Function

Zeyu Mi

2020-12-2

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

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

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

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]):
  M^0 ← M
  for i ← 1 to n:
    for j ← 1 to n:
      if M^(i-1)[j,i]==1:
        for k ← 1 to n:
          M^i[j,k] = M^(i-1)[j,k]∨M^(i-1)[i,k]
  return M^n
```

i=1,j=1

Transitive Closure: Warshall Algorithm

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

$i=1, j=1$

```
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

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

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

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

```
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

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

```
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

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

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

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

```
Warshall_Algorithm(M[1…n,1…n]):
  M^0 ← M
  for i ← 1 to n:
    for j ← 1 to n:
      if M^{i-1}[j,i]==1:
        for k ← 1 to n:
          M^i[j,k] = M^{i-1}[j,k]∨M^{i-1}[i,k]
  return M^n
```

i=2,j=1

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

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

```
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

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

```
Warshall_Algorithm(M[1…n,1…n]):
  M^0 ← M
  for i ← 1 to n:
    for j ← 1 to n:
      if M^{i-1}[j,i]==1:
        for k ← 1 to n:
          M^i[j,k] = M^{i-1}[j,k]∨M^{i-1}[i,k]
  return M^n
```

i=2,j=4

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

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

```
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

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

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

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

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

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

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

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

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
```

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

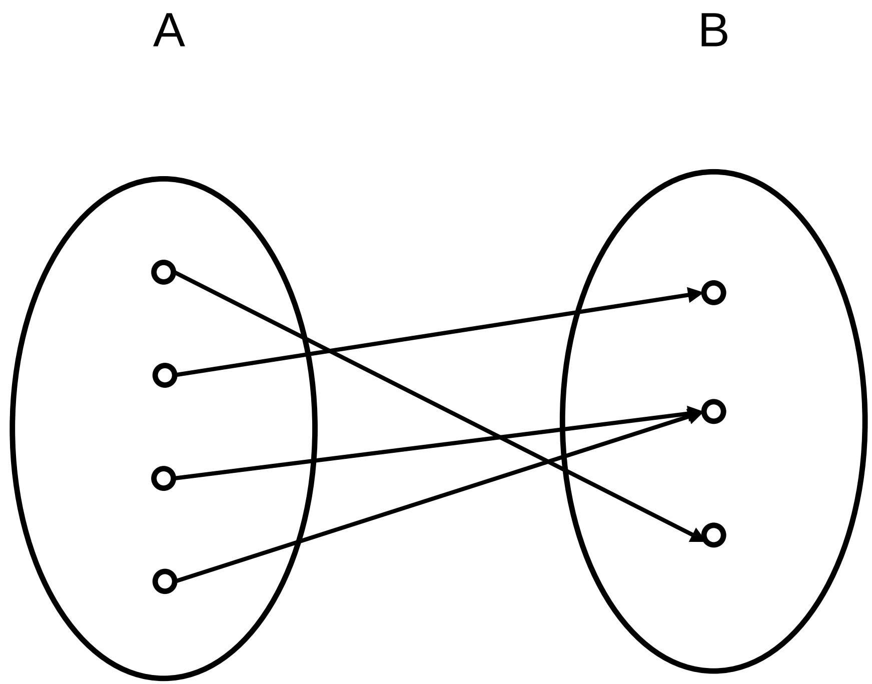
- For a relation $\boldsymbol{R}$ on non-empty set $\boldsymbol{A}$**,** $rs(R)$ means $r(s(R))$, the similar definition for other closure symbols, so we have:
  - $rs(R) = sr(R)$
  - $rt(R) = tr(R)$
  - $st(R) \subseteq ts(R)$

r is commutative with **t** and **s**

s and **t** are not commutative

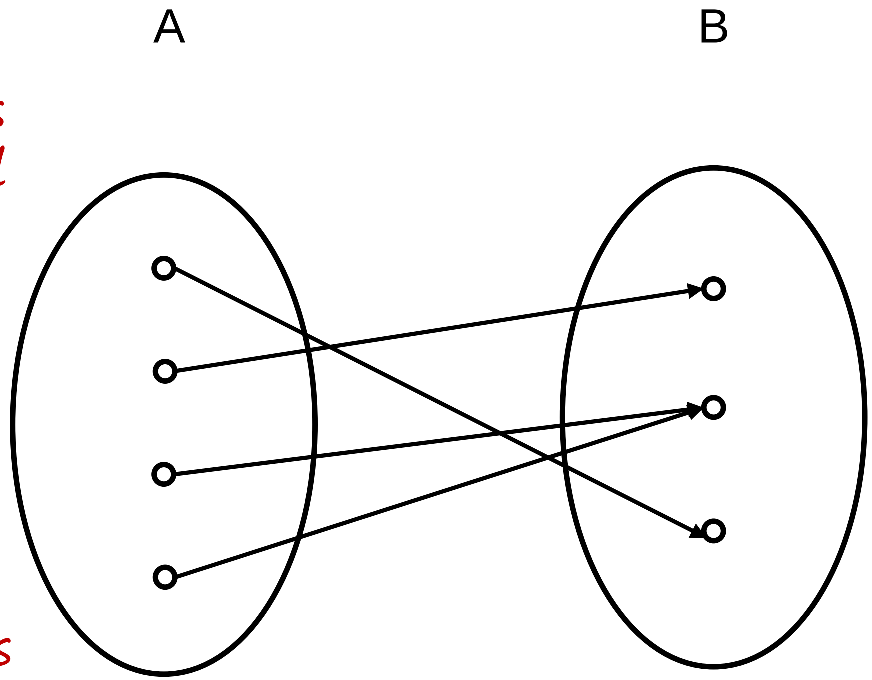
Binary Relation: Recap

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

Function

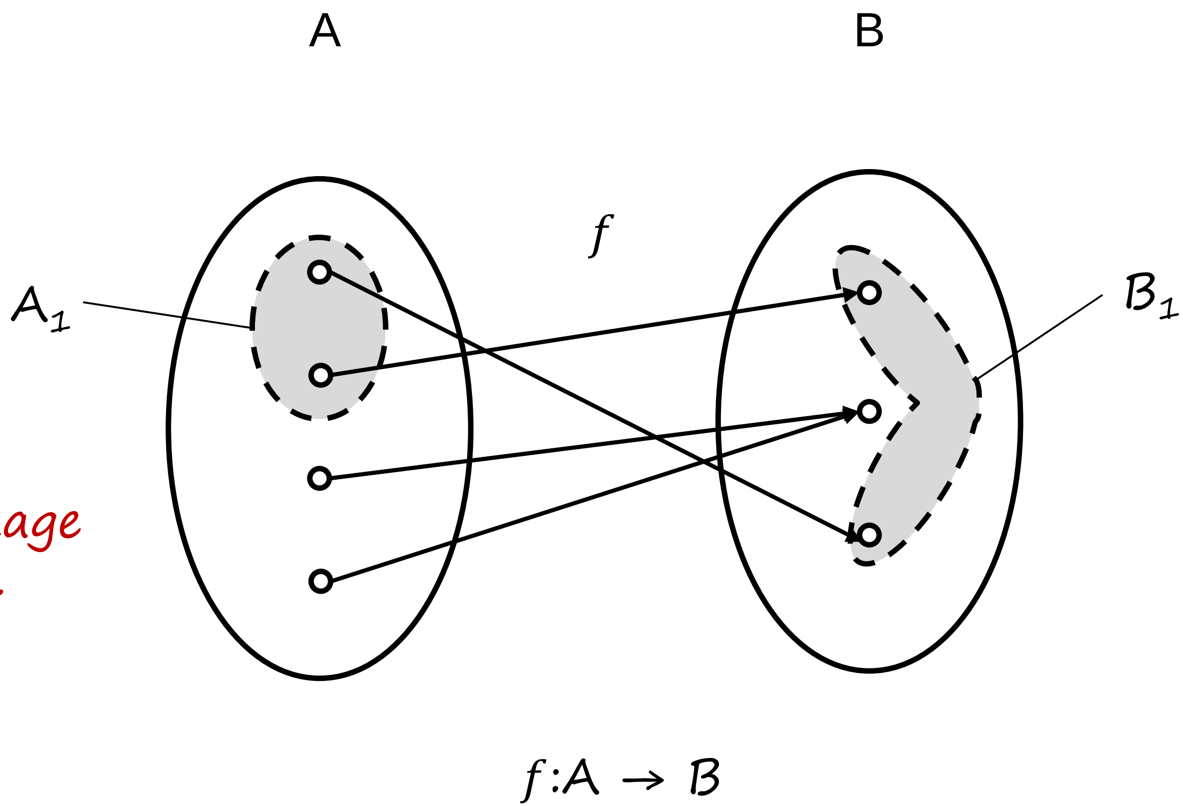
- All the function from set $\boldsymbol{A}$ to set $\boldsymbol{B}$ is $\boldsymbol{A_B}$

$$\boldsymbol{A_B} = \{f \mid f: \boldsymbol{A} \to \boldsymbol{B}\}$$

- If $|A| = n$, $|B| = m$
  - then $|A_B| = m^n$

Function

- For relation $\boldsymbol{R}$ on $\boldsymbol{A}$, for a set $\boldsymbol{B}$
  - the **restriction**(限制) of $\boldsymbol{R}$ on $\boldsymbol{B}$, donate as $\boldsymbol{R} \restriction \boldsymbol{B}$ is:
    $$R \restriction B = \{\langle x, y\rangle | \langle x, y\rangle \in R \land x \in B\}$$
  - The **image** (象) of R on B , donate as $R[B]$ is:
    $$R[B] = \{y | \exists x(\langle x, y\rangle \in R \land x \in B)\}$$
- Similarly for a function $f: \boldsymbol{A} \to \boldsymbol{B}$, $\forall \boldsymbol{A_1} \subseteq \boldsymbol{A}$ 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.



A_1 is the inverse image of B_1 on f.

B_1 is the image of A_1 on f.

Common Functions

- $f: \boldsymbol{A} \rightarrow \boldsymbol{B}$, then $\boldsymbol{f}$ is called **constant function** (常函数) if:
$$(\exists y)(\forall x)(x \in A \rightarrow f(x) = y)$$
- $f: \boldsymbol{A} \rightarrow \boldsymbol{A}$, then $\boldsymbol{f}$ is called **identity function** (恒等函数) if:
$$(\forall x)(x \in A \rightarrow f(x) = x)\ or\ f = I_A$$
- $f: \boldsymbol{A}^{\boldsymbol{n}} \rightarrow \boldsymbol{A}(\boldsymbol{n} \in \mathbb{N})$, then $\boldsymbol{f}$ 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

- $\boldsymbol{F}: \boldsymbol{A} \rightarrow \boldsymbol{B}_{\boldsymbol{C}}$ 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) $\boldsymbol{\chi_A}: \boldsymbol{E} \rightarrow \{\mathbf{0}, \mathbf{1}\}$ is defined by:

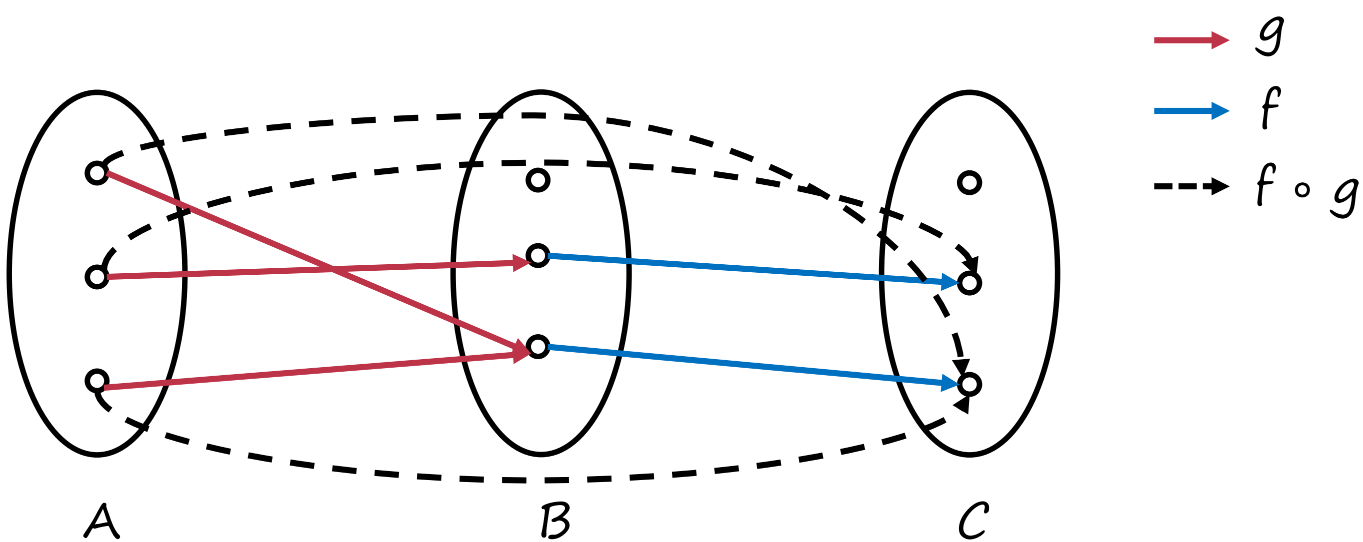
$$\chi_A(x) = \begin{cases} 1, x \in A \\ 0, x \notin A \end{cases}$$

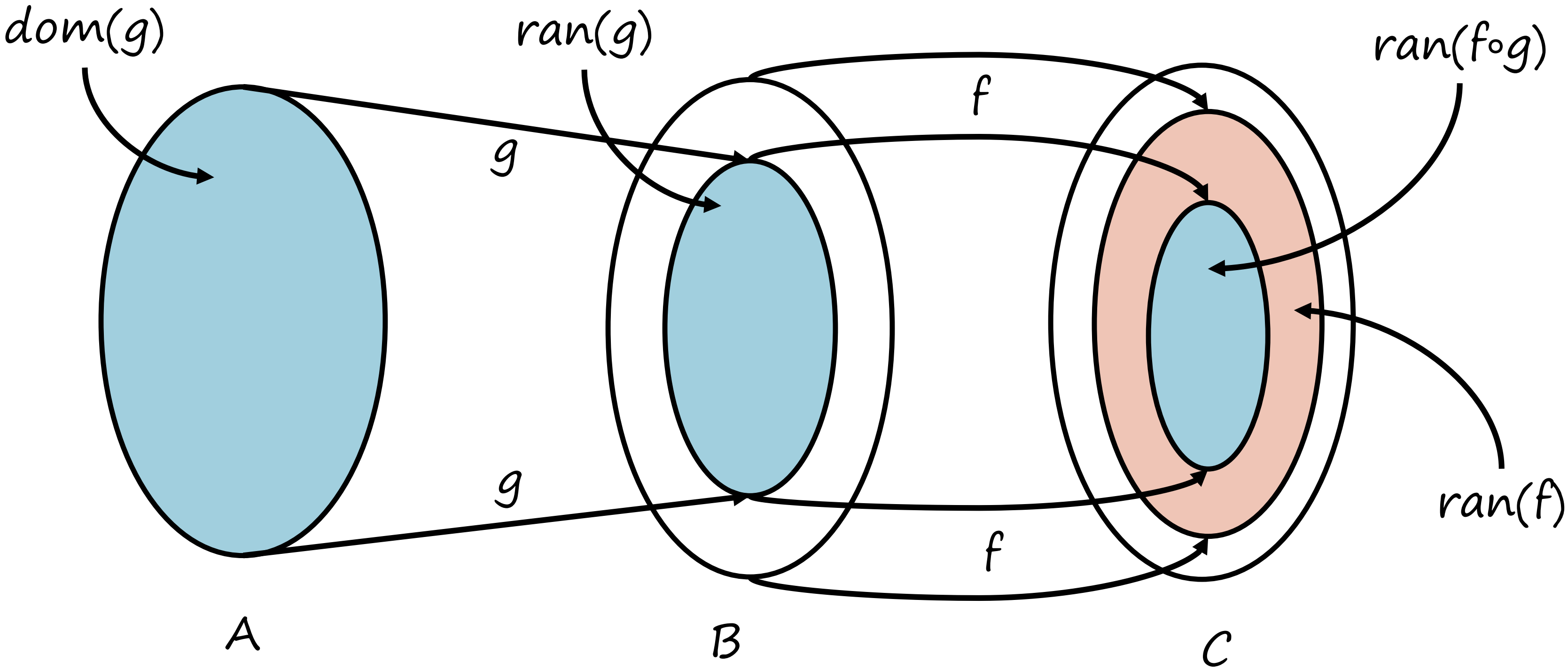
- Let $\boldsymbol{R}$ be an equivalence relation on $\boldsymbol{A}$, $\boldsymbol{g}$ is a **canonical map or natural map(典型映射或自然映射)** from $\boldsymbol{A}$ to quotient set $\boldsymbol{A}/\boldsymbol{R}$ if:

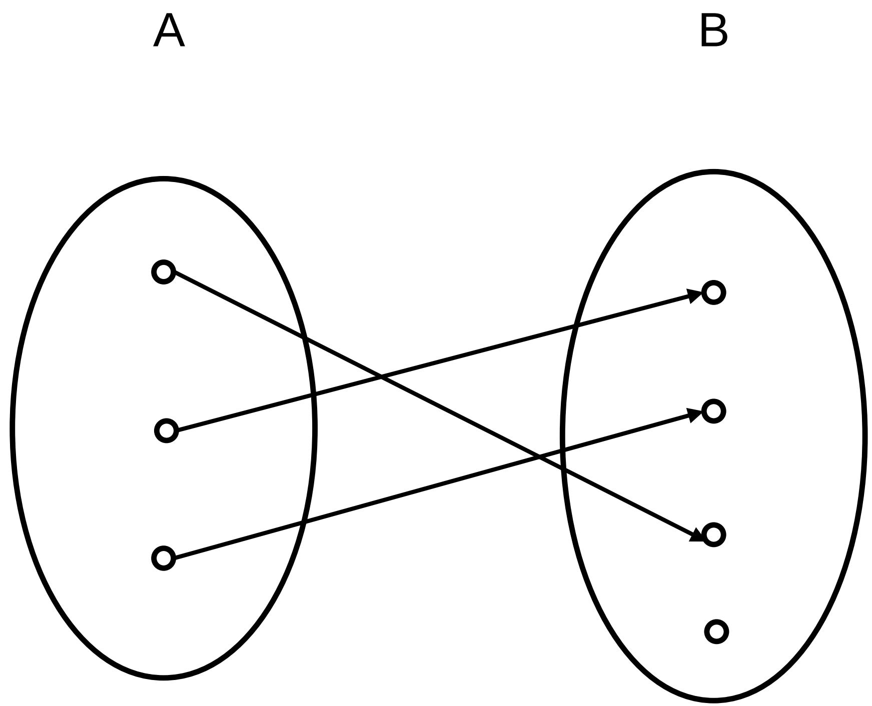
$$\boldsymbol{g}: \boldsymbol{A} \rightarrow \boldsymbol{A}/\boldsymbol{R}, \boldsymbol{g}(\boldsymbol{x}) = [\boldsymbol{x}]_{\boldsymbol{R}}$$

Compose of Function

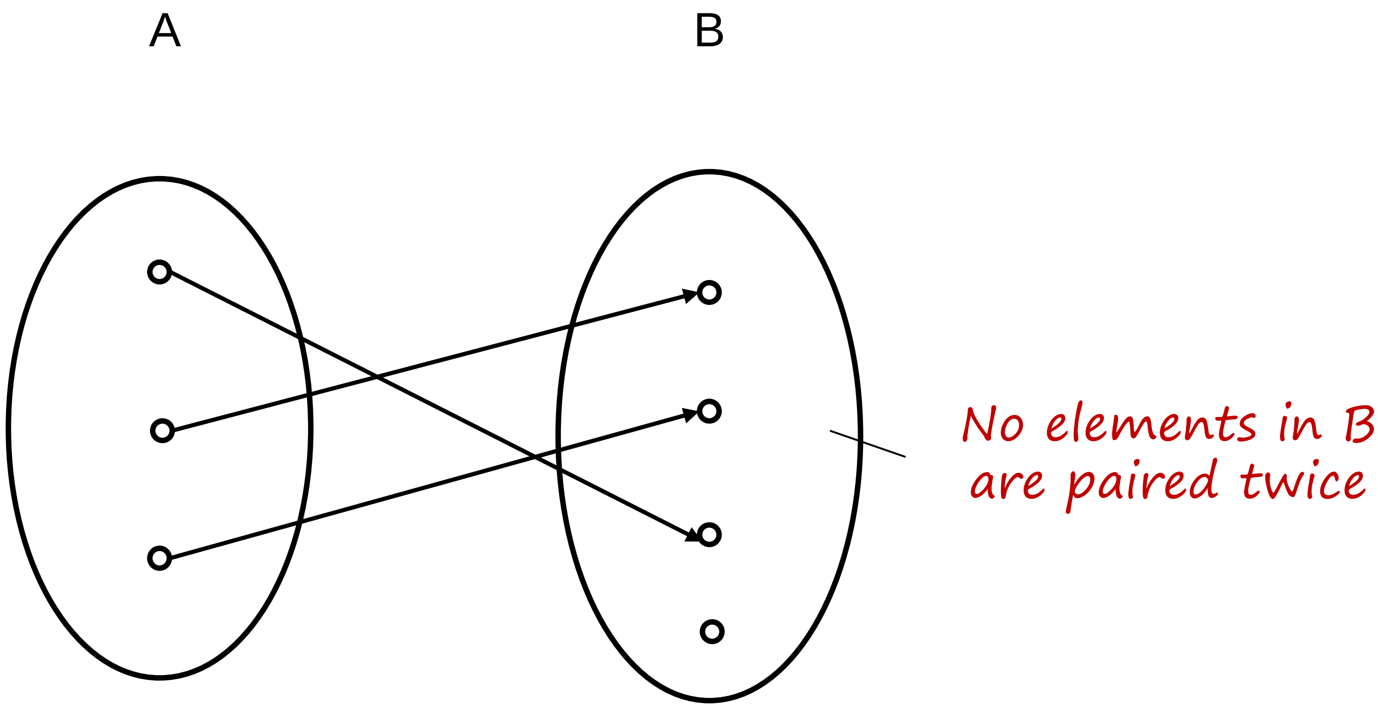
- Let $g: \boldsymbol{A} \to \boldsymbol{B},\ f: \boldsymbol{B} \to \boldsymbol{C}$, then:
  - $f \circ g$ is a function $f \circ g: \boldsymbol{A} \to \boldsymbol{C}$
  - For any $x \in A$, $(f \circ g)(x) = f(g(x))$




dom(g)
ran(g)
ran(f∘g)
g
g
f
f
ran(f)
A
B
C



A special function



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

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

Injection and Composition

Let $g: \boldsymbol{A} \rightarrow \boldsymbol{B},\ f: \boldsymbol{B} \rightarrow \boldsymbol{C}$, 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 $f\big(g(x_1)\big) \neq f(g(x_2))$ 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 $g\colon \boldsymbol{A} \rightarrow \boldsymbol{B},\ f\colon \boldsymbol{B} \rightarrow \boldsymbol{C}$, if $f \circ g$ is injective, then g is also injective.

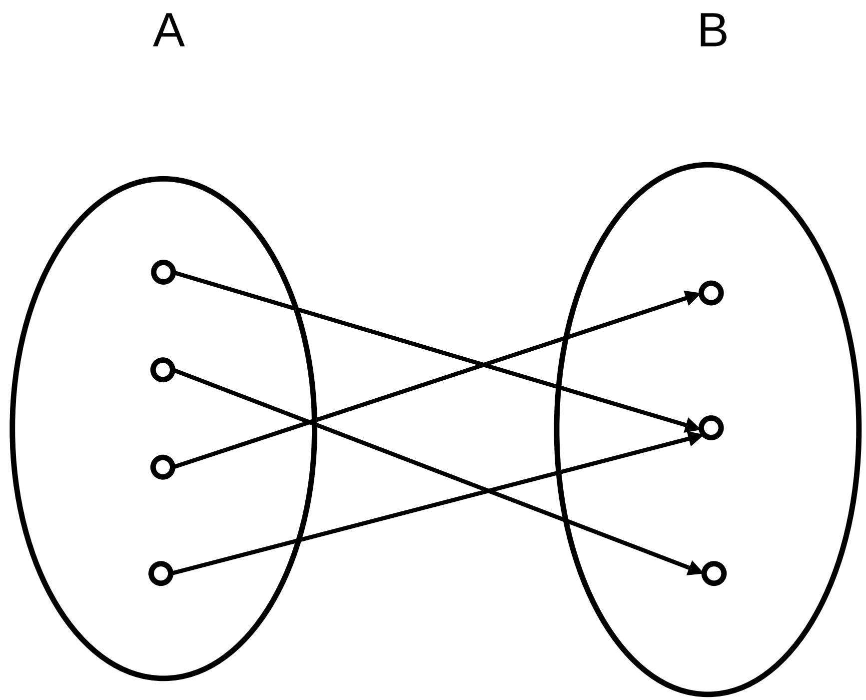
Proof:

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

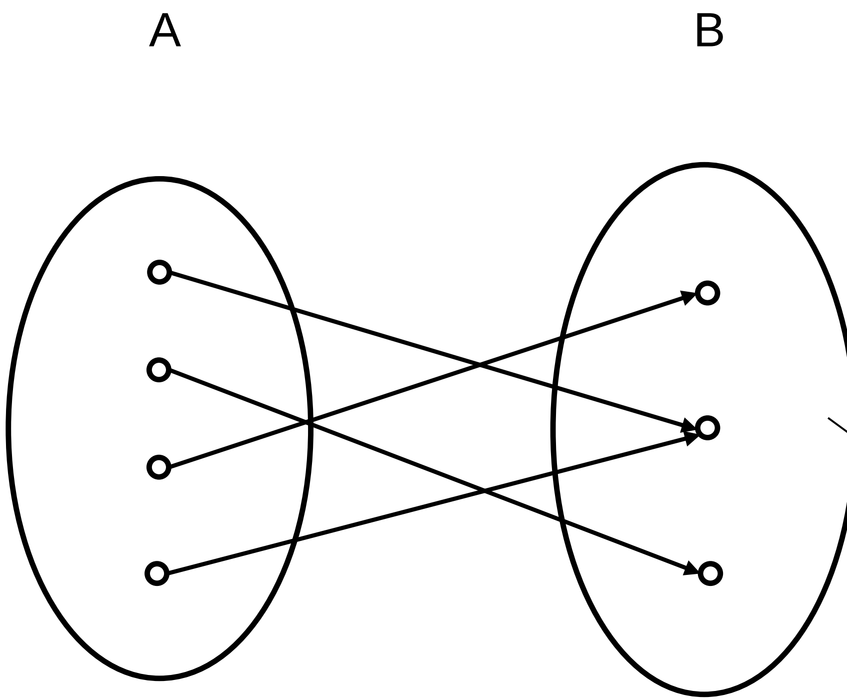
Now $g(x_1), g(x_2) \in B$, and $g(x_1) = g(x_2)$, then we have $f\big(g(x_1)\big) = f\big(g(x_2)\big)$.

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

Q.E.D



A special function



Every element in B are paired at least once.

A special function

Surjection(满射)

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

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

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 $g: \boldsymbol{A} \to \boldsymbol{B},\ f: \boldsymbol{B} \to \boldsymbol{C}$, 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 $f\big(g(x_0)\big) = y_1$, or $(f \circ g)(x) = y_1$, it means that $f \circ g$ is surjective

Q.E.D

Surjection and Composition

Let $g: \boldsymbol{A} \to \boldsymbol{B},\ f: \boldsymbol{B} \to \boldsymbol{C}$, if $f \circ g$ is surjective, then f is also surjective.

Proof:

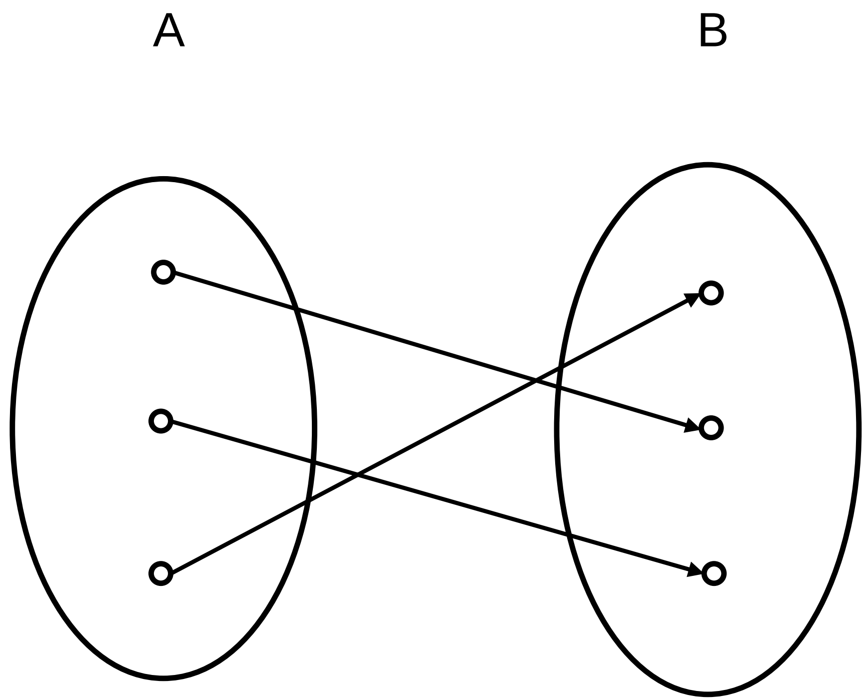
For any $y \in C$, we have $x \in A$ such that $(f \circ g)(x) = y$ as $f \circ g$ is surjective.

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

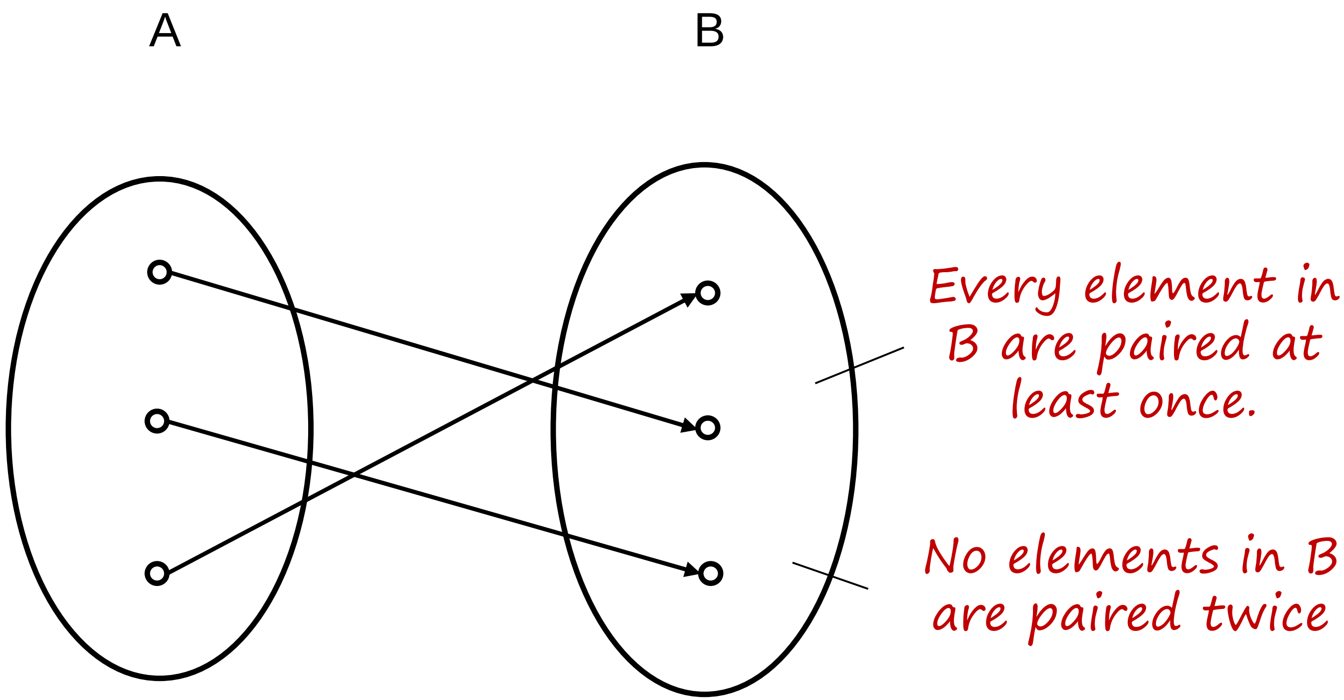
So for any $y \in C$, we have $x_0 \in B$ such that $f(x_0) = y$.

f is surjective.

Q.E.D



A special function



A special function

Bijection(双射)

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

Bijection

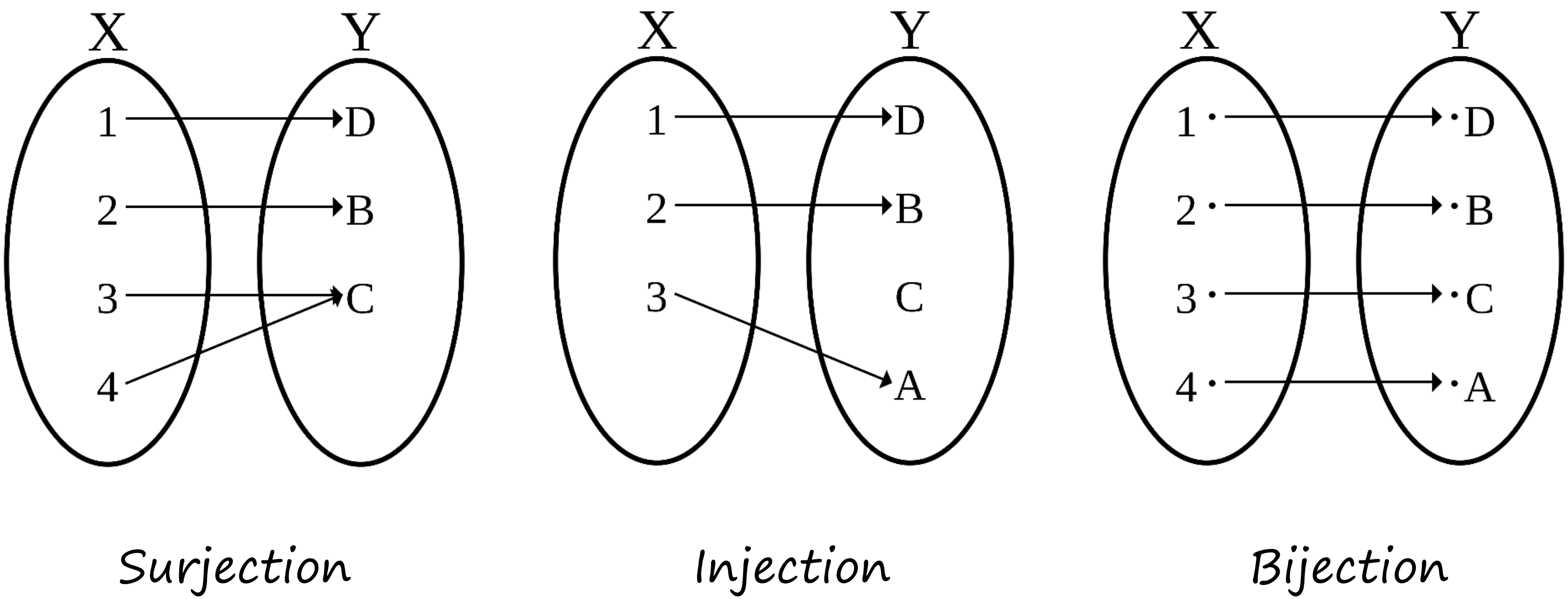
- $f(x) = x + 5$ $(\mathbb{R} \to \mathbb{R})$ is bijective.
- $f(x) = e^x$ $(\mathbb{R} \to \mathbb{R})$ is not bijective since it’s not surjective.
- $f(x) = x^3 - x$ $(\mathbb{R} \to \mathbb{R})$ is not bijective since it’s not injective.

Bijection and Composition

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

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

Bijection, injection and surjection



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 $\boldsymbol{A}$ to $\boldsymbol{B}$, marked as $A \approx B$. Otherwise marked as $\neg A \approx B$.

Comparing Cardinalities

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

Comparing Cardinalities

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

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

Comparing Cardinalities

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

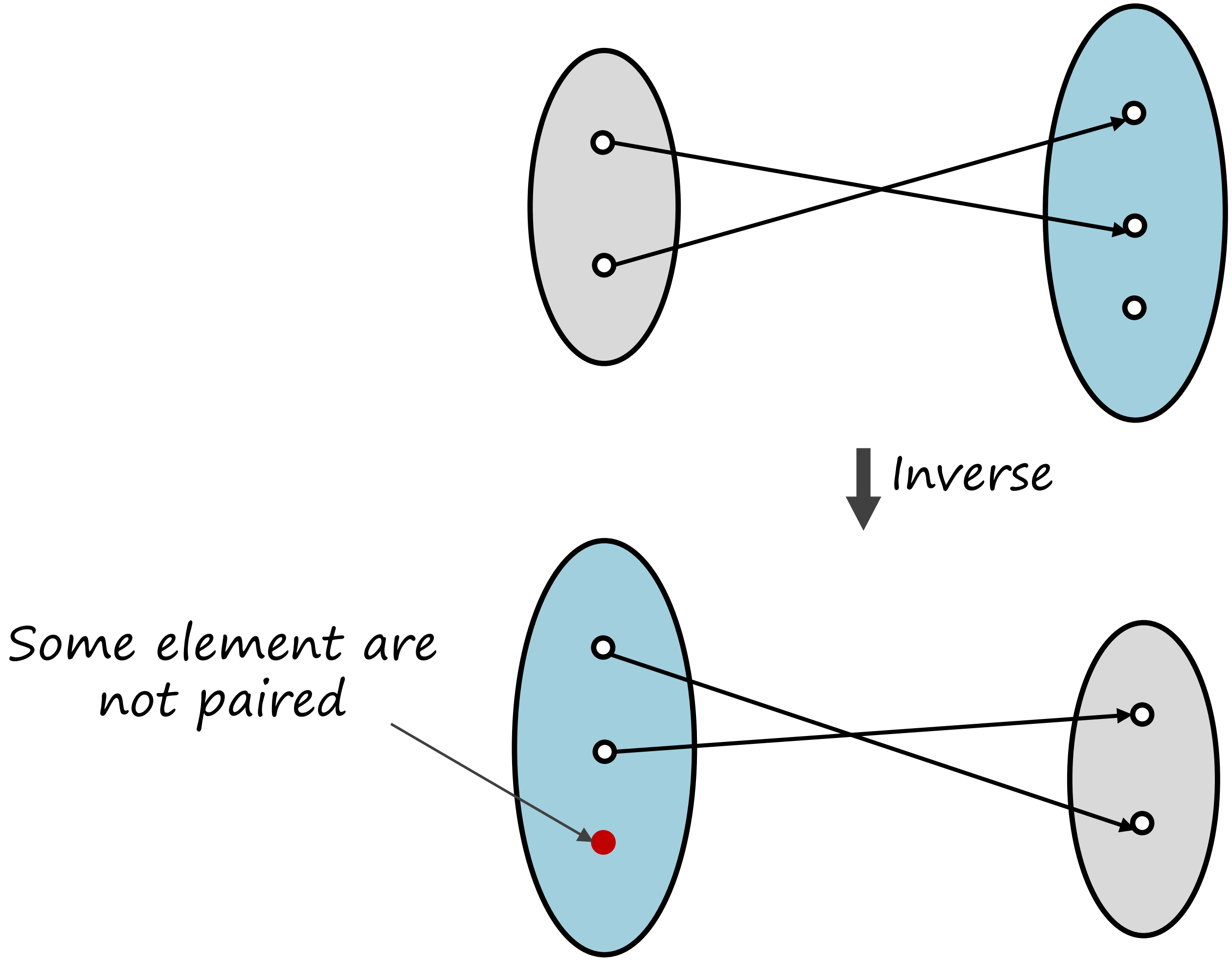
Axiom of Choice

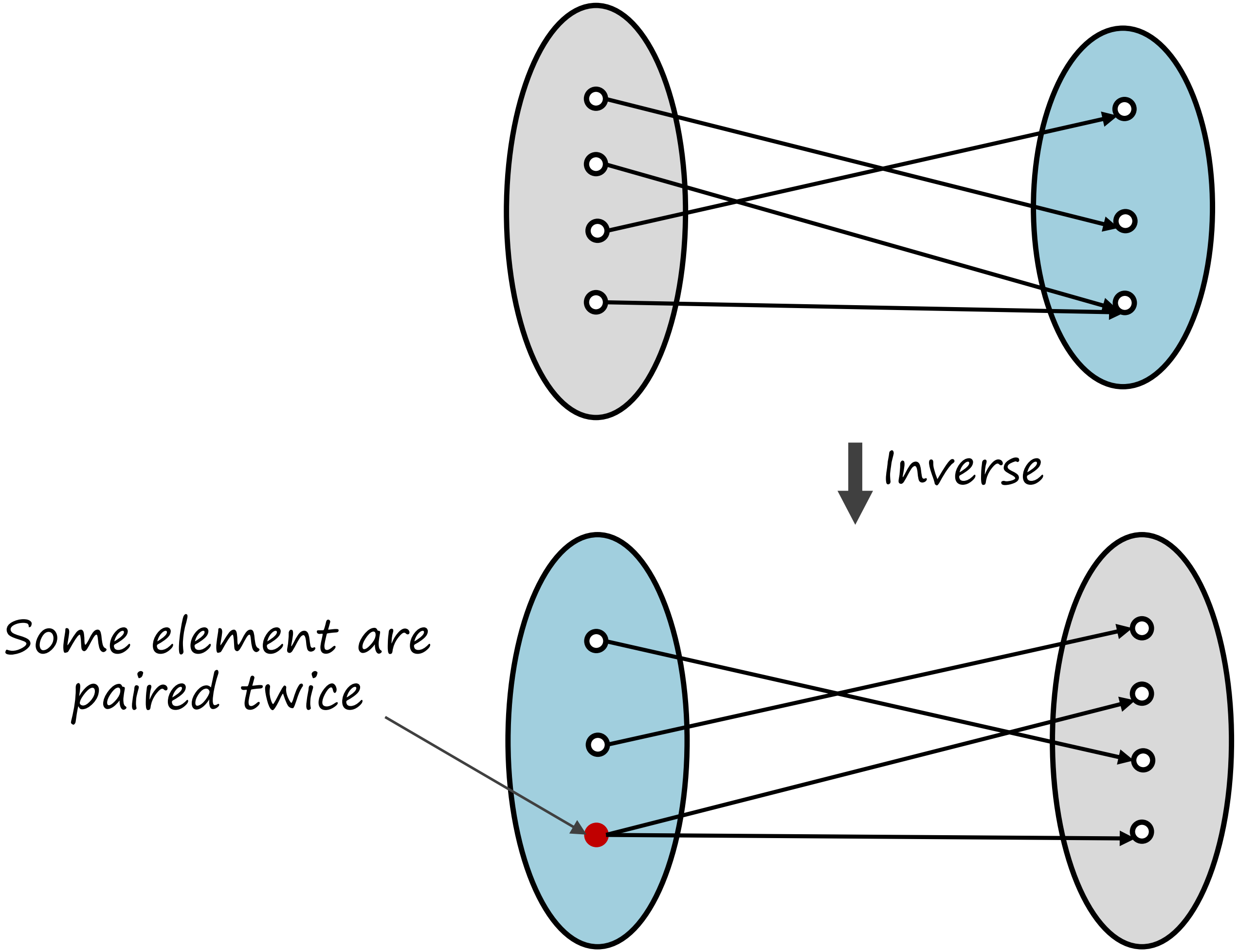
ZF.10 Axiom of choice(选择公理)

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.

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

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


Inverse
Some element are not paired


Inverse
Some element are paired twice

Inverse of Function

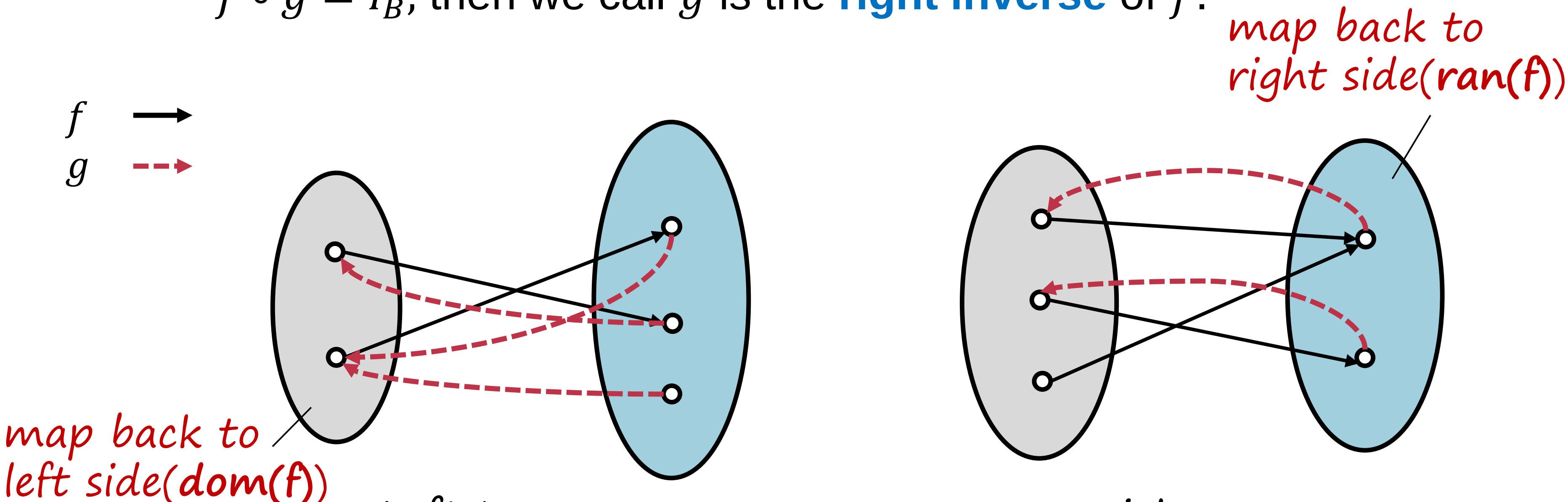
- Let $f: \boldsymbol{A} \to \boldsymbol{B}$
  - The inverse (defined by relation) of f may not be a function
  - If the inverse of f is a function, we call f is **inversible** （可逆的）

$f: A \to B$, if there exists $g: B \to A$, such that $g \circ f = I_A$ and $f \circ g = I_B$, we call that f is **inversible**, 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 $f: \boldsymbol{A} \to \boldsymbol{B}$, if there exists $g: \boldsymbol{B} \to \boldsymbol{A}$, 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.

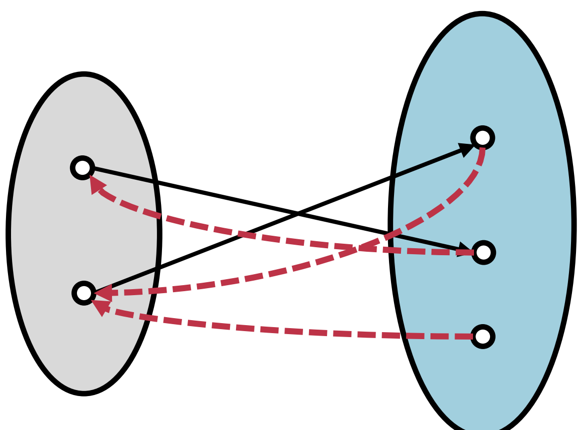


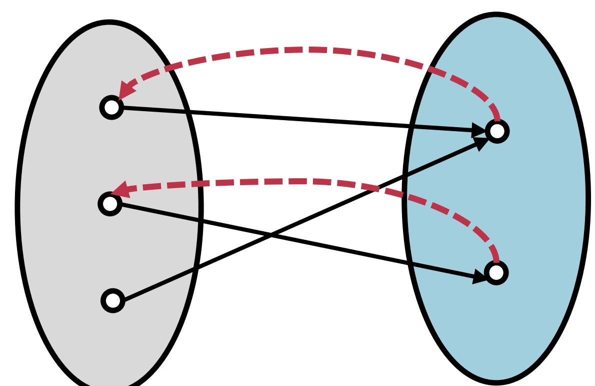
Left Inverse
$g \circ f = I_A$

Right Inverse
$f \circ g = I_B$

Left Inverse and Right Inverse

- Let $f: \boldsymbol{A} \rightarrow \boldsymbol{B}$, $A \neq \emptyset$
  - 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 $f: \boldsymbol{A} \rightarrow \boldsymbol{B}$, $A \neq \emptyset$, 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 $f: \boldsymbol{A} \to \boldsymbol{B}$
  - The inverse (defined by relation) of f may not be a function
  - If the inverse of f is a function, we call f is **inversible**（可逆的）

$f: A \to B$, if there exists $g: B \to A$, such that $g \circ f = I_A$ and $f \circ g = I_B$, we call that f is **inversible**, 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 inversible if and only if f is bijective.

Inverse of Function

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

Next

How to model the computation process.

- Introduce to Automata and DFA
- Turing Machine Basics.