第1章 绪论

内容

- 预备知识
- 中心概念

预备知识

--集合

--关系

--图

--证明方法

预备知识

--集合

--关系

--图

--证明方法

集合

集合是构造所有其他离散结构如关系、组合、图的基础。

集合用于把对象组织在一起。

集合是数学中最基本的概念之一，很难做出所谓严谨的、符合数学要求的定义。这里作如下的描述：

一些确切的对象汇集在一起视为一个单一的总体，称为**集合**。

从集合这一直观描述可导致悖论。

集合与对象

集合中的对象称为该集合的**元素**或**成员**。

若a是集合A的一个元素，则记为$a\in A$，读作“a属于A”或“a在A中”。

若a不是集合A的一个元素，则记为$a\notin A$，读作“a不属于A”或“a不在A中”。

任一对象对某一集合而言，或属于该集合，或不属于该集合，二者必居其一，不可兼得。

集合的表示

- 枚举法（列举法）：把集合中的元素逐个列出，或在能明确含意的情况下列出一部份，其余用“…”表示。通常只用于表示有穷集合或者至多包含全体自然数那么多个元素的集合。

例：A={红，黄，蓝}

S={1，2，…，k}

集合的表示

- 描述法：用集合中所有元素共有的性质或应该满足的条件表示集合。

例：$O=\{x \mid x\text{是小于}10\text{的奇数}\}$

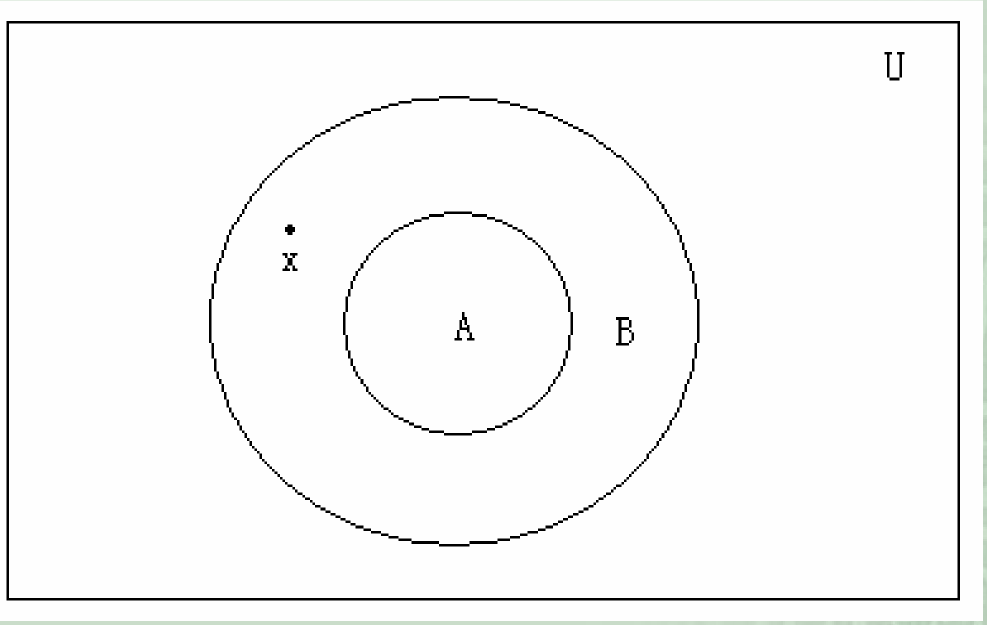
$$A=\{x \mid P(x)\}$$

- 归纳法：用递归定义的方法去描述集合。

例：$\{x_{i+1}=x_i+1, i=1, 2, 3, x_1=1\}$

集合的表示

- 文氏图。考虑的所有对象的集合用长方形表示，在其内部用圆或其他图形表示集合，用点表示元素。



几种集合

定义：不含任何元素的集合称为**空集**，记为∅或{}。

定义：仅含一个元素的集合称为**单元素集合**。

定义：若集合S中恰有n个不同的元素，n是非负整数，则称S是**有限集合**，n是S的**基数或势**，记为|S|。

定义：不是有限集合的集合称为**无限集合**或**无穷集**。

定义：整个论述域对应的集合称为**全集**，记为U。

集合间的关系

定义：两个集合A和B**相等**当且仅当它们有相同的元素，记为A=B。

形式化表示为：

$A=B \Leftrightarrow \forall x(x \in A \leftrightarrow x \in B)$

若两个集合含有相同的元素，则不管如何表示，它们都相等。

集合中元素的列举次序、是否重复都无关紧要。集合的表示不唯一。

集合间的关系

定义：集合A是集合B的**子集**当且仅当A的每个元素也是B的元素，记为$A\subseteq B$，读作“B包含A”或“A包含于B中”。

形式化表示为：

$$A\subseteq B\Leftrightarrow\forall x(x\in A\rightarrow x\in B)$$

集合间的关系

定义：集合A是集合B的**真子集**当且仅当$A\subseteq B$且$A\neq B$，读作“B真包含A”。

形式化表示为：

$$A\subset B \Leftrightarrow A\subseteq B\wedge A\neq B$$

$$\Leftrightarrow \forall x(x\in A\rightarrow x\in B)\wedge\exists x(x\in B\wedge x\notin A)$$

集合间的关系

对任意集合A、B、C：

- $\varnothing \subseteq A$ 。
- $A \subseteq U$。
- $A \subseteq A$。
- $A=B \Leftrightarrow A \subseteq B$ 且 $B \subseteq A$。
- 如果 $A \subseteq B$，则 $|A| \leqslant |B|$。
- 如果 $A \subset B$，则 $|A| \leqslant |B|$。
- 如果A是有穷集，且 $A \subset B$，则 $|A|<|B|$ 。
- 如果 $A \subseteq B$，则对 $\forall x \in A$，有 $x \in B$。
- 如果 $A \subset B$，则对 $\forall x \in A$，有 $x \in B$ 并且 $\exists x \in B$，但 $x \notin A$。
- 如果 $A \subseteq B$ 且 $B \subseteq C$，则 $A \subseteq C$。
- 如果 $A \subset B$ 且 $B \subset C$，或者 $A \subseteq B$ 且 $B \subset C$，或者 $A \subset B$ 且 $B \subseteq C$，则 $A \subset C$。
- 如果 $A=B$，则 $|A|=|B|$。

罗素悖论

从前文对集合的直观定义可导致悖论，即逻辑上的不一致。

罗素悖论：设论述域是所有集合的集合，定义 $S=\{A \mid A \notin A\}$。S是S的元素吗？

一个理发师只给所有不给自己理发的人理发。他给自己理发吗？

以称为公理的基本假设为起点建立集合理论可以避免悖论。

集合运算

集合可用不同的方式结合在一起。集合上的运算把给定的集合结合为一个新的集合。

例如：从选修音乐的学生集合和选修足球的学生集合可以构成选修音乐或足球的学生集合，既选修音乐又选修足球的学生集合等。

集合运算

定义：设A和B是集合。A和B的**并集**是A或B中或同时在A和B中的元素组成的集合，记为$A \cup B$。

元素x属于A和B的并集当且仅当x属于A或x属于B。

形式化表示为：$A \cup B = \{x \mid x \in A \vee x \in B\}$

集合运算

定义：设A和B是集合。A和B的**交集**是同时在A和B中的元素组成的集合，记为$A \cap B$。

元素x属于A和B的交集当且仅当x属于A且x属于B。

形式化表示为：$A \cap B = \{x \mid x \in A \land x \in B\}$

定义：若两个集合交集为空集，则称这两个集合**不相交**。

集合运算

- $S \cup T = T \cup S$ $S \cap T = T \cap S$
- $S \cup (T \cup V) = (S \cup T) \cup V$ $S \cap (T \cap V) = (S \cap T) \cap V$
- $\Phi \cap S = \Phi$
- $S \cap S = S \cup S = S \cup \Phi = S \cap U = S$
- $S \cup U = U$
- $S \subseteq S \cup T$ $S \cap T \subseteq S$
- $S \subseteq T \Leftrightarrow S \cup T = T \Leftrightarrow S \cap T = S$
- $|S \cap T| \leqslant \min\{|S|, |T|\}$
- $S \cup (T \cap V) = (S \cup T) \cap (S \cup V)$ $S \cap (T \cup V) = (S \cap T) \cup (S \cap V)$
- $S \cap (S \cup T) = S \cup (S \cap T) = S$
- $S \subseteq T \wedge V \subseteq W \Rightarrow S \cup V \subseteq T \cup W$ $S \subseteq T \wedge V \subseteq W \Rightarrow S \cap V \subseteq T \cap W$

集合运算

定义：设A和B是集合。A和B的**差集**是在A且不在B中的元素组成的集合，记为A-B。

元素x属于A和B的差集当且仅当x属于A且x不属于B。

形式化表示为：$A-B=\{x \mid x\in A\wedge x\notin B\}$

集合运算

- $A-\Phi=A$
- $A-B\subseteq A$
- $A-A = \Phi-A = \Phi$
- $A-B=A \Leftrightarrow A\cap B=\Phi$
- $A\cap(B-C)=(A\cap B)-(A\cap C)$

集合运算

定义：令U为全集。集合A的补集是全集中不在A中的元素组成的集合，记为 $\overline{A}$ 。

元素x属于$\overline{A}$当且仅当$x \notin A$。

形式化为 $\overline{A}=\{x \mid x \notin A\}=U-A$。

$$\overline{\Phi}=U$$

$$\overline{U}=\Phi$$

集合运算

- $A \cap \overline{A} = \varnothing$
- $A \cup \overline{A} = U$
- $B = \overline{A} \;\Leftrightarrow\; A \cup B = U \wedge A \cap B = \varnothing$
- $\overline{\varnothing} = U \qquad \overline{U} = \varnothing$
- $\overline{\overline{A}} = A$
- $\overline{A \cup B} = \overline{A} \cap \overline{B}$
- $\overline{A \cap B} = \overline{A} \cup \overline{B}$
- $A \subseteq B \Rightarrow \overline{B} \subseteq \overline{A}$

集合运算

定义：一组集合的**并集**是至少属于这组集合中一个集合的那些元素组成的集合。

用记号$A_1 \cup A_2 \cup \cdots \cup A_n = \bigcup_{i=1}^{n} A_i$表示集合$A_1$、$A_2$、…、$A_n$的并集。

集合运算

定义：一组集合的**交集**是属于这组集合中每个集合的那些元素组成的集合。

用记号$A_1 \cap A_2 \cap \cdots \cap A_n = \bigcap_{i=1}^{n} A_i$表示集合$A_1$、$A_2$、…、$A_n$的交集。

集合运算

定义：A、B两个集合的**环和**（对称差）记为 $A\oplus B$，定义为

$$A\oplus B=(A-B)\cup(B-A)$$

$$=\{x \mid x\in A\wedge x\notin B\vee x\in B\wedge x\notin A\}。$$

集合运算

定理： $A \oplus B = (A \cup B) \cap (\overline{A} \cup \overline{B})$

$= (A \cup B) - (A \cap B)$

推论： $\overline{A} \oplus \overline{B} = A \oplus B$

$A \oplus B = B \oplus A$

$A \oplus A = \varnothing$

定理： $(A \oplus B) \oplus C = A \oplus (B \oplus C)$

定理： $A \cap (B \oplus C) = (A \cap B) \oplus (A \cap C)$

集合运算

定义：A、B两个集合的**环积**记为A⊗B，定义为

$$A\otimes B=\overline{A\oplus B}$$

$$=\overline{A\cap\overline{B}\cup B\cap\overline{A}}$$

$$=(A\cup\overline{B})\cap(B\cup\overline{A})$$

$$=A\cap B\cup\overline{A}\cap\overline{B}$$

$$=\{x\mid x\in A\wedge x\in B\vee x\notin B\wedge x\notin A\}$$

集合运算

定理：设A、B和C是U的任意子集，则

$\overline{A}\otimes\overline{B}=A\otimes B$

$A\otimes B=B\otimes A$

$A\otimes A=U$

$(A\otimes B)\otimes C=A\otimes(B\otimes C)$

$A\cup(B\otimes C)=(A\cup B)\otimes(A\cup C)$

集合运算

定义：已知集合S，S的**幂集**是集合S所有子集组成的集合，记为P(S)或2^S 。

形式化表示为：

$P(S)=\{A \mid A\subseteq S\}$

n个元素的集合A，其幂集的元素个数为2^n。

集合运算

(1) $\Phi \in 2^A$。

(2) $\Phi \subseteq 2^A$。

(3) $\Phi \subset 2^A$。

(4) $2^{\Phi}=\{\Phi\}$。

(5) $A \in 2^A$。

(6) 如果A是有穷集，则$|2^A|=2^{|A|}$。

(7) $2^{A\cap B}=2^A \cap 2^B$。

(8) 如果$A\subseteq B$，则$2^A\subseteq 2^B$。

集合运算

集合是无序的一组元素。有时需要考虑元素的次序。有序n元组是有序的一组元素。

定义：**有序n元组**（n重组）$\langle a_1, a_2, \cdots, a_n \rangle$是以$a_1$为第一个元素，$a_2$为第二个元素，…, a_n 为第n个元素的有序组。有序二元组称为**序偶**。

两个有序n元组只有在每一对对应的元素都相等时它们才相等。

集合运算

定义：集合A和B的**笛卡尔积**是所有序偶$\langle a, b\rangle$的集合（其中$a \in A$, $b \in B$），用$A \times B$表示。

形式表示为：$A \times B = \{\langle a, b\rangle \mid a \in A \wedge b \in B\}$

例：设$A=\{1, 2\}$，$B=\{a, b, c\}$，求$A \times B$与$B \times A$。

$A \times B = \{\langle 1, a\rangle, \langle 1, b\rangle, \langle 1, c\rangle, \langle 2, a\rangle, \langle 2, b\rangle, \langle 2, c\rangle\}$

$B \times A = \{\langle a, 1\rangle, \langle a, 2\rangle, \langle b, 1\rangle, \langle b, 2\rangle, \langle c, 1\rangle, \langle c, 2\rangle\}$

集合运算

(1) $A\times B\neq B\times A$。

(2) $(A\times B)\times C\neq A\times(B\times C)$。

(3) $A\times A\neq A$。

(4) $A\times \Phi=\Phi$。

(5) $A\times(B\cup C)=(A\times B)\cup(A\times C)$。

(6) $(B\cup C)\times A=(B\times A)\cup(C\times A)$。

(7) $A\times(B\cap C)=(A\times B)\cap(A\times C)$。

(8) $(B\cap C)\times A=(B\times A)\cap(C\times A)$。

(9) $A\times(B-C)=(A\times B)-(A\times C)$。

(10) $(B-C)\times A=(B\times A)-(C\times A)$。

(11) 当A、B为有穷集时，$|A\times B|=|A|\times|B|$。

集合运算

定义：集合$A_1, A_2, \cdots, A_n$的笛卡尔积是有序n元组$\langle a_1, a_2, \cdots, a_n\rangle$的集合(其中对于i=1, 2, …, n, $a_i \in A_i$)，记为$A_1 \times A_2 \times \cdots \times A_n$ 。

形式表示为：

$$A_1 \times A_2 \times \cdots \times A_n = \{\langle a_1, a_2, \cdots, a_n\rangle \mid a_i \in A_i, i=1, 2, \cdots, n\}$$

集合运算

定理：若所有A_i（i=1, 2, …, n)都是有限集合，则

$$|A_1 \times A_2 \times \cdots \times A_n| = |A_1| \times |A_2| \times \cdots \times |A_n|。$$

预备知识

--集合

--关系

--图

--证明方法

关系

事物间存在广泛的联系。每天我们接触各种关系：亲属关系、选课关系、同学关系等。数学中有大于关系、整除关系、直线上点之间的关系等。

可用n个元素构成的n元组表示它们之间具有某种关系。给定集合上的一个关系就是具有该关系的所有元组汇集所得的集合。

二元关系

定义：设A、B是集合，一个从A到B的二元关系是$A \times B$的子集。A称为R的**前域**，B称为R的**陪域**。$D(R)=\{a \mid \exists b(\langle a, b\rangle \in R)\}$称为R的**定义域**，$R(R)=\{b \mid \exists a(\langle a, b\rangle \in R)\}$称为**R的值域**。

一个从A到B的二元关系是序偶的集合R，其中每个序偶的第一个元素取自A而第二个元素取自B。用记号aRb表示$\langle a, b\rangle \in R$。当$\langle a, b\rangle$属于R时叫做a与b有R关系。

二元关系

定义：集合A上的二元关系是从A到A的关系。

n元素的集合上有多少个二元关系？

2^{n^2}

n元关系

定义：设A_1、A_2、…、A_n是集合。$A_1 \times A_2 \times \cdots \times A_n$ 的子集称为在集合A_1、A_2、…、A_n上的**n元关系**。集合A_1、A_2、…、A_n称为关系的域，n称为关系的阶。

定义：$A \times A \times \cdots \times A$的子集称为**集合A上的n元关系**。

几种关系

定义：设R是 $\mathop{\times}\limits_{i=1}^{n} A_i$ 的子集，若R=∅，则称R为空关系，若R= $\mathop{\times}\limits_{i=1}^{n} A_i$ ，则称R为全域关系。

定义：A上的二元关系R=$\{\langle x, x\rangle \mid x\in A\}$称为相等关系，常记为$I_A$或$E_A$。

关系相等

定义：设R_1是$\mathop{\times}\limits_{i=1}^{n} A_i$上的n元关系，$R_2$是$\mathop{\times}\limits_{i=1}^{m} B_i$上的m元关系。$R_1=R_2$当且仅当n=m，且对所有i，$1\leqslant i\leqslant n$, $A_i=B_i$, 且R_1和R_2是相等的有序n元组集合。

关系的性质

定义：若对每个元素$a\in A$都有$\langle a,a\rangle\in R$，则R是**自反**的。若对每个元素$a\in A$都有$\langle a,a\rangle\notin R$，则R是**反自反**的。

形式化定义为：

A上的R是自反的$\Leftrightarrow\forall a(a\in A\rightarrow\langle a,a\rangle\in R)$

A上的R是反自反的$\Leftrightarrow\forall a(a\in A\rightarrow\langle a,a\rangle\notin R)$

关系的性质

定义：对于A中的元素a、b，若只要$\langle a, b\rangle \in R$就有$\langle b, a\rangle \in R$，则A上的关系R是**对称**的。若只有$a=b$时有$\langle a, b\rangle \in R$且$\langle b, a\rangle \in R$，则A上的关系R是**反对称**的。

形式化定义为：

A上的R是对称的

$$\Leftrightarrow \forall a \forall b (a \in A \wedge b \in A \wedge \langle a, b\rangle \in R \rightarrow \langle b, a\rangle \in R)$$

A上的R是反对称的$\Leftrightarrow \forall a \forall b (a \in A \wedge b \in A \wedge \langle a, b\rangle \in R \wedge \langle b, a\rangle \in R \rightarrow a=b)$

关系的性质

定义：对于A中的元素a、b、c，若$\langle a,b\rangle\in R$且$\langle b,c\rangle\in R$就有$\langle a,c\rangle\in R$，则A上的关系R是**传递**的。

形式化定义为：

A上的R是传递的$\Leftrightarrow\forall a\forall b\forall c(a\in A\wedge b\in A\wedge c\in A\wedge\langle a,b\rangle\in R\wedge\langle b,c\rangle\in R\rightarrow\langle a,c\rangle\in R)$

关系的合成

定义：设R是从集合A到集合B的关系，S是从集合B到集合C的关系。R和S的**合成**是由序偶$\langle a, c\rangle$汇集成的关系，其中$a\in A$、$c\in C$，且存在$b\in B$，使得$\langle a, b\rangle\in R$且$\langle b, c\rangle\in S$。R和S的合成记为$R\circ S$。

形式化定义为：

$$R\circ S=\{\langle a, c\rangle \mid a\in A\wedge c\in C\wedge \exists b(b\in B\wedge\langle a, b\rangle\in R\wedge \langle b, c\rangle\in S\}$$

关系的合成

定理：设R_1是从A到B的关系，R_2和R_3是从B到C的关系， R_4是从C到D的关系，则

1. $R_1\circ(R_2\cup R_3)=(R_1\circ R_2)\cup(R_1\circ R_3)$
2. $R_1\circ(R_2\cap R_3)\subseteq(R_1\circ R_2)\cap(R_1\circ R_3)$
3. $(R_2\cup R_3)\circ R_4=(R_2\circ R_4)\cup(R_3\circ R_4)$
4. $(R_2\cap R_3)\circ R_4\subseteq(R_2\circ R_4)\cap(R_3\circ R_4)$

关系的合成

定理：设R_1是从A到B的关系，R_2是从B到C的关系，R_3是从C到D的关系，则

$$(R_1 \circ R_2) \circ R_3 = R_1 \circ (R_2 \circ R_3)$$

关系的幂

可以用关系的合成递归定义关系的幂。

定义：设R是集合A上的关系。R的n次幂R^n，$n=1, 2, 3, \cdots$，递归地定义为$R^0 = E_A$，$R^{n+1} = R^n \circ R$。

关系的幂

定理：集合A上的关系R是传递的，当且仅当对 $n=1, 2, 3, \cdots$，有 $R^n \subseteq R$。

关系的闭包

集合A上的关系R可能具有或者不具有某种性质P，例如自反性、对称性或传递性。

定义：若存在包含R的具有性质P的关系S，并且S是包含R且具有性质P的每一个关系的子集，则S称为R的关于P的**闭包**。

若R自身已具有性质P，则R的关于P的闭包就是R。

下文将说明如何构造关系R的自反闭包r(R)、对称闭包s(R)和传递闭包t(R)。

关系的闭包

已知一个集合中的二元关系R，则r(R)(s(R)、t(R))是唯一的，它是包含R的最小的自反(对称、传递)关系。

若R是自反(对称，传递)的，则r(R)(s(R)、t(R))就是R本身。

若R不是自反(对称、传递)的，则可以补上最少的序偶，使之变为自反(对称、传递)关系，从而得到r(R)(s(R)、t(R))。

关系的闭包

定理：设R是集合S上的二元关系，则

1. R是自反的当且仅当r(R)=R。
2. R是对称的当且仅当s(R)=R。
3. R是传递的当且仅当t(R)=R。

逆关系

定义：设R是从A到B的二元关系，关系**R的逆(或R的逆关系)**记为R^{-1}或$\tilde{R}$，是从B到A的二元关系，定义为

$$R^{-1}=\{\langle y,x\rangle \mid \langle x,y\rangle \in R\}。$$

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

逆关系

定理：设R是从A到B的关系，而S是从B到C的关系，则$(R\circ S)^{-1}=S^{-1}\circ R^{-1}$。

逆关系

定理：设R、R_1和R_2都是从A到B的二元关系，则

1. $(R^{-1})^{-1}=R$
2. $(R_1\cup R_2)^{-1}=R_1^{-1}\cup R_2^{-1}$
3. $(R_1\cap R_2)^{-1}=R_1^{-1}\cap R_2^{-1}$
4. $(R_1-R_2)^{-1}=R_1^{-1}-R_2^{-1}$
5. $R_1\subseteq R_2\Rightarrow R_1^{-1}\subseteq R_2^{-1}$

逆关系

定理：设R是S上的二元关系，则R是对称的当且仅当$R=R^{-1}$。

自反闭包

定理：设R是集合A上的二元关系，则 $r(R)=R\cup E_A$。

对称闭包

定理：设R是集合A上的二元关系，则 $s(R)=R\cup R^{-1}$。

传递闭包

定理：设R是集合A上的二元关系，则 $\bigcup_{i=1}^{\infty} R^i$

t(R)= 。

定理：设 $\bigcup_{i=1}^{n} R^i$ R是集合A上的二元关系，|A|=n，则

t(R)= 。

关系的闭包

定理：

1. 若R是自反的，则s(R)和t(R)都是自反的。
2. 若R是对称的，则r(R)和t(R)都是对称的。
3. 若R是传递的，则r(R)是传递的。

关系的闭包

定理：设R是集合A上的二元关系，则

1. $rs(R)=sr(R)$
2. $rt(R)=tr(R)$
3. $ts(R)\supseteq st(R)$

等价关系

集合中某些元素可能在某一方面具有类似的性质，可以通过某种关系在这些相关元素间建立联系。例如同姓关系、三角形相似关系。

这种关系是自反的、对称的和传递的。它将集合划分成若干部分。

等价关系

定义：若集合A上的二元关系R是自反的、对称的和传递的，则称R是**等价关系**。

等价关系

定理：设R是A上的二元关系，R’=tsr(R)是R的自反对称传递闭包，则

1. R’是A上的等价关系，称为R诱导的等价关系。
2. R’是包含R的最小等价关系。

等价关系

定义：设R是集合A上的等价关系，$\forall a \in A$，令 $[a]_R=\{x \mid x \in A \wedge \langle a, x\rangle \in R\}$，则称$[a]_R$为a的关于R的**等价类**，简称为a的等价类。a称为$[a]_R$的表示元素。

不引起歧义时，可将$[a]_R$简记为$[a]$。

$[a]_R$是与a有关系R的所有元素的集合。

等价关系

定理：设R是非空集合A上的等价关系，a、b、c、$x\in A$，则

1. $[a]\neq\varnothing$且$[a]\subseteq A$；
2. 若$a\in A$，则$a\in[a]$；
3. $x\in[a]$当且仅当$\langle x,a\rangle\in R$；
4. 若$b\in[a]$且$\langle x,b\rangle\in R$，则$x\in[a]$；
5. 若$b\in[a]$且$c\in[a]$，则$\langle b,c\rangle\in R$；
6. 若$b\in[a]$，则$[b]=[a]$；
7. 若$\langle a,b\rangle\in R$，则$[a]=[b]$；
8. 若$\langle a,b\rangle\notin R$，则$[a]\cap[b]=\varnothing$；
9. $\cup\{[x]\mid x\in A\}=A$。

等价关系

定理：设R是非空集合A上的等价关系，下面的命题等价：

1. aRb
2. $[a]=[b]$
3. $[a]\cap[b]\neq\varnothing$

定理：设R是集合A上的等价关系，则对所有a、$b\in A$，或者$[a]=[b]$或者$[a]\cap[b]=\varnothing$。

等价关系

定义：设R是非空集合A上的等价关系，以关于R的全体不同等价类为元素的集合称为A关于R的**商集**，记为A/R。

等价关系

定理：设R_1和R_2是集合A上的等价关系，则$R_1=R_2$当且仅当$A/R_1=A/R_2$。

等价关系

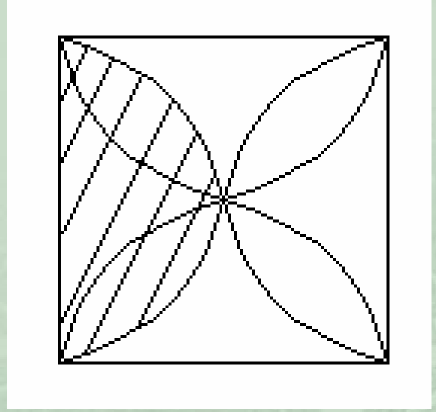
定义：给定非空集合A和集合族

$\Pi=\{A_1, A_2, \cdots, A_m\}$，若

1. $\varnothing \notin \Pi$；

2. $A=\bigcup_{i=1}^{m} A_i$

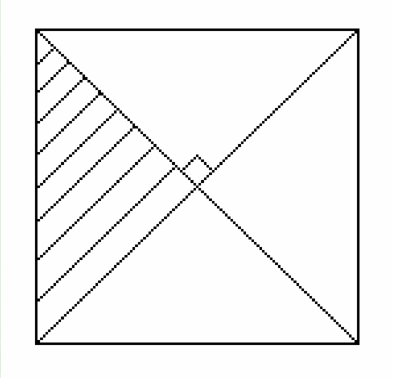
则Π称为A的覆盖。



等价关系

定义：给定非空集合A和集合族 $\Pi=\{A_1, A_2, \cdots, A_m\}$，若

1. $\varnothing \notin \Pi$；
2. $A=\bigcup_{i=1}^{m} A_i$；
3. $A_i \cap A_j=\varnothing$ 或 $A_i=A_j$



则Π称为A的**划分**。A_i称为划分Π的**块**。若划分是有限集合则块的数目称为**划分的秩**。若划分是无限集合则其秩是无限的。划分的秩即划分的大小。

等价关系

集合的划分与等价关系有深刻而本质的联系。

定理：设A是一个非空集合，R是A上的一个等价关系，则对应R的商集A/R是A的一个划分Π_R。（R诱导出Π）

定理：设A是一个非空集合，Π是A上的一个划分，令$R_\Pi=\{\langle x,y\rangle \mid x\in A\wedge y\in A\wedge x$与$y$属于$\Pi$的同一划分块$\}$，则$R_\Pi$是A上的一个等价关系。（$\Pi$诱导出R）

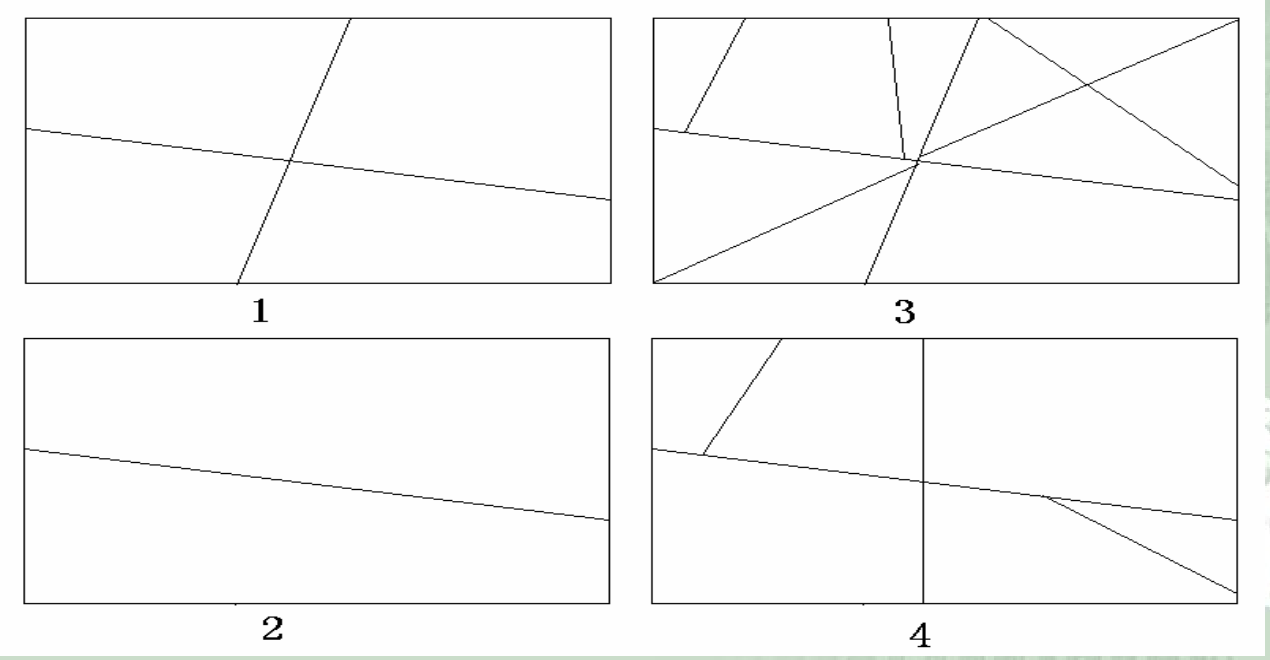
等价关系

给定A上的一个等价关系R，则R的所有不同等价类构成A的一个划分。

给定集合A的一个划分Π，存在着一个等价关系R，它以划分Π的元素作为它的等价类。

等价关系

定义：设Π和Π’都是非空集合A上的划分，若的Π’每个划分块都含于Π的某个划分块中，则称Π’是Π的细分。



等价关系

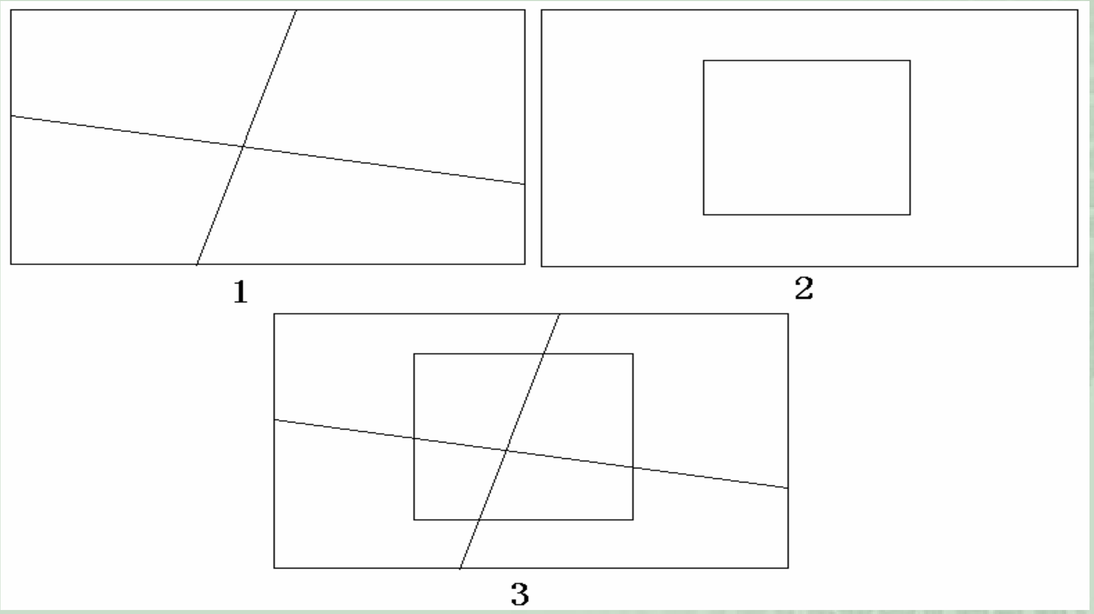
定理：设Π和Π’都是非空集合A上的划分，R和R’分别是由Π和Π’诱导的等价关系，则Π’细分Π当且仅当R’⊆R。

等价关系

定理：设F是非空集合A上划分的族，则细分关系是F上的偏序。

等价关系

定义：设Π_1和Π_2是非空集合A的划分，Π_1和Π_2的积是细分Π_1和Π_2的最小划分，记为$\Pi_1 \bullet \Pi_2$。



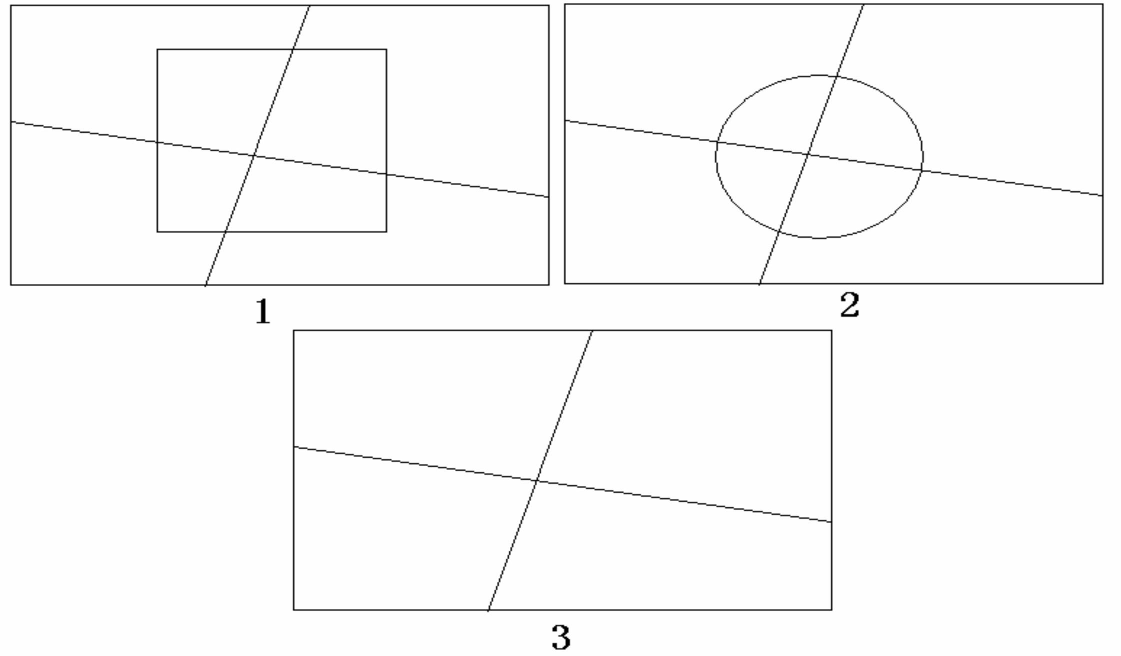
等价关系

定理：设Π_1和Π_2是非空集合A的划分，则$\Pi_1 \cdot \Pi_2$是唯一的。

定理：设R_1和R_2分别是非空集合A的划分Π_1和Π_2诱导的等价关系，则$R_1 \cap R_2$诱导出$\Pi_1 \cdot \Pi_2$ 。

等价关系

定义：设Π_1和Π_2是非空集合A的划分， Π_1和Π_2的和是Π_1和Π_2细分的最大划分，记为$\Pi_1+\Pi_2$ 。



等价关系

定理：设Π_1和Π_2是非空集合A的划分，则$\Pi_1+\Pi_2$是唯一的。

定理：设R_1和R_2分别是非空集合A的划分Π_1和Π_2诱导的等价关系，则$t(R_1\cup R_2)$诱导出$\Pi_1+\Pi_2$ 。

预备知识

--集合

--关系

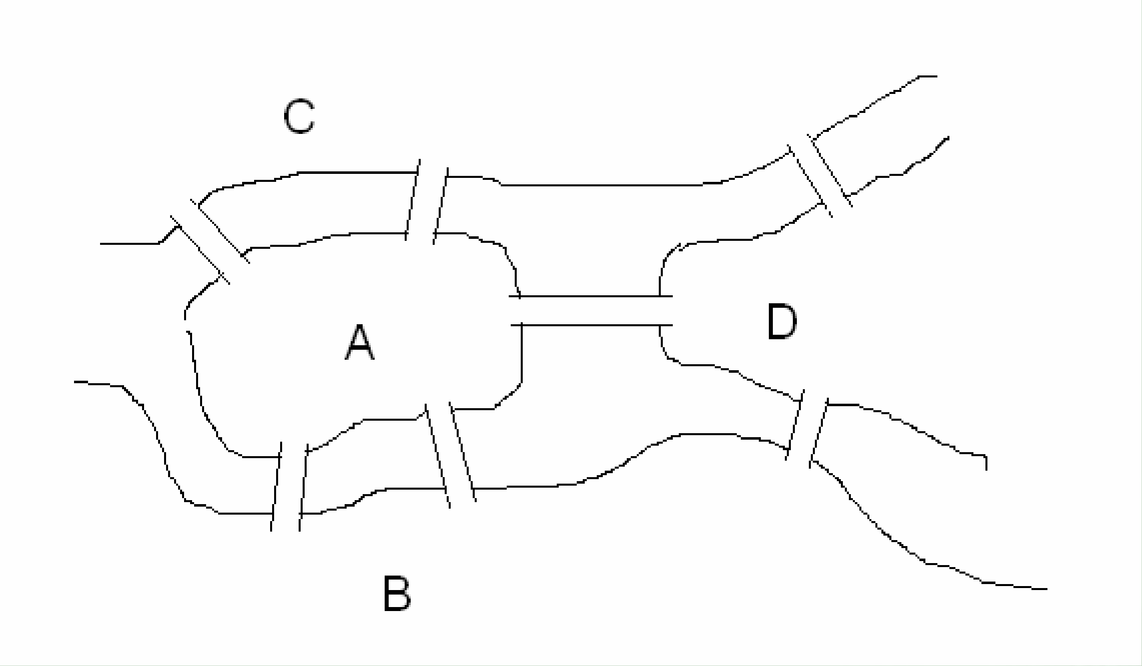
--图

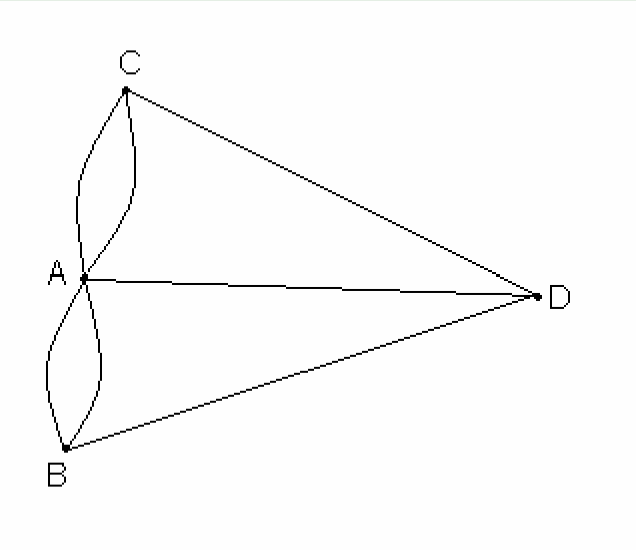
--证明方法

引子

- 源于数学游戏的难题研究：哥尼斯堡七桥问题、四色猜想、迷宫问题、哈密顿数学难题。
- 与群论、矩阵论、数值分析、概率论、拓扑学、组合学等数学分支密切有关。
- 在物理学、化学、通讯科学、计算机技术、电气和土木工程、建筑学、运筹学、生物遗传学、心理学、社会学、经济学、人类学和语言学等学科的某些领域都有应用。
- 为任何一个包含了一种二元关系的系统提供了一个数学模型。

引子





哥尼斯堡七桥问题

图

一个**图**G是一个三元组<V(G), E(G), Φ_G>，其中V(G)为有限非空结点（或称顶点、点）集合，E(G)是边（或称线）的集合，Φ_G是从边集到结点对集上的函数。

一个有p个点与q条线的图称为一个（p, q）图。（1, 0）图是平凡的。

图

若E中的边x对应结点对(u, v)，则称x**联结**u**和**v，记x=uv, 且称u和v**是邻接的点**，**点**u**与线**x**是互相关联的**。

若两条不同的线x和y与一个公共的点关联，则称x与y**是邻接的线**。

图中不与任何结点邻接的结点称为**孤立结点**。

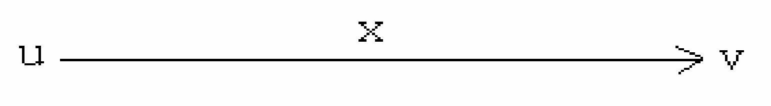
由图G**移去一个点**v产生的图G-v是由G的除v外的所有点和与v不关联的所有线组成的。

由图G**移去一条线**x产生的图G-x是由G的所有点和除x外所有线组成的。

图

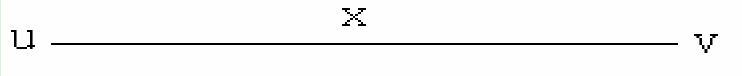
有向边（也称弧）对应有序结点对的边（在图解中带有箭头的边）

[例]



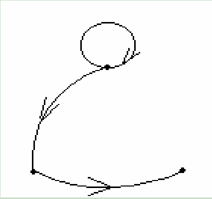
无向边（也称棱）对应无序结点对的边（在图解中不带箭头的边）。

[例]



环（也称自回路）图中联结结点到结点自身的边。

[例]



多重边 二个结点之间（方向相同）的多条边。

图

每一条边均为有向边的图称为**有向图**。

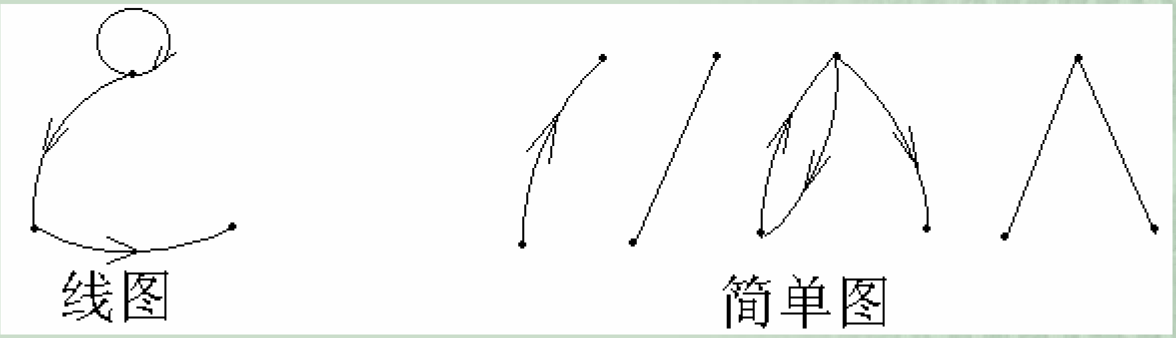
每一条边均为无向边的图称为**无向图**。

既有有向边又有无向边的图称为**混合图**。

把有向图的每条有向边都看作无向边，就得到无向图，这个无向图叫做该有向图的**底图**。

全由孤立节点构成的图称为**零图**。

含有多重边的图称为**多重图（也称伪图）**，非多重图称为**线图**。无环的线图称为**简单图**。



度

在有向图中，对于任何结点v，以v为始点的边的条数，称为结点v的**引出次数(或出度)**，记作$deg^{+}(v)$；以v点为终点的边的条数称为v的**引入次数(或入度)**，记作$deg^{-}(v)$；结点的v的引入次数和引出次数之和称为v的**次数（度数）**，用deg(v)表示。

在无向图中，点v作为G中边的端点的次数之和称为点v的**度**。

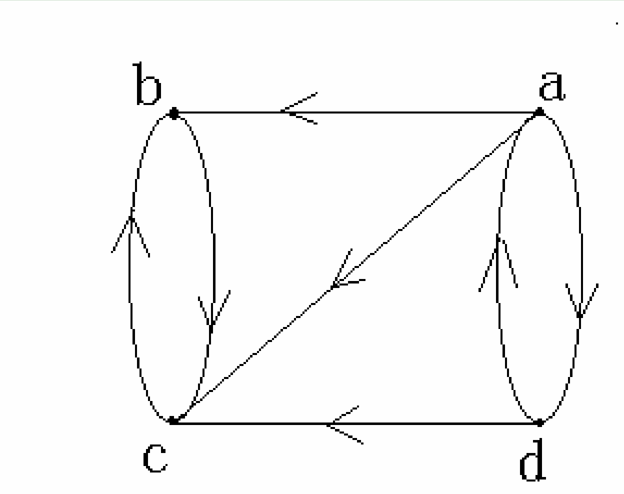
度为0的点称为**孤立结点**。度为1的点称为**端点**。

度

定理 一个图的各个点的度的和是线的数目的二倍。

[证]因为每1条线与2个点关联，所以加上1条线就使得各点的度的和增加2。

[例]



deg（a）＝4，

deg（b）＝3，

deg（c）＝4，

deg（d）＝3，

Σdeg ＝3+4+3+4＝14，

m＝7，2m＝14＝Σdeg。

度

定理 任何一个图中，度为奇数的点的数目是偶数。

在一个（p, q）简单无向图中，对每个点v，都有$0 \leqslant deg(v) \leqslant p-1$。

若图G有n个顶点, n+1条边，则G中至少有一个结点的度数≥3。

子图

设G=<V，E>, G’=<V’，E’>是两个图，

若V’$\subseteq$V， E’$\subseteq$E,则称G’是G的**子图**；

若V’$\subseteq$V， E’$\subseteq$E,并且G$\neq$G’则称G’是G的**真子图**；

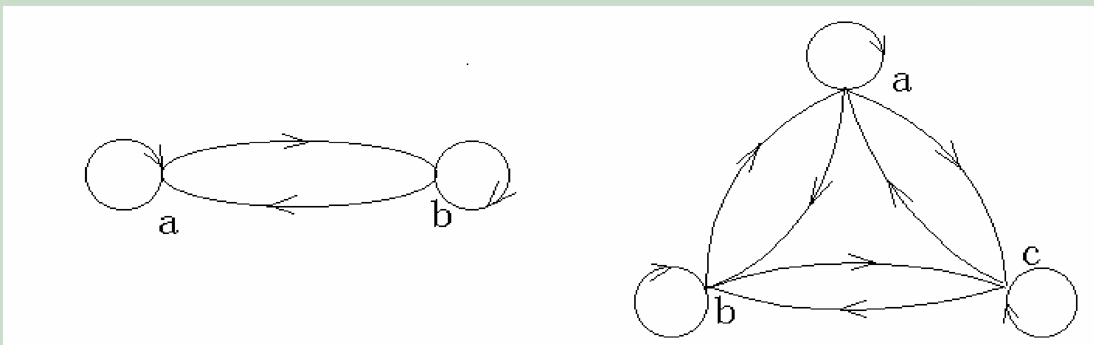
若V’=V， E’$\subseteq$E,则称G’是G的**生成子图**。

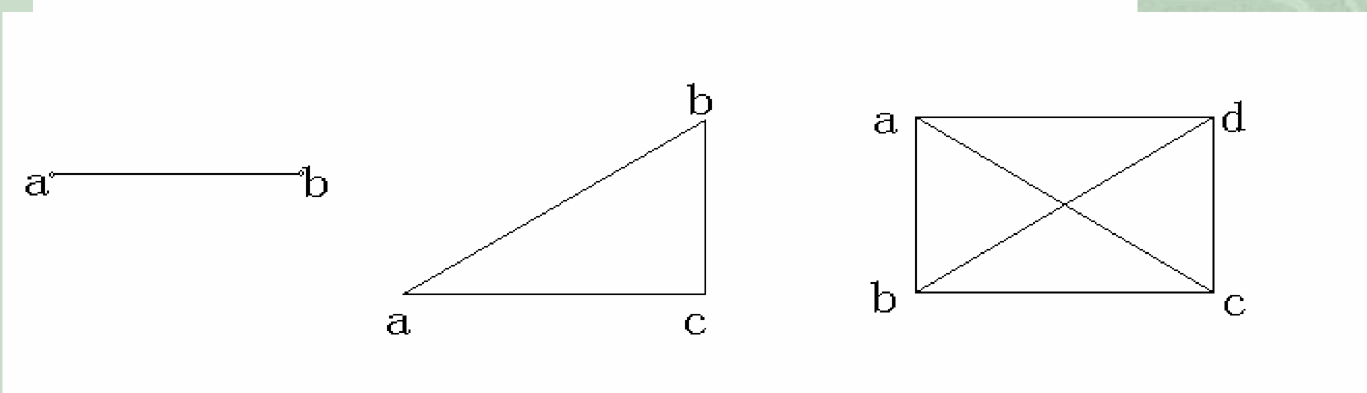
若子图G’中没有孤立结点，G’由E’唯一确定，则称G’**为由边集E’导出的子图**。

若在子图G’中，对V’中的任意二结点u和v，当[u, v]$\in$E时有[u, v]$\in$E’，则G’由V’唯一确定，此时称G’为**由结点集V’导出的子图**。

完全图

有向图G=<V,E>中，若E=V×V，则称G为**有向完全图**；无向图G=<V,E>中，若每二个不同结点之间均有一条边连接，则称G为**无向完全图**。

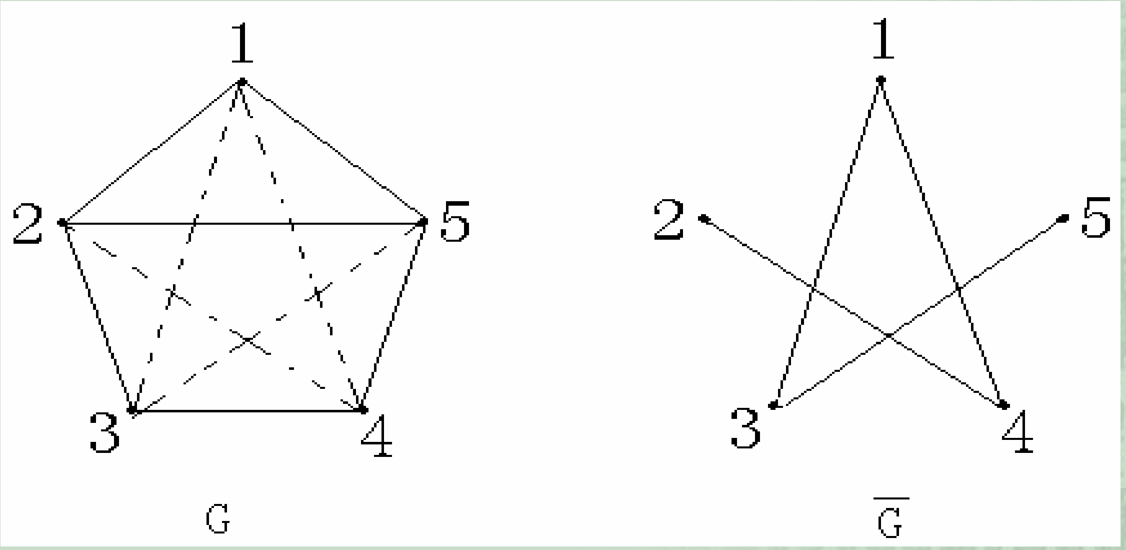




补图

设线图G= <V, E>有n个顶点，线图H= <V, E'>也有同样的顶点，且E'是由n个顶点的完全图的边删去E所得，则图H称为图G的**补图**，记为$H=\overline{G}$。

[例]



路径和回路

从结点v_0到结点v_n的一条**路径**是图的一个点边交替序列$L=v_0e_1v_1e_2v_2\ldots e_nv_n$，其中$v_{i-1}$和$v_i$分别是边$e_i$的始点和终点，i=1, 2, …, n。若路径的始点v_0和终点v_n重合，即$v_0=v_n$，则此路径称为**回路**。

在序列L中若同一条边不出现两次，则称此路径是**简单路径**。若简单路径的始点v_0和终点v_n重合，则此路径称为**简单回路**。

在序列L中若同一结点不出现两次，则称此路径是**基本路径**。若基本路径的始点v_0和终点v_n重合，则此路径称为**基本回路**（或称圈）。

[注]路径和回路可仅用边的序列表示($e_1e_2\ldots e_n$)，在非多重图时也可用结点序列表示($v_0v_1v_2\ldots v_n$)。

路径和回路

路径中所含的边的数目称为**路径的长度**。

对于G中的两个点u与v，若存在联结u和v的路径时，则其中最短路径的长度称为**u与v之间的距离**d(u, v)；若不存在联结u和v的路径，则d(u, v)=∞。

[注]对所有的点u、v和w，d(u, v)≥0并且当且仅当u=v时d(u, v)=0；d(u, v)+d(v, w)≥d(u, w)。无向图中d(u, v)=d(v, u)。

路径和回路

定理 在一个有n个结点的简单图G中，如果从u到v有一条路径，则从u到v有一条长度不大于n-1的基本路径。

定理 在一个有n个结点的简单图G中，如果经过v有一条简单回路，则经过v有一条长度不大于n的基本回路。

连通图

设G=<V, E>是图， 且u, v∈V。如果从u到v存在一条路径，则称u从v**可达**。规定v自身从v可达。

若无向图G是平凡图或G中任两结点可达，则称图G是**连通**的。若G的子图H是连通的且没有包含H的更大连通子图，则称H是G的**连通分图**（或称**连通支**）。

连通图

在有向图G中，如果任两结点至少从一个结点到另一个结点是可达的，则称图G是**单向连通**的；如果任两结点都互相可达，则称图G是**强连通**的；如果它的底图是连通的，则称图G是**弱连通**的。

[注]强连通的一定是单向连通的和弱连通的；单向连通的一定是弱连通的；弱连通的不一定是单向连通的。

连通图

在有向图G=<V,E>中,H是G的子图，若H是强连通（单向连通、弱连通）的且没有包含H的更大强连通（单向连通、弱连通）子图，则称H是G的**强（单向、弱）分图**。

连通图

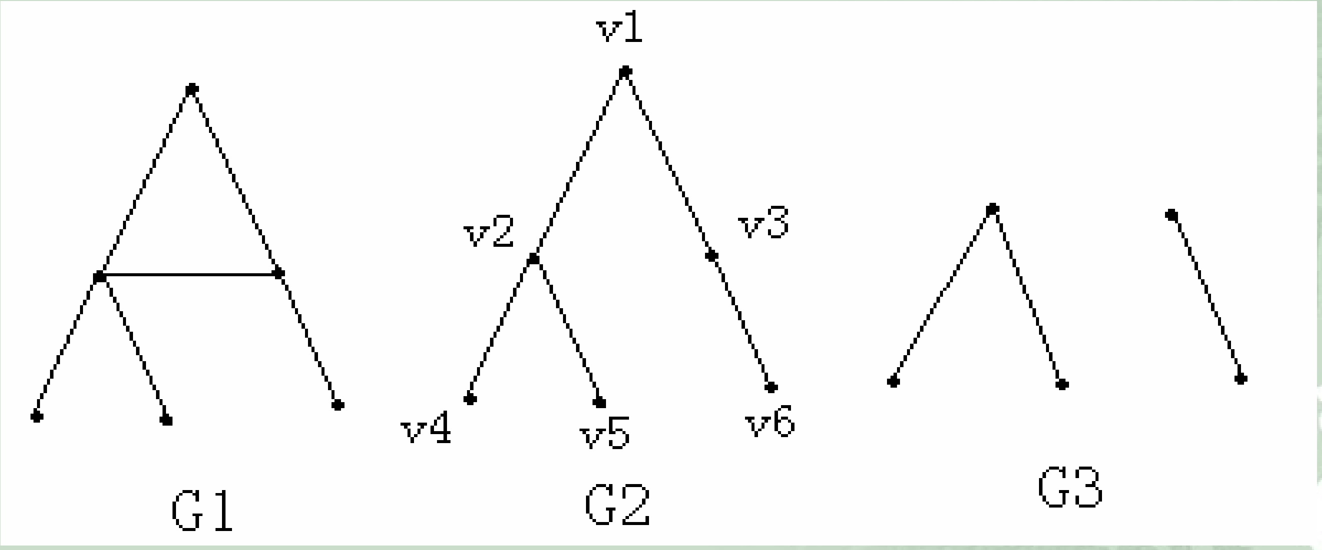
定理 在任一简单有向图G=<V,E>中，有向图的每一个结点恰好处于一个强分图（或弱分图）之中。

定理 在一个简单有向图Ｇ＝＜Ｖ，Ｅ＞中，每一个结点都处在一个或一个以上的单向分图中。

无向树

一个**树**是一个连通的无圈图。树中度为1的点称为**树叶**，度大于1的点称为**分枝点**（或内部结点）。

若一个无向图的各连通分图都是树，则称该图为**森林**。



无向树

定理 对一个(p, q)图G，下列陈述是等价的：

(1)G是一个树；

(2)G是无圈的，且p=q+1；

(3)G是连通的，且p=q+1；

(4)G是无圈的，且若G的任何两个不邻接的点联以一条线x，则G+x恰有一个圈；

(5)G是连通的，但删去任一线就不连通。

(6)G的任意两个点由唯一的一条道路联结。

无向树

定理 任何非平凡树至少有两片树叶。

［思路］显然树中每个点的度都不小于1，且所有点度之和为2p-2。假设某非平凡树T仅有一片树叶,则T中除该树叶外所有的度都不小于2。则T所有点度之和至少为2(p-1)+1，即2p-1，矛盾。

有向树的定义和性质

有向树是结点集合非空的，并符合以下条件的有向图：

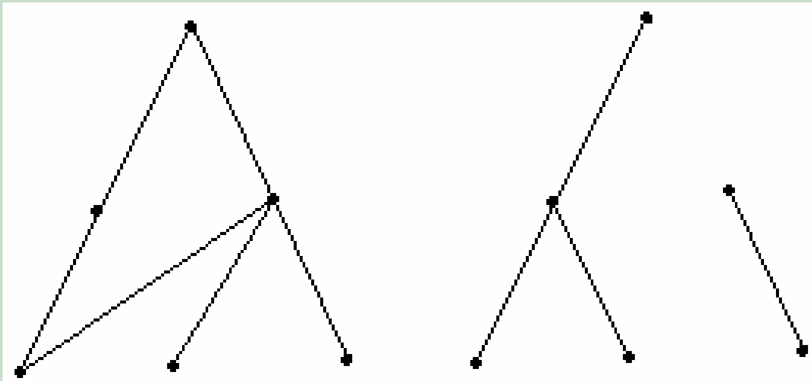
1. 有且仅有一个结点叫做树根，它的入度是0。
2. 除树根外每一结点的入度是1。
3. 树的每一结点a，都有从树根到a的一条有向路径。

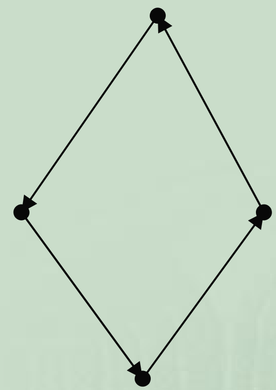
[注]有向树通常采用根在顶上，所有弧向下，弧的箭头略去的图解表示。

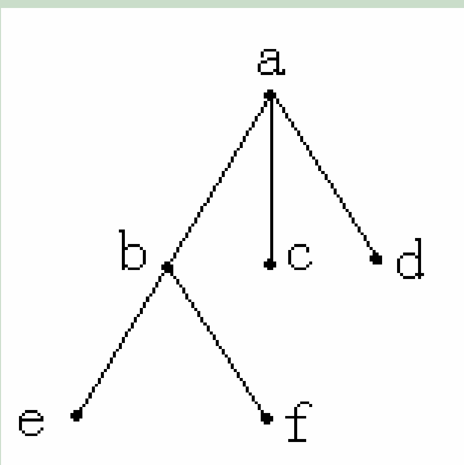
有向树的定义和性质

设a和b是有向树T的结点，如果有一弧从a到b，那么a称为b的**父亲**，而b称为a的**儿子**。父亲相同的结点称为**兄弟**。如果从结点a到结点b有一有向路径，那么a称为b的**祖先**，而b称为a的**后裔**；如果a≠b，那么a是b的一个**真祖先**而b是a的一个**真后裔**。由结点a和它的所有后裔导出的子有向图叫做**T的子树**，a叫做**子树的根**。如果a不是T的根，那么子树是T的**真子树**。出度是0的结点叫做树的**叶**；一结点若不是叶叫做**内部结点**。从树根r到一结点a的路径长度称为**a的路径长度**，亦称**a的层次**。树T中层次的最大值叫做**树T的高度**。

有向树的定义和性质







a是根，a有三个儿子b、c和d，结点b有两个儿子e和f，结点e没有儿子。结点b、c和d互为兄弟。e的父亲是b，e的真祖先是a和b。结点c、d、e和f是树的叶。a和b是内部结点。树的高度是2。结点集{b, e, f}导出的子有向图是根为b的子树。

有向树的定义和性质

引理 有向树中的每一有向路径是基本路径。

定理 有向树没有非零长度的任何回路。

定理 有向树成立公式$q=p-1$，这里q是边数，p是结点数。

定理 有向树的子树是有向树。

预备知识

--集合

--关系

--图

--证明方法

证明方法

- 演绎证明：通过列出已知为真的命题或从前面一些命题逻辑得出的命题来进行下去。
- 证明“若-则”命题：从前提开始，持续从前提和前面命题逻辑地推出命题直至推出结论是其中一个命题。
- 证明“当且仅当”命题：通过双向证明“若-则”命题来证明。

证明方法

- 证明逆否命题：通过证明 命题“若非C-则非H”来证明命题“若H-则C”。
- 反证法：通过证明命题“若H且非C-则假”来证明命题“若H-则C”。
- 反例：如果一个命题有参数，则只需给出一个反例（使命题为假的参数赋值）就可证明该命题为假。

证明方法

- 归纳证明：可用整数n上的归纳法来证明带参数n的命题。对基础（n的具体值的有穷多种情形）证明命题为真，然后证明归纳步骤：如果命题对直到n的值为真，则命题对n+1为真。
- 结构归纳法：证明关于某个递归定义结构（例如树）的定理。通过在构造所使用的步数上归纳，来证明关于所构造对象的定理。

数学归纳法

数学归纳法用来证明形如$\forall x P(x)$的命题。先看论域是正整数的情形。

用数学归纳法证明$\forall n(P(n))$包含两个步骤：

1. 基础步骤。证明命题$P(1)$为真。
2. 归纳步骤。证明对每个正整数n来说蕴涵式$P(n)\rightarrow P(n+1)$为真。

数学归纳法

表示成推理规则：

$P(1) \land \forall n(P(n) \to P(n+1)) \Rightarrow \forall n(P(n))$

首先证明P(1)为真。又因为P(1)蕴涵P(2)，故P(2)为真。又因为P(2)蕴涵P(3)，故P(3)为真。同理继续……可以得出对任意正整数n，P(n)为真。

数学归纳法

考虑一行多米诺骨牌，标记成1、2、3、…, 开始每张牌都立着。设P(n)是命题：多米诺骨牌n被推倒。若第一张牌被推倒，则P(1)为真。若第n张多米诺骨牌被推倒，则第n+1张也会被推倒，即$\forall n(P(n)\rightarrow P(n+1))$。显然，最后所有的骨牌都应被推倒，即$\forall n(P(n))$。

数学归纳法

在数学归纳法证明里并不假定对所有正整数来说P(n)为真。只是证明了：若P(n)为真，则P(n+1)为真。因此数学归纳法不属于回避问题或循环论证的情形。

数学归纳法

例：用数学归纳法证明对所有正整数n，有$n<2^n$。

证明：

(1) $n=1$时命题为真，因为$1<2^1$。

(2) 假设对n命题为真，即$n<2^n$，需要证明对$n+1$命题为真，即$n+1<2^{n+1}$。因为$n+1<n+2^n<2^n+2^n<2^{n+1}$。

由数学归纳法，对所有正整数n，有$n<2^n$。

中心概念

- 语言
- 文法
- 自动机

中心概念

- 语言
- 文法
- 自动机

语言的定义

- 词典定义：语言是一个用来表达想法、事实或概念的系统。
- 语言学家韦波斯特(Webster)：为相当大的团体的人所理解并使用的字和组合这些字的方法的统一体。

要想对语言的性质进行研究，用这些定义来建立语言的数学模型是不够精确的。必须有更形式化的定义。

语言的定义

- 语言学家乔姆斯基最初从产生语言的角度研究语言。他将语言L定义为一个字母表Σ中的字母组成的一些串的集合： $L \subseteq \Sigma^*$。可以在字母表上按照一定的规则定义一个文法，该文法所能产生的所有句子组成的集合就是该文法产生的语言。
- 克林从识别语言的角度研究语言：对于按照一定规则构造的任一自动机，该自动机就定义了一个语言，这个语言由该自动机所能识别的所有句子组成。

形式语言只研究语言的组成规则，不研究语言的含义。

语言的基本概念

- **字母表(alphabet)**
  - **字母表**是一个非空有穷集合，字母表中的元素称为该字母表的一个**字母(letter)**。又叫做**符号(symbol)**、或者**字符(character)**。
  - 非空性。
  - 有穷性。
- 例：

  {a，b，c，d}

  { a，b，c，…，z}

  {0,1}

语言的基本概念

- 字符的两个特性
  - 整体性**(monolith)**，也叫不可分性。
  - 可辨认性**(distinguishable)**，也叫可区分性。
- 例：

  {a，a′，b，b′}

  {aa，ab，bb}

  {∞，∧，∨，≥，≤}

语言的基本概念

- 字母表的乘积(**product**)

  $$\Sigma_1\Sigma_2=\{ab|a\in\Sigma_1,\ b\in\Sigma_2\}$$

- 例如：

  {0,1}{0,1}={00，01，10，00}

  {0,1}{a，b，c，d}={0a，0b，0c，0d，1a，1b，1c，1d}

  {a，b，c，d}{0,1}={a0，a1，b0，b1，c0，c1，d0，d1}

  {aa，ab，bb}{0,1}={ aa0，aa1，ab0，ab1，bb0，bb1}

语言的基本概念

- 字母表Σ的**n**次幂

  $\Sigma^0=\{\varepsilon\}$

  $\Sigma^n=\Sigma^{n-1}\Sigma$

  ε是由Σ中的**0**个字符组成的。

- Σ的正闭包

  $\Sigma^+=\Sigma\cup\Sigma^2\cup\Sigma^3\cup\Sigma^4\cup\ldots$

- Σ的克林闭包（星闭包）

  $\Sigma^*=\Sigma^0\cup\Sigma^+=\Sigma^0\cup\Sigma\cup\Sigma^2\cup\Sigma^3\cup\ldots$

语言的基本概念

- 例：

  $\{0,1\}^{+}=\{0,1,00,01,11,000,001,010,011,100,\ldots\}$

  $\{0,1\}^{*}=\{\varepsilon,0,1,00,01,11,000,001,010,011,100,\ldots\}$

- 习题：

  $\{a,b,c,d\}^{+}$

  $\{a,b,c,d\}^{*}$

语言的基本概念

- 结论：

Σ^*={x|x是Σ中的若干个，包括0个字符，连接而成的一个字符串}。

Σ^+={x|x是Σ中的至少一个字符连接而成的字符串}。

语言的基本概念

- 句子(sentence)

  Σ是一个字母表，$\forall x \in \Sigma^*$，x叫做Σ上的一个句子。

- 使用字母表中的若干符号可以构成符号串，这个符号串是字母表中符号的序列，是字母表上的句子。

- 别称：

  字(word)、(字符、符号)行(line)、(字符、符号)串(string)。

语言的基本概念

- 例：

  1、01、100都是{0, 1}上的句子。

  112不是{0, 1}上的句子。

语言的基本概念

- 句子相等

  两个句子被称为相等的，如果它们对应位置上的字符都对应相等。

语言的基本概念

- **出现(apperance)**
  - x，y∈∑*，a∈∑，句子xay中的a叫做a在该句子中的一个**出现**。
  - 当x= ε 时，a的这个出现为字符串xay的首字符
  - 如果a的这个出现是字符串xay的第n个字符，则y的首字符的这个出现是字符串xay的第n+1个字符。
  - 当y= ε 时，a的这个出现是字符串xay的尾字符

语言的基本概念

- 句子的长度(length)
  - $\forall x \in \Sigma^*$，句子x中字符出现的总个数叫做该句子的长度，记作$|x|$。
  - 长度为0的字符串叫空句子，记作ε。
- 例：

$$|abaabb|=6$$
$$|bbaa|=4$$
$$|\ \varepsilon\ |=0$$
$$|bbabaabbbaa|=11$$

语言的基本概念

- 注意
  - ε 是一个句子。
  - { ε }≠ Φ。这是因为{ ε }不是一个空集，它是含有一个空句子 ε 的集合。|{ ε }|=1，而| Φ|=0。

语言的基本概念

- 连接(concatenation)（又称并置）

  x，y∈Σ^*，x，y的连接是由串x直接相接串y所组成的，记作xy。

- 串x的n次幂

  $x^0=\varepsilon$

  $x^n=x^{n-1}x$

语言的基本概念

- 例：
  - 对x=001，y=1101

    $x^0=y^0=\varepsilon$

    $xy=0011101$

    $x^3=001001001$

    $y^4=1101110111011101$

语言的基本概念

- 习题：

对x=0101，y=110110，

x^0、y^0、x^2、y^4 、x^3y^2 =?

语言的基本概念

- Σ^*上的连接运算性质

  (1) 结合律：**(xy)z=x(yz)**。

  (2) 左消去律：如果**xy=xz**，则**y=z**。

  (3) 右消去律：如果**yx=zx**，则**y=z**。

  (4) 惟一分解性：存在惟一确定的a_1，a_2，…，$a_n \in \Sigma$，使得$x= a_1a_2\ldots a_n$。

  (5) 单位元素：$\varepsilon x=x\varepsilon=x$。

语言的基本概念

- **前缀与后缀**

  设x，y，z，w，v∈Σ^*，且x=yz，w=yv

  (1) y是x的前缀(prefix)。

  (2)如果z≠ε，则y是x的真前缀(proper prefix)。

  (3) z是x的后缀(suffix)；

  (4) 如果y≠ε，则z是x的真后缀(proper suffix)。

语言的基本概念

- **公共前缀与后缀**

(5) y是x和w的公共前缀(common Prefix)。

(6)如果x和w的任何公共前缀都是y的前缀，则y是x和w的最大公共前缀。

(7) 如果x=zy，w=vy，则y是x和w的公共后缀(common suffix)。

(8)如果x和w的任何公共后缀都是y的后缀，则y是x和w的最大公共后缀。

语言的基本概念

- 例 ：

字母表Σ={a，b}上的句子abaabb的前缀、后缀、真前缀和真后缀如下：

前缀：ε，a，ab，aba，abaa，abaab，abaabb

真前缀：ε，a，ab，aba，abaa，abaab

后缀：ε，b，bb，abb，aabb，baabb，abaabb

真后缀：ε，b，bb，abb，aabb，baabb

语言的基本概念

- 结论

(1) x的任意前缀y有惟一的一个后缀z与之对应，使得x=yz；反之亦然。

(2) x的任意真前缀y有惟一的一个后缀z与之对应，使得x=yz；反之亦然。

(3) $|\{w|w是x的后缀\}|=|\{w|w是x的前缀\}|$。

(4) $|\{w|w是x的真后缀\}|=|\{w|w是x的真前缀\}|$。

(5) $\{w|w是x的前缀\}=\{w|w是x的真前缀\}\cup\{x\}$，

$|\{w|w是x的前缀\}|=|\{w|w是x的真前缀\}|+1$。

语言的基本概念

- 结论

(6) {w|w是x的后缀}={w|w是x的真后缀}∪{x}，

|{w|w是x的后缀}|=|{w|w是x的真后缀}|+1。

(7) 对于任意字符串w，w是自身的前缀，但不是自身的真前缀；w是自身的后缀，但不是自身的真后缀。

(8) 对于任意字符串w，ε是w的前缀，且是w的真前缀；ε是w的后缀，且是w的真后缀。

语言的基本概念

- 约定

(1) 用小写字母表中较为靠前的字母a，b，c，…表示字母表中的字母。

(2) 用小写字母表中较为靠后的字母x，y，z，…表示字母表上的句子。

(3) 用x^T表示x的倒序。例如，如果x=abc，则x^T=cba。

语言的基本概念

- **子串(substring)**

  w，x，y，z$\in\Sigma^*$，且**w=xyz**，则称**y**是**w**的**子串。**

- **公共子串(common substring)**
  - **t，u，v，w，x，y，z**$\in\Sigma^*$，且**t=uyv，w=xyz，** 则称**y**是**t**和**w**的**公共子串(common substring)**。如果y_1，y_2，……，y_n是**t**和**w**的公共子串，且**max{|**y_1**|，|**y_2**|，…，|**y_n**|}=|**y_j**|**，则称y_j是**t**和**w**的**最大公共子串。**
  - 两个串的最大公共子串并不一定是惟一的。

语言的基本概念

- 语言(language)
  $\forall L \subseteq \Sigma^*$，L称为字母表$\Sigma$上的一个语言(language)，$\forall x \in L$，x叫做L的一个句子。
- 例：{0，1}上的不同语言
  {00，11}，{0，1}
  {0，1，00，11}，{0，1，00，11，01，10}
  $\{0^n \mid n \geq 1\}$
  $\{x \mid x \in \Sigma^+$且x中0和1的个数相同$\}$

语言的基本概念

一种语言通常定义为Σ*上的子集。语言中的一个符号串称为这个语言的一个句子。字母表上符号串的任意集合都可以看成是一种语言。

语言的基本概念

- 语言是集合，所以可以类似地定义语言的并、交和差。
- 在全集Σ^*上定义语言**L**的补集为：

  $\overline{L} = \Sigma^* - L$

- 语言的**乘积(product)**

  $L_1 \subseteq \Sigma_1^*$，$L_2 \subseteq \Sigma_2^*$，语言L_1与L_2的**乘积**是一个语言，该语言定义为：$L_1L_2=\{xy \mid x \in L_1, y \in L_2\}$，是字母表$\Sigma_1 \cup \Sigma_2$上的语言。

语言的基本概念

- **幂**

  $\forall L \subseteq \Sigma^*$，L的n次**幂**是一个语言，该语言定义为

  (1) 当**n=0**是，$L^n=\{\varepsilon\}$。

  (2) 当$n \geqslant 1$时，$L^n= L^{n-1}L$ 。

- **正闭包**

  $L^+=L \cup L^2 \cup L^3 \cup L^4 \cup \ldots$

- **克林闭包**

  $L^*= L^0 \cup L \cup L^2 \cup L^3 \cup L^4 \cup \ldots$

语言的基本概念

例：{0，1}上的不同语言

$\{00, 11\}^*$，$\{01, 10\}^*$，$\{00, 01, 10, 11\}^*$，

$\{0\}\{0, 1\}^*\{1\}$，$\{0, 1\}^*\{111\}\{0, 1\}^*$

语言的基本概念

- 例：

(1) $L_1=\{0, 1\}$。

(2) $L_2=\{00, 01, 10, 11\}$。

(3) $L_3=\{0, 1, 00, 01, 10, 11, 000, \cdots\}=\Sigma^+$。

(4) $L_4=\{\varepsilon, 0, 1, 00, 01, 10, 11, 000, \cdots\}=\Sigma^*$。

(5) $L_5=\{0^n \mid n\geqslant 1\}$。

(6) $L_6=\{0^n1^n \mid n\geqslant 1\}$。

(7) $L_7=\{1^n \mid n\geqslant 1\}$。

(8) $L_8=\{0^n1^m \mid n, m\geqslant 1\}$。

(9) $L_9=\{0^n1^n0^n \mid n\geqslant 1\}$。

(10) $L_{10}=\{0^n1^m0^k \mid n, m, k\geqslant 1\}$。

(11) $L_{11}=\{x \mid x\in\Sigma^+$且$x$中0和1的个数相同$\}$。

(12) $L_{12}=\{0^n1^m0^k \mid n, m, k\geqslant 0\}$。

语言的基本概念

- 上述几个语言的部分特点及相互关系：

  上述所有语言都是L_4的子集(子语言)；

  L_1，L_2是有穷语言；其他为无穷语言；其中L_1是Σ上的所有长度为1的句子组成的语言，L_2是Σ上的所有长度为2的句子组成的语言；

  L_3，L_4分别是Σ的正闭包和克林闭包；

  $L_5L_7 \neq L_6$，但$L_5L_7 = L_8$；同样$L_9 \neq L_{10}$，但是我们有：$L_6 \subset L_5L_7$，$L_9 \subset L_{10}$。

语言的基本概念

$L_6=\{0^n1^n|n\geq 1\}$中的句子中的0和1的个数是相同的，并且所有的0在所有的1的前面，$L_{11}=\{x|x\in\Sigma^+$且x中0和1的个数相同$\}$中的句子中虽然保持着0的个数和1的个数相等，但它并没要求所有的0在所有的1的前面。例如，0101，$1100\in L_{11}$，但是$0101\notin L_6$，$1100\notin L_6$。而对$\forall x\in L_6$，有$x\in L_{11}$。所以，$L_6\subset L_{11}$。

语言的基本概念

$L_1 \subset L_{12}$， $L_2 \subset L_{12}$

$L_5 \subset L_{12}$， $L_6 \subset L_{12}$

$L_7 \subset L_{12}$， $L_8 \subset L_{12}$

$L_9 \subset L_{12}$， $L_{10} \subset L_{12}$

$L_1 \not\subset L_{10}$， $L_2 \not\subset L_{10}$

$L_5 \not\subset L_{10}$， $L_6 \not\subset L_{10}$

$L_7 \not\subset L_{10}$， $L_8 \not\subset L_{10}$

$L_9 \subset L_{10}$， $L_{10} \subset L_{12}$

语言的基本概念

- 例：

(1) $\{x \mid x=x^T,\ x\in\Sigma^+\}$。

(2) $\{xx^T \mid x\in\Sigma^+\}$。

(3) $\{xx^T \mid x\in\Sigma^*\}$。

(4) $\{xwx^T \mid x,\ w\in\Sigma^+\}$。

(5) $\{xx^Tw \mid x,\ w\in\Sigma^+\}$。

中心概念

- 语言
- 文法
- 自动机

文法

- 对于Σ上的语言L，它的具体组成结构是什么样的？
- 一个给定的字符串是否为一个给定语言的句子?如果不是，它在结构的什么地方出了错?进一步地，这个错误是什么样的错？如何更正？……

文法

- 用文法作为相应语言的有穷描述，不仅可以描述出语言的结构特征，而且还可以产生这个语言的所有句子。

启示

- 文法的概念最早由语言学家们在研究自然语言理解中完成形式化。
- 归纳如下句子的描述：

(1) 哈尔滨是美丽的城市。

(2) 北京是祖国的首都。

(3) 集合是数学的基础。

(4) 形式语言是很抽象的。

(5) 教育走在社会发展的前面。

(6) 中国进入WTO。

启示

- 6个句子的主体结构

  <名词短语><动词短语><句号>

  <名词短语>={哈尔滨，北京，集合，形式语言，教育，中国}

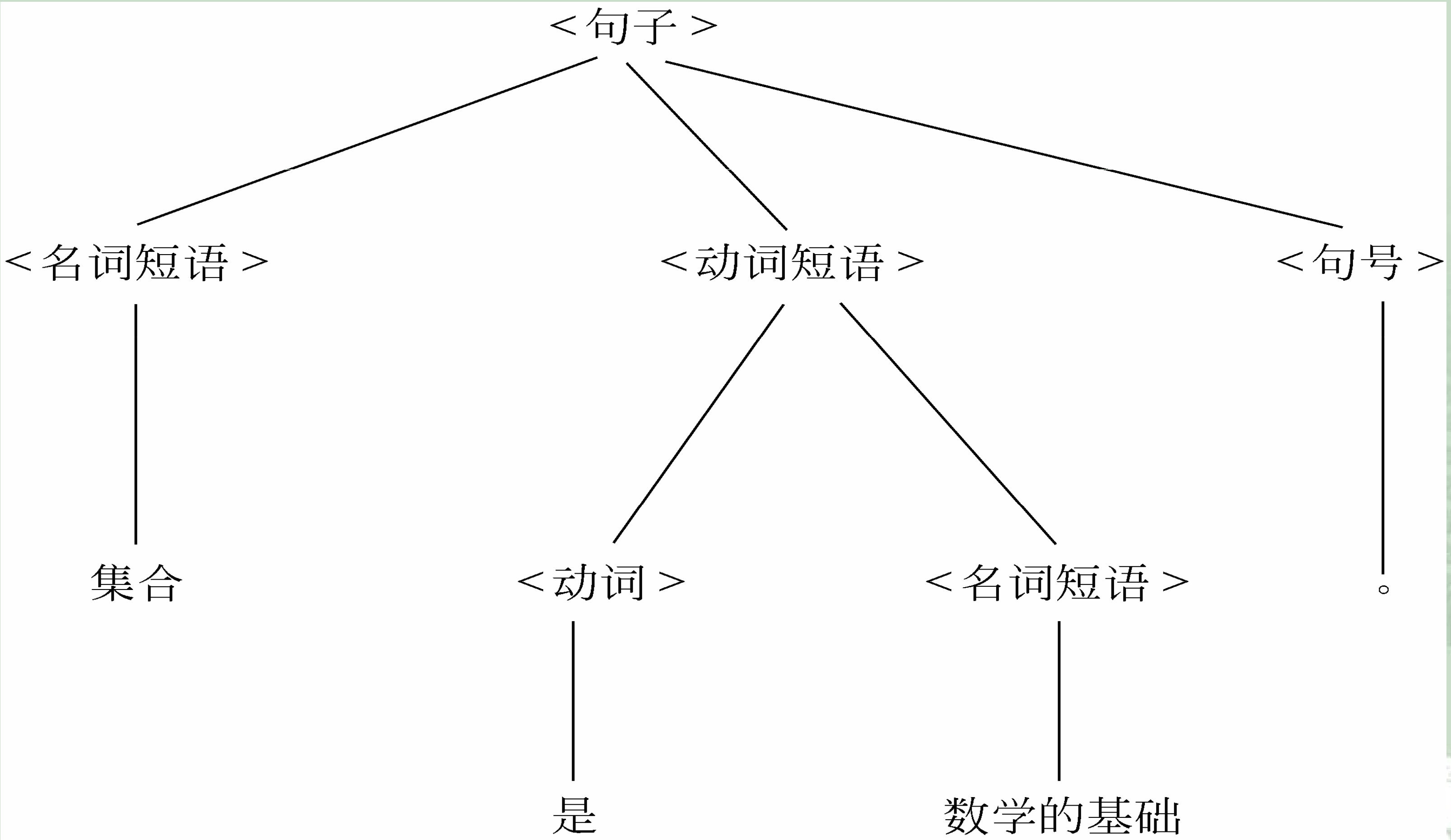
  <动词短语>={是美丽的城市，是祖国的首都，是数学的基础，是很抽象的，走在社会发展的前面，进入WTO}

  <句号>={。}

启示

- <动词短语>可以是<动词><形容词短语> 或者<动词><名词短语> 。
- <名词短语>={北京、哈尔滨、形式语言、中国、教育、集合、WTO、美丽的城市、祖国的首都、数学的基础、社会发展的前面}。
- <动词>={是、走在、进入}。
- <形容词短语>={很抽象的}。
- 把<名词短语><动词短语><句号>取名为<句子>。

启示



启示

- 表示成 α → β 形式

  <句子>→<名词短语><动词短语><句号>

  <动词短语>→<动词><形容词短语>

  <动词短语>→<动词><名词短语>

  <动词>→是

启示

<动词>→走在

<动词>→进入

<形容词短语>→很抽象的

<名词短语>→北京

<名词短语>→哈尔滨

<名词短语>→形式语言

启示

<名词短语>→中国

<名词短语>→教育

<名词短语>→集合

<名词短语>→ WTO

<名词短语>→美丽的城市

<名词短语>→祖国的首都

<名词短语>→数学的基础

<名词短语>→社会发展的前面

<句号>→。

启示

- 表示一个语言，需要4种东西

(1)形如<名词短语>的“符号”

它们表示相应语言结构中某个位置上可以出现的一些内容。每个“符号”对应的是一个集合，在该语言的一个具体句子中，句子的这个位置上能且仅能出现相应集合中的某个元素。所以，这种“符号”代表的是一个语法范畴。

(2) <句子>

所有的“规则”，都是为了说明<句子>的结构而存在，相当于说，定义的就是<句子>。

启示

(3)形如北京的“符号”

它们是所定义语言的合法句子中将出现的“符号”。仅仅表示自身，称为终极符号。

(4)所有的“规则”都呈 $\alpha \rightarrow \beta$ 的形式

在产生语言的句子中被使用，称这些“规则”为产生式。

形式定义

- 文法(grammar)G是一个四元组G=(V，T，P，S)，其中：
  - V——为变量(variable)的非空有穷集。∀A∈V，A叫做一个语法变量(syntactic Variable)，简称为变量，也可叫做非终极符号(nonterminal)。它表示一个语法范畴(syntactic Category)。

形式定义

- **T——为终极符(terminal)**的非空有穷集。$\forall$**a**$\in$**T**，**a**叫做终极符。由于**V**中变量表示语法范畴，**T**中的字符是语言的句子中出现的字符，所以，有**V**$\cap$**T**= Φ。
- **S——S**$\in$**V**，为文法**G**的**开始符号(start symbol)**。

形式定义

- **P——**为**产生式(production)**的非空有穷集合。**P**中的元素均具有形式α→β，被称为产生式，读作：α定义为β。其中α∈**(V∪T)**$^{+}$，且α中至少有**V**中元素的一个出现。β∈**(V∪T)***。α称为产生式α→β的**左部**，β称为产生式α→β的**右部**。产生式又叫做定义式或者语法规则。

形式定义

- 产生式规则是文法的核心，它们指出文法如何把一个符号串转化为另一个符号串。通过这个过程，产生式规则定义了一个和这个文法相关的语言。

形式定义

- 例：　以下四元组都是文法。

(1) ({A}，{0，1}，{A→01，A→0A1，A→1A0}，A)。

(2) ({A}，{0，1}，{A→0，A→0A}，A)。

(3) ({A，B}，{0，1}，{A→01，A→0A1，A→1A0，B→AB，B→0}，A)。

(4) ({A，B}，{0，1}，{A→0，A→1，A→0A，A→1A }，A)。

形式定义

(5) ({S， A， B， C， D}, {a， b， c， d, #}，{S→ABCD， S→abc#， A→aaA， AB→aabbB，BC→bbccC， cC→cccC， CD→ccd#， CD→d#，CD→#d}， S)。

(6) ({S}, {0， 1}， {S→00S， S→11S， S→00，S→11}， S)。

形式定义

- 约定

  (1) 对一组有相同左部的产生式

  $\alpha \to \beta_1$， $\alpha \to \beta_2$， …， $\alpha \to \beta_n$

  可以简单地记为：

  $\alpha \to \beta_1 | \beta_2 | \cdots | \beta_n$

  读作：α 定义为 β_1，或者 β_2，…，或者 β_n。并且称它们为 α 产生式。β_1，β_2，…，β_n 称为候选式(candidate)。

形式定义

(2) 使用符号

- 英文字母表较为前面的大写字母，如A，B，C，…表示语法变量；
- 英文字母表较为前面的小写字母，如a，b，c，…表示终极符号；
- 英文字母表较为后面的大写字母，如X，Y，Z，…表示该符号是语法变量或者终极符号；
- 英文字母表较为后面的小写字母，如x，y，z，…表示由终极符号组成的行；
- 希腊字母α，β，γ…表示由语法变量和终极符号组成的行

形式定义

- 例：四元组是否满足文法的要求。
  - ({A, B, C, E}, {a, b, c}, {S→ABC|abc, D→e|a, FB→c, A→A, E → abc| ε }, S)
  - 4种修改 ：

    (1) ({A, B, C, E, S, D, F}, {a, b, c, e}, {S→ABC|abc, D→e|a, FB→c, A→A, E → abc| ε }, S)。

    (2) ({A, B, C, E, S }, {a, b, c}, {S→ABC|abc, A→A, E → abc| ε }, S)。

    (3) ({A, B, C, E}, {a, b, c}, { A→A, E → abc| ε }, A)。

    (4) ({A, B, C, E}, {a, b, c}, { A→A, E → abc| ε }, E)。

形式定义

- 推导(derivation)

  设G=(V，T，P，S)是一个文法，若 $\alpha\rightarrow\beta\in P$， γ、$\delta\in(V\cup T)^*$，则称 $\gamma\alpha\delta$ 在G中直接推导出 $\gamma\beta\delta$，记作：$\gamma\alpha\delta\Rightarrow_G\gamma\beta\delta$，读作：$\gamma\alpha\delta$ 在文法G中直接推导出 $\gamma\beta\delta$。

  “直接推导”可以简称为推导，也称为派生。

形式定义

- 归约(reduction)

  若$\gamma\alpha\delta \Rightarrow_G \gamma\beta\delta$，则称$\gamma\beta\delta$在文法G中直接归约成$\gamma\alpha\delta$。

  在不特别强调归约的直接性时，“直接归约”可以简称为归约。

形式定义

- 应用文法产生式，我们可以以任意顺序推导出连续的符号串。如果$\alpha_1 \Rightarrow \alpha_2 \Rightarrow \cdots \Rightarrow \alpha_n$，我们就称$\alpha_1$推导出$\alpha_n$，记为$\alpha_1 \Rightarrow_G^n \alpha_n$。

形式定义

$\Rightarrow_G^n$、$\Rightarrow_G^+$、$\Rightarrow_G^*$可当成（V∪T)*上的二元关系。

(1) $\alpha \Rightarrow_G^n \beta$：表示α在G中经过n步推导出β；β在G中经过n步归约成α。即，存在α_1，α_2，…，$\alpha_{n-1} \in (V \cup T)^*$使得$\alpha \Rightarrow_G \alpha_1$，$\alpha_1 \Rightarrow_G \alpha_2$，…，$\alpha_{n-1} \Rightarrow_G \beta$。

(2) 当n=0时，有α=β。即$\alpha \Rightarrow_G^0 \alpha$。

(3) $\alpha \Rightarrow_G^+ \beta$：表示α在G中经过至少1步推导出β；β在G中经过至少1步归约成α。

(4) $\alpha \Rightarrow_G^* \beta$：表示α在G中经过若干步推导出β；β在G中经过若干步归约成α。

意义清楚时，可分别用$\Rightarrow$、$\Rightarrow^+$、$\Rightarrow^*$、$\Rightarrow^n$代替$\Rightarrow_G$、$\Rightarrow_G^+$、$\Rightarrow_G^*$、$\Rightarrow_G^n$。

形式定义

- 例：　设G=({A}，{a}，{A→a|aA}，A)

$A \Rightarrow a\underline{A}$　　使用产生式A→aA

$\Rightarrow aa\underline{A}$　　使用产生式A→aA

$\Rightarrow aaa\underline{A}$　　使用产生式A→aA

$\Rightarrow aaaa\underline{A}$　　使用产生式A→aA

…

$\Rightarrow a \cdots a\underline{A}$　　使用产生式A→aA

$\Rightarrow a \cdots aa$　　使用产生式A→a

形式定义

$A \Rightarrow a\underline{A}$	使用产生式A→aA
$\Rightarrow aa\underline{A}$	使用产生式A→aA
$\Rightarrow aaa\underline{A}$	使用产生式A→aA
$\Rightarrow aaaa\underline{A}$	使用产生式A→aA
…	
$\Rightarrow a\ldots a\underline{A}$	使用产生式A→aA
$\Rightarrow a\ldots aaA$	使用产生式A→aA

形式定义

$AAaaA\underline{A}A \Rightarrow A\underline{A}aaAaAA$ 使用产生式$A\rightarrow aA$

$\Rightarrow AaAaaAa\underline{A}A$ 使用产生式$A\rightarrow aA$

$\Rightarrow \underline{A}aAaaAaaA$ 使用产生式$A\rightarrow a$

$\Rightarrow aaAaaAaa\underline{A}$ 使用产生式$A\rightarrow a$

$\Rightarrow aa\underline{A}aaAaaa$ 使用产生式$A\rightarrow a$

$\Rightarrow aaa\underline{A}aaAaaa$ 使用产生式$A\rightarrow aA$

$\Rightarrow aaaaaa\underline{A}aaa$ 使用产生式$A\rightarrow a$

$\Rightarrow aaaaaaaaaa$ 使用产生式$A\rightarrow a$

形式定义

- 例：

设G=({S，A，B}，{0，1}，{S→A|AB，A→0|0A，B→1|11}，S)

对于$n \geq 1$，

$A \Rightarrow^{n} 0^{n}$　　首先连续n-1次使用产生式A→0A，最后使用产生式A→0；

$A \Rightarrow^{n} 0^{n}A$　　连续n次使用产生式A→0A；

$B \Rightarrow 1$　　使用产生式B→1；

$B \Rightarrow 11$　　使用产生式B→11。

形式定义

语法范畴A代表的集合L(A)={0，00，000，0000，……}={$0^n \mid n \geq 1$}；

语法范畴B代表的集合L(B)={1，11}

语法范畴S代表的集合L(S)=L(A)∪L(A)L(B)

={0，00，000，0000，…}∪{0，00，000，0000，…}{1，11}

={0，00，000，0000，…}∪

∪{01，001，0001，00001，…}∪

∪{011，0011，00011，000011，…}

形式定义

例：　设G=({A}，{0，1}，{A→01，A→0A1}，A)，

$A \Rightarrow^{n} 0^{n}A1^{n}$　　$n \geq 0$

$0^{n}A1^{n} \Rightarrow 0^{n+1}A1^{n+1}$　　$n \geq 0$

$0^{n}A1^{n} \Rightarrow 0^{n+1}1^{n+1}$　　$n \geq 0$

$0^{n}A1^{n} \Rightarrow^{i} 0^{n+i}A1^{n+i}$　　$n \geq 0，i \geq 0$

$0^{n}A1^{n} \Rightarrow^{i} 0^{n+i}1^{n+i}$　　$n \geq 0，i \geq 0$

$0^{n}A1^{n} \Rightarrow^{*} 0^{m}A1^{m}$　　$n \geq 0，m \geq n$

$0^{n}A1^{n} \Rightarrow^{+} 0^{m}1^{m}$　　$n \geq 0，m \geq n+1$

$0^{n}A1^{n} \Rightarrow^{+} 0^{m}A1^{m}$　　$n \geq 0，m \geq n+1$

$0^{n}A1^{n} \Rightarrow^{+} 0^{m}1^{m}$　　$n \geq 0，m \geq n+1$

形式定义

- 几点结论
  - 对任意的$x \in \Sigma^+$，我们要使语法范畴D代表的集合为$\{x^n|n \geq 0\}$，可用产生式组$\{D \to \varepsilon|xD\}$来实现。
  - 对任意的x，$y \in \Sigma^+$，我们要使语法范畴D代表的集合为$\{x^n y^n|n \geq 1\}$，可用产生式组$\{D \to xy|xDy\}$来实现。
  - 对任意的x，$y \in \Sigma^+$，我们要使语法范畴D代表的集合为$\{x^n y^n|n \geq 0\}$，可用产生式组$\{D \to \varepsilon|xDy\}$来实现。

形式定义

以不同的顺序应用产生式规则，一个给定的文法可以生成许多符号串，其中，所有从开始符号推导出的、仅由终极符构成的符号串的集合就是这个文法定义或生成的语言。

形式定义

- **语言(language)**

  $L(G)=\{w \mid w\in T^*$ 且 $S \Rightarrow^* w\}$

- **句子(sentence)**

  $\forall w\in L(G)$，w称为G产生的一个**句子**。

- **句型(sentential form)**

  G=(V，T，P，S)，对于$\forall \alpha \in (V\cup T)^*$，如果$S \Rightarrow^* \alpha$，则称α是G产生的一个**句型**。

形式定义

- 句子w是从S开始，在G中可以推导出来的终极符号行，它不含语法变量。
- 句型α是从S开始，在G中可以推导出来的符号行，它可能含有语法变量。
- 例：给定文法G=({S，A，B，C，D},{a，b，c，d,#}，{S→ABCD|abc#，A→aaA，AB→aabbB，BC→bbccC，cC→cccC，CD→ccd#，CD→d#，CD→#d}，S)

形式定义

$\underline{S} \Rightarrow \underline{A}BCD$	使用产生式$S \rightarrow ABCD$
$\Rightarrow aa\underline{A}BCD$	使用产生式$A \rightarrow aaA$
$\Rightarrow aaaa\underline{AB}CD$	使用产生式$A \rightarrow aaA$
$\Rightarrow aaaaaabb\underline{BC}D$	使用产生式$AB \rightarrow aabbB$
$\Rightarrow aaaaaabbbbc\underline{cC}D$	使用产生式$BC \rightarrow bbccC$
$\Rightarrow aaaaaabbbbcccc\underline{CD}$	使用产生式$cC \rightarrow cccC$
$\Rightarrow aaaaaabbbbcccc\#d$	使用产生式$CD \rightarrow \#d$

形式定义

S ⇒ ABCD	使用产生式**S→ABCD**
⇒AbbccCD	使用产生式**BC→bbccC**
⇒aaAbbccCD	使用产生式**A→aaA**
⇒aaAbbccccd#	使用产生式**CD→ccd#**
⇒aaaaAbbccccd#	使用产生式**A→aaA**
⇒aaaaaaAbbccccd#	使用产生式**A→aaA**
⇒aaaaaaaaAbbccccd#	使用产生式**A→aaA**

形式定义

- **例：** 构造产生标识符的文法

G=({<标识符>，<大写字母>，<小写字母>，<阿拉伯数字>}，{0，1，…，9，A，B，C，…，Z，a，b，c，…，z}，P, <标识符>)

P={ <标识符>→<大写字母>|<小写字母> ,

<标识符>→<标识符><大写字母>|<标识符><小写字母>|<标识符><阿拉伯数字> ,

<大写字母>→A|B|C|D|E|F|G|H|I|J|K|L|M|N|O,

<大写字母>→ P|Q|R|S|T|U|V|W|X|Y|Z ,

<小写字母>→a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z,

<阿拉伯数字>→0|1|2|3|4|5|6|7|8|9 }

形式定义

G′ =({<标识符>，<头>，<尾>}，{0，1，2，…，9，A，B，C，…，Z，a，b，c，…，z}，P′，<标识符>)

P′ ={<标识符>→<头><尾> ，

<头>→A|B|C|D|E|F|G|H|I|J|K|L|M|N

<头>→ O|P|Q|R|S|T|U|V|W|X|Y|Z ，

<头>→a|b|c|d|e|f|g|h|i|j|k|l|m|n

<头>→ o|p|q|r|s|t|u|v|w|x|y|z ，

形式定义

<尾>→ ε |0<尾>|1<尾>|2<尾>|3<尾>|4<尾>|5<尾>,

<尾>→ 6<尾>|7<尾>|8<尾>|9<尾> ,

<尾>→A<尾>|B<尾>|C<尾>|D<尾>|E<尾>|F<尾>,

<尾>→ G<尾>|H<尾>|I<尾>|J<尾>|K<尾> ,

<尾>→L<尾>|M<尾>|N<尾>|O<尾>|P<尾>|Q<尾>,

<尾>→ R<尾>|S<尾>|T<尾>|U<尾>|V<尾> ,

<尾>→ W<尾>|X<尾>|Y<尾>|Z<尾>|a<尾>|b<尾>,

<尾>→ c<尾>|d<尾>|e<尾>|f<尾>|g<尾> ,

<尾>→h<尾>|i<尾>|j<尾>|k<尾>|l<尾>|m<尾>,

<尾>→ n<尾>|o<尾>|p<尾>|q<尾>|r<尾> ,

<尾>→s<尾>|t<尾>|u<尾>|v<尾>|w<尾>|x<尾>

<尾>→ y<尾>|z<尾> }

形式定义

<标识符> ⇒<标识符><阿拉伯数字>

⇒ <标识符>3

⇒ <标识符><阿拉伯数字>3

⇒ <标识符>23

⇒ <标识符><小写字母>23

⇒ <标识符>n23

⇒ <标识符><阿拉伯数字>n23

⇒ <标识符>8n23

⇒ <标识符><小写字母>8n23

⇒ <标识符>d8n23

⇒ <小写字母>d8n23

⇒ id8n23

形式定义

〈标识符〉⇒〈头〉〈尾〉

⇒i〈尾〉

⇒id〈尾〉

⇒id8〈尾〉

⇒id8n〈尾〉

⇒id8n2〈尾〉

⇒id8n23〈尾〉

⇒id8n23

形式定义

- 为了证明某个语言L是由文法G生成的，我们必须证明：
  - 每个L中的句子w都可以使用G中的产生式由S推导出；
  - 每个使用G中产生式由S推导出的符号串都是L中的句子。

文法的构造

- 例：构造文法G，使L(G)={0，1，00，11}
  - 将文法的开始符号定义为这4个句子。
    - G_1=({S}，{0，1}，{S→0，S→1，S→00，S→11}，S)
  - 先用变量A表示0，用变量B表示1。
    - G_2=({S，A，B}，{0，1}，{S→A，S→B，S→AA，S→BB，A→0，B→1}，S)
  - 基于G_2，考虑“规范性”问题。
    - G_3=({S，A，B}，{0，1}，{S→0，S→1，S→0A，S→1B，A→0，B→1}，S)

文法的构造

- 可以在V、T、P中增加一些元素，以获得“不同的”文法。
  - G_4=({S，A，B，C}，{0，1，2}，{S→A，S→B， S→AA， S→BB，A→0，B→1}，S)
  - G_5=({S，A，B，C}，{0，1，2}，{S→A，S→B，S→AA，S→BB，A→0，B→1，CACS→21，C→11，C→2}，S)

$$L(G_1) = L(G_2) = L(G_3) = L(G_4) = L(G_5)$$

文法的构造

一个语言可以由多个文法生成。尽管这些文法是有区别的，但是它们在某种程度上是等价的。

文法的构造

- 等价(equivalence)

  设有两个文法G_1和G_2，如果$L(G_1)= L(G_2)$，则称G_1与G_2等价。

- 约定

  对一个文法，只列出该文法的所有产生式，且所列第一个产生式的左部是该文法的开始符号。

文法的构造

G_1： $S \to 0 \mid 1 \mid 00 \mid 11$

G_2： $S \to A \mid B \mid AA \mid BB,\ A \to 0,\ B \to 1$

G_3： $S \to 0 \mid 1 \mid 0A \mid 1B,\ A \to 0,\ B \to 1$

G_4： $S \to A \mid B \mid AA \mid BB,\ A \to 0,\ B \to 1$

G_5： $S \to A \mid B \mid BB,\ A \to 0,\ B \to 1,\ CACS \to 21,\ C \to 11,\ C \to 2$

文法的构造

- 例：
  - $L=\{0^n \mid n \geqslant 1\}$

    G_6： $S \to 0 \mid 0S$
  - $L=\{0^n \mid n \geqslant 0\}$

    G_7： $S \to \varepsilon \mid 0S$
  - $L=\{0^{2n}1^{3n} \mid n \geqslant 0\}$

    G_8： $S \to \varepsilon \mid 00S111$

文法的构造

- 例：构造文法G_9，使$L(G_9)=\{w|w\in\{a,b,\cdots,z\}^+\}$。

  用$S\to A|AS$生成 A^n。

  用$A\to a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z$从A产生可能出现在该位置的符号。

- G_9：$S\to A|AS$，

  $A\to a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z$

文法的构造

- 不可以用A→a|b|c|…|z表示
  A→a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z
- 不可以用A→a^8表示A→aaaaaaaa。
- 不能用A→a^n 表示A可以产生任意多个a。

文法的构造

- 例：构造文法G_{10}，使$L(G_{10})=\{ww^T|w\in\{0，1，2，3\}^+\}$。

文法：

$S\rightarrow HE$

$H\rightarrow 0|1|2|3|0H|1H|2H|3H$

$E\rightarrow 0|1|2|3|E0|E1|E2|E3$

难以生成$L(G_{10})$。

文法的构造

- $\{ww^T | w\in\{0，1，2，3\}^+\}$中句子的特点：
  设$w=a_1a_2\cdots a_n$，从而有$w^T= a_n\cdots a_2\ a_1$，
  故$ww^T=a_1a_2\cdots a_na_n\cdots a_2a_1$，
  满足$f(ww^T，i)=f(ww^T，|ww^T|-i+1)$。
  递归地定义L：
  (1)对$\forall a\in\{0，1，2，3\}$，$aa\in L$；
  (2)如果$x\in L$，则对$\forall a\in\{0，1，2，3\}$，$axa\in L$；
  (3)L中不含不满足(1)、(2)任何其他的串。

文法的构造

根据递归定义中的第一条，有如下产生式组：

$S \to 00 \mid 11 \mid 22 \mid 33$

再根据递归定义第二条，又可得到如下产生式组：

$S \to 0S0 \mid 1S1 \mid 2S2 \mid 3S3$

从而，

G_{10}：$S \to 00 \mid 11 \mid 22 \mid 33 \mid 0S0 \mid 1S1 \mid 2S2 \mid 3S3$

文法的构造

- 例：构造文法G_{12}，使$L(G_{12})=\{w|w$是我们习惯的十进制有理数$\}$。

  G_{12}：$S \to R|+R|-R$

  $R \to N|B$

  $B \to N.D$

  $N \to 0|AM$

  $D \to 0|MA$

  $A \to 1|2|3|4|5|6|7|8|9$

  $M \to \varepsilon\ |0M|1M|2M|3M|4M|5M|6M|7M|8M|9M$

文法的构造

- 例：构造产生算术表达式的文法：

(1)基础：常数是算术表达式，变量是算术表达式；

(2)归纳：如果E_1、E_2是表达式，则 $+E_1$、$-E_1$、E_1+E_2、E_1-E_2 、E_1*E_2 、E_1/E_2、E_1**E_2、$Fun(E_1)$是算术表达式。其中Fun为函数名。

(3)只有满足(1)和(2)的才是算术表达式。

G_{13}：$E \rightarrow id \mid c \mid +E \mid -E \mid E+E \mid E-E \mid E*E \mid E/E \mid E**E \mid Fun(E)$

文法的构造

- 例：构造产生语言$\{a^n b^n c^n | n \geq 1\}$的文法。

文法$G=(\{K_1\},\{a,b\},\{K_1 \to ab | aK_1b\},K_1)$

产生的语言$\{a^n b^n | n \geq 1\}$形式上看起来与语言$\{a^n b^n c^n | n \geq 1\}$比较接近。

$G=(\{K_2\},\{c\},\{K_2 \to c | cK_2\},K_2)$产生的语言是$\{c^n | n \geq 1\}$。

文法的构造

文法G：

$K \rightarrow K_1K_2$

$K_1 \rightarrow ab \mid aK_1b$

$K_2 \rightarrow c \mid cK_2$

不能产生语言

$\{a^nb^nc^n \mid n \geqslant 1\}$

而是产生语言

$\{a^nb^n \mid n \geqslant 1\}\ \{c^n \mid n \geqslant 1\}$

$\{a^nb^nc^n \mid n \geqslant 1\} \neq \{a^nb^n \mid n \geqslant 1\}\ \{c^n \mid n \geqslant 1\}$

文法的构造

文法G：$K \rightarrow abc \mid aKbc$

产生的语言为：

$$\{a^n(bc)^n \mid n \geq 1\}$$

焦点：交换b和c的位置。

文法的构造

G_{14}： $K \rightarrow aBC \mid aKBC$，

$CB \rightarrow BC$

$aB \rightarrow ab$

$bB \rightarrow bb$

$bC \rightarrow bc$

$cC \rightarrow cc$

G_{14}' ： $K \rightarrow abc \mid aKBc$，

$bB \rightarrow bb$

$cB \rightarrow Bc$

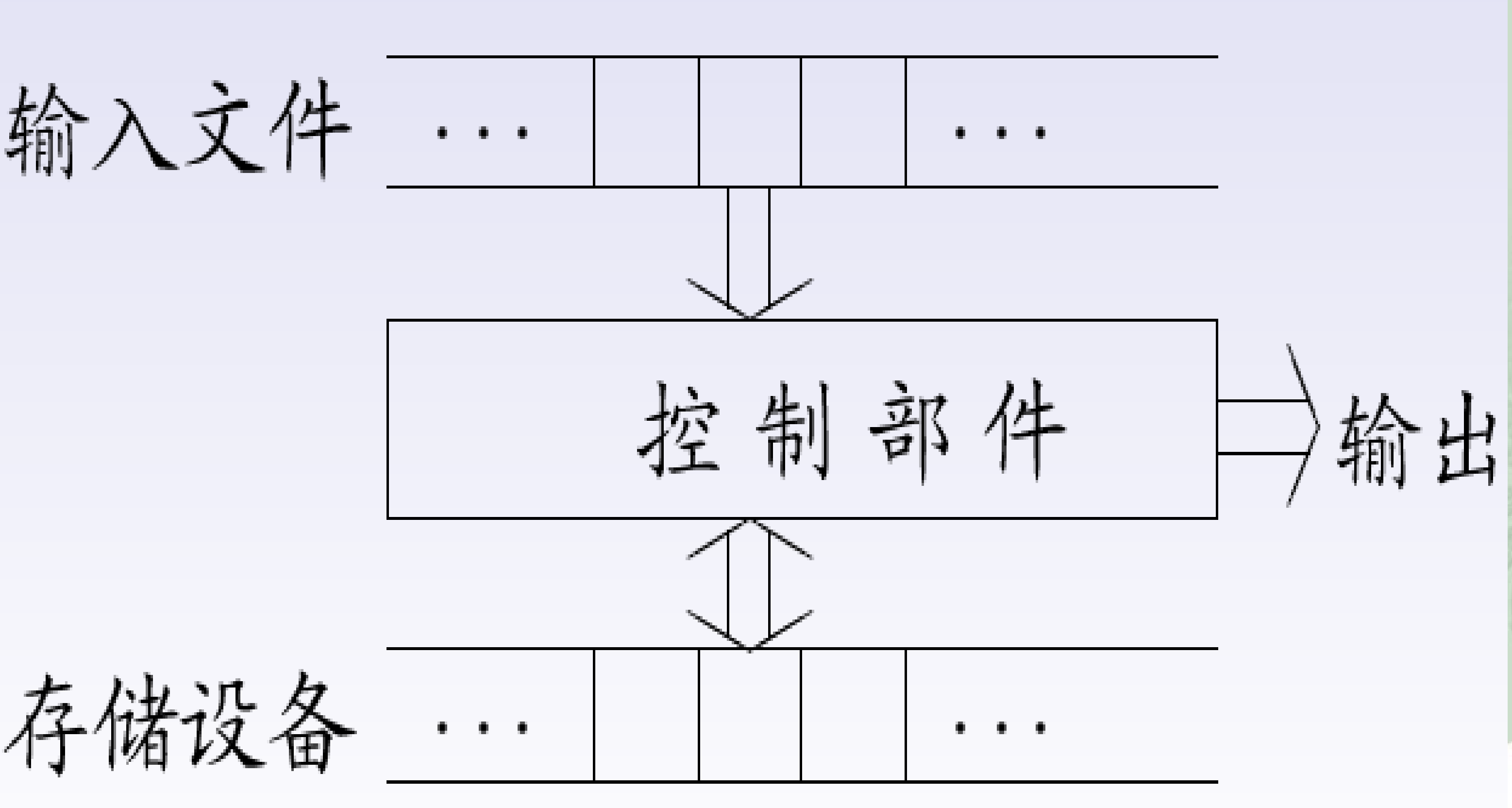
中心概念

- 语言
- 文法
- 自动机

自动机

- 自动机是数字计算机的抽象模型，包含一些本质特征：
  - 输入装置　　　读头+输入带
  - 控制部件　　　状态转换
  - 存储单元
  - 输出装置

自动机



自动机

- 自动机是在离散时间框架上操作的。
- 在任意给定的时间上，控制部件处于某个内部状态，读入装置正在搜索输入文件上的一个特定符号。
- 下一时刻，控制部件会处于哪个内部状态由转换函数决定。转换函数根据当时的状态、读入的符号和临时存储空间中的信息，来决定下一状态。

自动机

- 从一个时间间隔向下一时间间隔转换的过程中，自动机会产生输出，或者改变临时存储空间中的信息。
- 格局这个术语是控制部件、输入文件和临时存储空间的某一特定状态。自动机从一个格局到另一个格局的转换称为一个迁移。

自动机

- 自动机处于某个状态，接受一定的输入，输出一定的结果，执行一定的动作，转换到下一状态。使用转换函数描述整个工作过程。
- 自动机的本质：根据当前格局和规则决定下一个格局。

自动机

- 可能的状态、运行的规则都是事先确定的。一旦开始运行，就按照事先确定的规则工作，因此叫“自动机”。

自动机的分类

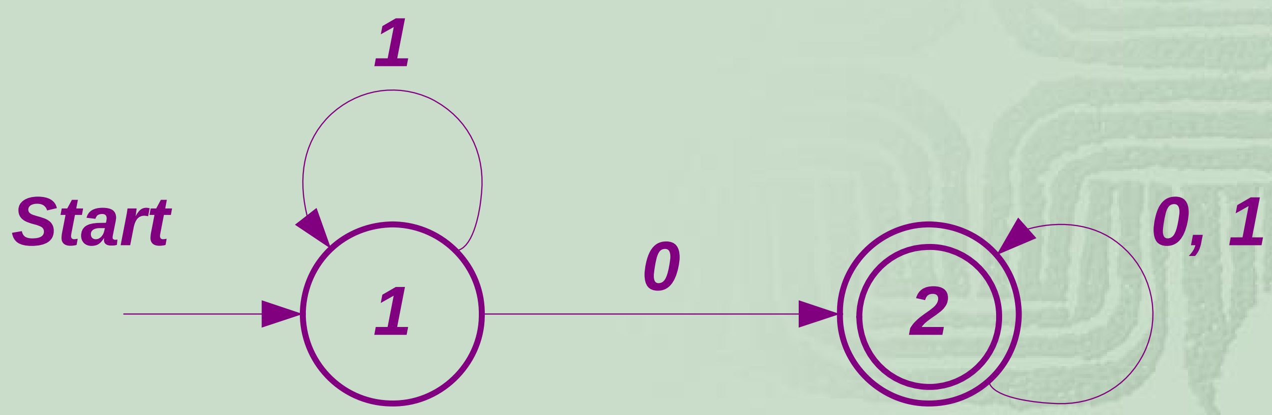
- 根据结构不同，自动机又可分为有限自动机、下推自动机、线性有界自动机、图灵机等。
- 图灵机的机制是非常简单的，但是足以解决非常复杂的问题。
- 使用自动机，可以形式化地描述现实世界中的一些问题。

语言、文法与自动机

- 形式语言和自动机是密切相关的。
  - 形式语言 —— 符号串的集合
  - 自动机 —— 符号串集合的识别系统

语言、文法与自动机

- 设 $\Sigma = \{ 0, 1 \}$， $L = \{ w \mid w$ 中至少有一个0 $\}$， 如 0011、10、110111$\in$L， 而11、ε、1111 $\notin$ L。
- 下图是一个可接受该语言的有限状态自动机:



语言、文法与自动机

- 形式语言描述的途径：
- 集合：列出语言中的句子或描述句子特征等。
- 文法：生成语言中的句子。
- 自动机：接受或识别语言中的句子。

语言、文法与自动机

- 文法是定义语言的一个数学模型，而自动机可看作是语言的识别系统。
- 通过对一些定理的证明，说明对于一个文法产生的语言，可以构造相应自动机接受该语言；一个自动机接受的语言，可以构造对应的文法产生该语言。一定类型的自动机和某种类型的文法具有等价性。

语言、文法与自动机

图灵机⇔递归可枚举语言

线性有界自动机⇔上下文相关语言

下推自动机⇔上下文无关语言

有限自动机⇔正则语言

语言、文法与自动机

语言、文法和自动机中蕴含的递归性可能是它们之间联系的根源。

习题

1 设Σ={a, b}, 求字符串aaabba的所有前缀的集合，后缀的集合，真前缀的集合和真后缀的集合。

2 设Σ={0, 1}, 请给出Σ上的下列语言的尽可能形式化的表示。

(1)所有以0开头，以1结尾的串。

(2)所有长度为奇数的串。

(3)所有含有3个连续0的串。

(4)所有正数第7个字符是0的串。

习题

3 设文法G的产生式集如下，试给出句子id+id*id的两个不同推导和两个不同归约。

$E \rightarrow id \mid c \mid +E \mid -E \mid E+E \mid E-E \mid E*E \mid E/E \mid E**E \mid Fun(E)$

4 设 $\Sigma=\{0,1\}$,请给出Σ上的下列语言的文法。

(1)所有以0开头，以1结尾的串。

(2)所有长度为奇数的串。

(3)所有以3个连续0结尾的串。

(4)所有含有子串010的串。

习题

5 设 $\Sigma=\{a, b, c\}$, 请给出Σ上的下列语言的文法。

(1) $L_1=\{a^nb^n \mid n\geqslant 0\}$。

(2) $L_2=\{a^nb^m \mid n, m\geqslant 1\}$。

(3) $L_3=\{a^nb^na^n \mid n\geqslant 1\}$。

(4) $L_4=\{awa \mid a\in \Sigma, w\in \Sigma^+\}$。

(5) $L_5=\{xwx^T \mid x, w\in \Sigma^+\}$。

(6) $L_6=\{w \mid w=w^T, w\in \Sigma^+\}$。