离散数学及其应用 第9章 关系

9. Relations

9.1 Relations and Their Properties

A `binary relation` （二元关系） R from a set A to a set B is a subset of $A \times B$. $R = \{(a,b)|a \in A, b \in B, aRb\}$

Let $A_1, A_2, \ldots, A_n$ be sets. An `n-ary Relation` on these sets is a subset of $A_1 \times A_2 \times \ldots \times A_n$. $A_1, A_2, \ldots, A_n$ are called the `domains` of relation, and n is called the `degree` of the relation.

A `relation on the set` A is a relation form A to A .

There are 2^{n^2} binary relations on a set A with n elements

9.2 Representing Relations

📋The Methods of Representing Relations

- List its all ordered pairs
- Using a set build notation / specification by predicates
- 2D table
- Connection matrix / zero-one matrix
- Directed graph / Digraph

📋2D table

🌰e.g.

$A = \{2,3,4\}$, $B = \{2,3,4,5,6\}$, $R = \{(a,b) | a \in A, b \in B, a|b\}$. Represent the relation R using a `2D table` .

Solution:

$R = \{(2,2),(2,4),(2,6),(3,3),(3,6),(4,4)\}$

	2	3	4	5	6
2	×		×		×
3		×			×
4			×		

📋Connection Matrices

Let R be a relation from $A = \{a_1, a_2, ..., a_m\}$ to $B = \{b_1, b_2, ..., b_n\}$. An $m \times n$ `connection matrix` $M_R = [m_{ij}]_{m \times n}$ for R is defined by

$m_{ij} = \begin{cases} 1, \text{if}(a_i, b_j) \in R \\ 0, \text{if}(a_i, b_j) \notin R \end{cases}$.

🌰e.g.

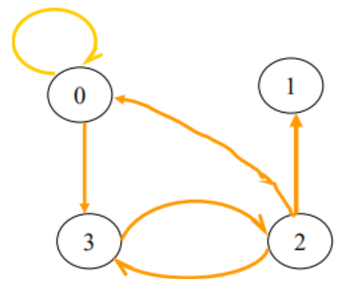
The connection matrix for the 2D table above is $\begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$

📋Directed graph / Digraph

A `directed graph` （有向图） or a `digraph` , consists of a set V of `vertices` （结点） together with a set E of ordered pairs of elements of V called `edges` (or arcs). The vertices a, b is called the `initial vertices` and `terminal vertices` of the edge (a, b) .

🌰e.g.

$A = \{0,1,2,3\}$, $R = \{(0,0),(0,3),(2,0),(2,1),(2,3),(3,2)\}$. Show the relation with a directed graph.



9.3 n–ary Relations and Their Applications

📋 Properties of Binary Relations

- Reflexive
- Irreflexive
- Symmetric
- Antisymmetric
- Transitive

A relation R on a set A is `reflexive` （自反） if $\forall x(x \in A \rightarrow (x, x) \in R)$. In matrices representing reflexive relations, all the elements on the main diagonal of M_R must be 1. In digraphs representing reflexive relations, there is a loop at every vertex.

A relation R on a set A is `irreflexive` （反自反） if $\forall x(x \in A \rightarrow (x, x) \notin R)$. In matrices representing irreflexive relations, all the elements on the main diagonal of M_R must be 0. In digraphs representing irreflexive relations, there is no loop at any vertex.

⚠️ A relation on a set can be neither reflexive nor irreflexive, for example, $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$.

A relation R on a set A is `symmetric` （对称） if $\forall x \forall y((x, y) \in R \rightarrow (y, x) \in R)$. Matrices representing symmetric relations M_R must be symmetric. In digraphs representing symmetric relations, if there is an arc (x, y) there must be an arc (y, x) .

A relation R on a set A is `antisymmetric` （反对称） if $\forall x \forall y((x, y) \in R \wedge (y, x) \in R \rightarrow x = y)$, or equivalently $\forall x \forall y((x, y) \in R \wedge x \neq y \rightarrow (y, x) \notin R)$. In digraphs representing symmetric relations, if there is an arc (x, y) connecting two different vertices, there can not be an arc (y, x) .

⚠️ A relation on a set can be both symmetric and antisymmetric, for example, $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ again.

A relation R on a set A is **`transitive`** （传递） if $\forall x \forall y \forall z((x,y) \in R \land (y,z) \in R \to (x,z) \in R)$. In matrices representing transitive relations, $\overline{(m_{ij} \land m_{jk})} \lor m_{jk} = 1$. In digraphs representing transitive relations, if there is an arc from x to y and one from y to z then there must be one from x to z .

🌰 e.g.

> Symmetric, Transitive ⇒ Reflexive?

Solution:

> Since $(a,b) \in R$ and R is symmetric, $(b,a) \in R$. Since $(a,b) \in R$, $(b,a) \in R$ and R is transitive, $(a,a) \in R$. But this is valid only for a satisfying $(a,b) \in R$. It is still possible that $(a,a) \notin R$ when $\nexists b((a,b) \in R)$.

Since relations from A to B is a subset of $A \times B$, two relations from A to B can be combined in any way two sets can be combined.

📋 Set Operations on Relations and Logical Operations of Matrices

We can solve the set operations on relations using exhaustive method, or logical operations of the matrices of the relations.

The Boolean Sum（布尔和） ∨: $0 \lor 0 = 0,\ 0 \lor 1 = 1,\ 1 \lor 0 = 1,\ 1 \lor 1 = 1$

The Boolean Product（布尔积） ∧: $0 \land 0 = 0,\ 0 \land 1 = 0,\ 1 \land 0 = 0,\ 1 \land 1 = 1$

The complement ─: $\overline{0} = 1,\ \overline{1} = 0$

Let $A = \{a_1, a_2, ..., a_m\}$, $B = \{b_1, b_2, ..., b_n\}$, $M_{R_1} = [c_{ij}]$, $M_{R_2} = [d_{ij}]$, the set operations of two relations are defined by

- $M_{R_1 \cup R_2} = M_{R_1} \lor M_{R_2} = [c_{ij} \lor d_{ij}]$
- $M_{R_1 R_2} = M_{R_1} \land M_{R_2} = [c_{ij} \land d_{ij}]$
- $M_{\overline{R_1}} = [\overline{c_{ij}}]$
- $M_{R_1 - R_2} = M_{R_1 \overline{R_2}} = [c_{ij} \land \overline{d_{ij}}]$

📋 Composition of Relations

$R = \{(a,b) | a \in A, b \in B, aRb\}$, $S = \{(b,c) | b \in B, c \in C, bSc\}$, then the **composite** （复合） of R and S , denoted as $S \circ R$, is $S \circ R = \{(a,c) | a \in A \wedge c \in C \wedge \exists b (b \in B \wedge aRb \wedge bSc)\}$.

To compute $S \circ R$, we can either use the definition directly or use the connection matrix.

🌰 e.g.

$$A = \{a, b\}, B = \{1, 2, 3, 4\}, C = \{5, 6, 7\}$$
$$R = \{(a, 1), (a, 2), (b, 3)\}, S = \{(2, 6), (3, 7), (4, 5)\}$$
$$S \circ R = ? \;\; R \circ S = ?$$

Solution:

(1) Using the definition directly

$$S \circ R = \{(a, 6), (b, 7)\}, \;\; R \circ S = \emptyset$$

(2) Using the connection matrix

$$\mathbf{M}_R = [r_{ij}]_{m \times n}, \mathbf{M}_S = [s_{jk}]_{n \times l}$$
$$\mathbf{M}_{S \circ R} = \mathbf{M}_R \cdot \mathbf{M}_S = [w_{ik}]_{m \times l}, \;\; w_{ik} = \bigvee_{j=1}^{n} (r_{ij} \wedge s_{jk})$$
$$A = \{a, b\}, B = \{1,2,3,4\}, C = \{5,6,7\}$$
$$R = \{(a,1),(a,2),(b,3)\}, S = \{(2,6),(3,7),(4,5)\}$$
$$\therefore \mathbf{M}_R = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad \mathbf{M}_s = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$
$$\mathbf{M}_{S \circ R} = \mathbf{M}_R \cdot \mathbf{M}_S = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
$$\therefore S \circ R = \{(a,6),(b,7)\}$$

Let R be a relation on the set A . The **powers** are defined inductively by $R, \, n = 1, 2, 3, \ldots$ are defined inductively by $R^1 = R, \, R^{n+1} = R^n \circ R$.

📋 The relation R on the set A is transive if and only if $R^n \subseteq R$, for $n = 1, 2, 3 \ldots$

Proof:

(1) $R^n \subseteq R, \quad for \quad n = 1,2,3,\ldots \quad \Rightarrow$ **R is transitive**

$(a,b) \in R, \ (b,c) \in R \Rightarrow (a,c) \in R$

$(a,c) \in R^2 \subseteq R$

(2) R is transitive $\Rightarrow \quad R^n \subseteq R, \quad for \quad n = 1,2,3,\ldots$

- **Inductive base** $\quad n = 1, \quad R \subseteq R$
- **Inductive step** $\quad R^n \subseteq R \quad \Rightarrow \quad R^{n+1} \subseteq R$

$\left.\begin{array}{l}(a,b) \in R^{n+1} \\ R^{n+1} = R^n \circ R\end{array}\right\} \Rightarrow \left.\begin{array}{l}(a,x) \in R, (x,b) \in R^n \subseteq R \\ R \text{ is transitive}\end{array}\right\} \Rightarrow (a,b) \in R$

For a relation $R = \{(a,b)|a \in A, b \in B, aRb\}$, the `inverse relation` from B to A is $R^{-1} = \{(b,a)|(a,b) \in R, a \in A, b \in B\}$.

📋 Methods to Get R^{-1}

1. Using the definition directly.
2. Reverse all the arcs in the digraph representation of R .
3. Take the transpose M_R^T of the connection matrix M_R of R .

📋 The Rroperties of Relation Operations

Suppose that R , S are the relations from A to B , T is the relation from B to C , P is the relation from C to D , then

- $(R \cup S)^{-1} = R^{-1} \cup S^{-1}$
- $(R|S)^{-1} = R^{-1}|S^{-1}$
- $(\overline{R})^{-1} = \overline{R^{-1}}$
- $(R - S)^{-1} = R^{-1} - S^{-1}$
- $(A \times B)^{-1} = B \times A$
- $\overline{R} = A \times B - R$
- $(S \circ T)^{-1} = T^{-1} \circ S^{-1}$
- $(R \circ T) \circ P = R \circ (T \circ P)$
- $(R \cup S) \circ T = (R \circ T) \cup (S \circ T)$

9.4 Closures of Relations

The `closure` of a relation R with respect to property P is the relation S with property P containing R such that S is a subset of every relation with property P containing R. Which means, S is the smallest relation with property P containing R .

📋Let R be a relation on A . The `reflexive closure` of R , denoted by $r(R)$, is $R \cup I_A$, where $I_A = \{(x, x) | x \in A\}$ is called the `diagnal relation` on A .

Corollary: $R = R \cup I_A \Leftrightarrow R$ is a reflexive relation.

Given R , to obtain its reflexive closure, we can:

- Add to R all ordered pairs of the form (a, a) with $a \in A$, not already in R .
- Add loops to all vertices on the digraph representation of R .
- Put 1's on the diagonal of the connection matrix of R .

📋Let R be a relation on A . The `symmetric closure` of R , denoted by $s(R)$, is $R \cup R^{-1}$.

Proof:

Obviously $R \cup R^{-1}$ contains R and is symmetric. Then we need to show it is the smallest symmetric relation which contains R .

Suppose that R' is a symmetric relation containing R , then

$$\text{If } (a,b) \in R \cup R^{-1}$$

$$\Rightarrow \left\{ \begin{array}{l} \left.\begin{array}{c} (a,b) \in R \\ R \subseteq R' \end{array}\right\} \Rightarrow (a,b) \in R' \\ \left.\begin{array}{l} (a,b) \in R^{-1} \Rightarrow (b,a) \in R \Rightarrow (b,a) \in R' \\ \qquad\qquad R' \text{ is a symmetric relation} \end{array}\right\} \Rightarrow (a,b) \in R' \end{array} \right\}$$

$$\Rightarrow R \cup R^{-1} \subseteq R'$$

Corollary: $R = R \cup R^{-1} \Leftrightarrow R$ is a symmetric relation.

Given R , to obtain its symmetric closure, we can:

- Add all ordered pairs of the form (b, a) where (a, b) is in the relation, that are not already in R .

- Add an edge from x to y whenever this edge is not already in directed graph but the edge from y to x is.
- $M_{s(R)} = M_R \vee M_R^T$

Then we go on to the `transive closure` （传递闭包）, denoted by $t(R)$. It can not be computed so simply as the two kinds of closures above. Firstly we need to introduce some terminologies.

A `path` of `length` n in a digraph G means a sequence of edges $(x_0, x_1), (x_1, x_2), \ldots, (x_{n-1}, x_n)$, denoted as $x_0, x_1, x_2, \ldots, x_{n-1}, x_n$. It is called a `cycle` or `circuit` if $x_0 = x_n$.

The term "path" also applies to relations. We say there is a path of length n from a to b in R if $\exists a, x_1, x_2, \ldots, x_{n-1}, b$ such that $(a, x_1) \in R, (x_1, x_2) \in R, \ldots, (x_{n-1}, b) \in R$.

📋Let R be a relation on A . There is a path of length n from a to b if and only if $(a, b) \in R^n$.

Proof: Using MI.

The `connectivity relation` denoted by R^* , is the set of ordered pairs (a, b) such that there is a path (in R) from a to b : $R^* = \bigcup\limits_{n=1}^{\infty} R^n$.

📋 $t(R) = R^*$.

Proof:

Obviously R^* contains R and is a transitive relation by its definition. Then we need to show that R^* is the smallest transitive relation which contains R .

Now suppose that S is any transitive relation which contains R . We need to show S contains R^* .

Since S is transitive, S^n is also transitive and $S^n \subseteq S$. It follows that $S^* \subseteq S$. If $R \subseteq S$, then $R^* \subseteq S^*$, because any path in R is also a path in S . So $R^* \subseteq S$.

Corollary: $R = t(R) \Leftrightarrow R$ is transitive.

💡 In fact, we need only consider paths of length n or less, where n is the cardinality of R .

📋If $|A| = n$, then any path of length $> n$ must contain a cycle.

Proof: Using the Pigeon Hole Principle.

📋If $|A| = n$, R is a relation on A, then $\exists k, k \le n, R^* = R \cup R^2 \cup ... \cup R^k$.

Corollary: If $|A| = n$ then $t(R) = R^* = R \cup R^2 \cup ... \cup R^n$.

Corollary: $M_{t(R)} = M_R \vee M_R^{[2]} \vee ... \vee M_R^{[n]}$.

📋A Basic Procedure for Computing t(R)

$A = \mathbf{M}_R;$

$B = A;$

for (*i*=2; *i*<=n; *i*++)

{

$A = A \cdot \mathbf{M}_R;$

$B := B \vee A;$

}

The complexity of algorithm:

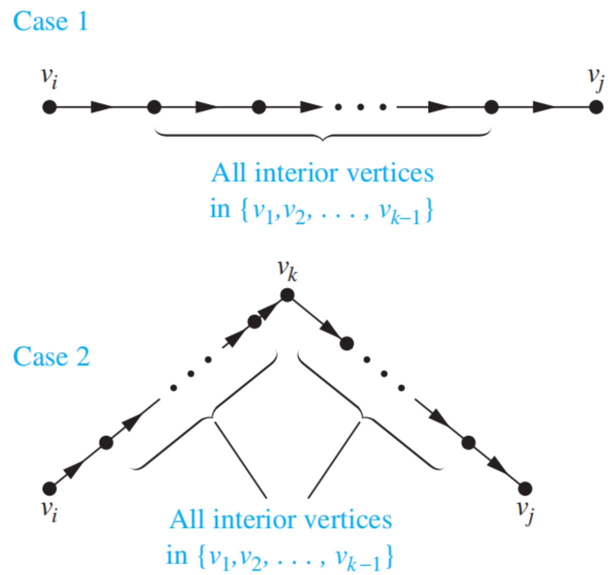
$$n^2(2n-1)(n-1) + (n-1)n^2 = 2n^3(n-1) = O(n^4)$$

📋 Warshall's Algorithm for Computing t(R)

> 别以为这个不会考!

If $a, x_1, x_2, \ldots, x_{m-1}, b$ is a path, its `interior vertices` are $x_1, x_2, \ldots, x_{m-1}$, that is, all the vertices of the path that occur somewhere other than as the first and last vertices in the path. Warshall's algorithm is based on the efficient construction of a sequence of zero—one matrices. These matrices are $W_0, W_1, \ldots, W_n$, where $W_0 = M_R$ is the zero—one matrix of relation R, and $W_k = [w_{ij}^{(k)}]$, where $w_{ij}^{(k)} = 1$ if there is a path from v_i to v_j such that all the interior vertices of this path are in the set $\{v_1, v_2, \ldots, v_k\}$ (the first k vertices in the list) and is 0 otherwise. Note that $W_n = M_{R^*}$, because the (i, j)th entry of M_{R^*} is 1 if and only if there is a path from v_i to v_j, with all interior vertices in the set $\{v_1, v_2, \ldots, v_n\}$.

We can compute W_k directly from W_{k-1} : There is a path from vi to vj with no vertices other than v1, v2, ... , vk as interior vertices if and only if either there is a path from v_i to v_j with its interior vertices among the first $(k-1)$ vertices in the list, or there are paths from v_i to v_k and from v_k to v_j that have interior vertices only among the first $(k-1)$ vertices in the list. That is, either a path from v_i to v_j already existed before v_k was permitted as an interior vertex, or allowing v_k as an interior vertex produces a path that goes from v_i to v_k and then from v_k to v_j .



The first type of path exists if and only if $w_{ij}^{[k-1]} = 1$, and the second type of path exists if

and only if both $w_{ik}^{[k-1]}$ and $w_{kj}^{[k-1]}$ are 1. Hence, $w_{ij}^{[k]} = w_{ij}^{[k-1]} \vee (w_{ik}^{[k-1]} \wedge w_{kj}^{[k-1]})$ whenever i, j and k are positive integers not exceeding n.

ALGORITHM 2 Warshall Algorithm.

procedure *Warshall* ($\mathbf{M}_R$: $n \times n$ zero–one matrix)
$\mathbf{W} := \mathbf{M}_R$
for $k := 1$ **to** n
  for $i := 1$ **to** n
    for $j := 1$ **to** n
      $w_{ij} := w_{ij} \vee (w_{ik} \wedge w_{kj})$
return $\mathbf{W}\{\mathbf{W} = [w_{ij}]$ is $\mathbf{M}_{R^*}\}$

The complexity of Warshall's algorithm is $2n^3$.

9.5 Equivalence Relation

A relation R on a set A is an `equivalence relation` （等价关系） if R is reflexive, symmetric and transitive. If elements a and b are related by an equivalence relation R , then we say a and b are equivalent, denoted by $a \sim b$. In an equivalence relation R , the set of all elements that are related to an element x of A is called the `equivalence class` （等价类） of x , denoted by $[x]_R$ or $[x]$.

$[x]_R = \{y|(y,x) \in R\}$

Let $\{A_1, A_2, ...\}$ be a collection of subsets of A . Then the collection forms a `partition` （分割） of A if and only if

- $A_i \neq \emptyset, i \in \mathbb{Z}$
- $i \neq j \Rightarrow A_i \cap A_j = \emptyset$
- $\forall a \in A, \exists i, a \in A_i$

denoted by $pr(A) = \{A_1, A_2, ...\}$. 就是说这一系列 A_i 的无交并为 A。

📋 Let R be an equivalence relation on a set A . The following statements are equivalent

- aRb
- $[a] = [b]$
- $[a] \cap [b] \neq \emptyset$

📋 Let R be an equivalence relation on a set A . Then the equivalence classes of R form a partition of A . Conversely, given a partition $\{A_i|i \in I\}$ of the set A , there is an equivalence relation R that has the sets $A_i, i \in I$, as its equivalence classes.

In short, an equivalence relation on a set A $\leftrightarrow$ a partition of A .

📋 Congruence modulo m forms a partition of all integers.

$$R = \{(a,b)|a \equiv b(\bmod m), a, b \in \mathbb{Z}\}$$

$$pr(\mathbb{Z}) = \{[0]_m, [1]_m, ..., [m-1]_m\}$$

🌰e.g.

$R = \{(a,b)|a \equiv b(\bmod 3), a, b \in \mathbb{Z}\}$. Show that R is an equivalence relation, and find its equivalence class.

Proof:

$a \equiv b(\bmod 3)$ if and only if $3|(a-b)$.

R is reflexive because $3|(a-a)$. We can also show it is symmetric and transitive using the qualities of division.

The equivalence classes are $[0]_3$, $[1]_3$ and $[2]_3$.

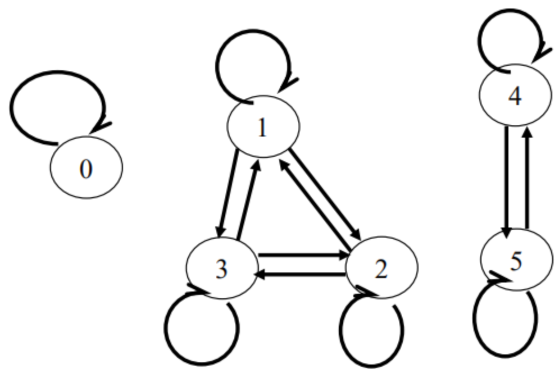
e.g.

Find the partition of the set A from R .

$$A = \{0,1,2,3,4,5\}$$

$$R = \{(0,0),(1,1),(2,2),(3,3),(4,4),(5,5),\\ (1,2),(1,3),(2,1),(2,3),(3,1),(3,2),\\ (4,5),(5,4)\}$$

Solution:



[0]={0}

[1]={1,2,3}

[4]={4,5}

pr(*A*)={ [0], [1], [4]}

📋 If R_1, R_2 are equivalence relations on A , then $R_1 \cap R_2$ is also an equivalence relation on A .

Proof:

1. It is reflexive. $\forall a \in A, (a,a) \in R_1, (a,a) \in R_2$, so $(a,a) \in R_1 \cap R_2$.
2. It is symmetric. If $(a,b) \in R_1 \cap R_2$, then $(a,b) \in R_1$ and $(a,b) \in R_2$, then $(b,a) \in R_1$ and $(b,a) \in R_2$, so $(b,a) \in R_1 \cap R_2$.
3. It is transitive.

📋 If R_1, R_2 are equivalence relations on A , then $R_1 \cup R_2$ is reflexive and symmetric relation on A . Not necessarily transitive.

📋 If R_1, R_2 are equivalence relations on A , then $(R_1 \cup R_2)^*$ is also an equivalence relation on A .

9.6 Partial Orderings

Let R be a relation on S . Then R is a `partial ordering` （偏序关系） or partial order if R is reflexive, antisymmetric and transitive, denoted by (S,R) . S is called a partially ordered set or a `poset` （偏序集）.

🌰e.g.

$$R_1 = \{(a,b)|a \le b, a,b \in \mathbb{Z}\} = (\mathbb{Z}, \le)$$
$$R_2 = \{(a,b)|\ a|b, a,b \in \mathbb{Z}^+\} = (\mathbb{Z}^+, |)$$
$$R_3 = \{(s_1,s_2)|s_1 \subseteq s_2, s_1,s_2 \in P(S)\} = (P(S), \subseteq)$$

We use the notation $\preceq$ to represent the partial orderings including $\le$, $|$ and $\subseteq$. $a \preceq b$ is read as " a is less than or equal to b ". $a \prec b$ means $a \preceq b$ and $a \ne b$, read as " a is less than b ".

The elements a and b of a poset $(S, \preceq)$ are called `comparable` if either $a \preceq b$ or $b \preceq a$. When a and b are elements of S such that neither $a \preceq b$ nor $b \preceq a$, a and b are called `incomparable` .

If $(S, \preceq)$ is a poset and every two elements of S are comparable, S is called a `totally ordered set` （全序集） or a `linearly ordered set` , $\preceq$ is called a `total order` or `linear order` . In this case $(S, \preceq)$ is called a `chain` （链）. 想象一下，全序集中的所有元素都可比较大小，因此它们可以按照大小顺序排成一条链，像数轴那样。

🌰e.g.

$(\mathbb{Z}, \le)$ is a poset and is a chain. $(\mathbb{Z}^+, |)$ and $(P(S), \subseteq)$ are posets but not totally ordered sets.

$(S, \preceq)$ is a `well-ordered` set if it is a poset such that $\preceq$ is a total order and every nonempty subset of S has a least element.

📋 The Principle of Well-Ordered Induction

Suppose that S is a well-ordered set. Then $\forall x \in S, P(x)$, if:

Inductive Step: for every $y \in S$, if $P(x)$ is true for all $x \in S$ with $x \prec y$, then $P(y)$.

Proof:

Suppose it is not the case that $P(x)$ is true for all $x \in S$. Then there is an element $y \in S$

such that $P(y)$ is false. Consequently, the set $A = \{x \in S \mid P(x) \text{ is false}\}$ is nonempty. Because S is well-ordered, A has a least element a . By the choice of a as a least element of A , we know that $P(x)$ is true for all $x \in S$ with $x \prec a$. This implies by the inductive step $P(a)$ is true. This contradiction shows that $P(x)$ must be true for all $x \in S$.

The `lexicographic order` $\preceq$ on $A_1 \times A_2$ is defined as following: given two posets $(A_1, \preceq_1)$ and $(A_2, \preceq_2)$, we construct an induced partial order R on $A_1 \times A_2$: $(x_1, y_1) \preceq (x_2, y_2)$ if $x_1 \preceq_1 x_2$ or $(x_1 = x_2 \ and \ y_1 \preceq_2 y_2)$. The definition of lexicographic order extends naturally to multiple Cartesian products of partially ordered sets.

🌰e.g. Lexicographic Order of String

The string $a_1a_2...a_m$ is less than $b_1b_2...b_n$ if and only if $(a_1a_2...a_m) \prec (b_1b_2...b_n) \ or \ [(a_1a_2...a_{min(m,n)}) = (b_1b_2...b_{min(m,n)}) \ and \ m < n]$.

discredit < discreet < discrete < discreteness < discretion

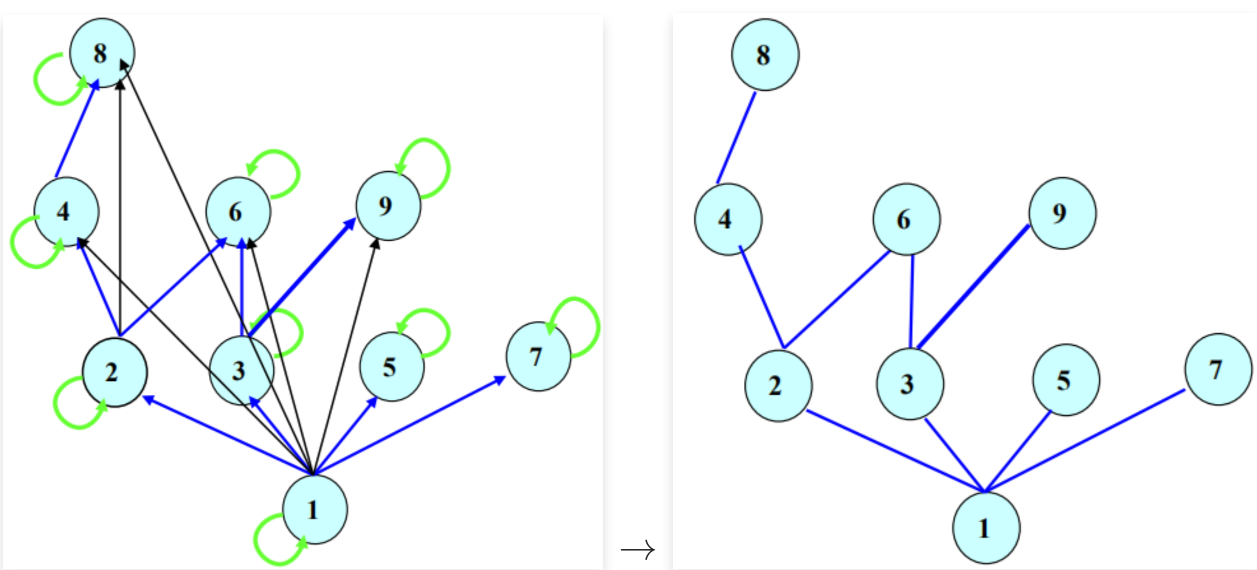
`Hasse diagram` （哈塞图） is a method used to represent a partial ordering clearly.

📋 To construct a Hasse diagram:

1. Construct a digraph representation of the poset (A, R) so that all arcs point up (except the loops).
2. Eliminate all loops.
3. Eliminate all arcs that are redundant because of transitivity.
4. Eliminate the arrows at the ends of arcs since everything points up.

🌰e.g.

$A = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}, R = \{(a, b) | \ a|b, a, b \in A\}$, and represent it using a Hasse diagram.



Let $(A, \preceq)$ be a poset. $B \subseteq A$, if $(B, \preceq)$ is a totally ordered set, then B is called a `chain` of $(A, \preceq)$. $|B|$ is called the `length` of chain. Otherwise, if $\forall a, b \in B(a \neq b \to (a, b) \notin R \wedge (b, a) \notin R)$, then B is called an `antichain` of $(A, \preceq)$.

Let $(A, \preceq)$ be a poset, then a is a `maximal element` （极大元） if there does not exist an element b in A such that $a \prec b$. Similarly for a `minimal element` （极小元）. Maximal and minimal elements are the "top" and "bottom" elements in the Hasse diagram.

⚠️ There can be more than one minimal and maximal element in a poset.

Let $(A, \preceq)$ be a poset. Then an element a in A is a `greatest element` （最大元） of A if $b \preceq a$ for every b in A , and a is a `least element` （最小元） of A if $a \preceq b$ for every b in A .

📋 The greatest and least element are unique when they exist.

Proof:

就是假设有两个最大元 / 最小元，然后证明它们必须是相等的，相当于只有一个。Suppose that a_1 is a greatest element in A . It follows that $x \leq a_1$ for every element in A . Suppose that a_2 is also a greatest element in A . It follows that $x \leq a_2$ for every element in A . It implies that $a_1 \leq a_2$ and $a_2 \leq a_1$. That is $a_1 = a_2$.

Let A be a subset of S in the poset $(S, \preceq)$. If there exists an element a in S such that $b \preceq a$ for all b in A , then a is called an `upper bound` （上界） of A . Similarly for `lower bounds` （下界）.

⚠️ Mind the differences of "maximal element", "greatest element" and "upper bound". “极大元”只要能比较大小的都小于等于它就行，“最大元”要所有元素都能和它比大小且都小于等于它，“上界”甚至可以不在子集 A 里面但子集 A 里所有元素必须小于等于它。

If a is an upper bound for P which is less than every other upper bounds, then it is the `least upper bound` （上确界）, denoted by $lub(S)$. Similarly for the `greatest lower bound` （下确界）, $glb(S)$.

A poset is called a `lattice` （格） if every pair of elements has a lub and a glb.

听说中文男足辱了北大中文系的系格

🌰e.g.

$(\{1,3,6,9,12\},|)$ is not a lattice because there is no multiple of both 6 and 9 in these numbers. If we add 36 into these numbers then it will form a lattice.

📋 Every totally ordered set is a lattice.

⚙️ The Lattices Model of Information Flow

The Lattices Model can be used to represent different information flow policies. For example, the multilevel security policy:

Each pieces of information is assigned to a security class. Each security class is represented by a pair (A,C) , where A is an authority level and C is a category. Order security classes by specifying that $(A_1,C_1) \preceq (A_2,C_2)$ iff $A_1 \leq A_2, C_1 \subseteq C_2$. For example, the typical authority levels used in the U.S. government is A={ unclassified(0), confidential(1), secret(2), top secret(3) }. If C={ spies, moles, double agents }, then their are $2^{|C|}$ = 8 different categories.

Information is permitted to flow from security classes (A_1,C_1) into (A_2,C_2) iff $(A_1,C_1) \preceq (A_2,C_2)$. Then the set of all security classes forms a lattice.

A total ordering $\preceq$ is said to be `compatible` （共生） with the partial ordering R if $a \preceq b$ whenever aRb . Constructing a compatible total ordering from a partial ordering is called `topological sorting` （拓扑排序）.

📋 Every finite nonempty poset $(S, \preceq)$ has at least one minimal element.

Proof:

Choose an element a_0 of S . If a_0 is not minimal, then there is an element a_1 with $a_1 \prec a_0$. If a_1 is not minimal, there is an element a_2 with $a_2 \prec a_1$. Continue this process, so that if a_n is not minimal, there is an element a_{n+1} with $a_{n+1} \prec a_n$.

Since there are only finite number of elements in the poset, this process must end with a minimal element.

📋 According to this lemma, to sort a poset (S, R) :

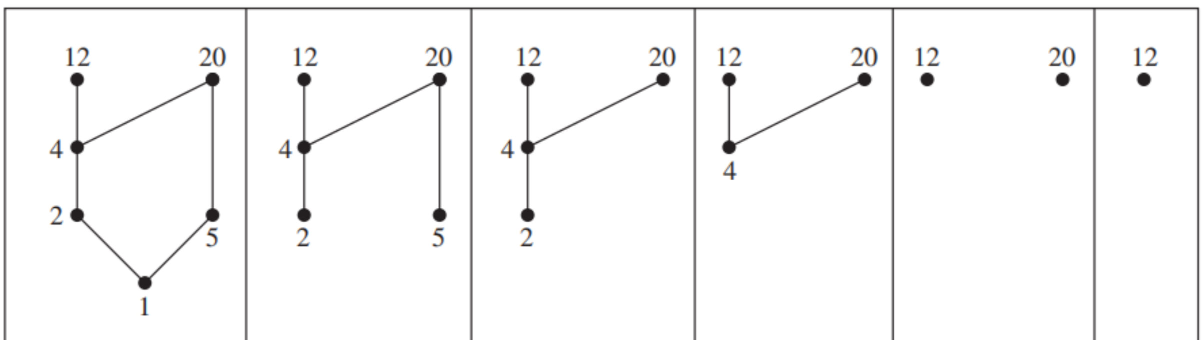
Basic Step: Select a (any) minimal element and put it in the list. Delete it from S .

Inducive Step: Continue until all elements appear in the list and S is void.

🌰e.g.

Find a compatible total ordering for the poset $(\{1, 2, 4, 5, 12, 20\}, |)$.

Solution:



Minimal element chosen	1	5	2	4	20	12

Delete one minimal element in each step. If there are more than one minimal element in one step, just choose a random one. This produces the total ordering 1 < 5 < 2 < 4 < 20 < 12.

⚙️ Topological sorting has an application to the scheduling of projects. The Hasse diagram means the order of completing a series of projects, where the less project need to be finished before the greater project is finished. By producing a total ordering, we can find a feasible order to finish all the projects.