5.1 Function Definition and Properties

同济经管 TONGJI SEM

- 5.1.1 Definition of a Function
- 5.1.2 Image and Preimage of a Function
- 5.1.3 Properties of a Function

- **Definition 5.7**: Let f:$A \to B$ be a function, and let $A_1 \subseteq A$, $B_1 \subseteq B$. Then:
  - $f(A_1)=\{f(x) \mid x \in A_1\}$ is called the **image of A_1 under f.**
  In particular, $f(A)$ is called the **image of the function.**
  - $f^{-1}(B_1)=\{x \mid x \in A \wedge f(x) \in B_1\}$ is called the **complete preimage of B_1 under f.**
- **Note :**
  - A function value $f(x) \in B$ is a point-to-point result, while an image $f(A_1) \subseteq B$ is a set-to-set transformation.
  - $A_1 \subseteq f^{-1}(f(A_1))$:The preimage of the image of A_1 may contain more elements than A_1 itself.
  - $f(f^{-1}(B_1)) \subseteq B_1$:Taking the image of the preimage of B_1 may not recover the original set B_1.
  - A **complete preimage** is the source of a set of outputs; a **preimage** refers to the source of a single output. Both follow the same set operation rules.

■ **Examples:**

(1) Let $f: \mathbf{N} \to \mathbf{N}$, and let $f(x) = \begin{cases} \frac{x}{2} & \text{if } x \text{ even} \\ x+1 & \text{if } x \text{ odd} \end{cases}$

$A=\{0,1\}$, $B=\{2\}$, then

$f(A) = f(\{0,1\}) = \{ f(0), f(1) \} = \{ 0, 2 \}$

$f(B) = \{ f(2) \} = \{ 1 \}$

(2) $A=\{1, 2, 3\}$, $B=\{a, b, c\}$, $f=\{\langle 1,a\rangle,\langle 2,a\rangle,\langle 3,b\rangle\}$, then

- $f^{-1}(\{a,b\}) =\{1,2,3\}$, $f^{-1}(\{b,c\}) =\{3\}$
- $\{1\}\subset\{1,2\}=f^{-1}(\{a\})= f^{-1}(f(\{1\}))$ (Non-injective functions expand preimages)
- $f(f^{-1}(\{b,c\}))=f(\{3\})=\{b\}\subset\{b, c\}$ (Non-surjective ones shrink images)

- 5.1.1 Definition of a Function
- 5.1.2 Image and Preimage of a Function
- **5.1.3 Properties of a Function**
    - Surjective, Injective, and Bijective Functions
    - Constructing a bijective function

- **Definitions of Surjective, Injective, and Bijective Functions**
- **Note:**
  - **Surjectivity** means: for $\forall y \in B$, here exists an $x \in A$ such that $f(x)=y$.
  - **Injectivity** means: if : $f(x_1)=f(x_2) \Rightarrow x_1=x_2$.
  - A surjection ensures full coverage of the codomain, an injection ensures no duplication in mapping, and a bijection guarantees reversibility.
- **Examples:** Determine whether the following functions are injective, surjective, or bijective, and explain why.

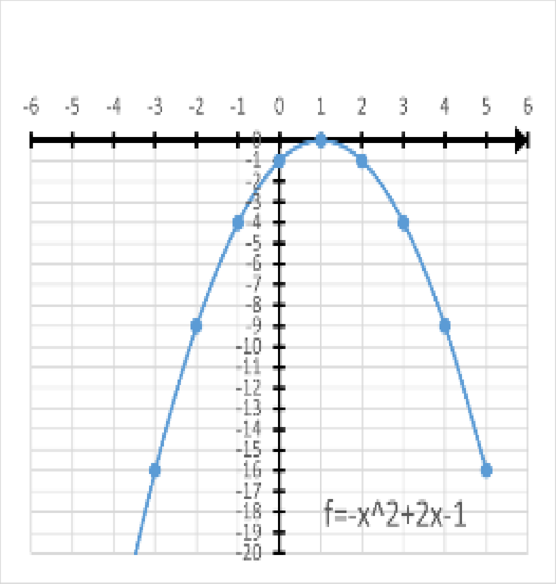
（1） f: $R \rightarrow R, f(x)=-x^2+2x-1$

Solution: When x=0,2 ， $f(x)$=-1， so it is not injective .

■ **Examples:** Determine whether the following functions are injective, surjective, or bijective, and explain why.

（1） f: $\mathbf{R}\to\mathbf{R}$, $f(x)=-x^2+2x-1$

Solution: When x=0,2， $f(x)$=-1， so it is ***not injective*** .

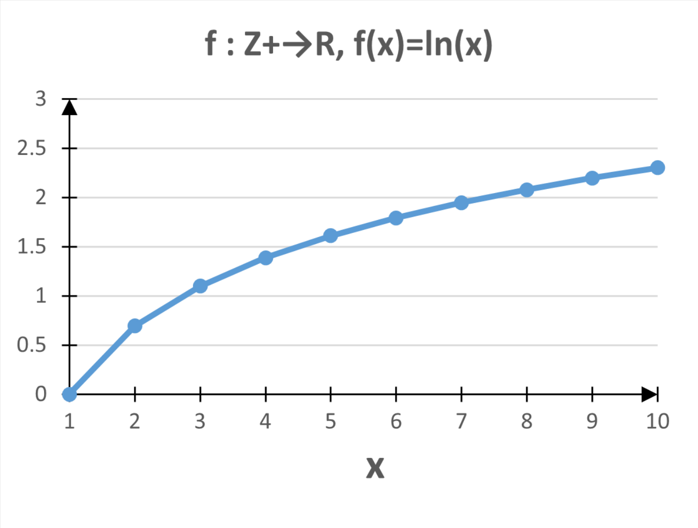


As shown in the figure, the function ***f*** cannot map to any positive real number, and thus it is ***neither surjective nor bijective***.

- **Examples:** Determine whether the following functions are injective, surjective, or bijective, and explain why.

（2） *f*: $Z^+ \rightarrow R$, f(x)=lnx, Z^+ is the set of positive integers.

Solution:



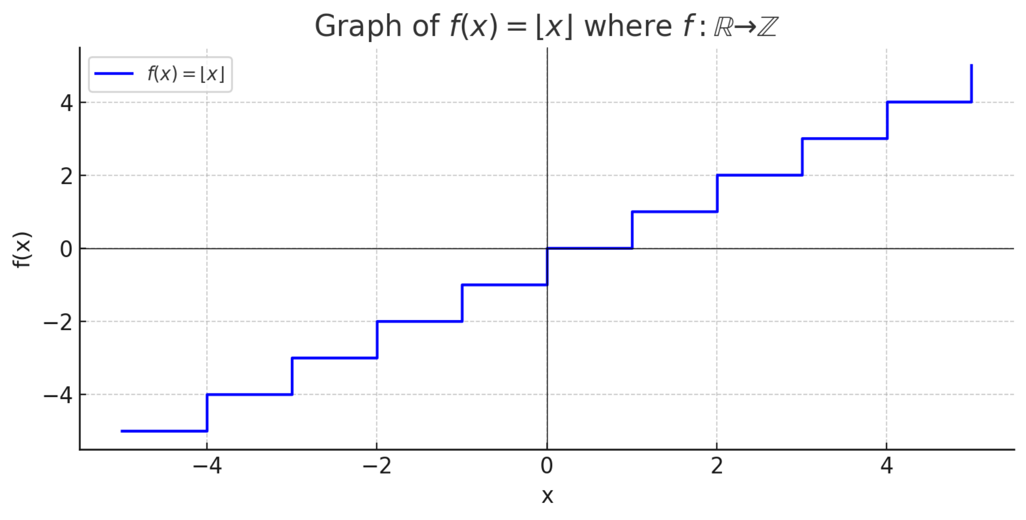
Solution: ***f*(*x*)** is monotonically increasing, so it is ***injective***.

Since **ran*f*={ln1,ln2,…}** cannot cover all values in the real number set **R**, ***f*** is not surjective.

- **Examples:** Determine whether the following functions are injective, surjective, or bijective, and explain why.

(3) $f: R \to Z, f(x) = \lfloor x \rfloor$

Solution:

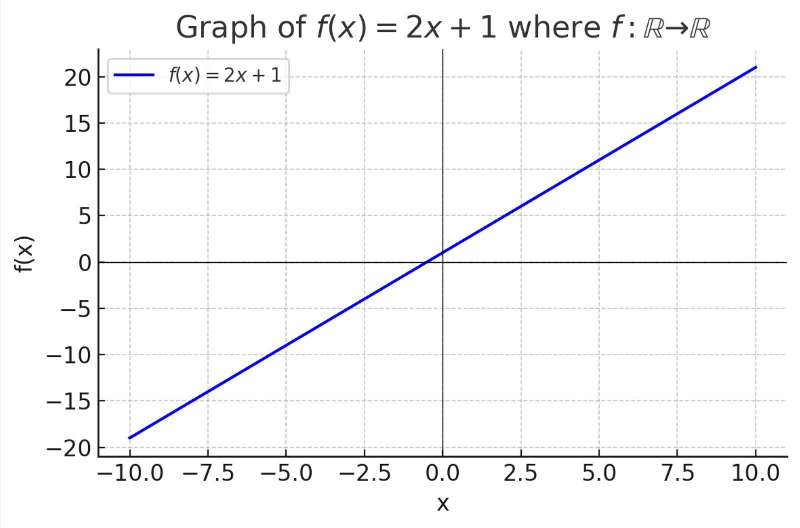


Every integer ***f*(*x*)** has a corresponding real number ***x***, so the function is ***surjective***. However, different real numbers may have the same floor value ***f*(*x*)**, so the function is ***not injective***.

- **Examples:** Determine whether the following functions are injective, surjective, or bijective, and explain why.

（4） ***f*** **: R→R, *f*(*x*)=2*x*+1**

Solution:

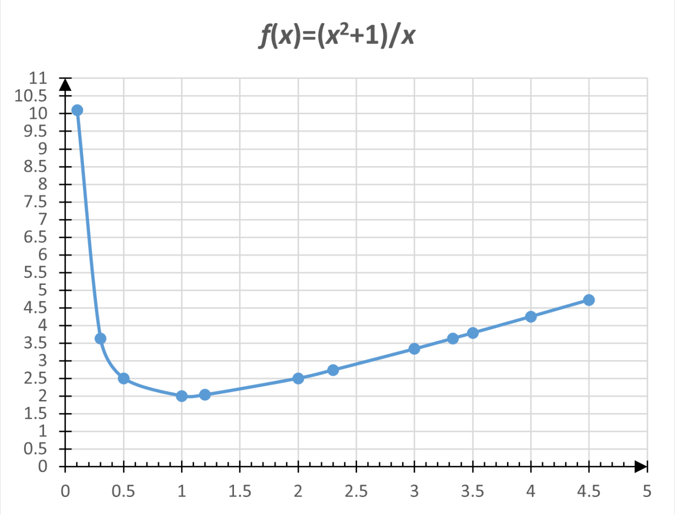


Surjective, ***injective***, ***bijective***, because it is monotonic and the range of **ran*f*=R**

- **Examples:** Determine whether the following functions are injective, surjective, or bijective, and explain why.

（5） $f: R^+ \rightarrow R^+, f(x)=(x^2+1)/x$, **where** R^+ is the set of positive real numbers.

Solution:



The function has a minimum value $f(1)=2$. This function is ***neither injective nor surjective***.

- Example: $A=P(\{1,2,3\})$, $B=\{0,1\}^{\{1,2,3\}}$

Solve: $A=\{\varnothing,\{1\},\{2\},\{3\},\{1,2\},\{1,3\},\{2,3\},\{1,2,3\}\}$.

$B=\{f_0, f_1, \ldots , f_7\}$.

$f_0=\{<1,0>,<2,0>,<3,0>\}$, $f_1=\{<1,0>,<2,0>,<3,1>\}$,
$f_2=\{<1,0>,<2,1>,<3,0>\}$, $f_3=\{<1,0>,<2,1>,<3,1>\}$,
$f_4=\{<1,1>,<2,0>,<3,0>\}$, $f_5=\{<1,1>,<2,0>,<3,1>\}$,
$f_6=\{<1,1>,<2,1>,<3,0>\}$, $f_7=\{<1,1>,<2,1>,<3,1>\}$.

Let $f: A \to B$,

$f(\varnothing)=f_0$, $f(\{1\})=f_1$, $f(\{2\})=f_2$, $f(\{3\})=f_3$,
$f(\{1,2\})=f_4$, $f(\{1,3\})=f_5$, $f(\{2,3\})=f_6$, $f(\{1,2,3\})=f_7$

- Each subset in A is mapped to its characteristic function. For example, $f(\{2\})=f_2$, since only element 2 is in the subset, its characteristic value is (0,1,0).

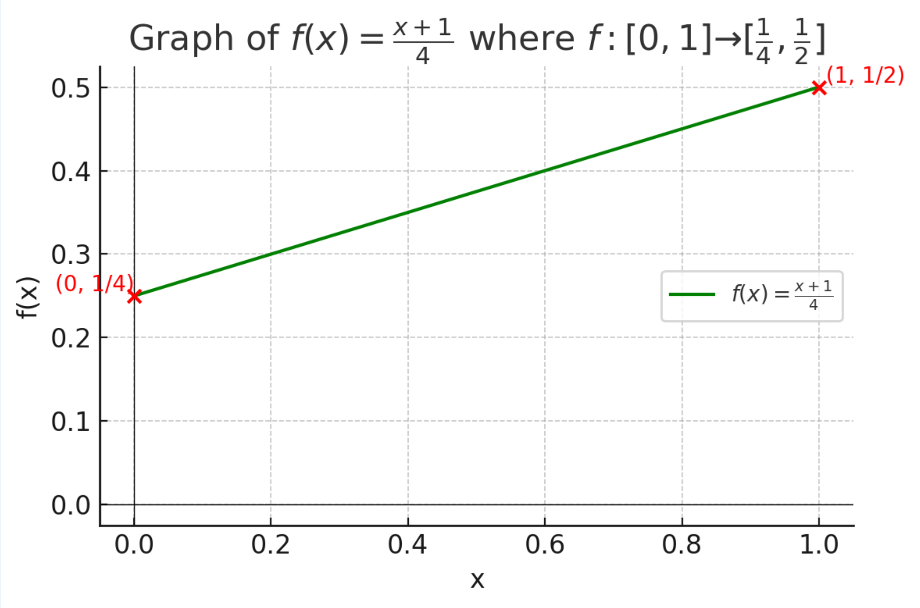
- **Construction Method: Linear Equation**
- **Example: A=[0,1] , B=[1/4,1/2]**

  Construct a bijection $f : A \to B$

Solve: To map A=[0,1] onto B=[1/4,1/2], match the endpoints and use a straight-line function to create a bijection.

Let $f : [0,1] \to [1/4,1/2]$

$$f(x)=(x+1)/4$$



- **Construction Method:** Arrange the elements of set A in a specific order based on a certain criterion. Then, starting from the first element, map them sequentially to the natural numbers.
- **Example:** $A=\mathrm{Z}$, $B=\mathrm{N}$, Construct a bijection $f: A \to B$

Solve : Arrange the elements of Z in the following order and correspond them with the elements of N:

$$
\begin{array}{llllllll}
\mathrm{Z}: & 0 & -1 & 1 & -2 & 2 & -3 & 3 \ldots \\
 & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
\mathrm{N}: & 0 & 1 & 2 & 3 & 4 & 5 & 6 \ldots
\end{array}
$$

The function represented by this correspondence is:

$$
f: \mathrm{Z} \to \mathrm{N}, f(x) = \begin{cases} 2x & x \geq 0 \\ -2x-1 & x < 0 \end{cases}
$$

Objective :

Key Concepts :

Discrete Mathematics 2025 Spring

同济经管
TONGJI SEM

魏可佶 kejiwei@tongji.edu.cn

CAMEA 中国高质量MBA教育认证 AACSB ACCREDITED EQUIS ACCREDITED

- 5.1 Function Definition and Properties
- 5.2 Composition of Functions and Inverse Functions
- 5.3 Relational Algebra

5.2.1 Composition of Functions

- The fundamental theorem of function composition and its corollaries
- Properties of function composition

5.2.2 Inverse Functions

- Conditions for the existence of an inverse function
- Properties of inverse function

↳ 5.2.1 Composition of Functions

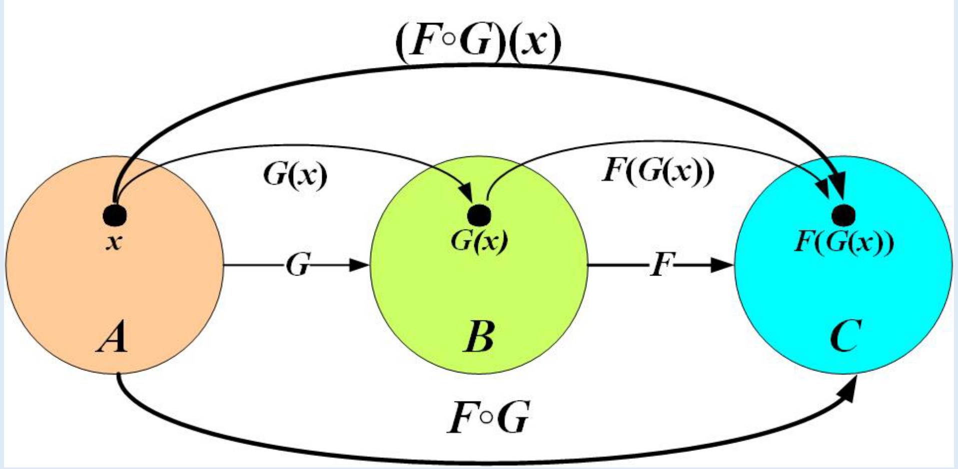
- **Theorem 5.1**: Let F,G be functions, **then the composition** $F \circ G$ is also a function and satisfies the following conditions:

  (1) **dom($F \circ G$)={ x | $x \in$ dom$G \wedge G(x) \in$ dom$F$}**

  (This describes the relationship between the domains and ranges of the functions.)

  (2) $\forall x \in$ **dom($F \circ G$)** , $F \circ G(x) = F(G(x))$

  (This specifies the order of computation for the composite function.)



Note: The composition of functions is **left composition** with **right-hand priority**, while the composition of relations is **right composition** with **left-hand priority**.

↳ Function Composition and Mapping Properties

- **Theorem 5.2:** Let $f: B \to C$, $g: A \to B$.

  (1) If f, g are surjective, then $f \circ g: A \to C$ is also ***surjective***.

  (2) If both f, g are injective, then the composition $f \circ g$: $A \to C$ is also ***injective***.

  (3) If both f, g are bijective, then the composition $f \circ g$: $A \to C$ is also ***bijective***.

↳ Function Composition and Mapping Properties

■ **Theorem 5.2:** Let $f : B \to C$, $g : A \to B$.

(1) If f , g are surjective, then $f \circ g : A \to C$ is also ***surjective***.

proof : **Goal:** Prove that for any $c \in C$, there exists at least one element $a \in A$ such that $(f \circ g)(a)=c$.

①Since f is surjective, for any $c \in C$, there exists some $b \in B$ such that $f(b)=c$.

②Since g is also surjective, there exists some $a \in A$ such that $g(a)=b$.

③Given that $g(a)=b$ and $f(b)=c$, by the definition of function composition, we have: $(f \circ g)(a)= f(g(a))=f(b)=c$.

Therefore, $f \circ g : A \to C$ is ***surjective***.

■ **Theorem 5.2:** Let $f : B \to C$, $g : A \to B$.

(2) If both f , g are injective, then the composition $f \circ g : A \to C$ is also ***injective***.

proof : **Goal**: We need to prove that if $x_1, x_2 \in A$, $x_1 \neq x_2$, *then* $f \circ g(x_1) \neq f \circ g(x_2)$.

①Since g injective, $g(x_1) \neq g(x_2)$, and $g(x_1)$, $g(x_2) \in B = dom f$。

②*Since* f injective, we know that $f(g(x_1)) \neq f(g(x_2))$, thus $f \circ g(x_1) \neq f \circ g(x_2)$.

so $f \circ g : A \to C$ *is injective*.

- 5.2.1 Composition of Functions
  - The fundamental theorem of function composition and its corollaries
  - Properties of function composition
- 5.2.2 Inverse Functions
  - Conditions for the existence of an inverse function
  - Properties of inverse function

- **Theorem 5.4: Let $f : A \rightarrow B$ *be bijective*, so $f^{-1}: B \rightarrow A$ *is also bijective*.**

Proof:

①Since f is a function, so f^{-1} is relation, and we have **domf^{-1}= ran$f$ =B** , **ranf^{-1}= dom$f$ =A.**

②For any $x \in B$, suppose have $y_1, y_2 \in A$ such that $<y_1,x> \in f^{-1} \wedge <y_2, x> \in f^{-1}$, then by the definition of inverse $<x,y_1> \in f \wedge <x,y_2> \in f$. Since f is injective, it follows that $y_1=y_2$, **hence** f^{-1} ***is a well-defined function***.

③Also, for every $x \in B$, there is a unique $a \in A$ such that $f(a)=x$, so $f^{-1}(x)=a$, therefore, f^{-1} ***is surjective***.

④Now, suppose exist $x_1, x_2 \in B$ such that $f^{-1}(x_1) = f^{-1}(x_2)=y$, then we have

$<y, x_1> \in f^{-1} \wedge <y, x_2> \in f^{-1} \Rightarrow <x_1,y> \in f \wedge <x_2,y> \in f \Rightarrow x_1=x_2$

(Since f is injective *function*),then proves that f^{-1} **is *injective*** .

Conclusion: Since f^{-1} is both **injective** and **surjective**, f^{-1} :B→A is a ***bijective function***.

- Example: Let f : R→R, g : R→R, $f(x)=\begin{cases} x^2 & x\geq 3 \\ -2 & x<3 \end{cases}$, $g(x)=x+2$
  - Find $f\circ g$, $g\circ f$.
  - If f and g have inverse functions, find their inverses.

Solve :

$g\circ f$: $\mathbf{R}\to\mathbf{R}$ $f\circ g$: $\mathbf{R}\to\mathbf{R}$

$$g\circ f(x)=\begin{cases} x^2+2 & x\geq 3 \\ 0 & x<3 \end{cases} \qquad f\circ g(x)=\begin{cases} (x+2)^2 & x\geq 1 \\ -2 & x<1 \end{cases}$$

f : R→R does not have an inverse function .

g : R→R inverse function is g^{-1}: R→R, $g^{-1}(x)=x-2$.

- **The composition of a bijective function and its inverse is the identity function.**
- **Theorem 5.5: Let $f : A \to B$** be a **bijective** function, then:

$$f^{-1} \circ f = I_A, \quad f \circ f^{-1} = I_B$$

Proof:

① According to Theorem **5.4** $f^{-1}: B \to A$ is also bijective.

② By the theorem on function composition, $f^{-1} \circ f : A \to A$, $f \circ f^{-1}: B \to B$, and both equal the corresponding **identity functions.**

③ By the above Therefore, we have: $f^{-1} \circ f = I_A$, $f \circ f^{-1} = I_B$.

- The left and right compositions of a bijective function and its inverse are identity functions on their respective sets, and I_A is generally not equal to I_B .

Discrete Mathematics 2025 Spring

魏可佶　kejiwei@tongji.edu.cn

- 5.1 Function Definition and Properties
- 5.2 Composition of Functions and Inverse Functions
- 5.3 Relational Algebra

5.3 Relational Algebra

- **5.3.1 Relational Algebra and Its Components**
  - Algebra, Algebraic Systems, and Branches of Algebra
  - Relational Algebra and Relational Database Operations
  - Relational Model and Relational Database
- **5.3.2** Relational Algebra and Its Operations
- **5.3.3** Applications of Relational Algebra

↳ Algebra, Algebraic Systems, and Branches of Algebra

- ***Algebra*** is the science of studying operational structures.
- ***Algebraic systems*** are the stage for these operations.
- ***Algebraic branches*** are the specific forms performed on that stage.

Algebra Branch	Algebraic System	Description	Applications
Relational Algebra	(R,σ,π,∪,∩,×,...)	Operations on relations (tables) in databases	SQL, database query languages
Boolean Algebra	(B,∨,∧,¬)	Logic system over two values (true/false or sets)	Logic design, digital circuits
Linear Algebra	(V,+,·)	Vector spaces over a field with scalar multiplication	Physics, machine learning, data science
Group Theory(G,·), Ring Theory(R,+,·), Field Theory(F,+,·), Automata Algebra State-transition structures,....			

↳ Relational Algebra and Relational Database Operations

- Relational algebra is an ***abstract query language*** used to operate on relations (tables) in a relational database.
- Relational algebra provides the ***theoretical foundation*** for query languages such as SQL.
- Relational algebra allows for ***query optimization*** and equivalence transformations.
- ***Relational algebra operates*** on relations (tables) using a set of operators—including selection, projection, union, and intersection—and produces new relations as output.

- The ***relational model***, proposed by E.F. Codd, is a data model that represents real-world entities and their relationships using relations (tables). Each relation consists of rows (tuples) and columns (attributes).
- The relational model is the theoretical foundation for building ***relational databases***. It represents data and relationships using relations (tables).
- ***Tables and Relations***: Each table in a database represents a **relation**. Each row in the table (also called a **tuple**) represents an **ordered tuple**, which is a member of the relation.
- ***Relationships Between Elements***: Each row in a table represents a specific **association or "relation"** between columns (also called **attributes**), forming a meaningful set of data.

5.3 Relational Algebra

- 5.3.1 Relational Algebra and Its Components
- 5.3.2 Relational Algebra Operations
  - Basic and Compound Operations in Relational Algebra
  - Algebraic Operations of Relational Databases
  - Relational Algebra Operations and SQL
- 5.3.3 Business Process Modeling and Its Tools (Languages)

↳ Basic and Compound Operations in Relational Algebra

- ***Relational algebra operations*** are a subset of relational operations. They form a formal, algebraic system for constructing and optimizing queries.
- Relational algebra operations are a set of ***closed operations***, where both the input and the result are relations (**tables**). They form the theoretical foundation of the SQL query language.
- Relational algebra consists of ***basic and derived operations***. Derived operations are constructed by combining basic ones.
- The ***basic operations include*** Selection, Projection, Union, Difference, Cartesian Product, and Rename.
- The ***Derived operations include*** Join, Natural Join, Theta Join, and Division.

- **Selection(σ)**: Selects all tuples from a relation that satisfy a given condition.
- **Projection(π)**: Creates a new relation by selecting specific columns from an existing relation.
- **Union(∪)**: Combines two relations with the same attributes into a single relation, denoted as **R ∪ S**。
- **Set Difference(-)**: Removes the elements of one relation from another, denoted as **R - S**。
- **Cartesian Product(×)**: Forms all possible combinations of tuples from two relations, denoted as **R × S**。

- **Rename**(ρ): Changes the names of attributes in a relation.
- **Natural Join**(⋈): Merges two relations based on their common attributes, denoted as **R ⋈ S**。
- **Intersection**(∩) ： **R ∩ S** is not a basic operation in relational algebra, but it can be constructed using union and set difference: **R ∩ S = （ R ∪ S） - （R - S） - （S - R） .**
- Structured Query Language (**SQL**) is a **practical implementation of relational algebra** and relational calculus, enabling operations such as intersection, querying, insertion, updating, and deletion in relational databases.

- **Data Tables in Databases as Relations**

  A relation is a ***set of tuples***, where each tuple $<A_1,A_2,\ldots,A_n>$ represents a relation with ***n*** attributes.

- **Example:** ⟨ID, Name, Age, Address, Phone, Email⟩

  Let ***R and S*** be ***m-ary*** relations with the same attributes, where the m attributes are denoted as $A_1,A_2,\ldots,A_m$, The basic operations are as follows:

  $R \cup S$ contains tuples from both ***R*** and ***S***.

  $R \cap S$ contains tuples that are present in both ***R and S***;

  $R - S$ contains tuples that are in ***R but not in S.***

  Projection $\pi_{A_{i_1},A_{i_2},\ldots,A_{i_n}}(R)$ Select only certain columns $A_{i_1},A_{i_2},\ldots,A_{i_n}$ from ***R*** form a new relation.

■ Example: *R*:Employee Information

Name	Age	Address	Phone	Email
Zhang Wei	28	Zhongguancun, Beijing	13812345678	zhangwei@exa.com
Li Ting	32	Pudong, Shanghai	13987654321	liting@exa.com
Wang Qiang	45	Tianhe, Guangzhou	13722223333	wangqiang@exa.com
Zhao Xiaolin	26	Jinjiang, Chengdu	13611114444	zhaoxiaolin@exa.com

$\pi_{\text{Name,Phone,Email}}(R)$ The query result is selecting the columns **Name**, **Phone**, and **Email**.

Name	Phone	Email
Zhang Wei	13812345678	zhangwei@exa.com
Li Ting	13987654321	liting@exa.com
Wang Qiang	13722223333	wangqiang@exa.com
Zhao Xiaolin	13611114444	zhaoxiaolin@exa.com

- **Example:** ⟨ID, Name, Age, Address, Phone, Email⟩

  Let ***R and S*** be ***m-ary*** relations with the same attributes.

  $R \cup S$:

  SELECT * FROM ***R*** UNION SELECT * FROM ***S***;

  $R \cap S$:

  SELECT * FROM ***R*** INTERSECT SELECT * FROM ***S***;

  $R - S$:

  SELECT * FROM ***R*** EXCEPT SELECT * FROM ***S***;

  $\pi_{Name,Phone,Email}(R)$:

  SELECT Name, Phone, Email FROM ***R***;

5.3.2 Applications of Relational Algebra

↳ Algebraic Operations of Relational Databases(e.g.)

- The **Cartesian Product** $R \times S$ is a set consisting of $m \times n$ tuples of the form $<A_1,\ldots,A_m,B_1,\ldots,B_n>$ where the tuples have $m+n$ attributes.
- **Example:** *R*={<1,abc>,<2,cabel>},

  S={<cabel,300,25>,<sin,190,15>,<cod,60,5>},

  R×*S*={<1,abc,cabel,300,25>,<1,abc,sin,190,15>,
  <1,abc,cod,60,5>,<2,cabel,cabel,300,25>,
  <2,cabel,sin,190,15>,<2,cabel,cod,60,5>}

R×*S* : SELECT * FROM R CROSS JOIN *S*;

5.3 Relational Algebra

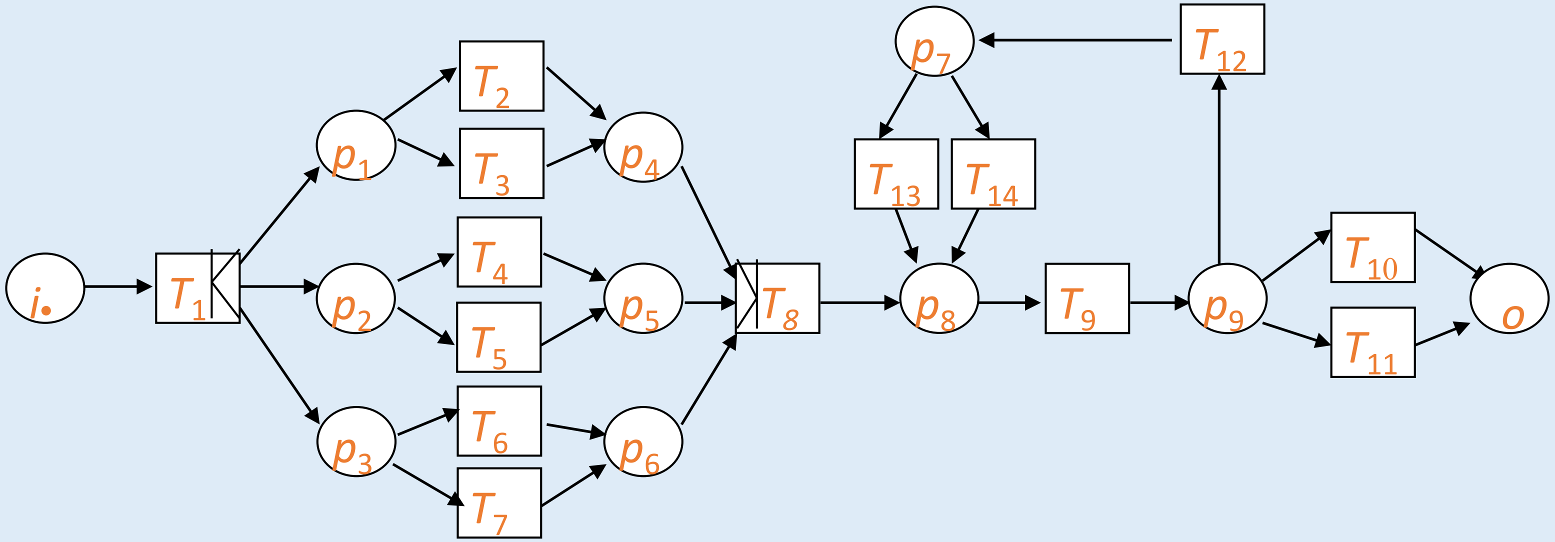
- 5.3.1 Relational Algebra and Its Components
- 5.3.2 Relational Algebra Operations
- 5.3.3 Business Process Modeling and Its Tools (Languages)
  - Business Process Modeling Tools
  - Workflow Net and Its Components
  - WF-net Transition Types
  - Formal Definition of WF-net

- **Business Process Modeling (BPM)** is a method for graphically representing internal processes within an organization.
- BPM is an interdisciplinary application of several branches of discrete mathematics—such as graph theory, logic, and automata theory—and represents an advanced application level of **system behavior modeling** based on discrete mathematics.
- Popular Business Process Modeling **Tools and Languages:**
  - Petri Nets
  - Workflow Net
  - UML Activity Diagrams
  - BPMN（Business Process Model and Notation）
  - EPC（Event-driven Process Chain）

- ***Petri Net*** is a mathematical modeling language consisting of **places**, **transitions**, and **tokens**. It can accurately describe and analyze the dynamic behavior of complex systems.
- A ***WF-net*** is a specialized Petri net that adds start and end constraints for workflows, using sets, relations, and functions to enhance business process modeling and analysis.
- ***WF-net Components***:

Component	Symbol / Notation	Description
Place ***P***	○	condition or state
Transition ***T***	▭	activity or task
Flow Relation ***F***	→	control flow connection
Start Place ***i***	○（**no input**）	start place
End Place ***o***	○（**no output**）	end place
Net Definition	$N = (P, T, F)$	triple of ***P***, ***T***, and ***F***

↳ Workflow Net(e.g.)



T_1: Receive the paper, invite three reviewers.

T_2, T_4, T_6, T_{13}: Receive review comments on time.

T_3, T_5, T_7, T_{14}: Do not receive review comments on time; ;

T_8: Summarize review comments.

T_9: Decide whether to accept the paper; ;

T_{10}: Accept the paper; ;

T_{11}: Reject the paper; ;

T_{12}: Re-invite other reviewers ;

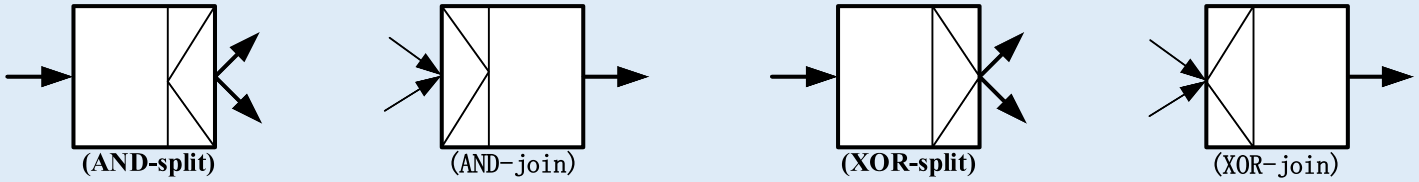
5.3.4 Business Process Modeling and Its Tools (Languages)

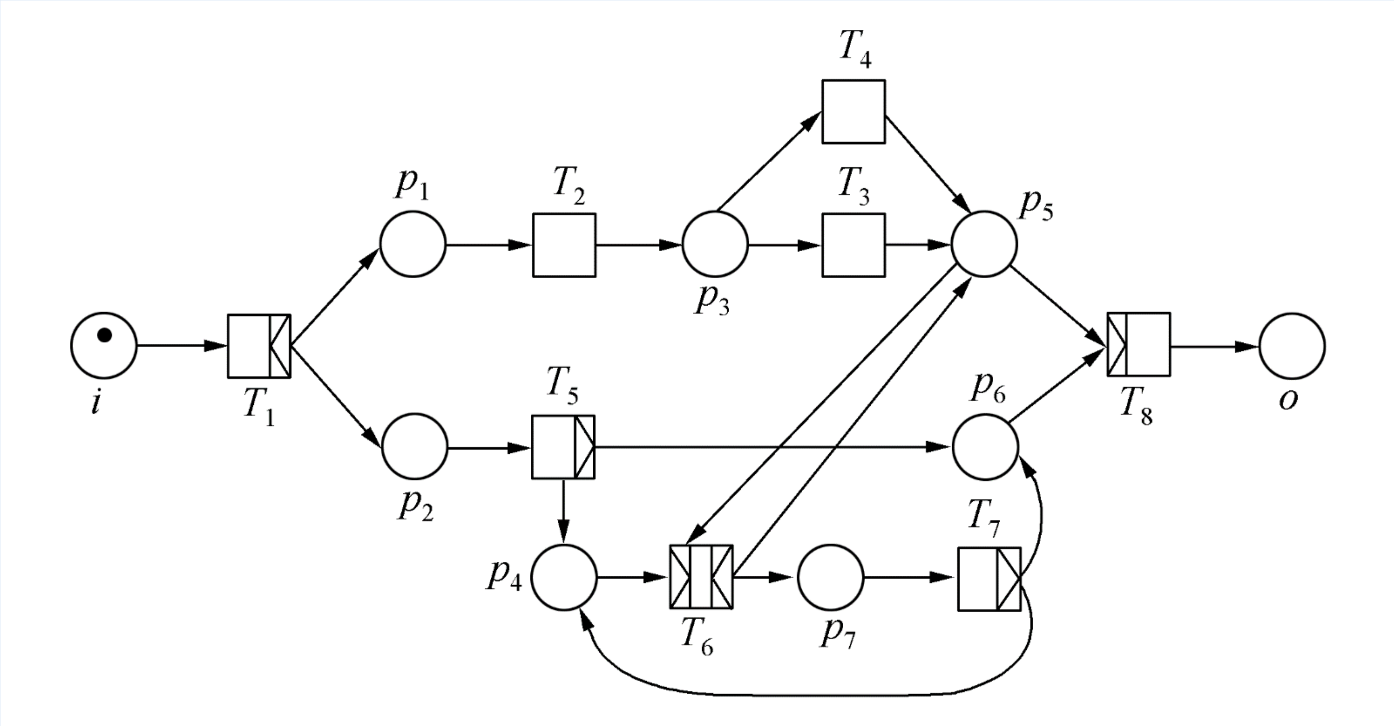
↳ WF-net Transition Types

- WF-net Transition Types:

Type	Structure	Description
Normal	One input → One output	Executes a single task in sequence
AND-split	One input → Multiple outputs	Starts parallel branches
AND-join	Multiple inputs → One output	Waits for all branches to continue
XOR-split	One input → One of several outputs	Selects one path based on condition
XOR-join	One of several inputs → One output	Any path completes to proceed
Loop	Output reconnects to earlier place	Repeats a task until condition met
Cancel/Interrupt	Interrupts flow	End a process branch prematurely

- WF-net Control-flow Operator Symbols:





Meaning of Transition Set ***T***:

T_1: Register

T_2: Send out survey

T_3: Process survey

T_4: Handle expiration

T_5: Evaluate complaints

T_6: Handle complaints

T_7: Check processing results

T_8: Archive

Place Set ***P***:

i: Start ;

o: End

p_1, p_2, p_3, …, p_7: Intermediate places

同济经管 TONGJI SEM

- **WF_net** is a triple $<P,T,F>$, P: Set of places, T: Set of transitions, F: Flow relation.

(1) $P \cap T = \varnothing$; (2) $P \cup T \neq \varnothing$; (3) $F \subseteq P \times T \cup T \times P$;

(4) **dom**$F \cup$ **ran**$F = P \cup T$, **where dom**$F = \{x \mid \exists y(<x,y> \in F)\}$,

ran$F = \{y \mid \exists x(<x,y> \in F)\}$;

(5) There exists a starting place $i \in P$, **such that** $\bullet i = \varnothing$,

$\bullet i = \{j \mid <j,i> \in F\}$ is the **pre-set** of i;

(6) There exists an ending place $o \in P$, **such that** $o\bullet = \varnothing$,

$o\bullet = \{j \mid <o,j> \in F\}$ is the **post-set** of o;

(7) Each node $x \in P \cup T$, lies on a path from ***I to o***.

Objective :

Key Concepts :

Objective :

Key Concepts ：