(典型映射或自然映射) from A to quotient set A/R if:

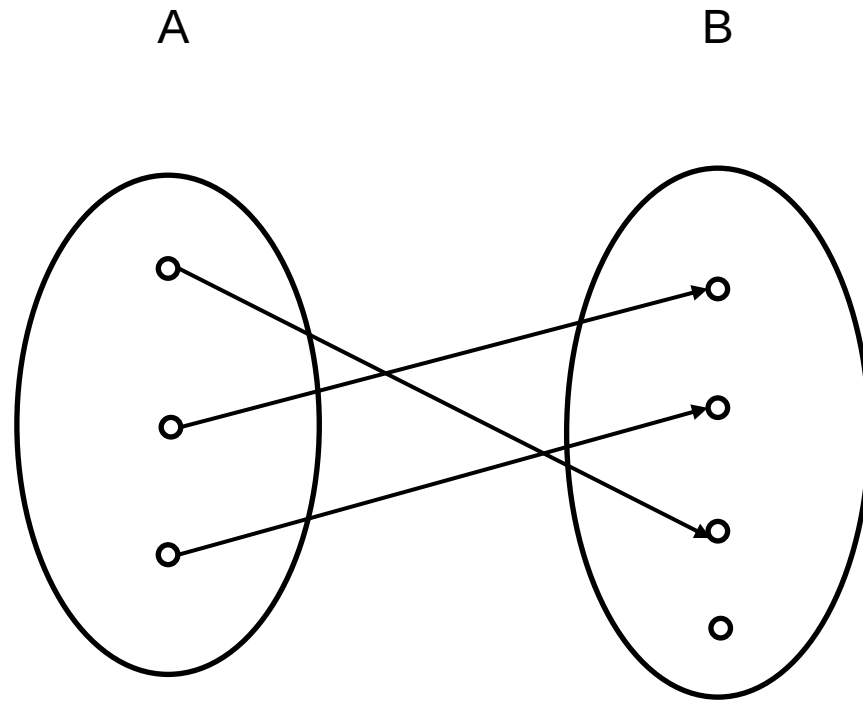
$$g: A \rightarrow A/R, g(x) = [x]_R$$

Compose of Function

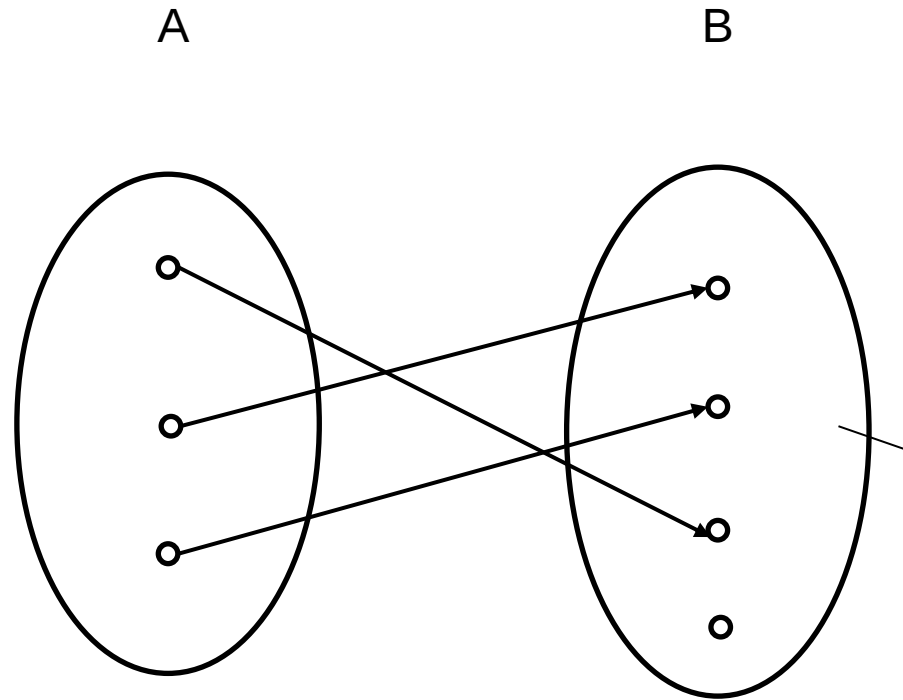
- Let $g: A \rightarrow B$, $f: B \rightarrow C$, then:
 - $f \circ g$ is a function $f \circ g: A \rightarrow C$
 - For any $x \in A$, $(f \circ g)(x) = f(g(x))$







A special function



*No elements in B
are paired twice*

A special function

Injection(单射)

- $f: A \rightarrow B$ is an **injection** or **injective** if for any $x_1, x_2 \in A, x_1 \neq x_2$, then $f(x_1) \neq f(x_2)$.

$$(\forall x_1)(\forall x_2)(x_1 \in A \wedge x_2 \in A \wedge x_1 \neq x_2 \rightarrow f(x_1) \neq f(x_2))$$

Different inputs has different outputs

- Or if $f(x_1) = f(x_2)$, then $x_1 = x_2$

$$(\forall x_1)(\forall x_2)(x_1 \in A \wedge x_2 \in A \wedge f(x_1) = f(x_2) \rightarrow x_1 = x_2)$$

Same outputs have the same input

Injection

- $f(x) = x \ (\mathbb{R} \rightarrow \mathbb{R})$
 - injective
 - For any $x_1 \neq x_2$, $f(x_1) = x_1 \neq x_2 = f(x_2)$
- $f(x) = x^2 \ (\mathbb{R} \rightarrow \mathbb{R})$
 - not injective.
 - $f(-1) = f(1) = 1$

Injection and Composition

Let $g: A \rightarrow B$, $f: B \rightarrow C$, if f , g are injective, then $f \circ g$ is also injective.

Proof:

For any $x_1, x_2 \in A$, $x_1 \neq x_2$, we have $g(x_1) \neq g(x_2)$ since g is injective.

Now $g(x_1), g(x_2) \in B$, and $g(x_1) \neq g(x_2)$, then we have $f(g(x_1)) \neq f(g(x_2))$ since f is injective.

$(f \circ g)(x) = f(g(x))$ by definition, so $(f \circ g)(x_1) \neq (f \circ g)(x_2)$

Q.E.D

Injection and Composition

Let $g: A \rightarrow B$, $f: B \rightarrow C$, if $f \circ g$ is injective, then g is also injective.

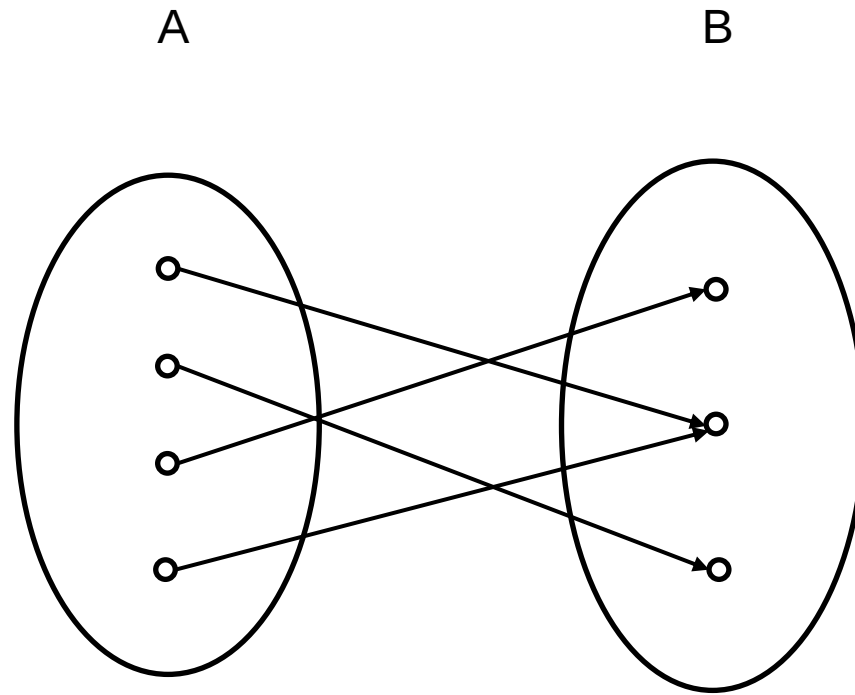
Proof:

Assume that g is not injective, then there exists $x_1, x_2 \in A$, $x_1 \neq x_2$ and $g(x_1) = g(x_2)$.

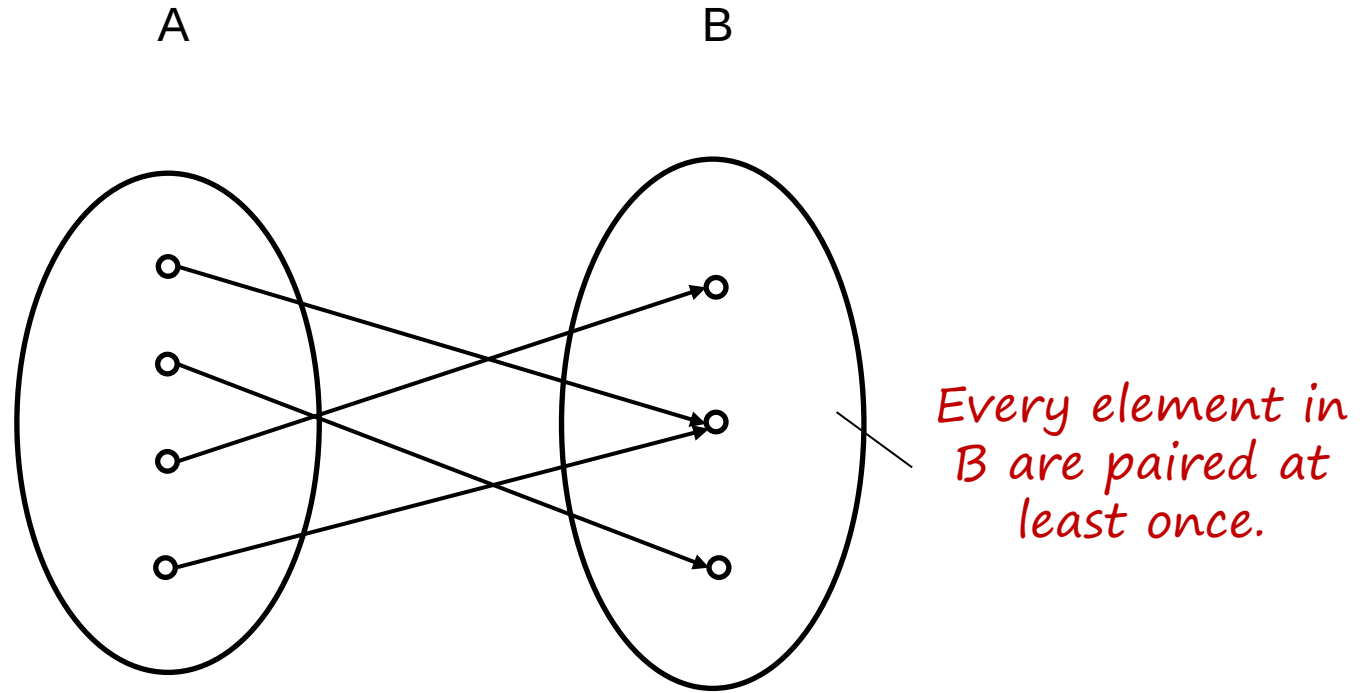
Now $g(x_1), g(x_2) \in B$, and $g(x_1) = g(x_2)$, then we have $f(g(x_1)) = f(g(x_2))$.

But $f \circ g$ is injective, it means $f(g(x_1)) \neq f(g(x_2))$. Contradiction

Q.E.D



A special function



A special function

Surjection(满射)

- $f: A \rightarrow B$ is a **surjection** or **surjective** if $\text{ran}(f) = B$, or for any $y \in B$, there exists $x \in A$ such that $f(x) = y$

$$(\forall y)(\exists x)(y \in B \rightarrow (x \in A \wedge f(x) = y))$$

Every possible output has an input

Surjection

- $f(x) = x + 5$ ($\mathbb{R} \rightarrow \mathbb{R}$)
 - surjective.
 - For any $x_0 \in \mathbb{R}$, $f(x_0 - 5) = x_0$
- $f(x) = x^2$ ($\mathbb{R} \rightarrow \mathbb{R}$)
 - not surjective.
 - There's no element $x_0 \in \mathbb{R}$ such that $f(x_0) = x_0^2 = -1$

Surjection and Composition

Let $g: A \rightarrow B$, $f: B \rightarrow C$, if f , g are surjective, then $f \circ g$ is also surjective.

Proof:

For any $y_1 \in C$, we have $x_1 \in B$ such that $f(x_1) = y_1$ since f is surjective.

Now $x_1 \in B$, so we have $x_0 \in A$ such that $g(x_0) = x_1$ since g is surjective.

So for any $y_1 \in C$, we have $x_0 \in A$ such that $f(g(x_0)) = y_1$, or $(f \circ g)(x) = y_1$, it means that $f \circ g$ is surjective

Q.E.D

Surjection and Composition

Let $g: A \rightarrow B$, $f: B \rightarrow C$, if $f \circ g$ is surjective, then f is also surjective.

Proof:

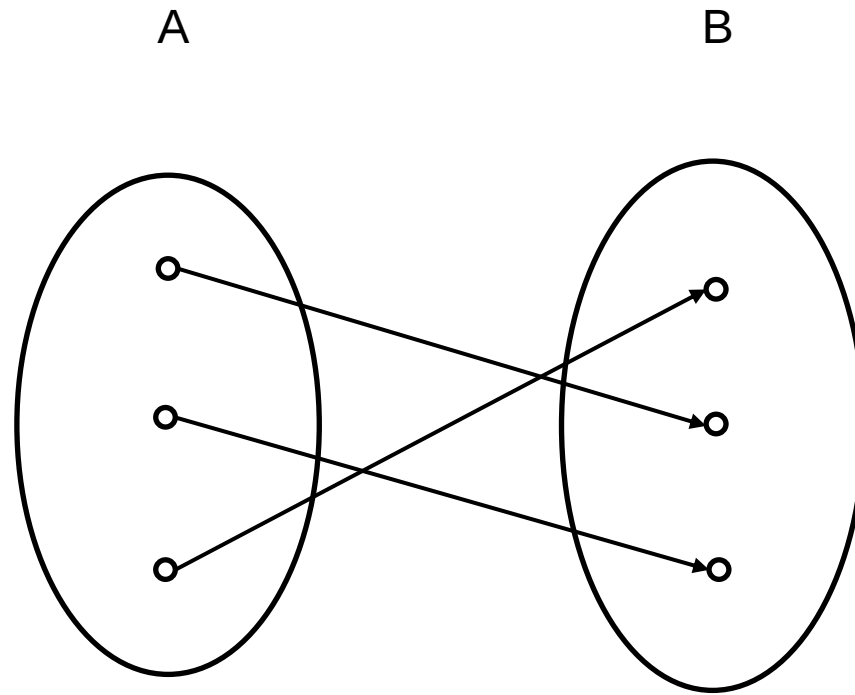
For any $y \in C$, we have $x \in A$ such that $(f \circ g)(x) = y$ as $f \circ g$ is surjective.

Let $x_0 = g(x) \in B$, so we have $(f \circ g)(x) = f(g(x)) = f(x_0) = y$.

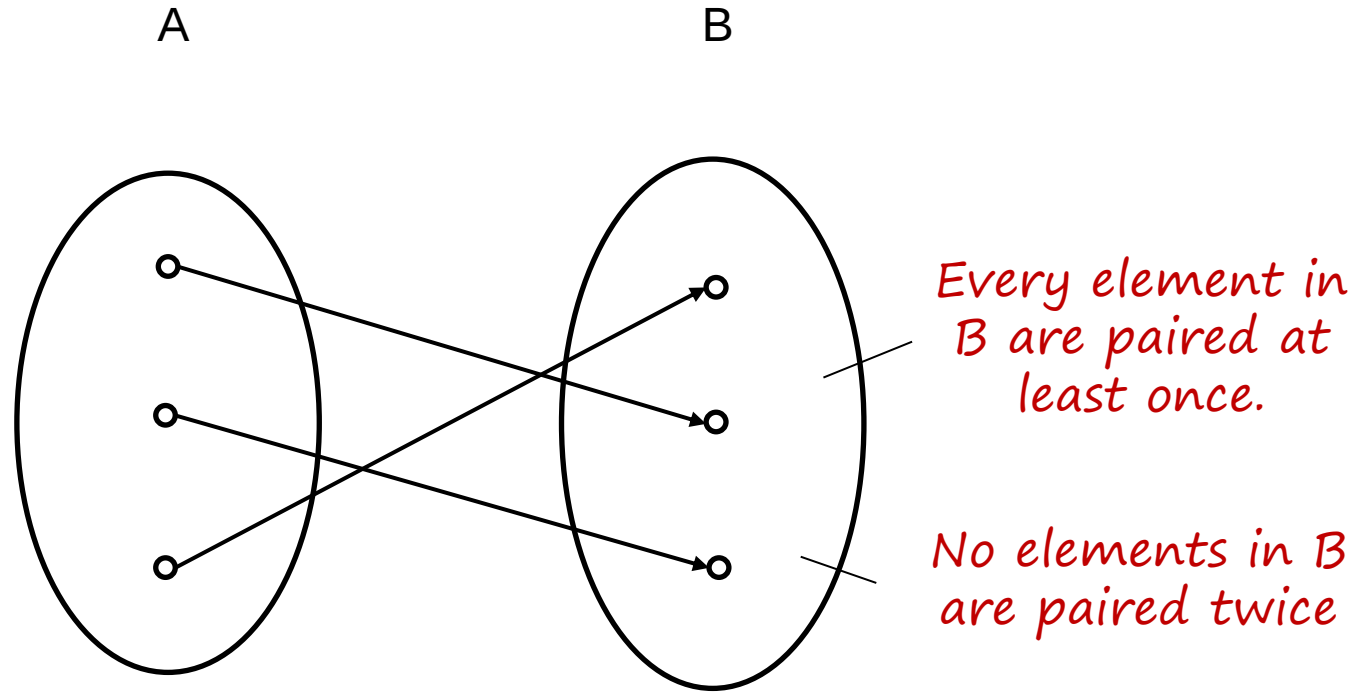
So for any $y \in C$, we have $x_0 \in B$ such that $f(x_0) = y$.

f is surjective.

Q.E.D



A special function



A special function

Bijection(双射)

- $f: A \rightarrow B$ is a **bijection** or **bijjective** if f is **both** injective and surjective.
- Bijections are sometimes called **one-to-one correspondences(一一对应)**.

Bijection

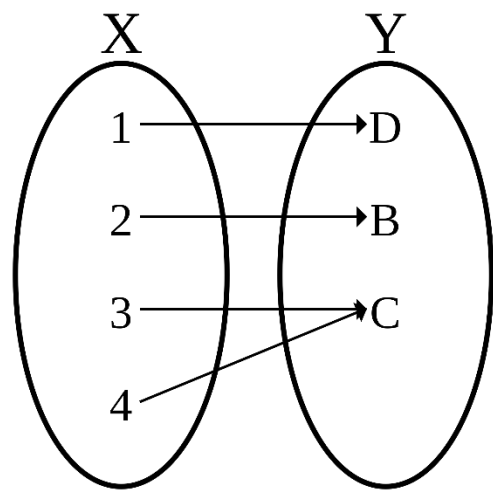
- $f(x) = x + 5$ ($\mathbb{R} \rightarrow \mathbb{R}$) is bijective.
- $f(x) = e^x$ ($\mathbb{R} \rightarrow \mathbb{R}$) is not bijective since it's not surjective.
- $f(x) = x^3 - x$ ($\mathbb{R} \rightarrow \mathbb{R}$) is not bijective since it's not injective.

Bijection and Composition

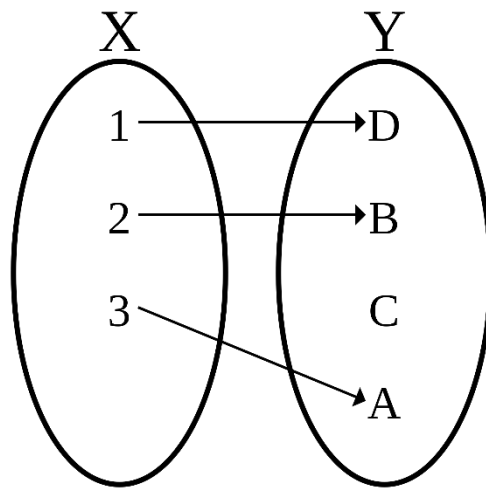
Let $g: A \rightarrow B$, $f: B \rightarrow C$, if f, g is bijective, then $f \circ g$ is also bijective.

Let $g: A \rightarrow B$, $f: B \rightarrow C$, if $f \circ g$ is bijective, then g is injective and f is surjective.

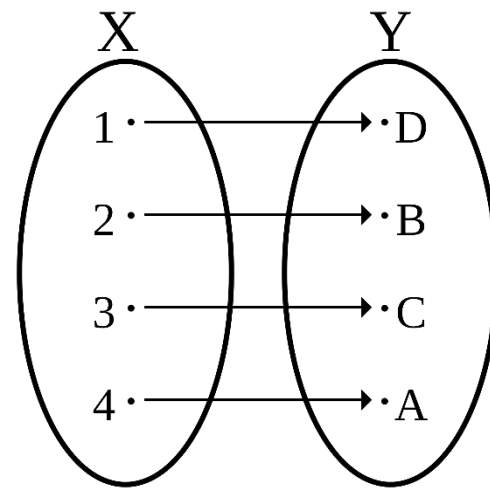
Bijection, injection and surjection



Surjection



Injection



Bijection

Comparing Cardinalities

For set A and B , A and B are **equianumerous (等势)** if there is a way to pair their elements off without leaving any elements uncovered, marked as $A \approx B$. Otherwise marked as $\neg A \approx B$.

Comparing Cardinalities

~~For set A and B , A and B are **equianumerous (等势)** if there is a way to pair their elements off without leaving any elements uncovered, marked as $A \approx B$. Otherwise marked as $\neg A \approx B$.~~

For set A and B , A and B are **equianumerous (等势)** if there is a **bijection from A to B** , marked as $A \approx B$. Otherwise marked as $\neg A \approx B$.

Comparing Cardinalities

For set A and B , $\text{card}(A) \leq \text{card}(B)$ if there is a way to pair their elements off without leaving **any elements of A** uncovered.

Comparing Cardinalities

~~For set A and B , $\text{card}(A) \leq \text{card}(B)$ if there is a way to pair their elements off without leaving **any elements of A** uncovered.~~

For set A and B , $\text{card}(A) \leq \text{card}(B)$ if there **is an injection from A to B .**

Comparing Cardinalities

- $f: A \rightarrow B$
 - If f is injective, then $|A| \leq |B|$
 - If f is surjective, then $|A| \geq |B|$
 - If f is bijective, then $|A| = |B|$

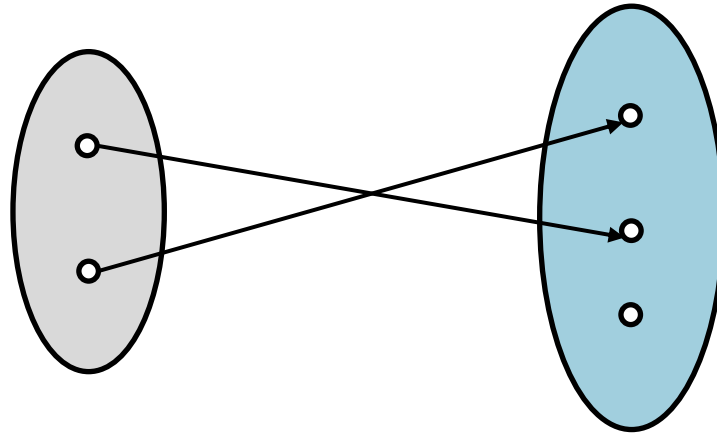
Axiom of Choice

ZF.10 Axiom of choice(选择公理)

For any relation R , there exists a function F , such that F is a subset of R and the domain of F and R are equal.

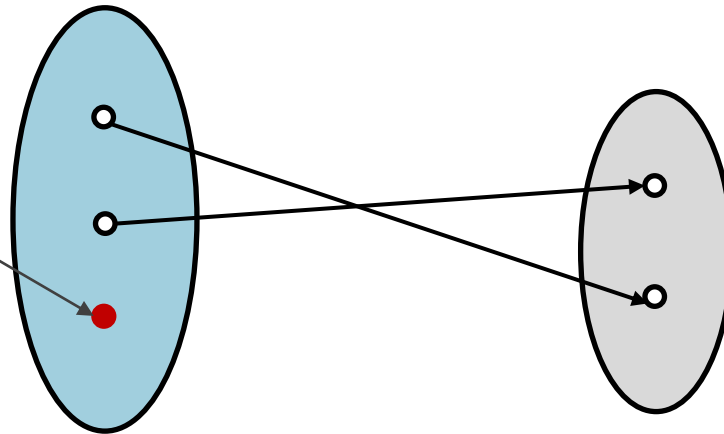
$$(\forall \text{ Relation } R)(\exists \text{ Function } F)(F \subseteq R \wedge \text{dom}(R) = \text{dom}(F))$$

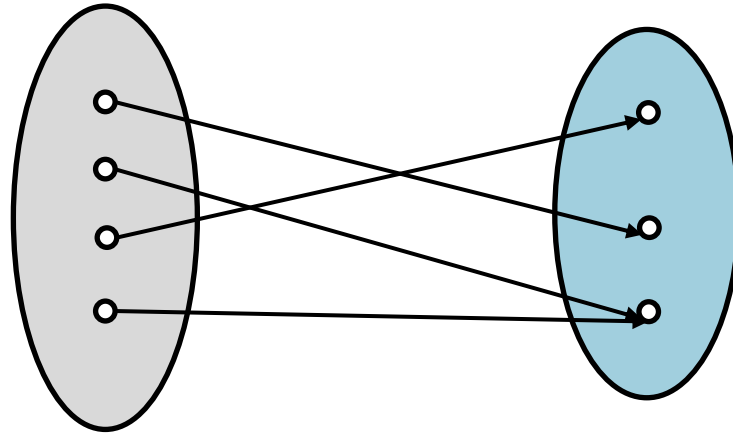
- $R = \{\langle 1, a \rangle, \langle 1, b \rangle, \langle 2, a \rangle\}$
- $f_1 = \{\langle 1, a \rangle, \langle 2, a \rangle\}$
- $f_2 = \{\langle 1, b \rangle, \langle 2, a \rangle\}$



↓ Inverse

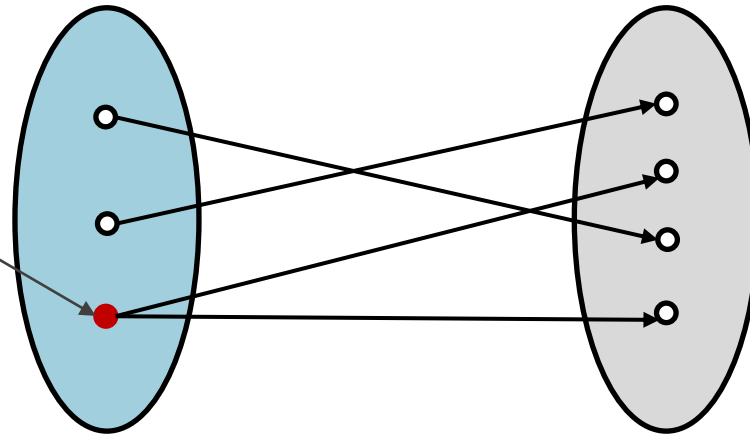
Some element are
not paired





↓ Inverse

Some element are
paired twice



Inverse of Function

- Let $f: A \rightarrow B$
 - The inverse (defined by relation) of f may not be a function
 - If the inverse of f is a function, we call f is **invertible** (可逆的)

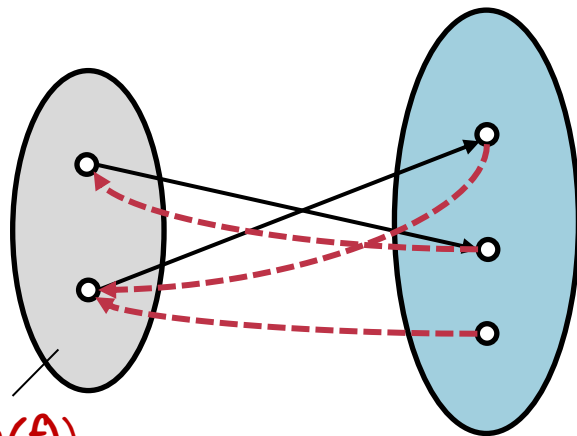
$f: A \rightarrow B$, if there exists $g: B \rightarrow A$, such that $g \circ f = I_A$ and $f \circ g = I_B$, we call that f is **invertible**, and g is the **inverse function**(反函数) of f .

Some times $g \circ f = I_A$ and $f \circ g = I_B$ may not hold simultaneously.

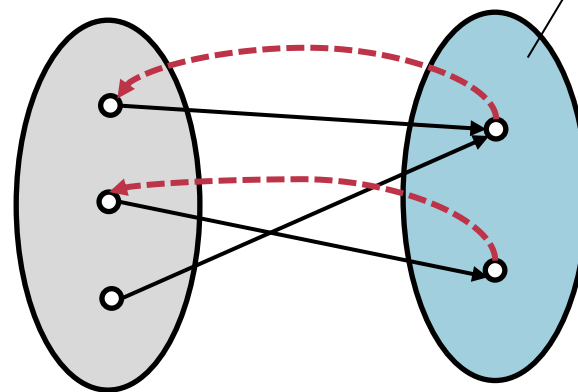
Left Inverse(左逆) and Right Inverse(右逆)

- Let $f: A \rightarrow B$, if there exists $g: B \rightarrow A$, such that
 - $g \circ f = I_A$, then we call g is the **left inverse** of f .
 - $f \circ g = I_B$, then we call g is the **right inverse** of f .

f $\longrightarrow$
 g $\dashrightarrow$



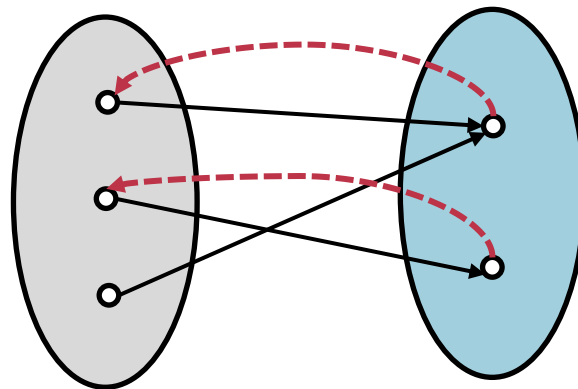
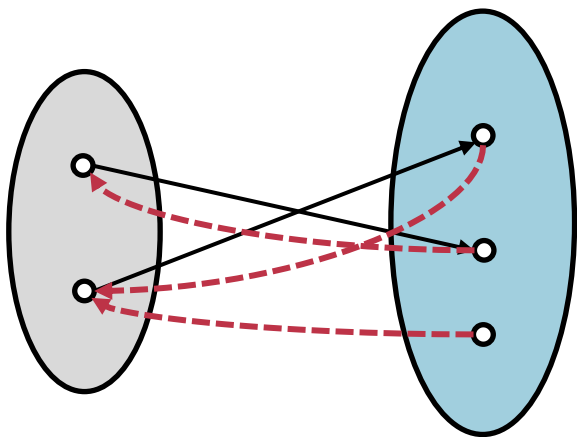
Left Inverse
 $g \circ f = I_A$



Right Inverse
 $f \circ g = I_B$

Left Inverse and Right Inverse

- Let $f: A \rightarrow B$, $A \neq \emptyset$
 - f has left inverse if and only if f is injective.
 - f has right inverse if and only if f is surjective.
 - f has both left and right inverse if and only if f is bijective.
 - f is bijective, then the left and right inverse of f are equal.



Left Inverse and Right Inverse

- Let $f: A \rightarrow B$, $A \neq \emptyset$, assume that g is the left inverse of f , and h is the right inverse of f .
 - We have $g \circ f = I_A$, $f \circ h = I_B$
 - Then $g = g \circ I_B = g \circ (f \circ h) = (g \circ f) \circ h = I_A \circ h = h$
 - So if f has both left and right inverse, then they are equal.

Inverse of Function

- Let $f: A \rightarrow B$
 - The inverse (defined by relation) of f may not be a function
 - If the inverse of f is a function, we call f is **invertible** (可逆的)

$f: A \rightarrow B$, if there exists $g: B \rightarrow A$, such that $g \circ f = I_A$ and $f \circ g = I_B$, we call that f is **invertible**, and g is the **inverse function**(反函数) of f .

- f has both left and right inverse if and only if f is bijective.



f is invertible if and only if f is bijective.

Inverse of Function

- Let $f: A \rightarrow B$ be a bijection
 - f^{-1} is a function and is the inverse function of f .
 - f^{-1} is bijective too.
 - $\forall x \in A, f^{-1}(f(x)) = x. \forall y \in B, f(f^{-1}(y)) = y$



Next

How to model the computation process.

- Introduce to Automata and DFA
- Turing Machine Basics.