

第 5 章

正则语言的性质

前面曾经提到, $\{0^n 1^n \mid n \geq 1\}$ 不是正则语言(RL)。这并不是因为某人构造不出产生该语言的正则文法(RG)、有穷状态自动机(FA)或者正则表达式(RE)等,而是从 RL 的性质出发,可以证明该语言不具有 RL 的性质,所以不存在这些形式的描述。那么,RL 具有什么样的性质呢? 本章将首先对此进行讨论,包括 RL 的泵引理及其应用;RL 关于并、乘积、闭包、补、交、正则代换、同态、逆同态等运算的封闭性。

另一个问题是对一个给定的 RL,存在多个产生该语言的正则文法,多个识别该语言的有穷状态自动机和多个表达该语言的正则表达式。由于它们描述的是同一个语言,所以,它们“本质上”应该是一样的。因此,人们希望能找到一种方法,给出 RL“本质上”的描述——该语言所决定的相应字母表的克林闭包的等价分类,然后根据这个分类,构造出接受这个给定 RL 的、状态最少的、确定的有穷状态自动机——最小 DFA。由于语言是给定的,所以相应的最小 DFA 在规定的意义(如同构)下应该是唯一的。最后将讨论如何根据 DFA 去构造其相应的最小 DFA,并且此方法是可以自动化的。这一部分的内容包括:右不变的等价关系、DFA 所确定的等价关系与语言确定的等价关系的右不变性,Myhill-Nerode 定理的证明与应用,DFA 的极小化。

在有了语言 L 的最小 DFA 后,去掉其相应的陷阱状态,可以方便地得到较为简洁的正则文法。当然,从最小 DFA 出发,寻求 L 的正则表达式也可能对构造是非常有利的。

最后,本章还将讨论 RL 的判定算法。

5.1 正则语言的泵引理

从前面的讨论知道,任何有穷语言都是 RL。所以,非 RL 一定是无穷语言。因此,我们只讨论无穷语言是否为 RL 的判定问题。如何判定一个语言不是 RL 呢? 这需要从 RL 的“本质”特性出发进行讨论。

DFA 是 RL 的识别模型。一个 DFA 只有有穷个状态,这就是说,当该 DFA 识别的语言 L 是无穷语言时, L 中必定存在一个足够长的句子,使得 DFA 在识别该句子的过程中,肯定要重复地经过某些状态。

例如句子 $z = a_1 a_2 \cdots a_m$,不妨假设 DFA 在识别它的过程中需要经过的状态依次为 $q_0, q_1, \cdots, q_m$ 。根据重叠原理,当 m 大于或等于 DFA 所有可达状态的个数时,这些状态中至少

有一对是重复的,如 q_k 和 q_j 。显然,这里的 $k \neq j$ 成立,不妨设 $k < j$ 。设 $v = a_{k+1} \cdots a_j$ 是引导 DFA 从状态 q_k 到达状态 q_j 的子串,则它就是该 DFA 中从 q_0 到 q_m 的标记为 x 的路中从 q_k 到 q_j 的标记为 v 的回路,因此, v 在它出现的位置无论重复多少次,所构成的字符串都一定是 DFA 所识别的语言的句子,如图 5-1 所示。

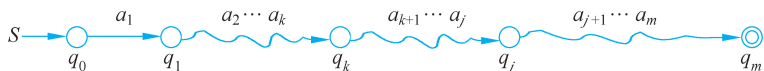


图 5-1 DFA 处理句子中经历的状态序列

由于 q_k 和 q_j 是同一个状态,为了便于理解,将该图重画成如图 5-2 所示的样子。

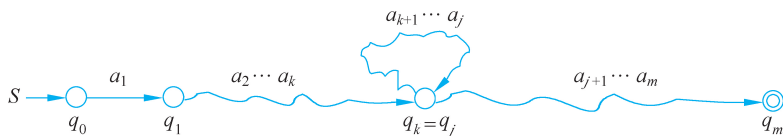


图 5-2 DFA 处理句子经历的状态序列中存在重复的状态

根据重叠原理,这样的 q_k 与 q_j 在序列 $q_0, q_1, \dots, q_N$ 中一定是存在的,其中, N 为相应 DFA 的可达状态个数。下面给出这种现象的严格描述,并给出判定一个语言不是 RL 的一般方法。

设 L 是一个 RL,DFA

$$M = (Q, \Sigma, \delta, q_0, F)$$

满足

$$L(M) = L, |Q| = N$$

不失一般性,不妨假设 Q 中不含任何不可达状态,并且 $|Q| = N$ 。取 L 的句子

$$z = a_1 a_2 \cdots a_m (m \geq N)$$

对 $\forall h, 1 \leq h \leq m$, 令

$$\delta(q_0, a_1 a_2 \cdots a_h) = q_h$$

由于 $m \geq N$, 所以,在状态序列 $q_0, q_1, \dots, q_N$ 中至少有两个状态是相同的。不妨假设 q_k 与 q_j 是该状态序列中最早出现的相同状态:

$$q_k = q_j$$

显然,

$$k < j \leq N$$

此时有

$$\delta(q_0, a_1 a_2 \cdots a_k) = q_k$$

$$\delta(q_k, a_{k+1} \cdots a_j) = q_j = q_k$$

$$\delta(q_j, a_{j+1} \cdots a_m) = q_m$$

注意到 $q_j = q_k$, 所以对于任意整数 $i \geq 0$,

$$\begin{aligned} \delta(q_k, (a_{k+1} \cdots a_j)^i) &= \delta(q_k, (a_{k+1} \cdots a_j)^{i-1}) \\ &\vdots \\ &= \delta(q_k, a_{k+1} \cdots a_j) \\ &= q_k \end{aligned}$$

因为

$$z \in L(M)$$

所以

$$q_m \in F$$

故,对于任意整数 $i \geq 0$,

$$\delta(q_0, a_1 a_2 \cdots a_k (a_{k+1} \cdots a_j)^i a_{j+1} \cdots a_m) = q_m$$

也就是说,

$$a_1 a_2 \cdots a_k (a_{k+1} \cdots a_j)^i a_{j+1} \cdots a_m \in L(M)$$

取

$$u = a_1 a_2 \cdots a_k$$

$$v = a_{k+1} \cdots a_j$$

$$w = a_{j+1} \cdots a_m$$

对于任意整数 $i \geq 0$,

$$uv^i w \in L$$

注意到 $k < j \leq N$, 所以 u, v 满足如下条件:

$$|uv| \leq N$$

$$|v| \geq 1$$

再注意到讨论中的 DFA M 的任意性, 所以, 该结论对最小 DFA 也成立。从而得到如下引理。

引理 5-1 设 L 为一个 RL, 则存在仅依赖于 L 的正整数 N , 对于 $\forall z \in L$, 如果 $|z| \geq N$, 则存在 u, v, w , 满足:

- (1) $z = uvw$ 。
- (2) $|uv| \leq N$ 。
- (3) $|v| \geq 1$ 。
- (4) 对于任意整数 $i \geq 0$, $uv^i w \in L$ 。
- (5) N 不大于接受 L 的最小 DFA M 的状态数。

引理 5-1 被称为关于 RL 的**泵引理**(pumping lemma)。由于它是如此的重要和有影响, 以至于人们一直将其称为引理而未改称其为定理。

例 5-1 证明 $\{0^n 1^n \mid n \geq 1\}$ 不是 RL。

证明: 设 $L = \{0^n 1^n \mid n \geq 1\}$ 。假设 L 是 RL, 则它满足泵引理。不妨设 N 是泵引理所指的仅依赖于 L 的正整数, 取

$$z = 0^N 1^N$$

显然, $z \in L$ 。按照泵引理所述, 必存在 u, v, w 。由于 $|uv| \leq N$, 并且 $|v| \geq 1$, 所以 v 只可能是由 0 组成的非空串。不妨设

$$v = 0^k, \quad k \geq 1$$

此时有

$$u = 0^{N-k-j}$$

$$w = 0^j 1^N$$

从而有

$$uv^i w = 0^{N-k-j} (0^k)^i 0^j 1^N = 0^{N+(i-1)k} 1^N$$

当 $i=2$ 时,有

$$uv^2 w = 0^{N+(2-1)k} 1^N = 0^{N+k} 1^N$$

注意到 $k \geq 1$, 所以

$$N+k > N$$

这就是说,

$$0^{N+k} 1^N \notin L$$

这与泵引理矛盾, 所以, L 不是 RL。

例 5-2 证明 $\{0^n \mid n \text{ 为素数}\}$ 不是 RL。

证明: 设 $L = \{0^n \mid n \text{ 为素数}\}$ 。假设 L 是 RL, 则它满足泵引理。不妨设 N 是泵引理所指的仅依赖于 L 的正整数, 取

$$z = 0^{N+p}$$

其中, $N+p$ 是素数, 即 $z \in L$ 。

按照泵引理所述, 必存在 u, v, w 。由于 $|v| \geq 1$, 所以 v 必定是由 0 组成的非空串。不妨设

$$v = 0^k, \quad k \geq 1$$

此时有

$$u = 0^{N+p-k-j}$$

$$w = 0^j$$

从而有

$$uv^i w = 0^{N+p-k-j} (0^k)^i 0^j = 0^{N+p+(i-1)k}$$

由于是要证明 L 不是 RL, 因此这里要能找到 i 的一个值, 它能表明串 $0^{N+p+(i-1)k}$ 不是 L 的句子。对本例而言, 就是要表明 $N+p+(i-1)k$ 不是素数。显然, 当 $i = N+p+1$ 时, 有

$$\begin{aligned} N+p+(i-1)k &= N+p+(N+p+1-1)k \\ &= N+p+(N+p)k \\ &= (N+p)(1+k) \end{aligned}$$

注意到 $k \geq 1$, 所以

$$N+p+(N+p+1-1)k = (N+p)(1+k)$$

不是素数。故当 $i = N+p+1$ 时,

$$uv^{N+p+1} w = 0^{(N+p)(1+k)}$$

不是 L 的句子:

$$uv^{N+p+1} w \notin L$$

这与泵引理矛盾。所以, L 不是 RL。

例 5-3 证明 $\{0^n 1^m 2^{n+m} \mid m, n \geq 1\}$ 不是 RL。

证明: 设 $L = \{0^n 1^m 2^{n+m} \mid m, n \geq 1\}$ 。假设 L 是 RL, 则它满足泵引理。不妨设 N 是泵引理所指的仅依赖于 L 的正整数(实际上, 这个 N 是不存在的), 取

$$z = 0^N 1^N 2^{2N}$$

显然, $z \in L$ 。按照泵引理所述, 必存在 u, v, w 。由于 $|uv| \leq N$, 并且 $|v| \geq 1$, 所以 v 只能是由 0 组成的非空串。不妨设

$$v = 0^k, \quad k \geq 1$$

此时有

$$u = 0^{N-k-j}$$

$$w = 0^j 1^N 2^{2N}$$

从而有

$$uv^i w = 0^{N-k-j} (0^k)^i 0^j 1^N 2^{2N} = 0^{N+(i-1)k} 1^N 2^{2N}$$

与前面两个例题一样, 由于是要证明 L 不是 RL, 因此, 这里要能找到 i 的一个值, 它能表明串 $0^{N+(i-1)k} 1^N 2^{2N}$ 不是 L 的句子。对本例而言, 也就是要找 i 的一个值, 它使得所得字符串中 0 和 1 的个数之和不等于 2 的个数, 即 $N+(i-1)k+N \neq 2N$ 。显然, 当 $i=0$ 时, 有

$$uv^0 w = 0^{N+(0-1)k} 1^N 2^{2N} = 0^{N-k} 1^N 2^{2N}$$

注意到 $k \geq 1$, 所以

$$N-k+N = 2N-k < 2N$$

这就是说,

$$0^{N-k} 1^N 2^{2N} \notin L$$

这与泵引理矛盾。所以, L 不是 RL。

通过上述 3 个例子, 可以更清楚地看出, 引理 5-1 是用来证明一个语言不是 RL 的。因为它只是说 RL 必定满足这些条件, 并没有说满足条件的语言是 RL。即引理给出的是 RL 的“必要条件”而不是“充分条件”。所以, 不能用泵引理去证明一个语言是 RL。此外, 在使用泵引理证明一个给定语言不是 RL 时, 需要注意如下几方面的问题:

(1) 由于泵引理给出的是 RL 的必要条件, 所以, 在它证明一个语言不是 RL 时, 使用反证法。

(2) 泵引理说的是对 RL 都成立的条件, 而要用它证明给定语言不是 RL。也就是说, 相应语言的“仅仅依赖于 L 的正整数 N ”实际上是不存在的。所以, 一定是无法给出一个具体数的。因此, 往往就用符号 N 来表示这个“假定存在”而实际并不存在的数。

(3) 泵引理指出, 如果 L 是 RL, 则对任意 $z \in L$, 只要 $|z| \geq N$, 一定会存在 u, v, w , 使得 $uv^i w \in L$ 对所有的 i 成立。因此, 在选择 z 时, 就需要注意到论证时的简洁和方便。

例如, 例 5-3 中如果不取 $z = 0^N 1^N 2^{2N}$, 而是取 $z = 0^n 1^m 2^{n+m}$, 并且只要求 $2(n+m) \geq N$, 那么要证明不存在这样的 u, v, w , 除了需要讨论 $v = 0^k$ 的情况外, 还需要分别讨论 $v = 1^k, v = 2^k, v = 0^h 1^k, v = 1^h 2^k$ 等情况。这里, $h, k \geq 1$ 。显然, 这些附加的讨论在取 $z = 0^N 1^N 2^{2N}$ 时, 都被避免了。

再例如, 设

$$L = \{x \mid x \text{ 为 0 的个数和 1 的个数相等的 0, 1 串}\}$$

L 不是 RL。如果取 $z = (01)^N$, 就无法用泵引理证明 L 不是 RL。因为, 只要取 $v = (01)^k, N \geq k \geq 1$, 就能保证对于任意整数 $i \geq 0, uv^i w \in L$ 成立。这样的 z 还很多。但是, 如果取 $z = 0^N 1^N$, 相应的证明就变得与例 5-1 一样容易了。由此可见, 要力求寻找这样的 z , z 中不存在满足条件的子串 v , 使得当 v 被“泵进”或者“泵出”时, 字符串 $uv^i w$ 总保持 L 的句

子特征。

综上所述,在选择 z 时,应该选择那些既可以找出矛盾,又能使证明尽可能简单的特殊的 z (对“所有 z ”的否定)。事实上,在用泵引理证明一个语言不是 RL 的过程中, z 的选取是最困难的。一旦选出了一个恰当的 z ,证明基本上就可以顺利地进行下去,因为剩余的叙述基本上是“按部就班”地进行。

(4) 当一个特意被选来用作“发现矛盾”的 z 确定以后,就必须说明满足条件 $|uv| \leq N$ 和 $|v| \geq 1$ 的所有 v 都不能使 $uv^i w \in L$ 对所有 $i \geq 0$ 成立 (对“存在 u, v, w ”的否定)。

(5) 与选 z 时类似,在寻找 i 时,也仅需要找到一个表明矛盾的“具体”值就可以了 (对“所有 i ”的否定)。

(6) 一般地,在证明一个语言不是 RL 的时候,并不使用泵引理的第(5)条。

(7) 事实上,引理所要求的 $|uv| \leq N$ 并不是必需的。但它为简化相应的证明提供了良好的支撑。但是,从引理的证明可以看出,对于一个 RL,当它的句子 z 足够长时,对 z 的任意一个长度为 N 的子串 z_1 ,同样可以在这个子串中找到可以被“泵进”或者“泵出”的非空子串 v 。这会给证明带来更多的方便。例如,关于 $\{0^n 1^m 2^m \mid n, m \geq 1\}$ 不是 RL 的证明就是如此。这种证明方法被称为扩充了的泵引理 (见相关习题)。当然,有时会需要使用 RL 所具有的其他性质来完成相应的证明。有关内容稍后讨论。

(8) 除了需要证明一个语言不是 RL 外,有时也希望证明一个语言是 RL。此时,最直接的方法是给出该语言的正则文法描述,或者 FA 和 RE 描述。但是有时候直接用一些已有的结果和 RL 的性质会更有效,毕竟语言的正则文法描述或者 FA 和 RE 描述并不总是那么容易给出的。

5.2 正则语言的封闭性

本节讨论 RL 对有关运算的封闭性。

定义 5-1 如果任意的、属于某一语言类的语言在某一特定运算下所得的结果仍然是该类语言,则称该语言类对此运算是**封闭**的,并称该语言类对此运算具有**封闭性** (closure property)。

给定一个语言类的若干语言的描述。如果存在一个算法,它可以构造出这些语言在给定运算下所获得的运算结果的相应形式的语言描述,则称此语言类对相应的运算是**有效封闭**的,并称此语言类对相应的运算具有**有效封闭性** (valid closure property)。

根据正则表达式的定义,立即可以得到以下定理。

定理 5-1 RL 在并、乘积、闭包运算下是封闭的。

定理 5-2 RL 在补运算下是封闭的。

证明: 设 L 是 Σ 上的一个 RL,从而存在一个 DFA

$$M = (Q, \Sigma, \delta, q_0, F)$$

满足

$$L(M) = L$$

取 DFA

$$M' = (Q, \Sigma, \delta, q_0, Q - F)$$

显然, 对于任意 $x \in \Sigma^*$,

$$\delta(q_0, x) = f \in F \Leftrightarrow \delta(q_0, x) = f \notin Q - F$$

即

$$x \in L(M) \Leftrightarrow x \notin L(M')$$

这就是说,

$$L(M') = \Sigma^* - L(M)$$

所以, RL 在补运算下是封闭的。定理得以证明。

定理 5-3 RL 在交运算下封闭。

证明: 该定理由定理 5-1、定理 5-2 以及 De Morgan 定理可以推得。

对于给定的 DFA $M_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1)$ 和 DFA $M_2 = (Q_2, \Sigma, \delta_2, q_{02}, F_2)$, 如何构造出 DFA M , 使得 $L(M) = L(M_1) \cap L(M_2)$ 呢? 显然, 对于一个输入字符串, 不可能让其中一个 DFA “先工作”, 等它确认该字符串是它接受的句子后, 再让另一个 DFA 启动。因为, DFA 并不是 2DFA, 所以, 对一个输入串, 不可能进行两次扫描。实际上, 可以按照如下思路去考虑: 对 Σ^* 中的任意字符串 $a_1 a_2 \cdots a_m$, M_1 和 M_2 在处理 $a_1 a_2 \cdots a_m$ 的过程中一定分别经过状态序列 $q_{01}, q_{11}, \cdots, q_{m1}$ 和 $q_{02}, q_{12}, \cdots, q_{m2}$ 。这两个状态序列依据 $a_1 a_2 \cdots a_m$ 中字符的出现顺序和相应的移动函数有如图 5-3 所示的对应关系。根据状态及状态变换的这种对应关系, 就可以知道如何根据 M_1 和 M_2 去构造 M 。事实上, 对于任意一个输入字符串, 所构造的 M 必须同时能够模拟 M_1 和 M_2 对该输入字符串的处理过程。而且要求, 对 M_1 行为的模拟不应该受到对 M_2 行为的模拟的影响; 反之, 对 M_2 行为的模拟也不应该受到对 M_1 行为的模拟的影响。当 $a_1 a_2 \cdots a_m$ 是 $L(M_1) \cap L(M_2)$ 中的句子时, 有 $q_{m1} \in F_1, q_{m2} \in F_2$ 。当 $a_1 a_2 \cdots a_m \notin L(M_1) \cap L(M_2)$, 但 $a_1 a_2 \cdots a_m \in L(M_1) \cup L(M_2)$ 时, $q_{m1} \in F_1, q_{m2} \in F_2$ 有且仅有一个成立。当 $a_1 a_2 \cdots a_m \notin L(M_1) \cup L(M_2)$ 时, $q_{m1} \notin F_1$ 和 $q_{m2} \notin F_2$ 同时成立。根据这里的讨论, 读者不难给出 M 的详细描述。

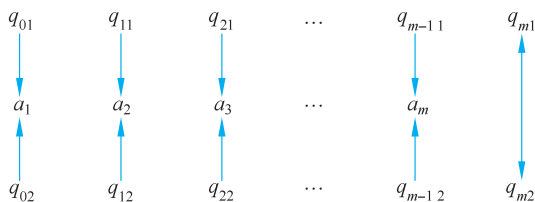


图 5-3 M_1 和 M_2 对应于同一个输入串的状态变换过程

上述讨论实际上涉及利用已有的(子)系统去构造一个新系统, 按照上述讨论, 很容易构造出 DFA M , 使得 $L(M) = L(M_1) \cap L(M_2)$ 。

下面再考虑另外一个问题。对字母表 Σ 上的 RL L , 令 DFA $M = (Q, \Sigma, \delta, q_0, F)$ 识别 L 。对于任意 $(q, a) \in Q \times \Sigma$, 设 $\delta(q, a) = p$, $f(a)$ 是 Δ 上的一个 RL, 假定 DFA $M_a = (Q_a, \Delta, \delta_a, q_{0a}, F_a)$ 识别 $f(a)$ 。不妨假设: 对 $a, b \in \Sigma$, 如果 $a \neq b$, 则 $Q_a \cap Q_b = \emptyset, Q_a \cap Q = \emptyset$ 。容易构造一个 FA M_{RS} , 它的“主框架”是 M , 而它的“分支子模块”为 M_a 。如果 M 在状态 q 读入字符 a 时进入状态 p , 则让 M_{RS} 在状态 q 做一个空移动转到 M_a 的开始状态 q_{0a} , M_{RS} 在

“分支子模块” M_a 中处理 $f(a)$ 的句子。当这个句子被处理完后,它一定处在 M_a 的某一个终止状态。从 M_a 的每一个终止状态到 M 的状态 p 引一条标记为 ϵ 的弧,使 M_{RS} 回到状态 p 。如此下去,直到最后到达 M 的终止状态。可见, M_{RS} 的状态集为 $Q \bigcup_{a \in \Sigma} Q_a$, 它的终止状态即为 M 的终止状态集 F 。从而,有 FA

$$M_{RS} = (Q \bigcup_{a \in \Sigma} Q_a, \Delta, \delta_{RS}, q_0, F)$$

使得 $L(M_{RS}) = f(L)$ 。可见,RL 在这种运算下也是封闭的。称这种运算为正则代换。

定义 5-2 设 Σ, Δ 是两个字母表,映射

$$f: \Sigma \rightarrow 2^{\Delta^*}$$

称为是从 Σ 到 Δ 的**代换**(substitution)。如果对于 $\forall a \in \Sigma, f(a)$ 是 Δ 上的 RL,则称 f 为**正则代换**(regular substitution)。

先将 f 的定义域扩展到 Σ^* 上, $f: \Sigma^* \rightarrow 2^{\Delta^*}$ 。

$$(1) f(\epsilon) = \{\epsilon\}。$$

$$(2) f(xa) = f(x)f(a)。$$

再将 f 的定义域扩展到 2^{Σ^*} 上, $f: 2^{\Sigma^*} \rightarrow 2^{\Delta^*}$ 。对于 $\forall L \subseteq \Sigma^*$,

$$f(L) = \bigcup_{x \in L} f(x)$$

例 5-4 设 $\Sigma = \{0, 1\}, \Delta = \{a, b\}, f(0) = a, f(1) = b^*$, 则

$$f(010) = f(0)f(1)f(0)$$

$$= ab^*a$$

$$f(\{11, 00\}) = f(11) \cup f(00)$$

$$= f(1)f(1) \cup f(0)f(0)$$

$$= b^*b^* + aa$$

$$= b^* + aa$$

$$f(L(0^*(0+1)1^*)) = L(a^*(a+b^*)(b^*)^*)$$

$$= L(a^*(a+b^*)b^*)$$

$$= L(a^*ab^* + a^*b^*b^*)$$

$$= L(a^*b^*)$$

定义 5-3 设 Σ, Δ 是两个字母表,映射

$$f: \Sigma \rightarrow 2^{\Delta^*}$$

为正则代换,则

$$(1) f(\emptyset) = \emptyset。$$

$$(2) f(\epsilon) = \epsilon。$$

(3) 对于 $\forall a \in \Sigma, f(a)$ 是 Δ 上的正则表达式。

(4) 如果 r, s 是 Σ 上的正则表达式,则

$$f(r+s) = f(r) + f(s)$$

$$f(rs) = f(r)f(s)$$

$$f(r^*) = f(r)^*$$

是 Δ 上的正则表达式。

定理 5-4 设 L 是 Σ 上的一个 RL, $f: \Sigma \rightarrow 2^{\Delta^*}$ 是正则代换, 则 $f(L)$ 也是 RL。

证明: 前面曾经非形式地简要介绍了根据接受 L 的 DFA M 以及接受 $f(a)$ 的 DFA M_a 构造接受 $f(L)$ 的 FA 的思路, 有关构造和证明的严格描述请读者给出。在这里, 用正则表达式来进行定理的证明。

因为 L 是 RL, 则存在正则表达式 r , 使得 $L(r) = L$ 。

下面对 r 中运算符的个数 n 实施归纳, 证明 $f(r)$ 是表示 $f(L)$ 的正则表达式。即

$$f(L(r)) = L(f(r))$$

当 $n=0$ 时, 由定义 5-2 和定义 5-3, 结论成立。

设当 $n \leq k$ 时定理成立, 即当 r 中运算符的个数不大于 k 时, $f(L(r)) = L(f(r))$ 。当 $n = k+1$ 时有如下 3 种情况:

(1) $r = r_1 + r_2$ 。

$$\begin{aligned} f(L) &= f(L(r)) \\ &= f(L(r_1 + r_2)) \\ &= f(L(r_1) \cup L(r_2)) && \text{正则表达式的定义} \\ &= f(L(r_1)) \cup f(L(r_2)) && \text{正则代换的定义} \\ &= L(f(r_1)) \cup L(f(r_2)) && \text{归纳假设} \\ &= L(f(r_1) + f(r_2)) && \text{正则表达式的定义} \\ &= L(f(r_1 + r_2)) && \text{正则表达式的正则代换的定义} \\ &= L(f(r)) \end{aligned}$$

(2) $r = r_1 r_2$ 。

$$\begin{aligned} f(L) &= f(L(r)) \\ &= f(L(r_1 r_2)) \\ &= f(L(r_1) L(r_2)) && \text{正则表达式的定义} \\ &= f(L(r_1)) f(L(r_2)) && \text{正则代换的定义} \\ &= L(f(r_1)) L(f(r_2)) && \text{归纳假设} \\ &= L(f(r_1) f(r_2)) && \text{正则表达式的定义} \\ &= L(f(r_1 r_2)) && \text{正则表达式的正则代换的定义} \\ &= L(f(r)) \end{aligned}$$

(3) $r = r_1^*$ 。

$$\begin{aligned} f(L) &= f(L(r)) \\ &= f(L(r_1^*)) \\ &= f(L(r_1))^* && \text{正则表达式的定义} \\ &= (f(L(r_1)))^* && \text{正则代换的定义} \\ &= (L(f(r_1)))^* && \text{归纳假设} \\ &= L(f(r_1)^*) && \text{正则表达式的定义} \\ &= L(f(r_1^*)) && \text{正则表达式的正则代换的定义} \\ &= L(f(r)) \end{aligned}$$

所以, 结论对 $n = k+1$ 成立。由归纳法原理, 结论对任意正则表达式成立。

定理得到证明。

注意,定理 5-1 可以作为本定理的直接推论。

例 5-5 设 $\Sigma = \{0, 1, 2\}$, $\Delta = \{a, b\}$, 正则代换 f 定义为

$$\begin{aligned} f(0) &= ab \\ f(1) &= b^* a^* \\ f(2) &= a^* (a + b) \end{aligned}$$

则

- (1) $f(00) = abab$ 。
- (2) $f(010) = abb^* a^* ab = ab^+ a^+ b$ 。
- (3) $f((0+1+2)^*) = (ab + b^* a^* + a^* (a + b))^* = (b^* a^* + a^* (a + b))^* = (a + b)^*$ 。
- (4) $f(0(0+1+2)^*) = ab(ab + b^* a^* + a^* (a + b))^* = ab(a + b)^*$ 。
- (5) $f(012) = abb^* a^* a^* (a + b) = ab^+ a^* (a + b)$ 。
- (6) $f((0+1)^*) = (ab + b^* a^*)^* = (ab + b + a + b^* a^*)^* = (a + b)^*$ 。

定义 5-4 设 Σ, Δ 是两个字母表, $f: \Sigma \rightarrow \Delta^*$ 为映射。如果对于 $\forall x, y \in \Sigma^*$, 有

$$f(xy) = f(x)f(y)$$

则称 f 为从 Σ 到 Δ^* 的**同态映射**(homomorphism)。

对于 $\forall L \subseteq \Sigma^*$, L 的同态像

$$f(L) = \bigcup_{x \in L} \{f(x)\}$$

对于 $\forall w \subseteq \Delta^*$, w 的同态原像是一个集合:

$$f^{-1}(w) = \{x \mid f(x) = w \text{ 且 } x \in \Sigma^*\}$$

对于 $\forall L \subseteq \Delta^*$, L 的同态原像是一个集合:

$$f^{-1}(L) = \{x \mid f(x) \in L\}$$

例 5-6 设 $\Sigma = \{0, 1\}$, $\Delta = \{a, b\}$, 同态映射 f 定义为

$$\begin{aligned} f(0) &= aa \\ f(1) &= aba \end{aligned}$$

则

- (1) $f(01) = aaaba$ 。
- (2) $f((01)^*) = (aaaba)^*$ 。
- (3) $f^{-1}(aab) = \emptyset$ 。
- (4) $f^{-1}(aa) = \{0\}$ 。
- (5) $f^{-1}(\{aaa, aba, abaaaaa, abaaaaaa\}) = \{1, 100\}$ 。
- (6) $f^{-1}((ab+ba)^* a) = \{1\}$ 。
- (7) $f(f^{-1}((ab+ba)^* a)) = f(\{1\}) = \{aba\}$ 。

令 $L = (ab + ba)^* a$, 上述(7)表明, $f(f^{-1}(L)) \neq L$ 。但是, 不难证明, 对任意语言 L 和同态映射 f ,

$$f(f^{-1}(L)) \subseteq L$$

注意到同态映射是正则代换的特例, 所以有以下推论和定理。