Chapter 4

正则语言的性质

4.1 证明语言的非正则性

例 1. $L = \{0^m1^n \mid m, n \geq 0\}$ 是否是正则语言?

例 2. $L = \{0^m1^n \mid m \geq 2, n \geq 4\}$ 是否是正则语言?

例 3. $L_{01} = \{0^n1^n | n \geq 0\}$ 是否是正则语言?

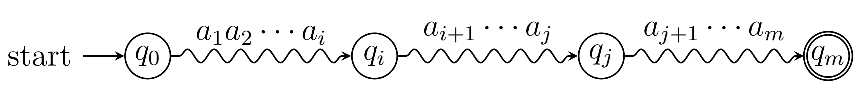
4.1.1 正则语言的泵引理

定理 5 (正则语言的泵引理). *如果语言 L 是正则的, 那么存在正整数 N, 对 $\forall w \in L$, 只要 $|w| \geq N$, 就可以将 w 分为三部分 $w = xyz$ 满足:*

1. *$y \neq \varepsilon$ ($|y| > 0$);*
2. *$|xy| \leq N$;*
3. *$\forall k \geq 0$, $xy^kz \in L$.*

证明:

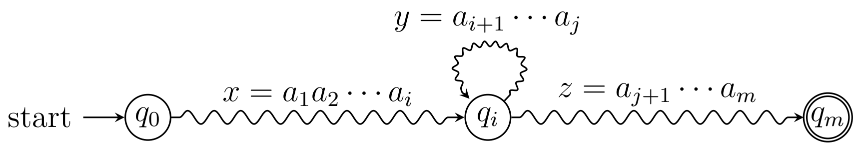
1. 如果 L 正则, 那么存在有 n 个状态 DFA A 使 $\mathbf{L}(A) = L$;
2. 取 $w = a_1 \dots a_m \in L$ ($m \geq n$), 定义 $q_i = \hat{\delta}(q_0, a_1 \dots a_i)$; q_0 是开始状态, 当 A 输入 w 的前 n 个字符时, 经过的状态分别是 $q_0, q_1, \cdots, q_n$ 共 $n + 1$ 个;



3. 由鸽巢原理, 必有两状态相同 $q_i = q_j$ $(0 \le i < j \le n)$; 由 q_i 和 q_j 将 w 分为

$$x = a_1 a_2 \cdots a_i$$
$$y = a_{i+1} a_{i+2} \cdots a_j$$
$$z = a_{j+1} a_{j+2} \cdots a_m$$

4. 那么 $w = xyz$ 如图, 且有 $\forall k > 0,\ xy^k z \in L$;



   因为如果从 q_i 出发, 输入 y, 会到达 q_j, 而 $q_i = q_j$, 所以当输入 $y^k (k \ge 0)$ 时, 始终会回到 q_i. 所以当 DFA A 输入 $xy^k z$ 时, 由 q_0 始终会达到 q_m. 那么, 如果 $xyz \in \mathbf{L}(A)$, 一定有 $xy^k z \in \mathbf{L}(A)$ 对所有 $k \ge 0$ 成立.

5. 而因为 $i < j$ 所以 $y \ne \varepsilon$ (即 $|y| > 0$), 因为 $j \le n$ 所以 $|xy| \le n$. □

任何从开始状态到接受状态的路径, 如果长度超过 n, 一定会经过 $n+1$ 个状态, 必定有一个重复状态, 因此会形成一个循环 (loop); 那么, 这个循环可以被重复多次后, 沿原路径还会到达接收状态. 泵引理中的 N, 是正则语言固有存在的.

泵引理可以用来确定特定语言不在给定语言类 (正则语言) 中. 但是它们不能被用来确定一个语言在给定类中, 因为满足引理是类成员关系的必要条件, 但不是充分条件.

4.1.2 泵引理的应用

续例 3. 证明 $L_{01} = \{0^n 1^n | n \ge 0\}$ 不是正则语言.

证明:

1. 假设 L_{01} 是正则的.
2. 那么, 存在 $N \in \mathbb{Z}^+$, 对 $\forall w \in L_{01}$ $(|w| \ge N)$ 满足泵引理.
3. 从 L_{01} 中取 $w = 0^N 1^N$, 显然 $w \in L_{01}$ 且 $|w| = 2N \ge N$.
4. 那么, w 可被分为 $w = xyz$, 且 $|xy| \le N$ 和 $y \ne \varepsilon$.
5. 因此 y 只能是 0^m 且 $m > 0$.
6. 那么 $xy^2 z = 0^{N+m} 1^N \notin L_{01}$, 而由泵引理 $xy^2 z \in L_{01}$, 矛盾.
7. 所以假设不成立, L_{01} 不是正则的. □

例 4. 证明 $L_{\text{eq}} = \{w \in \{0,1\}^* \mid w \text{ 由数量相等的 } 0 \text{ 和 } 1 \text{ 构成}\}$ 不是正则的.

思考题

刚刚已经证明了

$$L_{01} = \{0^n 1^n \mid n \geq 0\}$$

不是正则语言, 那么能否使用

$$L_{01} \subseteq L_{\text{eq}}$$

来说明 L_{eq} 也不是正则的呢?

续例 4. 证明 $L_{\text{eq}} = \{w \in \{0,1\}^* \mid w \text{ 由数量相等的 } 0 \text{ 和 } 1 \text{ 构成}\}$ 不是正则的.

证明:

1. 假设 L_{eq} 是正则的.
2. 那么, 存在 $N \in \mathbb{Z}^+$, 对 $\forall w \in L_{\text{eq}}\,(|w| \geq N)$ 满足泵引理.
3. 从 L_{eq} 中取 $w = 0^N 1^N$, 显然 $w \in L_{\text{eq}}$ 且 $|w| = 2N \geq N$.
4. 那么, w 可被分为 $w = xyz$, 且 $|xy| \leq N$ 和 $y \neq \varepsilon$.
5. 因此 y 只能是 0^m 且 $m > 0$.
6. 那么 $xy^2z = 0^{N+m}1^N \notin L_{\text{eq}}$, 而由泵引理 $xy^2z \in L_{\text{eq}}$, 矛盾.
7. 所以假设不成立, L_{eq} 不是正则的. □

例 5. 证明 $L = \{0^i 1^j \mid i > j\}$ 不是正则的.

证明:

1. 假设 L 是正则的.
2. 那么, 存在 $N \in \mathbb{Z}^+$, 对 $\forall w \in L\,(|w| \geq N)$ 满足泵引理.
3. 从 L 中取 $w = 0^{N+1}1^N$, 则 $w \in L$ 且 $|w| = 2N+1 \geq N$.
4. 由泵引理, w 可被分为 $w = xyz$, 且 $|xy| \leq N$ 和 $y \neq \varepsilon$.
5. 那么, y 只能是 0^m 且 $m \geq 1$.
6. 那么, $xz = xy^0z = 0^{N+1-m}1^N \notin L$, 因为 $N+1-m \leq N$, 而由泵引理 $xy^0z \in L$, 矛盾.
7. 所以假设不成立, L 不是正则的. □

例 6. Prove $L = \{a^3b^nc^{n-3} \mid n \geq 3\}$ is not regular.

证明:

1. 假设 L 是正则的.
2. 那么, 存在 $N \in \mathbb{Z}^+$, 对 $\forall w \in L\,(|w| \geq N)$ 满足泵引理.
3. 从 L 中取 $w = a^3b^Nc^{N-3}$, 则 $w \in L$ 且 $|w| = 2N \geq N$.
4. 由泵引理, w 可被分为 $w = xyz$, 且 $|xy| \leq N$ 和 $y \neq \varepsilon$.
5. 那么, 则 y 只可能有 3 种情况 $(m > 0, r > 0, s > 0)$:
   (a) $y = a^m$, 则 $xy^2z = a^{3+m}b^Nc^{N-3} \notin L$;
   (b) $y = b^m$, 则 $xy^2z = a^3b^{N+m}c^{N-3} \notin L$;
   (c) $y = a^rb^s$, 则 $xy^2z = a^3b^sa^rb^Nc^{N-3} \notin L$.
6. 无论 y 为何种情况, xy^2z 都不可能在 L 中, 与泵引理矛盾.
7. 所以假设不成立, L 不是正则的. □

例. 证明 $L = \{a^{n!} \mid n > 0\}$ 不是正则的.

... 取 $w = a^{N!}$, $|y| = m$, $|xy^2z| = N! + m$, 而 $0 \leq m \leq N < N!$, 所以 $N! < |xy^2z| = N! + m < N! + N! < N \cdot N! + N! = (N+1)!$, 即 $|xy^2z|$ 不是阶乘数, ...

思考题

- $L = \{0^n1^n \mid 0 \leq n \leq 100\}$ 是否是正则语言?
- 有限的语言, 是否符合泵引理呢?如 $\emptyset$, $\{\varepsilon\}$, $\{0, 00\}$ 等.

4.1.3 泵引理只是必要条件

即“正则 $\Rightarrow$ 泵引理成立”, 所以“$\neg$ 泵引理成立 $\Rightarrow$ $\neg$ 正则”.

- 泵引理只是正则语言的必要条件
- 只能用来证明某个语言不是正则的
- 与正则语言等价的定理 — Myhill-Nerode Theorem

例 7. 语言 L 不是正则的, 但每个串都可以应用泵引理

$$L = \{ca^n b^n \mid n \geq 1\} \cup \{c^k w \mid k \neq 1, w \in \{a,b\}^*\}$$

- 其中 $A = \{ca^n b^n \mid n \geq 1\}$ 部分不是正则的
- 而 $B = \{c^k w \mid k \neq 1, w \in \{a,b\}^*\}$ 部分是正则的
- 而 A 的任何串 $w = ca^i b^i$, 都可应用泵引理, 因为

  $$w = (\varepsilon)(c)(a^i b^i)$$

  重复字符 c 生成的新串都会落入 B 中

思考题

泵引理中的正整数 N 与 DFA 的状态数 n 之间有何关系? 与 NFA 的状态数之间呢?

思考题

语言

$$L = \{0^n x 1^n \mid n \geq 1, x \in \{0,1\}^*\}$$

是否是正则语言?

4.2 正则语言的封闭性

定义. 正则语言经某些运算后得到的新语言仍保持正则, 称正则语言在这些运算下封闭.

4.2.1 并/连接/闭包

定理 6 (并/连接/闭包的封闭性). 正则语言在并, 连接和闭包运算下保持封闭.

证明: 由正则表达式的定义得证. □

4.2.2 补

定理 7 (补运算封闭性). 如果 L 是 Σ 上的正则语言, 那么 $\overline{L} = \Sigma^* - L$ 也是正则的.

证明:

设接受语言 L 的 DFA

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

即 $\mathbf{L}(A) = L$. 构造 DFA

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

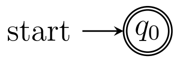
则有 $\overline{L} = \mathbf{L}(B)$, 因为 $\forall w \in \Sigma^*$

$$w \in \overline{L} \iff \hat{\delta}(q_0, w) \notin F \iff \hat{\delta}(q_0, w) \in Q - F \iff w \in \mathbf{L}(B). \quad \square$$

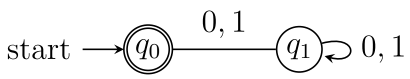
注意

使用这种方法求正则语言的补时, DFA 不能有缺失状态.

例 8. 若 $\Sigma = \{0, 1\}$, $L = \{\varepsilon\}$ 的 DFA 如图, 请给出 $\overline{L}$ 的 DFA.



应使用完整的 DFA 去求补:



思考题

如何求正则表达式的补？

例 9. 证明 L_{neq} =$\{w \mid w$ 由数量不相等的 0 和 1 构成$\}$ 不是正则的.

- 由泵引理不易直接证明 L_{neq} 不是正则的;
- 因为无论如何取 w, 将其分为 $w = xyz$ 时, 都不易产生 L_{neq} 之外的串;
- 而证明 $L_{\text{eq}} = \overline{L_{neq}}$ 非正则很容易;
- 由补运算的封闭性, 所以 L_{neq} 也不是正则的.

定理 8. *若 DFA A_L, A_M 和 A 的定义如下*

$$A_L = (Q_L, \Sigma, \delta_L, q_L, F_L)$$
$$A_M = (Q_M, \Sigma, \delta_M, q_M, F_M)$$

$$A = (Q_L \times Q_M, \Sigma, \delta, (q_L, q_M), F_L \times F_M)$$

其中

$$\begin{aligned} \delta : (Q_L \times Q_M) \times \Sigma &\to Q_L \times Q_M \\ \delta((p, q), a) &= (\delta_L(p, a), \delta_M(q, a)). \end{aligned}$$

则对任意 $w \in \Sigma^*$,

$$\hat{\delta}((q_L, q_M), w) = (\hat{\delta}(q_L, w), \hat{\delta}(q_M, w)).$$

证明: 通过对 w 的归纳来证明.

归纳基础：当 $w = \varepsilon$ 时

$$\begin{aligned} \hat{\delta}((q_L, q_M), \varepsilon) &= (q_L, q_M) && \hat{\delta}\text{ 的定义} \\ &= (\hat{\delta}_L(q_L, \varepsilon), \hat{\delta}_M(q_M, \varepsilon)) && \text{同理} \end{aligned}$$

归纳递推：当 $w = xa$ 时

$$\begin{aligned} \hat{\delta}((q_L, q_M), xa) &= \delta(\hat{\delta}((q_L, q_M), x), a) && \hat{\delta}\text{ 的定义} \\ &= \delta((\hat{\delta}(q_L, x), \hat{\delta}(q_M, x)), a) && \text{归纳假设} \\ &= (\delta_L(\hat{\delta}_L(q_L, x), a), \delta_M(\hat{\delta}_M(q_M, x), a)) && \delta\text{ 的构造} \\ &= (\hat{\delta}_L(q_L, xa), \hat{\delta}_M(q_M, xa)) && \hat{\delta}\text{ 的定义} \quad \square \end{aligned}$$

4.2.3 交

定理 9 (交运算封闭性)**.** *如果 L 和 M 是正则语言, 那么 $L \cap M$ 也是正则语言.*

证明 1: 由 $L \cap M = \overline{\overline{L} \cup \overline{M}}$ 得证. □

证明 2: 由定理 8 构造识别 $L \cap M$ 的 DFA A, 则 $\forall w \in \Sigma^*$,

$$\begin{aligned} w \in L \cap M &\iff \hat{\delta}_L(q_L, w) \in F_L \wedge \hat{\delta}_M(q_M, w) \in F_M \\ &\iff (\hat{\delta}_L(q_L, w), \hat{\delta}_M(q_M, w)) \in F_L \times F_M \\ &\iff \hat{\delta}((q_L, q_M), w) \in F_L \times F_M \\ &\iff w \in \mathbf{L}(A). \end{aligned}$$

因此 $\mathbf{L}(A) = L \cap M$, 所以 $L \cap M$ 也是正则的. □

例 10. 如果已知语言
$$L_{01} = \{0^n 1^n \mid n \geq 0\}$$
不是正则的, 请用封闭性证明语言
$$L_{\text{eq}} = \{w \mid w \text{ 由数量相等的 } 0 \text{ 和 } 1 \text{ 构成}\}$$
也不是正则的.

证明:

1. 首先, 因为 $\mathbf{0}^*\mathbf{1}^*$ 是正则语言;
2. 而 $L_{01} = \mathbf{L}(\mathbf{0}^*\mathbf{1}^*) \cap L_{\text{eq}}$;
3. 如果 L_{eq} 是正则的, L_{01} 必然也是正则的;
4. 因为已知 L_{01} 不是正则的, 所以 L_{eq} 一定不是正则的. □

思考题

为什么又能用 L_{eq} 的子集 L_{01} 是非正则的, 来证明 L_{eq} 是非正则的呢?

例 11. 如果 L_1 和 L_2 都不是正则的, 那么 $L_1 \cap L_2$ 一定不是正则的吗?

不一定. 因为, 如果令
$$L_1 = \{0^n 1^n \mid n \geq 0\}$$
$$L_2 = \{a^n b^n \mid n \geq 0\}$$
显然两者都不是正则语言, 但
$$L_1 \cap L_2 = \{\varepsilon\}$$
是正则语言.

4.2.4 差

定理 10 (差运算封闭性)**.** *如果 L 和 M 都是正则语言, 那么 $L - M$ 也是正则的.*

证明: $L - M = L \cap \overline{M}$. □

例 12. 证明正则语言在以下运算下封闭
$$\min(L) = \{w \mid w \text{ is in } L, \text{but no proper prefix of } w \text{ is in } L\}$$

证明 1: 设 L 的 DFA 为 $M = (Q, \Sigma, \delta, q_0, F)$, 构造 $\min(L)$ 的 DFA $M' = (Q, \Sigma, \delta', q_0, F)$ 其中 δ' 如下, 往证 $L(M') = \min(L)$:

$$\delta'(q, a) = \begin{cases} \delta(q, a) & \text{if } q \notin F \\ \emptyset & \text{if } q \in F \end{cases}$$

1. $\forall w \in L(M')$, 存在转移序列 $q_0 q_1 \cdots q_n \in F$ 使 M' 接受 w, 其中 $q_i \notin F\,(0 \le i \le n-1)$, $\therefore w \in \min(L)$.
2. $\forall w \in \min(L)$, 有 $w \in L$, M 接受 w 的状态序列为如果 $q_0 q_1 \cdots q_n \in F$, 则显然 $q_i \notin F\,(0 \le i \le n-1)$, 否则 w 有 L 可接受的前缀, $\therefore w \in L(M')$ $\square$

证明 2:

由封闭性

$$\min(L) = L - L\Sigma^+,$$

得证. $\square$

4.2.5 反转

定义. 字符串 $w = a_1 a_2 \dots a_n$ 的反转, 记为 w^R, 定义为

$$w^R = a_n a_{n-1} \dots a_1.$$

定义. 语言 L 的反转, 记为 L^R, 定义为

$$L^R = \{w^R \in \Sigma^* \mid w \in L\}.$$

定理 11 (反转的封闭性)**.** 如果 L 是正则语言, 那么 L^R 也是正则的.

两种证明方法:

- 对正则表达式 E 的结构归纳, 往证

$$\mathbf{L}(E^R) = (\mathbf{L}(E))^R.$$

- 由识别 L 的 DFA $A = (Q, \Sigma, \delta_A, q_0, F)$, 构造识别 L^R 的 ε-NFA

$$B = (Q, \Sigma, \delta_B, q_s, \{q_0\})$$

  1. 将 A 的初始状态 q_0, 改为唯一的接受状态;
  2. 将 A 的边调转方向: 如果 $\delta_A(q, a) = p$, 那么 $\delta_B(p, a) = q$;

3. 新增初始状态 q_s, 且令 $\delta_B(q_s, \varepsilon) = F$;
4. 往证 $\mathbf{L}(B) = L^R$.

例 13. 语言 L 及其反转 L^R 分别为

$$L = \{w \in \{0,1\}^* \mid w \text{ ends in 01.}\}$$

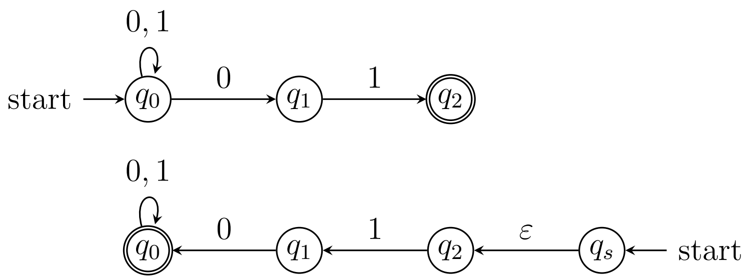
$$L^R = \{w \in \{0,1\}^* \mid w \text{ starts with 10.}\}$$

正则表达式分别为

$$L = (\mathbf{0} + \mathbf{1})^*\mathbf{01}$$

$$L^R = \mathbf{10}(\mathbf{0} + \mathbf{1})^*.$$

自动机分别为



证明: 往证如果有正则表达式 E, 则存在正则表达式 E^R 使

$$\mathbf{L}(E^R) = (\mathbf{L}(E))^R.$$

归纳基础:

1. 当 $E = \emptyset$ 时, 有 $\emptyset^R = \emptyset$;
2. 当 $E = \boldsymbol{\varepsilon}$ 时, 有 $\boldsymbol{\varepsilon}^R = \boldsymbol{\varepsilon}$;
3. $\forall a \in \Sigma$, 当 $E = \mathbf{a}$ 时, 有 $\mathbf{a}^R = \mathbf{a}$;

都满足 $\mathbf{L}(E^R) = (\mathbf{L}(E))^R$, 因此命题成立.

归纳递推:

1. 当 $E = E_1 + E_2$ 时, 有 $(E_1 + E_2)^R = E_1^R + E_2^R$

$$\begin{aligned} &(\mathbf{L}(E_1 + E_2))^R \\ &= (\mathbf{L}(E_1) \cup \mathbf{L}(E_2))^R && \text{正则表达式的加} \end{aligned}$$

$$
\begin{aligned}
&= \{w^R \mid w \in \mathbf{L}(E_1) \cup w \in \mathbf{L}(E_2)\} && \text{语言的反转} \\
&= (\mathbf{L}(E_1))^R \cup (\mathbf{L}(E_2))^R && \text{同上} \\
&= \mathbf{L}(E_1^R) \cup \mathbf{L}(E_2^R) && \text{归纳假设} \\
&= \mathbf{L}(E_1^R + E_2^R) && \text{正则表达式的加}
\end{aligned}
$$

2. 当 $E = E_1E_2$ 时, 有 $(E_1E_2)^R = E_2^R E_1^R$

$$
\begin{aligned}
&(\mathbf{L}(E_1E_2))^R = (\mathbf{L}(E_1)\mathbf{L}(E_2))^R && \text{正则表达式的连接} \\
&= \{w_1w_2 \mid w_1 \in \mathbf{L}(E_1), w_2 \in \mathbf{L}(E_2)\}^R && \text{语言的连接} \\
&= \{(w_1w_2)^R \mid w_1 \in \mathbf{L}(E_1), w_2 \in \mathbf{L}(E_2)\} && \text{语言的反转} \\
&= \{w_2^R w_1^R \mid w_1 \in \mathbf{L}(E_1), w_2 \in \mathbf{L}(E_2)\} && \text{字符串的反转} \\
&= \{w_2^R \mid w_2 \in \mathbf{L}(E_2)\}\{w_1^R \mid w_1 \in \mathbf{L}(E_1)\} && \text{语言的连接} \\
&= (\mathbf{L}(E_2))^R(\mathbf{L}(E_1))^R && \text{语言的反转} \\
&= \mathbf{L}(E_2^R)\mathbf{L}(E_1^R) = \mathbf{L}(E_2^R E_1^R) && \text{正则表达式的连接}
\end{aligned}
$$

3. 当 $E = E_1^*$ 时, 有 $(E_1^*)^R = (E_1^R)^*$

$$
\begin{aligned}
&(\mathbf{L}(E_1^*))^R \\
&= \{w_1w_2\dots w_n \mid n \ge 0, w_i \in \mathbf{L}(E_1)\}^R && \text{正则表达式的闭包} \\
&= \{(w_1w_2\dots w_n)^R \mid n \ge 0, w_i \in \mathbf{L}(E_1)\} && \text{语言的反转} \\
&= \{w_n^R w_{n-1}^R \dots w_1^R \mid n \ge 0, w_i \in \mathbf{L}(E_1)\} && \text{字符串的反转} \\
&= \{w_n^R w_{n-1}^R \dots w_1^R \mid n \ge 0, w_i^R \in \mathbf{L}(E_1^R)\} && \text{归纳假设} \\
&= \{w_1w_2\dots w_n \mid n \ge 0, w_i \in \mathbf{L}(E_1^R)\} && \text{变量重命名} \\
&= \mathbf{L}((E_1^R)^*) && \text{正则表达式的闭包}
\end{aligned}
$$

都满足 $(\mathbf{L}(E))^R = \mathbf{L}(E^R)$, 因此命题成立, 所以 L^R 也是正则语言. □

4.2.6 同态与逆同态

同态

定义. 若 Σ 和 Γ 是两个字母表, 同态定义为函数 $h : \Sigma \to \Gamma^*$

$$\forall a \in \Sigma,\ h(a) \in \Gamma^*.$$

扩展 h 的定义到字符串,

$$
\begin{aligned}
&(1)\quad h(\varepsilon) = \varepsilon \\
&(2)\quad h(xa) = h(x)h(a)
\end{aligned}
$$

再扩展 h 到语言, 对 $\forall L \subseteq \Sigma^*$,

$$h(L) = \{h(w) \mid w \in L\}.$$

例 14. 若由 $\Sigma = \{0, 1\}$ 到 $\Gamma = \{a, b\}$ 的同态函数 h 为

$$h(0) = ab, \quad h(1) = \varepsilon.$$

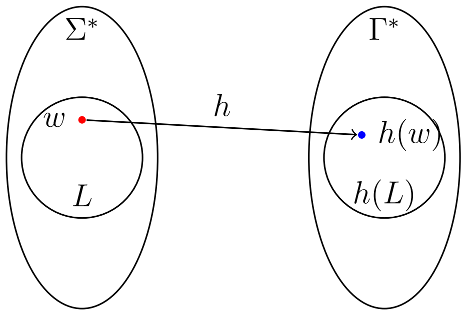
则 Σ 上的字符串 0011, 在 h 的作用下

$$\begin{aligned} h(0011) &= h(0)h(0)h(1)h(1) \\ &= abab\varepsilon\varepsilon = abab. \end{aligned}$$

语言 $L = \mathbf{1}^*\mathbf{0} + \mathbf{0}^*\mathbf{1}$, 在 h 的作用下, $h(L)$ 为:

$$\begin{aligned} h(\mathbf{1}^*\mathbf{0} + \mathbf{0}^*\mathbf{1}) &= (h(\mathbf{1}))^*h(\mathbf{0}) + (h(\mathbf{0}))^*h(\mathbf{1}) \\ &= (\varepsilon)^*(ab) + (ab)^*(\varepsilon) \\ &= (ab)^* \end{aligned}$$

定理 12 (同态的封闭性)**.** *若 L 是字母表 Σ 上的正则语言, h 是 Σ 上的同态, 则 $h(L)$ 也是正则的.*



- 若 L 的正则表达式为 E, 即 $L = \mathbf{L}(E)$, 按如下规则构造表达式 $h(E)$

$$\begin{aligned} h(\emptyset) &= \emptyset & h(\mathbf{r} + \mathbf{s}) &= h(\mathbf{r}) + h(\mathbf{s}) \\ h(\boldsymbol{\varepsilon}) &= \boldsymbol{\varepsilon} & h(\mathbf{rs}) &= h(\mathbf{r})h(\mathbf{s}) \\ \forall a \in \Sigma,\ h(\mathbf{a}) &= h(a) & h(\mathbf{r}^*) &= (h(\mathbf{r}))^* \end{aligned}$$

- 往证 $\mathbf{L}(h(E)) = h(\mathbf{L}(E))$, 而 $h(E)$ 显然也是正则表达式, 因此 $h(L)$ 正则

证明: 对 E 的结构归纳, 往证 $\mathbf{L}(h(E)) = h(\mathbf{L}(E))$.

归纳基础:

- 当 $E = \varepsilon$ 时

$$h(\mathbf{L}(\boldsymbol{\varepsilon})) = h(\{\varepsilon\}) = \{\varepsilon\} = \mathbf{L}(\boldsymbol{\varepsilon}) = \mathbf{L}(h(\boldsymbol{\varepsilon}))$$

- 当 $E = \emptyset$ 时
$$h(\mathbf{L}(\emptyset)) = h(\emptyset) = \emptyset = \mathbf{L}(\emptyset) = \mathbf{L}(h(\emptyset))$$
- $\forall a \in \Sigma$, 当 $E = \mathbf{a}$ 时
$$h(\mathbf{L}(\mathbf{a})) = h(\{a\}) = \{h(a)\} = \mathbf{L}(h(a)) = \mathbf{L}(h(\mathbf{a}))$$

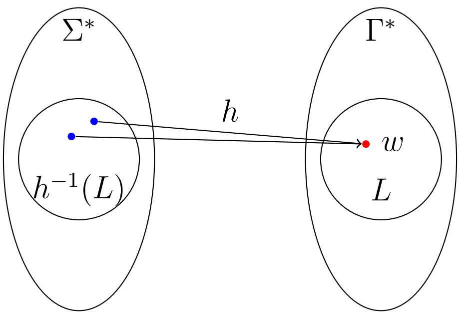
所以命题成立.

归纳递推: 假设对正则表达式 F, G 分别有
$$\mathbf{L}(h(F)) = h(\mathbf{L}(F)),\quad \mathbf{L}(h(G)) = h(\mathbf{L}(G))$$

- 当 $E = F + G$ 时:
$$\begin{aligned} h(\mathbf{L}(F+G)) &= h(\mathbf{L}(F) \cup \mathbf{L}(G)) && \text{正则表达式的加} \\ &= h(\mathbf{L}(F)) \cup h(\mathbf{L}(G)) && h\,\text{作用在每个集合的串上} \\ &= \mathbf{L}(h(F)) \cup \mathbf{L}(h(G)) && \text{归纳假设} \\ &= \mathbf{L}(h(F) + h(G)) && \text{正则表达式的加} \\ &= \mathbf{L}(h(F+G)) && h(F+G)\,\text{的定义} \end{aligned}$$
- 当 $E = FG$ 时: 略
- 当 $E = F^*$ 时: 略 □

逆同态

定义. *若 h 是字母表 Σ 到 Γ 的同态, 且 L 是 Γ 上的语言, 那么使 $h(w) \in L$ 的 w $(w \in \Sigma^*)$ 的集合, 称为语言 L 的 h 逆, 记为 $h^{-1}(L)$, 即*
$$h^{-1}(L) = \{w \in \Sigma^* \mid h(w) \in L\}.$$



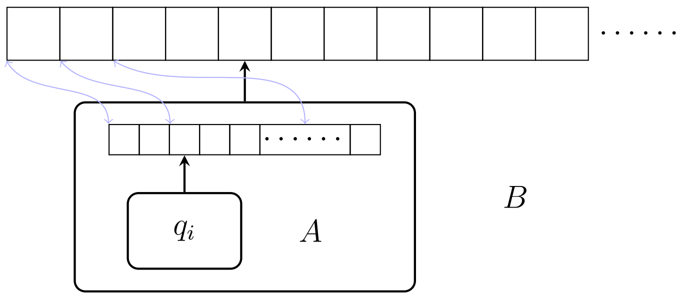
定理 13 (逆同态的封闭性)**.** *如果 h 是字母表 Σ 到 Γ 的同态, L 是 Γ 上的正则语言, 那么 $h^{-1}(L)$ 也是正则语言.*

证明: 由 L 的 DFA $A = (Q, \Gamma, \delta, q_0, F)$, 构造识别 $h^{-1}(L)$ 的 DFA

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

其中

$$\delta'(q, a) = \hat{\delta}(q, h(a)).$$



为证明 $\mathbf{L}(B) = h^{-1}(L)$, 先证明 $\hat{\delta}'(q, w) = \hat{\delta}(q, h(w))$.

对 $|w|$ 归纳, 往证 $\hat{\delta}'(q, w) = \hat{\delta}(q, h(w))$.

1. 归纳基础:若 $w = \varepsilon$
$$\hat{\delta}(q, h(\varepsilon)) = \hat{\delta}(q, \varepsilon) = q = \hat{\delta}'(q, \varepsilon),$$
2. 归纳递推: 若 $w = xa$

$$\begin{aligned}
\hat{\delta}'(q, xa) &= \delta'(\hat{\delta}'(q, x), a) && \hat{\delta}' \text{ 定义} \\
&= \delta'(\hat{\delta}(q, h(x)), a) && \text{归纳假设} \\
&= \hat{\delta}(\hat{\delta}(q, h(x)), h(a)) && \delta' \text{ 构造} \\
&= \hat{\delta}(q, h(x)h(a)) && \text{DFA 节例 5} \\
&= \hat{\delta}(q, h(xa)).
\end{aligned}$$

所以 $\forall w \in \Sigma^*$, $\hat{\delta}'(q_0, w) = \hat{\delta}(q_0, h(w)) \in F$, 即 w 被 B 接受当且仅当 $h(w)$ 被 A 接受, B 是识别 $h^{-1}(L)$ 的 DFA, 因此 $h^{-1}(L)$ 是正则的. □

例 15. Prove that $L = \{0^n 1^{2n} \mid n \geq 0\}$ is a language not regular.

证明: 设同态 $h : \{0, 1\} \to \{0, 1\}^*$ 为

$$h(0) = 0,$$

$$h(1) = 11,$$

那么

$$h^{-1}(L) = \{0^n 1^n \mid n \geq 0\} = L_{01},$$

我们已知 L_{01} 非正则, 由封闭性, L 不是正则的. □

例 16. 若语言 $L = (\mathbf{00}+\mathbf{1})^*$, 同态 $h : \{a, b\} \to \{0, 1\}^*$ 为

$$h(a) = 01, \quad h(b) = 10,$$

请证明 $h^{-1}(L) = (\mathbf{ba})^*$.

证明: 往证 $h(w) \in L \iff w = (ba)^n$.

($\Leftarrow$) 若 $w = (ba)^n$, 而 $h(ba) = 1001$, 因此 $h(w) = (1001)^n \in L$.

($\Rightarrow$) 若 $h(w) \in L$, 假设 $w \notin (\mathbf{ba})^*$, 则只能有四种情况:

1. w 以 a 开头, 则 $h(w)$ 以 01 开头, 显然 $h(w) \notin (\mathbf{00}+\mathbf{1})^*$;
2. w 以 b 结尾, 则 $h(w)$ 以 10 结尾, 显然 $h(w) \notin (\mathbf{00}+\mathbf{1})^*$;
3. w 有连续的 a, 即 $w = xaay$, 则 $h(w) = z1010v$, 则显然 $h(w) \notin (\mathbf{00}+\mathbf{1})^*$;
4. w 有连续的 b, 即 $w = xbby$, 则 $h(w) = z0101v$, 则显然 $h(w) \notin (\mathbf{00}+\mathbf{1})^*$;

因此 w 只能是 $(ba)^n, n \geq 0$ 的形式. □

例 17. For a language L, define head(L) to be the set of all prefixes of strings in L. Prove that if L is regular, so is head(L).

证明. 设 L 是 Σ 上的正则语言且 $\Sigma = \{0, 1\}$, $\Gamma = \{0, 1, a, b\}$. 定义同态 $h : \Gamma \to \Sigma^*$ 和 $g : \Sigma \to \Gamma^*$ 分别为:

$$\begin{array}{llll} h(0) = 0 & h(a) = 0 & g(0) = 0 & g(a) = \varepsilon \\ h(1) = 1 & h(b) = 1 & g(1) = 1 & g(b) = \varepsilon \end{array}$$

则因为 $(\mathbf{0}+\mathbf{1})^*(\mathbf{a}+\mathbf{b})^*$ 是 Γ 上的正则语言, 所以

$$(\mathbf{0}+\mathbf{1})^*(\mathbf{a}+\mathbf{b})^* \cap h^{-1}(L)$$

是 Γ 上的正则语言, 所以

$$\mathrm{head}(L) = g((\mathbf{0}+\mathbf{1})^*(\mathbf{a}+\mathbf{b})^* \cap h^{-1}(L))$$

是 Σ 上的正则语言, 因此 head(L) 是正则的. □

4.3 正则语言的判定性质

正则语言, 或任何语言, 典型的 3 个判定问题:

1. 以某种形式化模型描述的语言是否为空?是否无穷?
2. 某个特定的串 w 是否属于所描述的语言?
3. 以两种方式描述的语言, 是否是相同的? — 语言的等价性

我们想知道, 要回答这类问题的具体算法, 是否存在.

4.3.1 空性, 有穷性和无穷性

正则语言的空, 有穷和无穷 (Emptiness, finiteness and infiniteness), 可以通过定理 14 来判定.

定理 14. *具有 n 个状态的有穷自动机 M 接受的集合 S:*

1. *S 是非空的, 当且仅当 M 接受某个长度小于 n 的串;*
2. *S 是无穷的, 当且仅当 M 接受某个长度为 m 的串, $n \le m < 2n$.*

所以，对于正则语言:

- 存在算法, 判断其是否为空, 只需检查全部长度小于 n 的串;
- 存在算法, 判断其是否无穷, 只需检查全部长度由 n 到 $2n-1$ 的串.

证明: 设接受正则语言 S 的 DFA 为 M.

1. 必要性: 显然成立. 充分性:

   i 如果 S 非空, 设 w 是 M 接受的串中长度最小者之一;

   ii 必然 $|w| < n$, 否则由泵引理 $w = xyz$, 接受 xz 更短.

2. 必要性: 由泵引理, 显然成立. 充分性:

   i 如果 S 无穷, 假设没有长度 n 到 $2n-1$ 之间的串;

   ii 那么取 $w \in \mathbf{L}(M)$ 是长度 $\ge 2n$ 中最小者之一;

   iii 由泵引理 $w = xyz$, 且 M 会接受更短的串 xz;

   iv 于是, 或者 w 不是长度最小的, 或者长度 n 到 $2n-1$ 之间有被接受的串, 因此假设不成立. □

4.3.2 等价性

定理 15. *存在算法, 判定两个有穷自动机是否等价 (接受语言相同).*

证明:

1. 设 M_1 和 M_2 是分别接受 L_1 和 L_2 的有穷自动机;
2. 则 $(L_1 \cap \overline{L_2}) \cup (\overline{L_1} \cap L_2)$ 是正则的, 所以可被某个有穷自动机 M_3 接受;
3. 而 M_3 接受某个串, 当且仅当 $L_1 \neq L_2$;
4. 由于存在算法判断 $\mathbf{L}(M_3)$ 是否为空, 因此得证. □

4.4 自动机的最小化

4.4.1 状态的等价性

定义. *DFA $A = (Q, \Sigma, \delta, q_0, F)$ 中两个状态 p 和 q, 对 $\forall w \in \Sigma^*$:*

$$\hat{\delta}(p, w) \in F \Leftrightarrow \hat{\delta}(q, w) \in F,$$

则称这两个状态是等价的, 否则称为可区分的.

- 等价性只要求 $\hat{\delta}(p, w)$ 和 $\hat{\delta}(q, w)$ 同时在或不在 F 中, 而不必是相同状态.

4.4.2 填表算法与 DFA 最小化

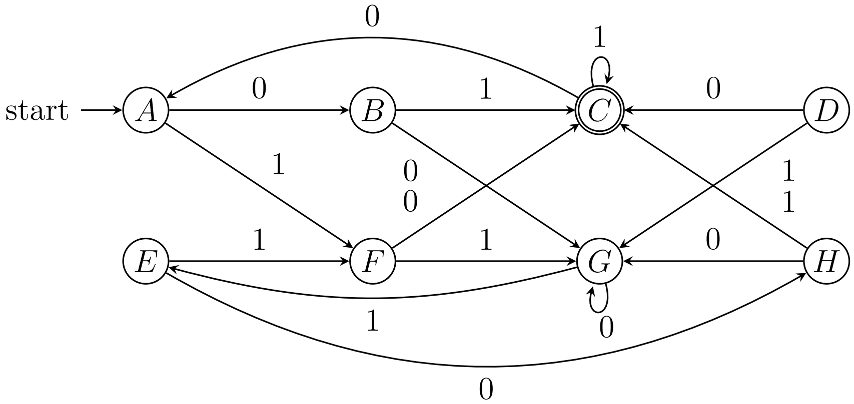
填表算法

递归寻找 DFA 中全部的可区分状态对:

1. 如果 $p \in F$ 而 $q \notin F$, 则 $[p, q]$ 是可区分的;
2. $\forall a \in \Sigma$, 如果
$$[r = \delta(p, a), s = \delta(q, a)]$$
是可区分的, 则 $[p, q]$ 是可区分的.

定理 16. *如果填表算法不能区分两个状态, 则这两个状态是等价的.*

例 18. 用填表算法找到如图 DFA 中全部可区分状态对.



	A	B	C	D	E	F	G
B	×						
C	×	×					
D	×	×	×				
E		×	×	×			
F	×	×	×		×		
G	×	×	×	×	×	×	
H	×		×	×	×	×	×

1. 直接标记终态和非终态之间的状态对:

$$\{C\}\times\{A, B, D, E, F, G, H\}.$$

2. 标记所有经过字符 0 到达终态和非终态的状态对:

$$\{D, F\}\times\{A, B, C, E, G, H\}.$$

3. 标记所有经过字符 1 到达终态和非终态的状态对:

$$\{B, H\}\times\{A, C, D, E, F, G\}.$$

4. 此时还有 [A,E], [A,G], [B,H], [D,F], [E,G] 未标记, 只需逐个检查.

   × [A,G] 是可区分的, 因为经串 01 到可区分的 [C,E];

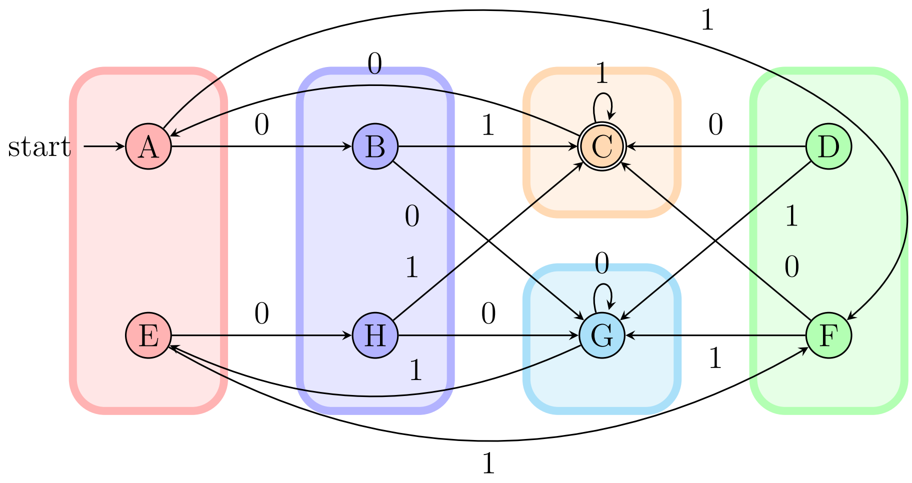
   × [E,G] 是可区分的, 因为经串 10 到可区分的 [C,H].

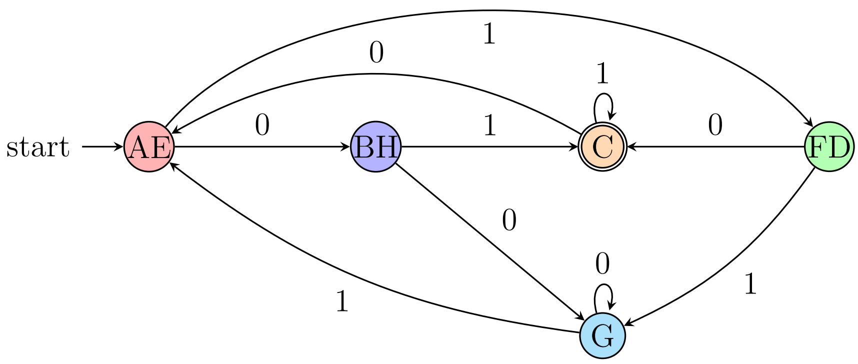
5. 而 [A,E], [B,H] 和 [D,F] 在经过很短的字符串后, 都会到达相同状态, 因此都是等价的.

DFA 最小化

根据等价状态, 将状态集划分成块, 构造等价的最小化 DFA. 根据填表算法取得的 DFA A 状态间的等价性, 将状态集进行划分, 得到不同的块; 利用块构造新的 DFA B, B 的开始状态的为包含 A 初始状态的块, B 的接受状态为包含 A 的接收状态的块, 转移函数为块之间的转移; 则 B 是 A 的最小化 DFA.

续例 18. 构造其最小化的 DFA.





思考题

NFA 能否最小化?

正则语言部分总结

计算机, 也可以看作是 DFA.