Acronyms

Use the template *acronym.tex* together with the Springer document class SVMono (monograph-type books) or SVMult (edited books) to style your list(s) of abbreviations or symbols in the Springer layout.

Lists of abbreviations, symbols and the like are easily formatted with the help of the Springer-enhanced `description` environment.

suffices to 就可以...,e.g. In order to check whether a nonempty word belongs to L, it suffices to verify that its first letter is in P

BABI Spelled-out abbreviation and definition

CABR Spelled-out abbreviation and definition

Symbols

$\in$	$x \in S$	x is an element of set S
$\notin$	$x \notin S$	x is not an element of set S
$\subseteq$	$A \subseteq B$	A is a subset of B
$\mathbb{N}$	$\{0,1,2,\ldots\}$	Set of non-negative integers
$\mathbb{N}^+$	$\{1,2,\ldots\}$	Set of positive integers
$\mathbb{N}_k$	$\{1,2,\ldots,k\}$	Index set
$\mathbb{Z}$	$\{\ldots,-2,-1,0,1,2,\ldots\}$	Set of integers
$\mathbb{Q}$	$\{p/q \mid p \in \mathbb{Z}, q \in \mathbb{Z}, q \neq 0\}$	Set of rational numbers
$\mathbb{Q}^+$		Set of rational positive numbers
$\mathbb{R}$		Set of real numbers
$\mathbb{C}$		Set of complex numbers
$\Leftrightarrow$	$\ldots \Leftrightarrow$ —	… if and only if —
$\Rightarrow$	$\ldots \Rightarrow$ —	… implies —

Chapter 1
automata abstract

P P ¶ P $\mathscr{P}$ P P P **P**

[WATSON93a, p6] **the signatures of the transition relations:**
$T \in \mathbb{P}(Q \times V \times Q)$
$T \in V \to P(Q \times Q)$
$T \in Q \times Q \to P(V)$
$T \in Q \times V \to P(Q)$
$T \in Q \to P(V \times Q)$
for example, the function $T \in Q \to P(V \times Q)$ is defined as $T(p) = \{(a,q) : (p,a,q) \in T\}$

ε-transition relation:
$E \in P(Q \times Q)$
$E \in Q \to P(Q)$

$T \in P(Q \times V \times Q), T = \{(s,a,q)\}$
$T(s) \in Q \to P(V \times Q), T(s) = \{(a,q) : (s,a,q) \in T\}$
$Q_{map} : P(Q \times V), Q_{map} = \{(q,a) : (s,a,q) \in T\}$

$Q_{map}(q) = \{a : (s,a,q) \in T\}$
$Q_{map}{}^{-1} : V \nrightarrow P(Q), Q_{map}{}^{-1} = \{(a,q) : (s,a,q) \in T\}$
According to [WATSON93a, Convention A.4] (Tuple projection):
$\bar{\pi}_2(T) = \{(s,q) : (s,a,q) \in T\}$
$Q_{map} = (\bar{\pi}_1(T))^R, Q_{map} = \{(a,q) : (s,a,q) \in T\}^R = \{(q,a) : (s,a,q) \in T\}$

$f(a) = (f(a^R))^R$

Definition 1.1 (Prefix-closure[Chrison2007]). Let $L \subseteq V^*$,then

$$\overline{L} := \{s \in V^* : (\exists t \in V^*)[st \in L]\}$$

In words, the prefix closure of L is the language denoted by $\overline{L}$ and consisting of all the prefixes in L. In general, $L \subseteq \overline{L}$.

L is said to be prefix-closed if $L = \overline{L}$. Thus language L is prefix-closed if any prefix of any string in L is also an element of L.

$L_1 = \{\varepsilon, a, aa\}, L_1 = \overline{L_1}, L_1$ is prefix-closed.
$L_2 = \{a, b, ab\}, \overline{L_2} = \{\varepsilon, a, b, ab\}, L_2 \subset \overline{L_2}, L_2$ is not prefix closed.

Definition 1.2 (Post-closure[Chrison2007]). Let $L \subseteq V^*$ and $s \in L$. Then the post-language of L after s, denoted by L/s, is the language

$$L/s := \{t \in V^* : st \in L\}$$

By definition, $L/s = \emptyset$ if $s \notin \overline{L}$.

Definition 1.3 (Left derivatives[WATSON93a]). Given language $A \subseteq V^*$ and $w \in V^*$ we define the left derivative of A with respect to w as:

$$w^{-1}A = \{x \in V^* : wx \in A\}$$

A 关于 w 的左导数，就是 A 中：{w 的后缀组成的字符串集合}。

Sometimes derivatives are written as $D_w A$ or as $\frac{dA}{dw}$. Right derivatives are analogously defined. Derivatives can also be extended to $B^{-1}A$ where B is also a language.

Example 1.1. $A=\{a,aab,baa\}, a^{-1}A=D_aA=\frac{dA}{da}=\{\varepsilon,ab,\emptyset\}=\{\varepsilon,ab\}$ □

Example 1.2. $L=\{ba,baa,baab,ca\}, w=\{ba\},$

$w^{-1}L=\{\varepsilon,a,ab,\emptyset\}=\{\varepsilon,a,ab\}$

$(wa)^{-1}L=(baa)^{-1}L=\{\emptyset,\varepsilon,b,\emptyset\}=\{\varepsilon,b\}$

$a^{-1}(w^{-1}L)=a^{-1}\{\varepsilon,a,ab\}=\{\emptyset,\varepsilon,b\}=\{\varepsilon,b\}$

$w\in L\equiv\varepsilon\in w^{-1}L, and (wa)^{-1}L=a^{-1}(w^{-1}L)$ □

Example 1.3. $a^{-1}\{a\}=\{\varepsilon\};\quad a^{-1}\{b\}=\emptyset,\quad \Leftarrow if(a\neq b)$ □

Example 1.4. $L_0=\{ab\}, L_1=\{ac\}, L_0L_1=\{abac\}$

$a^{-1}(L_0L_1)=\{bac\}$

$a^{-1}(L_0L_1)=(a^{-1}L_0)L_1\cup\emptyset \quad \Leftarrow(\varepsilon\notin L_0)$

$=\{b\}L_1=\{bac\}$ □

Example 1.5. $L_0=\{\varepsilon,ab\}, L_1=\{ac\}, L_0L_1=\{ac,abac\}$

$a^{-1}(L_0L_1)=\{c,bac\}$

$a^{-1}(L_0L_1)=(a^{-1}L_0)L_1\cup a^{-1}L_1 \quad \Leftarrow(\varepsilon\in L_0)$

$=\{\emptyset,b\}L_1\cup\{c\}=\{c,bac\}$ □

证明. $a^{-1}(L_0L_1)$

$1. if(\varepsilon\in L_0)\Rightarrow a^{-1}(L_0L_1)=(a^{-1}L_0)L_1\cup a^{-1}L_1$

$L_0=(L_0\backslash\{\varepsilon\})\cup\{\varepsilon\}$

$a^{-1}(L_0L_1)=a^{-1}(((L_0\backslash\{\varepsilon\})\cup\{\varepsilon\})L_1)$

$=a^{-1}(L_0L_1\cup L_1)$

$a^{-1}L_0=a^{-1}((L_0\backslash\{\varepsilon\})\cup\{\varepsilon\})$

$=a^{-1}(L_0\backslash\{\varepsilon\})\cup a^{-1}\{\varepsilon\}$

$=a^{-1}L_0\cup\emptyset=a^{-1}L_0$

From [Hopcroft2008, p99]

(1) 如果 L 是一个语言，a 是一个符号，则 L/a(称作 L 和 a 的商) 是所有满足如下条件的串 w 的集合：wa 属于 L。例如，如果 $L=\{a,aab,baa\}$, 则 $L/a=\{\varepsilon,ba\}$, 证明：如果 L 是正则的，那么 L/a 也是。提示：从 L 的 *DFA* 出发，考虑接受状态的集合。

(2) 如果 L 是一个语言，a 是一个符号，则 $a\backslash L$ 是所有满足如下条件的串 w 的集合: aw 属于 L。例如，如果 $L=\{a,aab,baa\}$, 则 $a\backslash L=\{\varepsilon,ab\}$, 证明：如果 L 是正则的，那么 $a\backslash L$ 也是。提示：记得正则语言在反转运算下是封闭的，又由 (1) 知，正则语言的商运算下是封闭的。

Definition 1.4 (Kleene-closure[Chrison2007]). Let $L\subseteq V^*$, then

$$L^* := \{\varepsilon\}\cup L\cup LL\cup LLL\cup\cdots$$

This is the same operation that we defined above for the set V, except that now it is applied to set L whose elements may be strings of length greater than one. An element of L^* is formed by the concatenation of a finite (but possibly arbitrarily large) number of elements of L; this includes the concatenation of "zero" elements, that is the empty string ε. Note that $*$ operation is idempotent: $(L^*)^* = L^*$.

$$\begin{aligned} L^* &= \{\varepsilon\}+L^+ \\ &= \{\varepsilon\}\cup(L\backslash\{\varepsilon\})L^* \\ &= \{\varepsilon\}+L+LL+LLL+\cdots \end{aligned}$$

1.1 Linear equation

see [Jean2018, 5.3,p64].

We give an algorithm to covert an automaton to a rational(regular) expression. The algorithm amounts to solving a system of linear equations on languages. We first consider an equation of the form

$$X = KX + L \tag{1.1}$$

Proposition 1.1 (Arden's Lemma). *if K does not contain the empty word, then $X = K^\star L$ is the unique solution of the equation $X = KX + L$.*

where K and L are languages and X is the unknown. When K does not contain the empty word, the equation admits a unique solution.

证明. Replacing X by K^*L in the expression $KX+L$,one gets

$$K(K^*)L+L = K^+L+L = (K^+L+L) = K^*L,$$

and hence $X = K^*L$ is a solution of (1.1). see[1]

To Prove uniqueness, consider two solutions X_1 and X_2 of (1.1). By symmetry,it suffices to show that each word u of X_1 also belongs to X_2. Let us prove this result by induction on the length of u.

If $|u| = 0$, u is the empty word[2] and if $u \in X_1 = KX_1 + L$, then necessarily $u \in L$ since $\varepsilon \notin K$. But in this case, $u \in KX_2 + L = X_2$. see[3]

For the induction step, consider a word u of X_1 of length $n+1$. Since $X_1 = KX_1 + L, u$ belongs either to L or to KX_1. if $u \in L$, then $u \in KX_2 + L = X_2$. If $u \in KX_1$ then $u = kx$ for some $k \in K$ and $x \in X_1$. Since k is not the empty word, one has necessarily $|x| \leq n$ and hence by induction $x \in X_2$.[see[4]] It follows that $u \in KX_2$ and finally $u \in X_2$. This conclude the induction and the proof of the proposition. □

From [Wonham2018, p74] The *length* $|s|$ of a string $s \in \Sigma^*$ is defined according to

$$|\varepsilon| = 0; |s| = k, \text{if } s = \sigma_1\sigma_2\cdots\sigma_k \in \Sigma^+$$

Thus $|cat(s,t)| = |s| + |t|$.

---

[1]

$$\begin{aligned} K^* &= \{\varepsilon\} + K^+ \\ &= \{\varepsilon\} + (K \backslash \{\varepsilon\})K^* \\ &= \{\varepsilon\} + K + KK + KKK + \cdots \end{aligned}$$

[2] The empty word $= \varepsilon, |\varepsilon| = 0$; if a language $M = \{\varepsilon\}, |M| = 1$, The empty language $M = \emptyset, |M| = 0$. 文献 [Jean2018] 用 1 表示 ε, 因为 $\varepsilon K = K\varepsilon = K$, 因此,$\varepsilon$ 是连接运算的单位元，正是 1 表示的用意。0 表示 $\emptyset$, 它是并运算的单位元,$K \cup \emptyset = \emptyset \cup K = K$.

[3] In this case,$|u| = 0, X = \{\varepsilon\}, |X| = 1$. *i.e.* $\varepsilon = K\varepsilon + L, \varepsilon = K + L$

[4] $u = kx, |u| = |kx| = n+1, \varepsilon \notin K, |k| \geq 1, |x| \leq n$, 由假设知，$u$ 属于 X_1, 归纳 $|x| = 0, |x| = 1, \cdots, n, x \in X_2$.

A *language* over Σ is any subset of Σ^*, i.e. an element of the power set $Pwr(\Sigma^*)$; thus the definition includes both the empty language $\emptyset$, and Σ^* itself.

Note the distinction between $\emptyset$ (the language with no strings) and ε (the string with no symbols). For instance the language $\{\varepsilon\}$ is nonempty, but contains only the empty string.

From [Wonham2018, p78]

Proposition 1.2 ([Wonham2018]).

1. If $L = M^*N$ *then* $L = ML + N$
2. If $\varepsilon \notin M$ *then* $L = ML + N$ *implies* $L = M^*N$ □

Part(2) is Known as Arden's rule. Taken with Part(1) it says that if $\varepsilon \notin M$ then $L = M^*N$ is the unique solution of $L = ML + N$; in particular if $L = ML$ (with $\varepsilon \notin M$) then $L = \emptyset$

Exercise 1.1. Show by counterexample that the restriction $\varepsilon \notin M$ in Arden's rule cannot be dropped.

Solution 1.1. Examples text goes here.

Exercise 1.2. Prove Arden's rule. Hint: If $L = ML + N$ then for every $k \geq 0$

$$L = M^{k+1}L + (M^k + M^{k-1} + \cdots + M + \varepsilon)N$$

Solution 1.2.

Preliminaries :

$$\begin{aligned}
M^* &= M^k + M^{k-1} + \cdots + M^1 + M^0 \qquad (k \geq 0) \\
&= M^k + M^{k-1} + \cdots + M^1 + \varepsilon \\
&= M^+ + \varepsilon \\
&= MM^* + \varepsilon \\
&= (M \backslash \{\varepsilon\})M^* + \varepsilon \\
M^+ &= M^k + M^{k-1} + \cdots + M^1 \qquad (k > 0) \\
&= M(M^k + M^{k-1} + \cdots + M^1 + M^0) \\
&= MM^* \\
M^0 &= \{\varepsilon\} = 1 \\
M\varepsilon &= \varepsilon M = M \\
\varepsilon + \varepsilon &= \varepsilon \\
M + M &= M
\end{aligned}$$

证明.

$$L = ML + N \Rightarrow$$

$$M^0 L = M^1 L + M^0 N \tag{1.2}$$

$$M^1 L = M^2 L + M^1 N \tag{1.3}$$

$$M^2 L = M^3 L + M^2 N \tag{1.4}$$

$$\tag{1.5}$$

$$\cdots$$

$\Rightarrow$

$(M^0 + M^1 + M^2 + \cdots)L = (M^1 + M^2 + M^3 + \cdots)L + (M^0 + M^1 + M^2 + \cdots)N$

$\Rightarrow$

so,if $L = ML + N$,then for every $k \geq 0$

$L = M^{k+1}L + (M^k + M^{k-1} + \cdots + M + M^0)N$

$\Rightarrow$

$$L = M^{k+1}L + (M^k + M^{k-1} + \cdots + M + \varepsilon)N \tag{1.6}$$

(1) $k=0$

$L=ML+(\varepsilon)N=ML+N$

$\Rightarrow (1-M)L=N$

$(\varepsilon-M)L=N$

由于 $\varepsilon \notin M$, 左端不会消去 $\{\varepsilon\}$. 因此, 只能在 N 中找 L, 仅有唯一解: $L=\{\varepsilon\}=\{\text{empty word}\}\subseteq N$.

From [R.Su and Wonham2004, definition 2.3]

Definition 1.5. Let

$G_A=(X_A,\Sigma,\xi_A,x_{A,0},X_{A,m})$

$G_A=(X_B,\Sigma,\xi_B,x_{B,0},X_{B,m})$

G_B is a DES-epimorphic image(满射像) of G_A under DES-epimorphism $\theta: X_A \to X_B$ if

1. $\theta: X_A \to X_B$ is surjective(满射)
2. $\theta(x_{A,0})=x_{B,0}$ and $\theta(X_{A,m})=X_{B,m}$
3. $(\forall x\in X_A)(\forall \sigma\in\Sigma)\xi_A(x,\sigma)!\Rightarrow[\xi_B(\theta(x),\sigma)!\&\xi_B(\theta(x),\sigma)=\theta(\xi_A(x,\theta))]$
4. $(\forall x\in X_B)(\forall \sigma\in\Sigma)\xi_B(x,\sigma)!\Rightarrow[(\exists x'\in X_A)\xi_A(x',\sigma)!\&\theta(x')=x]$

In particular,G_B is DES-isomorphic(同构) to G_A if $\theta: X_A \to X_B$ is bijective(双射).

see figure 1.1.

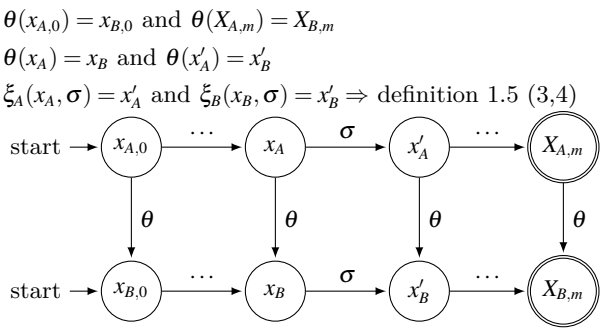


图 1.1: definition 1.5, G_B is a DES-epimorphic image(满射像) of G_A under DES-epimorphism $\theta: X_A \to X_B$

References

[Hopcroft2008] John E. Hopcroft,Rajeev Motwani,Jeffrey D. Ullman 著, 孙家骕等译, 自动机理论、语言和计算机导论,Third Edition, 机械工业出版社,2008.7

[WATSON93a] WATSON, B. W. *A taxonomy of finite automata construction algorithms*, Computing Science Note 93/43, Eindhoven University of Technology, The Netherlands, 1993. Available by ftp from ftp.win.tue.nl in pub/techreports/pi.

[WATSON93b] WATSON, B. W. *A taxonomy of finite automata minimization algorithms*, Computing Science Note 93/44, Eindhoven University of Technology, The Netherlands, 1993. Available by ftp from ftp.win.tue.nl in pub/techreports/pi.

[WATSON94a] WATSON, B. W. *An introduction to the FIRE engine: A C++ toolkit for FInite automata and Regular Expressions*, Computing Science Note 94/21, Eindhoven University of Technology, The Netherlands, 1994. Available by ftp from ftp.win.tue.nl in pub/techreports/pi

[WATSON94b] WATSON, B.W. *The design. and implementation of the FIRE engine: A C++ toolkit for FInite automata and Regular Expressions*, Computing Science Note 94/22, Eindhoven University of Technology, The Netherlands, 1994. Available by ftp from ftp.win.tue.nl in pub/techreports/pi.

[Chrison2007] Christos G. Cassandras and Stéphane Lafortune, *Introduction to Discrete Event Systems*,Second Edition,New York,Springer,2007

[Wonham2018] W. M. Wonham and Kai Cai,*Supervisory Control of Discrete-Event Systems*, Revised 2018.01.01

[Jean2018] Jean-Éric Pin, *Mathematical Foundations of Automata Theory*,Version of June 15,2018

[蒋宗礼 2013] 蒋宗礼, 姜守旭, 形式语言与自动机理论（第 3 版）, 清华大学出版社,2013.05

[Lipschutz2007] S. Lipschutz and M. L. Lipson, *Schaum's Outline of Theory and Problems of Discrete Mathematics*, Third Edition, New York: McGraw-Hill, 2007.

[Rosen2007] K. H. Rosen, *Discrete Mathematics and Its Applications*, Seventh Edition, New York: McGraw-Hill, 2007.

[R.Su and Wonham2004] R. Su and W. M. Wonham, *Supervisor reduction for discrete-event systems*, Discrete Event Dyn. Syst., vol. 14, no. 1, pp. 31–53, Jan. 2004.

1. S. Maclane and G. Birkhoff, *Algebra*, Third Edition, New York: Bchelsea Publishing Company, 1988.

Chapter 2
[蒋宗礼 2013](第 1 章绪论)

- 集合：集合的表示、集合之间的关系、集合的基本运算。
- 关系：主要介绍了二元关系相关的内容。包括等价关系、等价分类、关系合成、关系闭包。
- 递归定义与归纳证明。
- 图：无向图、有向图、树的基本概念。
- 语言与形式语言：自然语言的描述，形式语言和自动机理论的出现，形式语言和自动机理论对计算机科学与技术学科人才能力培养的作用
- 基本概念：字母表、字母、句子、字母表上的语言、语言的基本运算

2.1 集合的基础知识

2.1.1 集合之间的关系

Definition 2.1. 设 A,B 是两个集合，如果集合 A 中的元素都是集合 B 的元素，则称集合 A 是集合 B 的子集 *(subset)*，集合 B 是集合 A 的包集 *(container)*。记作 $A \subseteq B$, 也可以记作 $B \supseteq A$。

由定义可知，$A \subseteq B$ 的充要条件是：对于 A 中的每一个元素 a，均有 $a \in B$。为了简洁起见，P_1 是 P_2 的充要条件记为

$$P_1 \iff P_2$$

或者

$$P_2 \text{ iff } P_1$$

经常使用全称量词和存在量词：$\forall x$ 表示对所有的 x; $\exists x$ 表示存在一个 x。按照此约定，有

$A \subseteq B \iff \forall x \in A, x \in B$ 成立，

也就是

$A \subseteq B$ iff $\forall x \in A, x \in B$ 成立。

Definition 2.2. 设 A,B 是两个集合，如果 $A \subseteq B$, 且$\exists x \in B$, 但$x \notin B$，则称 A 是 B 的*真子集 (proper subset)*，记作 $A \subset B$。

Definition 2.3. 如果集合 A,B 含有的元素完全相同，则称集合 A 与集合 B 相等 *(equivalence)*，记作 $A = B$。

对任意集合 A,B,C, 不难得出下列结论：

1. $A = B$ iff $A \subseteq B$ 且 $B \subseteq A$。
2. 如果 $A \subseteq B$，则 $|A| \leq |B|$。
3. 如果 $A \subset B$，则 $|A| \leq |B|$。
4. 如果 A 是有穷集，且 $A \supset B$，则 $|B| > |A|$。
5. 如果 $A \subseteq B$，则 $\forall x \in A$, 有 $x \in B$。
6. 如果 $A \subset B$，则 $\forall x \in A$, 有 $x \in B$ 并且 $\exists x \in B$, 但 $x \notin A$。
7. 如果 $A \subseteq B$ 且 $B \subseteq C$，则 $A \subseteq C$。
8. 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 $A \subset C$。
9. 如果 $A = B$，则 $|A| = |B|$。

Definition 2.4. 设 A,B 是两个集合，A 与 B 的*对称差 (symmetric difference)* 是一个集合，该集合由属于 A 但不属于 B，以及属于 B 但不属于 A 的所有元素组成，记作 $A \otimes B$。

$$A \otimes B = \{a | a \in A \text{且} a \notin B \text{或者} a \notin B \text{且} a \in B\}$$

显然，对集合 A,B, 有

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

$\otimes$ 为对称差运算符，$A \otimes B$ 读作 A 对称减 B(A 与 B 的对称差)。

Definition 2.5. 设 A,B 是两个集合，A 与 B 的笛卡儿积 *(Cartesian product)* 是一个集合，该集合该集合是由所有这样的有序对 (ab) 组成的：其中 $a \in A, b \in B$ ，记作 $A \times B$。

$$A \times B = \{(a,b) | a \in A \& b \in B\}$$

$\times$ 为笛卡儿乘运算符。$A \times B$ 读作 A 叉乘 B。

对任意集合 A,B,C, 不难得出下列结论：

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. $A \times \emptyset = \emptyset$。
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，$|A \times B| = |A||B|$。

Definition 2.6. 设 A 是一个集合，A 的幂集 *(power set)* 是一个集合，该集合由 A 的所有子集组成，记作 2^A。

$$2^A = \{B | B \subseteq A\}$$

2^A 读作 A 的幂集。

对于任意集合 A,B，则有下列结论：

1. $\emptyset \in 2^A$。
2. $\emptyset \subseteq 2^A$。
3. $\emptyset \subset 2^2$。
4. $2^\emptyset = \{\emptyset\}$。
5. $A \in 2^A$。
6. 如果 A 是有穷集合，则 $|2^A| = 2^{|A|}$。
7. $2^{A \cap B} = 2^A \cap 2^B$。
8. if $A \subseteq B$, then $2^A \subseteq 2^B$。

Definition 2.7. 设 A 是论域 U 上的一个集合，A 的补集 *(complementary set)* 是一个集合，该集合由在 U 中，但不在 A 中的所有元素组成，记作 $\bar{A}$。

$$\bar{A} = U - A$$

$\bar{A}$ 读作 A(关于论域 U) 的补集 (U 中子集 A 的补集)。

在实际工作和生活中，人们都会在一定的范围内讨论问题，讨论问题的这个范围叫作论域。如果集合 A 是论域 U 上的一个集合，则 $A \subseteq U$。

设 U 是论域，A,B 是 U 上的集合，则有下列结论：

1. $\bar{\emptyset} = U$。
2. $\bar{U} = \emptyset$。
3. if $A \subseteq B$, then $\bar{B} \subseteq \bar{A}$。
4. $A \cup \bar{A} = U$。
5. $A \cap \bar{A} = \emptyset$。
6. $B = \bar{A} \iff A \cup B = U \& A \cap B = \emptyset$。
7. De Morgan 定律
   $\overline{A \cap B} = \bar{A} \cup \bar{B}$。
   $\overline{A \cup B} = \bar{A} \cap \bar{B}$。

2.2 关系

- 二元关系
- 递归定义与归纳证明
- 关系的闭包

2.2.1 二元关系

Definition 2.8. 设 A,B 是两个集合，任意的 $R \subset A \times B$,R 是 A 到 B 的*二元关系 (binary relation)*。

$(a,b) \in R$, 表示 a 与 b 满足关系 R, 按照中缀形式，也可以表示为 aRb。

A 称为定义域 *(domain)*, B 称为值域 *(range)*。
当 $A = B$ 时，则称 R 是 A 上的二元关系。

Definition 2.9. R 是 A 上的二元关系,

1. 如果对任意一个 $a \in A$, 有 $(a,a) \in R$, 则称 R 是自反的 *(reflexive)*。
2. 如果对任意一个 $a \in A$, 有 $(a,a) \notin R$, 则称 R 是反自反的 *(irreflexive)*。
3. 如果对任意的 $a,b \in A$, 当 $(b,a) \in R$ 时, 必有 $(a,b) \in R$, 则称 R 是对称的 *(symmetric)*。
4. 如果对任意的 $a,b \in A$, 当 $(b,a) \in R$ 和 $(a,b) \in R$ 同时成立, 必有 $a = b$，则称 R 是反对称的 *(asymmetric)*。
5. 如果对任意的 $a,b,c \in A$, 当 $(a,b) \in R$ 和 $(b,c) \in R$ 同时成立, 必有 $(a,c) \in R$，则称 R 是传递的 *(transition)*。

条件 (1),(3),(5) 合并在一起，叫作关系的三歧性：自反性，对称性，传递性。

Example 2.1. 关系的性质。

1. $=$ 关系是自反的，对称的，传递的。
2. $>, <$ 关系是反自反的，传递的。
3. $\geq, \leq$ 关系是自反的，反对称的，传递的。
4. 集合之间的包含关系是自反的，反对称的，传递的。
5. 整数集上的模 n 同余关系是自反的，对称的，传递的。
6. 通常意义下的父子关系是反自反的，非传递的。
7. 通常意义下的兄弟关系是反自反的，传递的。
8. 通常意义下的祖先关系是反自反的，传递的。

2.2.2 等价关系与等价类

Definition 2.10. 如果集合 A 上的二元关系 R 是自反的，对称的，传递的，则称 R 是等价关系 *(equivalence relation)*。

Example 2.2. 等价关系示例。

1. 实数集上的“$=$”关系。

2. 整数集上的模 n 同余关系。
3. 通常意义下的“在同一个学校工作”的关系。
4. “户口在同一省、市、自治区”的关系。

由此，可以考虑利用集合 S 上的等价关系 R 将 S 划分成若干个等价类。

Definition 2.11. 设 R 是集合 S 上的等价关系，则满足如下要求的 S 的划分 $S_1, S_2, S_3, \cdots, S_n, \cdots$ 称为 S 关于 R 的等价划分，S_i 称为等价类 *(equivalence class)*。它们满足以下各条：

1. $S = S_1 \cup S_2 \cup S_3 \cup \cdots \cup S_n \cup \cdots$。
2. if $i \neq j$, then $S_i \cap S_j = \emptyset$。
3. 对任意的 i, S_i 中的任意两个元素 a, b, aRb 恒成立。
4. 对任意的 $i, j, i \neq j, S_i$ 中的任意元素 a 和 S_j 中的任意元素 b, aRb 恒不成立。

R 将 S 分成的等价类的个数称为 R 在 S 上的指数 *(index)*。有时候，R 将 S 分成有穷多个等价类，此时称 R 具有有穷指数，反之称为无穷指数。

Example 2.3. 等价类。

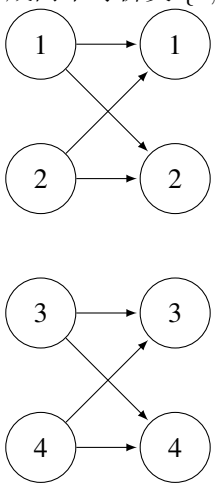
1. "=" 关系将自然数 $\mathbb{N}$ 分成无穷多个等价类：$\{1\}, \{2\}, \{3\}, \cdots$。
2. 非负整数集上的模 5 同余关系将 $\{0, 1, 2, 3, \cdots\}$ 分成 5 个等价类：
   $\{0, 5, 10, 15, 20, \cdots\}$
   $\{1, 6, 11, 16, 21, \cdots\}$
   $\{2, 7, 12, 17, 22, \cdots\}$
   $\{3, 8, 13, 18, 23, \cdots\}$
   $\{4, 9, 14, 19, 24, \cdots\}$
3. 某计算机学院 2001 年招收本科生 420 名，分成 12 个班，按同班同学关系划分，这 420 名学生分成 12 个等价类，每个等价类对应一个班。

Note 2.1. 值得注意的是，给定集合 S 上的一个等价关系 R,R 就确定了 S 的一个等价划分。当给定另一个不同的等价关系时，它会确定 S 的一个新的等价划分。

Example 2.4. 令 $S = \{1, 2, 3, 4\}$.

1. 通常意义的"=" 关系将 S 分成 4 个等价类：$\{1\}, \{2\}, \{3\}, \{4\}$。

2. 如果取 $R=\{(1,1),(2,1),(1,2),(2,2),(3,3),(3,4),(4,3),(4,4)\}$, 则将 S 分成两个等价类:$\{1,2\},\{3,4\}$。



2.2.3 关系的合成 *(composition)*

Definition 2.12. 设 $R_1 \subseteq A\times B$ 是 A 到 B 的关系，$R_2 \subseteq B\times C$ 是 B 到 C 的关系，则 R_1 与 R_2 的合成 *(composition)* $R_1\circ R_2$ 是 A 到 C 的关系。

$$R_1\circ R_2=\{(a,c)|\exists(a,b)\in R_1\&(b,c)\in R_2\}$$

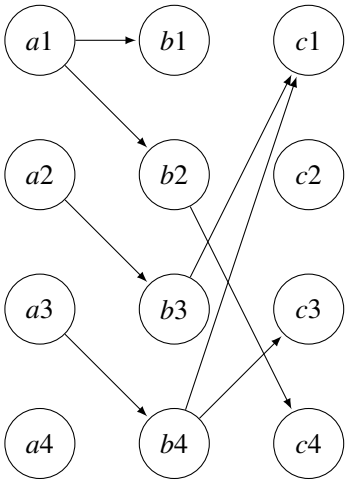
Example 2.5. 设 R_1,R_2 是集合 $\{1,2,3,4\}$ 上的关系, 其中

$R_1=(1,1),(1,2),(2,3),(3,4)$

$R_1=(2,4),(4,1),(4,3),(3,1)$

则

$R_1\circ R_2=\{(1,4),(2,1),(3,1),(3,3)\}$



设 R_1, R_2, R_3 分别是 S 上的二元关系，可以证明以下结论:

1. $R_1 \circ R_2 \neq R_2 \circ R_1$。
2. $(R_1 \circ R_2) \circ R_3 = R_1 \circ (R_2 \circ R_3)$。(结合律)
3. $(R_1 \cup R_2) \circ R_3 = (R_1 \circ R_3) \cup (R_2 \circ R_3)$。(合成对 $\cup$ 的右分配律)
4. $R_3 \circ (R_1 \cup R_2) = (R_3 \circ R_1) \cup (R_3 \circ R_2)$。(合成对 $\cup$ 的左分配律)
5. $(R_1 \cap R_2) \circ R_3 \subseteq (R_1 \circ R_3) \cap (R_2 \circ R_3)$。(合成对 $\cap$ 的右分配律)
6. $R_3 \circ (R_1 \cap R_2) \subseteq (R_3 \circ R_1) \cap (R_3 \circ R_2)$。(合成对 $\cap$ 的左分配律)

2.2.4 递归定义 *(recurisive definition)* 与归纳证明

- 递归定义 (recurisive definition)
  - 又称为归纳定义 (indutive definition), 它来定义一个集合。
  - 集合的递归定义由三个部分组成:
    - 基础 (basis): 用来定义该集合的最基本的元素。
    - 归纳 (induction): 指出用集合中的元素来构造集合的新元素的规则。
    - 极小性限定: 指出一个对象是所定义集合中的元素的充要条件是它可以通过有限次的使用基础和归纳条款中所给的规定构造出来。

- 归纳证明
  - 与递归定义相对应
  - 归纳证明方法包括三大步:
    - 基础 (basis): 证明最基本元素具有相应性质。
    - 归纳 (induction): 证明如果某些元素具有相应性质，则根据这些元素用所规定的方法得到新元素也有相应的性质。
    - 根据归纳法原理，所有的元素具有相应的性质。

Example 2.6. 算术表达式的递归定义：

1. 基础 (basis): 常数是算术表达式，变量是算术表达式；
2. 归纳 (induction): 如果 E_1, E_2 是算术表达式，则 $+E_1, -E_1, E_1+E_2, E_1-E_2, E_1*E_2, E_1/E_2, E_1 \wedge E_2, Fun(E_1)$ 是算术表达式。其中 Fun 为函数名。
3. 只有满足 (1) 和 (2) 的式子才是算术表达式。

Definition 2.13. 设 R 是 S 上的关系，我们递归地定义 R^n 的幂：

1. $R^0 = \{(a,a)|a \in S\}$
2. $R^i = R^{i-1}R \quad (i = 1,2,3,\cdots)$

Example 2.7. 对有穷集合 A, 证明 $|2^A| = 2^{|A|}$。

证明. 设 A 为一个有穷集合，施归纳于 $|A|$:

1. 基础 (basis): 当 $|A| = 0$ 时，由幂集定义，$|2^A| = |\{\emptyset\}| = 1$。而 $2^{|A|} = 2^0 = 1$。所以有 $|2^A| = 2^{|A|}$ 对 $|A| = 0$ 成立。
2. 归纳 (induction): 假设 $|A| = n$ 时结论成立, 这里 $n \geq 0$, 往证当 $|A| = n+1$ 时结论成立。
   为此，不妨设 $A = B \cup \{a\}, a \notin B$, 即
   $|2^A| = B \cup \{a\} = |B| + |\{a\}| = |B| + 1$
   由幂集的定义知
   $2^A = 2^B \cup \{C \cup \{a\} | C \in 2^B\}$
   由于 $a \notin B$, 所以
   $2^B \cap \{C \cup \{a\} | C \in 2^B\} = \emptyset$
   由 $\{C \cup \{a\} | C \in 2^B\}$ 的构造方法知道，可以按如下方法构造一个一一对应的映射 $f : \{C \cup \{a\} | C \in 2^B\} \to 2^B$, 使
   $f(C \cup \{a\}) = C$

所以

$|\{C\cup\{a\}|C\in 2^B\}|=|2^B|$

故

$$\begin{aligned}|2^A| &= |2^B\cup\{C\cup\{a\}|C\in 2^B\}|\\ &= |2^B|+|\{C\cup\{a\}|C\in 2^B\}|\\ &= |2^B|+|2^B|\\ &= 2|2^B|\end{aligned}$$

显然，$|B|=n$, 由归纳假设知

$|2^B|=2^{|B|}$

从而有

$|2^A|=2|2^B|=2\times|2^B|=2^{|B|+1}=2^{|A|}$

这就是说，结论对 $|A|=n+1$ 成立。

3. 由归纳法原理，结论对任意有穷集合成立。

Example 2.8. 表达式的前缀形式是指将运算符写在前面，后跟相应的运算对象。如：$+E_1$ 的前缀形式为 $+E_1$，E_1+E_2 的前缀形式为 $+E_1E_2$，E_1*E_2 的前缀形式为 $*E_1E_2$，$E_1\wedge E_2$ 的前缀形式为 $\wedge E_1E_2$，$Fun(E_1)$ 的前缀形式为 $FunE_1$。证明 *Example 2.6*所定义的表达式可以用这里定义的前缀形式表示。

证明. 设 E 为 *Example 2.6*所定义的算术表达式，现对 E 中所含的运算符 (包括函数引用) 的个数实施归纳。设 E 中含 n 个运算符。

1. 基础 (basis): 当 $n=0$ 时，表达式为一个常数或者变量，结论显然成立。
2. 归纳 (induction): 假设 $n\le k$ 时结论成立，这里 $k\ge 0$, 往证当 $n=k+1$ 时结论成立。

   由于 E 中含有 $k+1$ 个运算符, 所以必须是如下情况中的一种：

   a. 当 $E=+E_1$ 时，我们知道 E_1 中的运算符个数为 k，由归纳假设，E_1 有对应的前缀形式 F_1, 从而 E 的前缀形式为 $+F_1$。
   b. 当 $E=-E_1$ 时，类似地，E 的前缀形式为 $-F_1$。
   c. 当 $E=E_1+E_2$ 时，E_1,E_2 中的运算符个数分别小于等于 k，由归纳假设，E_1,E_2 有对应的前缀形式 F_1,F_2, 从而 E 的前缀形式为 $+F_1F_2$。

d. 对 $E = E_1 - E_2, E = E_1 * E_2, E = E_1 / E_2, E = E_1 \wedge E_2$ 的情况进行类似的讨论，它们的前缀形式分别为: $-F_1F_2, *E_1E_2, /E_1E_2, \wedge E_1E_2$。

e. 当 $E = Fun(E_1)$ 时，我们知道 E_1 中的运算符个数为 k，由归纳假设，E_1 有对应的前缀形式 F_1, 从而 E 的前缀形式为 $FunF_1$。

综上所述，结论对 $n = k+1$ 成立。

3. 由归纳法原理，结论对 *Example 2.6*所定义的所有表达式成立。

2.2.5 关系的闭包

Definition 2.14. 设 P 是关于关系的性质的集合，关系 R 的P 闭包 *(closure)* 是包含 R 并且具有 P 中所有性质的最小关系。

Definition 2.15. 设 R 是 S 上二元关系，R 的正闭包 *(positive closure)*R^+ 的定义为:

(1) $R \subseteq R^+$。

(2) 如果 $(a,b),(b,c) \in R^+$, 则 $(a,c) \in R^+$。

(3) 除 (1)、(2) 外，R^+ 不包含有其他任何元素。

可以证明，R^+ 具有传递性，因此又称其为传递闭包 *(transitive closure)*。还可以证明，对于任意的二元关系 R, 有

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

且当 S 为有穷集时, 有

$$R^+ = R \cup R^2 \cup R^3 \cup \cdots \cup R^{|S|}$$

Definition 2.16. 设 R 是 S 上二元关系，R 的克林闭包 *(Kleene closure)*R^* 的定义为:

(1) $R^0 \subseteq R^*, R \subseteq R^*$.

(2) 如果 $(a,b),(b,c) \in R^*$ 则 $(a,c) \in R^*$.

(3) 除 (1),(2) 外，R^* 不再含有其他任何元素.

可以证明，R^* 具有自反性和传递性，因此又称其为自反传递闭包 *(reflexive and transitive closure)*。

由定义 2.15和 2.16可知，对于任意二元关系 R，有

$$\begin{aligned} R^* &= R^0 \cup R^+ \\ &= R^0 \cup R \cup R^2 \cup R^3 \cup \ldots \end{aligned}$$

而且当 S 为有穷集时:

$$R^* = R^0 \cup R \cup R^2 \cup R^3 \cup \cdots \cup R^{|S|}$$

设 R_1, R_2 是 S 上的两个二元关系，则

(1) $\emptyset^+ = \emptyset$

(2) $(R_1^+)^+ = R_1^+$

(3) $(R_1^*)^* = R_1^*$

(4) $R_1^+ \cup R_2^+ \subseteq (R_1 \cup R_2)^+$

(5) $R_1^* \cup R_2^* \subseteq (R_1 \cup R^2)^*$

2.3 语言

2.3.1 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
- 字符的两个特性
  - 整体性 (monolith), 也叫不可分性
  - 可辨认性 (distinguishable)，也叫可去区分性
- 字母表的乘积 (product)

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

- 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

$$\Sigma^+ = \Sigma \cup \Sigma^2 \cup \Sigma^3 \cup \cdots$$

$$\Sigma^+ = \{x | x\text{是}\Sigma\text{中的至少一个字符连接而成的字符串}\}$$

- Σ 的克林闭包

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

$$\Sigma^* = \{x | x\text{是}\Sigma\text{中的若干个，包括 0 个字符，连接而成的字符串}\}$$

Example 2.9. {alphabet}

{a,b,c,d}
{a,b,c,...,z}
{0,1}
{a,a',b,b' }
{aa,ab,bb }
$\{\infty, \wedge, \vee, \geq, \leq\}$

Example 2.10. product

{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}

Example 2.11. Σ^0, Σ^*

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

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

$$\{a,b,c,d\}^+ = \{a,b,c,d,aa,ab,ac,ad,ba,bb,bc,bd,\dots,aaa,aab,aac,aad,aba,abb,abc,\dots\}$$

$$\{a,b,c,d\}^* = \{\varepsilon,a,b,c,d,aa,ab,ac,ad,ba,bb,bc,bd,\dots,aaa,aab,aac,aad,aba,abb,abc,\dots\}$$

2.3.2 句子 *(sentence)*/字 *(word)*/字符串 *(string)*

- 别称
  句子 (sentence),(字符、符号) 行 (line),(字符、符号) 串 (string).

- 句子 (sentence)
  Σ 是一个字母表,$\forall \in \Sigma^*$,x 叫做 Σ 上的一个句子。
- 句子相等
  两个句子被认为相等的，如果它们对应位置上的字符都对应相等。
- 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
- 串 x 的 n 次幂

$$x^0 = \varepsilon$$
$$x^n = x^{n-1}x$$

Note 2.2. 注意事项

- ε 是一个句子
- $\{\varepsilon\} \neq \emptyset$。这是因为 $\{\varepsilon\}$ 不是一个空集，它是含有一个空句子的 ε 的集合。$|\{\varepsilon\}| = 1, |\emptyset| = 0$

Example 2.12.

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

Example 2.13. x=001,y=1101

$$x^0 = y^0 = \varepsilon$$
$$x^4 = 001001001001$$
$$y^4 = 1101110111011101$$

2.3.3 并置/连结 *(concatenation)*

- 并置/连结 (concatenation)
  - $x, y \in \Sigma^*, x, y$ 的并置是由串 x 直接相接串 y 组成的。记作 xy.
- Σ^* 上的并置运算性质

1. 结合律: $(xy)z = x(yz)$
2. 左消去律: if $xy = xz$,then $y = z$
3. 右消去律: if $yx = zx$,then $y = z$
4. 惟一分解性: 存在惟一确定的 $a_1, a_2, \ldots, a_n \in \Sigma$, 使得 $x = a_1 a_2 \cdots a_n$.
5. 单位元素: $\varepsilon x = x\varepsilon = x$

2.3.4 前缀与后缀

设 $x, y, z, w, v \in \Sigma^*$, 且 $x = yz, w = yv$

1. y 是 x 的前缀 (prefix)
2. 如果 $z \neq \varepsilon$, 则 y 是 x 的真前缀 (proper prefix).
3. z 是 x 的后缀 (suffix)
4. 如果 $y \neq \varepsilon$, 则 z 是 x 的真后缀 (proper suffix)
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 $x = zy$ 和 $w = vy$, 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。

Example 2.14. $\Sigma = \{a, b\}$ 上的句子 abaabb:
前缀: $\varepsilon, a, ab, aba, abaa, abaab, abaabb$
真前缀: $\varepsilon, a, ab, aba, abaa, abaab$
后缀: $\varepsilon, b, bb, abb, aabb, baabb, abaabb$
真后缀: $\varepsilon, b, bb, abb, aabb, baabb$

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 有惟一的一个真后缀 z 与之对应, 使得 $x = yz$; 反之亦然。
3. $|\{w | w\text{是}x\text{的后缀}\}| = |\{w | w\text{是}x\text{的前缀}\}|$
4. $|\{w | w\text{是}x\text{的真后缀}\}| = |\{w | w\text{是}x\text{的真前缀}\}|$

5. $\{w|w是x的前缀\} = |\{w|w是x的真前缀 \cup \{x\}\}$
6. $|\{w|w是x的前缀\}| = |\{w|w是x的真前缀+1\}|$
7. $\{w|w是x的后缀\} = |\{w|w是x的真后缀 \cup \{x\}\}$
8. $|\{w|w是x的后缀\}| = |\{w|w是x的真后缀+1\}|$
9. 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。
10. 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。

约定

- 用小写字母表中较为靠前的字母 $a,b,c,\ldots$ 表示字母表中的字母
- 用小写字母表中较为靠后的字母 $x,y,z,\ldots$ 表示字母表中的句子 (字)
- 用 x^T 表示 x 的倒序。例如, 如果 $x = abc$, 则 $X^T = cba$

2.3.5 子串 *(substring)*

- 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 $y_1,y_2,\ldots,y_n$ 是 t 和 w 的公共子串, 且 $max\{|y_1|,|y_2|,\ldots,|y_n|\} = |y_j|$, 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。

2.3.6 语言 *(language)*

$\forall \in \Sigma^*, L$ 称为字母表 Σ 上的一个语言 *(language)*,$\forall x \in L, x$ 叫做 L 的一个句子 (sentence)/字 (word)/字符串 (string)。

Example 2.15. $\Sigma = \{0,1\}$ 上的不同语言 {00,11}
{0,1}
{0,1,00,11}

$\{0,1,00,11,01,10\}$
$\{00,11\}^*$
$\{01,10\}^*$
$\{00,01,10,11\}^*$
$\{0\}\{0,1\}^*\{1\}$
$\{0,1\}^*\{111\}\{0,1\}^*$

2.3.7 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 *(product)* 是一个语言, 该语言定义为:

$$L_1L_2 = \{xy | x \in L_1, y \in L_2\}$$

是字母表 $\Sigma_1 \cup \Sigma_2$ 上的语言。

Example 2.16. $\Sigma = \{0,1\}$

$$\begin{aligned}
L_1 &= \{0,1\} \\
L_2 &= \{00,01,10,11\} \\
L_3 &= \{0,1,00,01,10,11,000,\dots\} = \Sigma^+ \\
L_4 &= \{\varepsilon,0,1,00,01,10,11,000,\dots\} = \Sigma^* \\
L_5 &= \{0^n | n \geq 1\} \\
L_6 &= \{0^n 1^n \geq 1\} \\
L_7 &= \{1^n | n \geq 1\} \\
L_8 &= \{0^n 1^m | n,m \geq 1\} \\
L_9 &= \{0^n 1^n 0^n | n \geq 1\} \\
L_{10} &= \{0^n 1^m 0^k | n,m,k \geq 1\} \\
L_{11} &= \{x | x \in \Sigma^+ \text{且 } x \text{ 中 0 和 1 的个数相同}\}
\end{aligned}$$

- 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 的字组成的语言, L_2 是 Σ 上的所有长度为 2 的字组成的语言;

- 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 $L_9 \neq L_{10}$, 但我们有 $L_6 \subset L_5L_7, L_9 \subset L_{10}$.
- L_6 中的 word 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} 中的 word 中虽然保持着 0 和 1 的个数相同，但它并没有要求所有的 0 在所有的 1 的前面。例如，$0101, 1100 \in L_{11}$, 但是 $0101 \notin L_6$。而对 $\forall x \in L_6$, 有 $x \in L_{11}$。所以 $L_6 \subset L_{11}$。

Example 2.17. x^T example

1. $\{x|x = x^T, x \in \Sigma\}$
2. $\{xx^T|x \in \Sigma^+\}$
3. $\{xx^T|x \in \Sigma^*\}$
4. $\{xwx^T|x, w \in \Sigma^+\}$
5. $\{xx^Tw|x, w \in \Sigma^+\}$

- 幂 $\forall L \in \Sigma^*, L$ 的 n 次幂是一个语言，该语言定义为
  1. 当 $n = 0$ 时，$L^n = \{\varepsilon\}$
  2. 当 $n \geq 1$ 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
$$L^* = L^0 \cup L \cup L^2 \cup L^3 \cup L^4 \cup \ldots$$

2.4 Exercise and Solution

Exercise 2.1. 设 L 是 Σ 上的一个语言，Σ^* 上的二元关系 R_L 定义为: 对任给的 $x, y \in \Sigma^*$，如果对于 $\forall z \in \Sigma^*$，均有 $xz \in L$ 与 $yz \in L$ 同时成立或者同时不成立，则 xR_Ly。请证明 R_L 是 Σ^* 上的一个等价关系。将 R_L 称为由语言 L 所确定的等价关系。即

$$xR_Ly \Leftrightarrow (\forall z \in \Sigma^*, xz \in L \Leftrightarrow yz \in L)$$

实际上，R_L 还有另外一个性质：如果对任给的 $x, y \in \Sigma^*$, 当 xR_Ly 成立时必有 xzR_Lyz, 对 $\forall z \in \Sigma^*$ 都成立。这将被称为 R_L 的“右不变”性，你能证明此性质成立吗？

Solution 2.1. 分两步证明 R_L 是“右不变”的等价关系。

证明.

1. R_L 是等价关系。
   - 自反性：$\forall x \in \Sigma^*$, 显然对于 $\forall z \in \Sigma^*, xz$ 要么是 L 的字符串, 要么不是 L 的字符串。由 R_L 的定义知，xR_Lx
   - 对称性：不难看出，$xR_Ly \Leftrightarrow (\forall z \in \Sigma^*, xz \in L \Leftrightarrow yz \in L) \Leftrightarrow yR_Lx$
   - 传递性：设 xR_Ly, yR_Lz。
     $\because$ (由 R_L 的定义知)
     $xR_Ly \Leftrightarrow (\forall w \in \Sigma^*, xw \in L \Leftrightarrow yw \in L)$
     $yR_Lz \Leftrightarrow (\forall w \in \Sigma^*, yw \in L \Leftrightarrow zw \in L)$
     $\therefore$
     $\forall w \in \Sigma^*, xw \in L \Leftrightarrow yw \in L$ and $yw \in L \Leftrightarrow zw \in L$
     $\Rightarrow$
     $\forall w \in \Sigma^*, xw \in L \Leftrightarrow zw \in L$

   故
   xR_Lz, 即 R_L 是等价关系。
2. R_L 是右不变的。
   设 xR_Ly。由 R_L 的定义，对 $\forall w, v \in \Sigma^*, xwv \in L \Leftrightarrow ywv \in L$。注意到 v 的任意性，知 xwR_Lyw
   所以，R_L 是右不变的等价关系。

Exercise 2.2. 设 $\{0,1\}^*$ 上的语言 $L = \{0^n1^n | n \geq 0\}$, 请给出 $\{0,1\}^*$ 的关于 L 所确定的等价关系 R_L 的等价分类。

Solution 2.2. 根据 **Exercise 2.1**对 R_L 的定义及其证明，考虑 R_L 对 $\{0,1\}^*$ 的等价分类时，主要需根据语言 L 的结构，分析哪些串按照 L 的要求具有相同的特征。

1. 取 $0^n1^n, 0^m1^m \in L, 0^n1^n\varepsilon \in L, 0^m1^m\varepsilon \in L$ 同时成立，但是，对所有 $x \in \{0,1\}^+, 0^n1^nx \in L, 0^m1^mx \in L$ 同时不成立，符合 R_L 的定义。所以，L 中的元素是属于同一类的。即 $0^n1^nR_L0^m1^m \Leftrightarrow (0^n1^n \in L \Leftrightarrow 0^m1^m \in L), n, m \geq 0$。
2. 分析是否还有其他的元素与 L 中的元素属于同一类。根据 L 的结构，一种类型的串不含子串 10，这种串可以表示成 $0^k1^h, k \neq h$; 另一种是

含子串 10(L 的字符串不含这种子串)，这种串可以表示成 $x10y$。显然，$01\varepsilon \in L$，但是 $0^k1^h \notin L(k \neq h), x10y \notin L(x,y \in \{0,1\}^*)$。根据等价分类的性质，$\{0,1\}^*$ 中的不在 L 中的串与在 L 的串不在同一个等价类中。

3. 考察不含子串 10 的串。这些串有如下几种形式：

   a. $0^n, n \geq 1$。

   b. $1^n, n \geq 1$。

   c. $0^m1^n, m,n \geq 1$，且$m > n$。

   d. $0^m1^n, m,n \geq 1$，且$m < n$。

   对于 $0^n, n \geq 1$，1^n 接在它后面时，构成串 0^n1^n。显然，当 $m \neq n$ 时，$0^n1^n \in L$, 但 $0^m1^n \notin L$。所以,0^n 和 0^m 一定不在同一个等价类中。

   类似的讨论可知，对于 $0^n, n \geq 1$:

   0^n 不可能与形如 $0^m1^n(m,n \geq 1, 且m < n)$ 的串在同一等价类中；

   0^n 不可能与含有子串 10 的串在同一等价类中。

   下面再考虑形如 0^k 的串和形如 $0^m1^n(m,n \geq 1, 且m > n)$ 是否可能在同一等价类中。

   注意到当 $m-n=h$ 时,

   $$0^h1^h \in L, 0^m1^n1^h \in L$$

   同时成立，但是当 $n \geq 1, x = 01^{h+1}$ 时 $(x \neq 1^h)$,

   $$0^hx \in L, 0^m1^nx \notin L$$

   成立。所以，对应 $h > 0$, 令

   $$[h] = \{0^m1^n | m-n = h 且 n \geq 1\}$$

   [h] 中的元素在同一个等价类中，而且所有其他的元素都不在这个等价类中。实际上，当 $h=0$ 时，有

   $$[0] = L$$

4. 形如 1^m 的串和形如 $0^m1^n(m,n \geq 1且m < n)$ 的串应该在同一等价类中。事实上，对于 $\{0,1\}^*$ 中的任意字符串 x,

   $$1^mx \notin L, 0^m1^nx \notin L(m,n \geq 1且m < n)$$

   恒成立。所以，这些字符串在同一个等价类中。

5. 所有含子串 10 的串在同一等价类中。事实上，设 y,z 是含子串 10 的串，对于 $\{0,1\}^*$ 中的任意字符串 x,

$$yx \notin L, zx \notin L(m,n \geq 1 \text{且} m < n)$$

恒成立。所以，这些字符串在同一个等价类中。

6. 形如 1^m 的串和含子串 10 的串应该在同一等价类中。事实上，设 y 是含有子串 10 的串，对于 $\{0,1\}^*$ 中的任意字符串 x,

$$1^m x \notin L, yx \notin L(m \geq 1)$$

恒成立。所以，这些字符串在同一个等价类中。

综上所述，R_L 确定的 $\{0,1\}^*$ 的等价分类为

$$[10] = \{x10y | x,y \in \{0,1\}^*\} \cup \{0^m 1^n | n-m \geq 1\}$$
$$[0] = \{0^m 1^n | n-m = 0\} = \{0^n 1^n | n \geq 0\}$$
$$[1] = \{0^m 1^n | n-m = 1\}$$
$$[2] = \{0^m 1^n | n-m = 2\}$$
$$\vdots$$
$$[h] = \{0^m 1^n | n-m = h\}$$
$$\vdots$$
$$\{0\}$$
$$\{00\}$$
$$\vdots$$
$$\{0^n\}$$
$$\vdots$$

其中,n,m 均为非负整数。

Exercise 2.3. 使用归纳法证明：

对字母表 Σ 的任意字符串 x,x 的前缀有 $|x|+1$ 个。

Solution 2.3. 证明. 设 $x \in \Sigma^*$, 现对 x 的长度施归纳。为了叙述方便用 $prefix(x)$ 表示字符串 x 的所有前缀组成的集合。

当 $|x| = 0$ 时，有 $x = \varepsilon$，由字符串的前缀定义知，$prefix(x) = \{x\}$。ε 就是 x 的唯一前缀，而

$$\begin{aligned}|prefix(x)| &= |\{\varepsilon\}| \\ &= 1 \\ &= 0+1 \\ &= |x|+1\end{aligned}$$

所以，结论对 $|x|=0$ 成立。

设 $|x|=n$ 时结论成立，$n\geq 0$。即

$|prefix(x)|=|x|+1$

现在考察 $|x|=n+1$ 的情况。为了叙述方便，不妨设 $x=ya$，其中 $|y|=n, a\in\Sigma$。由归纳假设，

$|prefix(y)|=|y|+1$

首先证明 y 的任何前缀都是 x 的前缀。事实上，设

$|prefix(y)|=\{u_1,u_2,\cdots,u_n\}$

对于 $\forall u\in prefix(y)$，根据前缀的定义，存在 $v\in\Sigma^*$，使得 $uv=y$，注意到 $uva=x$，所以，u 也是 x 的前缀，它对应 x 的后缀为 va。

再注意到 $x=ya$，所以，一方面，对于 $\forall u\in prefix(x)$，均有

$u\neq x$

从而

$x\notin prefix(y)$，

然而，由

$x\varepsilon=x$

可知，x 是 x 的一个前缀。另一方面，由 $x=ya$ 知道，如果 u 是 x 的一个前缀，v 是 u 对应的 x 的后缀，则有如下两种情况：

(1) $v\geq 1$，此时必有 $u\in prefix(y)$。

(2) $|v|=0$，此时必有 $v=\varepsilon$ 并且 $u=u\varepsilon=ya=x$。

由此可见，

$$\begin{aligned}prefix(x) &= prefix(y)\cup\{x\} \\ &= \{u_1,u_2,\cdots,u_n,x\}\end{aligned}$$

由 $x\notin prefix(y)$ 可知，

$$\begin{aligned}|prefix(x)| &= |prefix(y) \cup \{x\}| \\ &= |prefix(y)| + |\{x\}| \\ &= |prefix(y)| + 1\end{aligned}$$

再由归纳假设，

$$\begin{aligned}|prefix(x)| &= |prefix(y)| + 1 \\ &= |y| + 1 + 1 \\ &= |x| + 1\end{aligned}$$

表明结论对 $|x| = n+1$ 成立。
由归纳法原理，结论对于任意 $x \in \Sigma^*$ 成立。 □

另一种往证 |x|=n+1 时结论成立方法：

设 $|x| = n$ 时结论成立，$n \geq 0$。即 $|prefix(x) = |x| + 1$
现在考察 $|x| = n+1$ 的情况。为了叙述方便，不妨设 $x = ya$，其中 $|y| = n, a \in \Sigma \& a \neq y$。
$x = ya \Rightarrow$

$$\begin{aligned}prefix(x) &= prefix(ya) \\ &= \{\varepsilon\} \cup prefix(y) \cup prefix(\{ya\}) && \text{(前缀定义)} \\ &= \{\varepsilon\} \cup prefix(y) \cup prefix(a) && (prefix(\{ya\}) = \{\varepsilon\} \cup prefix(y) \cup prefix(a)) \\ &= \{\varepsilon\} \cup prefix(y) \cup \{a\} && (prefix(a) = \{\varepsilon, a\})\end{aligned}$$

$\because \varepsilon \in prefix(y), a \neq y,$
$\therefore |prefix(x)| = |prefix(y)| + |\{a\}| = |prefix(y)| + 1$
由归纳假设有，$|prefix(y)| = |y| + 1$
所以，$|prefix(x)| = |prefix(y)| + 1 = (|y| + 1) + 1 = |x| + 1$
表明结论对 $|x| = n+1$ 成立。

Exercise 2.4. 设 $\Sigma = \{a, b\}$, 求字符串 *aaaaabbbba* 的所有前缀的集合，后缀的集合，真前缀的集合，真后缀的集合。

Solution 2.4. 见表 (2.1)。

从表中可以看出，按照前缀与后缀的对应关系构成原来的字符串的对应关系，前缀和后缀是一一对应的，但是，按照这种对应关系，真前缀和真后缀不是一一对应的。不过，若将真后缀和真前缀中的空串 ε 去掉，则这种一一对应关系是仍然存在的。

另外，从对这一问题的讨论知，并不是所有的字符串都有真前缀和真后缀。

表 2.1: $\Sigma=\{a,b\}$,*aaaaabbbba* 的前缀，后缀，真前缀，真后缀的集合

前缀长度	前缀	是真前缀	对应后缀	是真后缀
0	ε	√	*aaaaabbbba*	
1	*a*	√	*aaaabbbba*	√
2	*aa*	√	*aaabbbba*	√
3	*aaa*	√	*aabbbba*	√
4	*aaaa*	√	*abbbba*	√
5	*aaaaa*	√	*bbbba*	√
6	*aaaaab*	√	*bbba*	√
7	*aaaaabb*	√	*bba*	√
8	*aaaaabbb*	√	*ba*	√
9	*aaaaabbbb*	√	*a*	√
10	*aaaaabbbba*		ε	√

Exercise 2.5. 设 L_1,L_2,L_3,L_4 分别是 $\Sigma_1,\Sigma_2,\Sigma_3,\Sigma_4$ 上的语言，能否说 L_1,L_2,L_3,L_4 是某个字母表 Σ 上的语言？如果能，请问这个字母表 Σ 是什么样的?。

Solution 2.5. 可以说 L_1,L_2,L_3,L_4 是同一个字母表 Σ 上的语言。这里

$$\Sigma=\Sigma_1\cup\Sigma_2\cup\Sigma_3\cup\Sigma_4$$

Exercise 2.6. 设 L_1,L_2,L_3,L_4 分别是 $\Sigma_1,\Sigma_2,\Sigma_3,\Sigma_4$ 上的语言，证明下列等式成立。

$$(L_1\cup L_2\cup L_3\cup L_4)^*=(L_1^*L_2^*L_3^*L_4^*)^*$$

Solution 2.6. 证明. 考虑到语言就是一系列字符串的集合，所以，证明两个语言相等，实际上就是证明相应的两个集合相等。因此，为证明

$$(L_1 \cup L_2 \cup L_3 \cup L_4)^* = (L_1^* L_2^* L_3^* L_4^*)^*$$

就是证明下列 (1),(2) 式同时成立。

$$(L_1 \cup L_2 \cup L_3 \cup L_4)^* \subseteq (L_1^* L_2^* L_3^* L_4^*)^* \tag{1}$$

$$(L_1 \cup L_2 \cup L_3 \cup L_4)^* \supseteq (L_1^* L_2^* L_3^* L_4^*)^* \tag{2}$$

首先证明 (1) 式成立。为此, 设

$x \in (L_1 \cup L_2 \cup L_3 \cup L_4)^*$

从而存在非负整数 n 和 $x_1, x_2, \cdots, x_n, \{x_1, x_2, \cdots, x_n\} \subseteq (L_1 \cup L_2 \cup L_3 \cup L_4)$, 使得

$x = x_1 x_2 \cdots x_n$

注意到 $\{x_1, x_2, \cdots, x_n\} \subseteq (L_1 \cup L_2 \cup L_3 \cup L_4)$, 所以，对于 $1 \leq j \leq n$

$x_j \in L_1, x_j \in L_2, x_j \in L_3, x_j \in L_4$

中至少一个成立，这表明

$x_j \in L_1^*, x_j \in L_2^*, x_j \in L_3^*, x_j \in L_4^*$

中至少一个成立，再注意到

$\varepsilon \in L_1^*, \varepsilon \in L_2^*, \varepsilon \in L_3^*, \varepsilon \in L_4^*$

并且

$L_1 \subseteq L_1^*, L_2 \subseteq L_2^*, L_3 \subseteq L_3^*, L_4 \subseteq L_4^*$

使得

$L_1 \subseteq L_1^* L_2^* L_3^* L_4^*, L_2 \subseteq L_1^* L_2^* L_3^* L_4^*, L_3 \subseteq L_1^* L_2^* L_3^* L_4^*, L_4 \subseteq L_1^* L_2^* L_3^* L_4^*$

从而

$x_j \in L_1^* L_2^* L_3^* L_4^*$

故

$x = x_1 x_2 \cdots x_n \in (L_1^* L_2^* L_3^* L_4^*)^n$

亦即

$x = x_1 x_2 \cdots x_n \in (L_1^* L_2^* L_3^* L_4^*)^*$

所以 (1) 式成立。

类似易证 (2) 式亦成立。

综上所述 (1),(2) 式同时成立。所以，$(L_1 \cup L_2 \cup L_3 \cup L_4)^* = (L_1^* L_2^* L_3^* L_4^*)^*$。

□

Exercise 2.7. 设 L_1, L_2, L_3, L_4 分别是 $\Sigma_1, \Sigma_2, \Sigma_3, \Sigma_4$ 上的语言，证明下列等式成立否。

$$L_2(L_1L_2 \cup L_2)^* L_1 = L_1L_1^*L_2(L_1L_1^*L_2)^*$$

Solution 2.7. 证明. 此式不成立，仅举一个反例即可完成证明。

令 $L_1 = \{a\}, L_2 = \{b\}$, 此时，

$L_2(L_1L_2 \cup L_2)^* L_1 = \{b\}(\{a\}\{b\} \cup \{b\})^*\{a\} = \{b\}\{ab, b\}^*\{a\}$

它含的串都是以 $\{b\}\{a\}$。

$L_1L_1^*L_2(L_1L_1^*L_2)^* = \{a\}\{a\}^*\{b\}(\{a\}\{a\}^*\{b\})^* = \{a\}^+\{b\}(\{a\}^+\{b\})^*$

它含的串都都是以 $\{a\}$ 开头的串。

所以，此式不成立。 □

Exercise 2.8. 设 $\Sigma = \{0,1\}$ 请给出 Σ 上的下列语言的形式化表示。

1. 所有长度为偶数的串。
2. 所有含有 3 个连续 0 的串。
3. 所有的倒数第 10 个字符是 0 的串。

Solution 2.8. $\Sigma = \{0,1\}$

1. 所有长度为偶数的串。可以用以下任一种表示方法 (含 $\varepsilon, |\varepsilon| = 0$, 认为是偶数)：

   a. $(\{0,1\}\{0,1\})^*$

   b. $(\{00,01,10,11\})^*$

   c. $\{00,01,10,11\} \cup (\{0,1\}\{0,1\})^*$

2. 所有含有 3 个连续 0 的串。

   $\{0,1\}^*000\{0,1\}^*$

3. 所有的倒数第 10 个字符是 0 的串。

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