第 3 次作业 – 正则语言的性质

2154312 郑博远

习题 4.1.2 证明下列语言都不是正则的：

e） 由 0 和 1 构成的 ww 形式的串的集合，也就是某个串重复的串集合。

f） 由 0 和 1 构成的 ww^R 形式的串的集合，也就是由某个串后面跟着它的反转所构成的串的集合。（一个串的逆的形式化定义见 4.2.2 节。）

g） 由 0 和 1 构成的 $w\overline{w}$ 形式的串的集合，其中 $w\overline{w}$ 是把 w 中所有的 0 都换成 1 同时把所有的 1 都换成 0 而得到的串，例如，$\overline{011} = 100$，因此 011100 是该语言中的一个串。

h） 所有由 0 和 1 构成的 $w1^n$ 形式的串的集合，其中 w 是由 0 和 1 构成的长度为 n 的串。

解答：

e) 假设该语言 L 是正则的，且 L 的泵长度为 p。考察字符串 $s = 0^p1^p0^p1^p \in L$，由泵引理可知$|xy| \le p$，因此设 $y = 0^k$。$0^{p+k}1^p0^p1^p \notin L$，与假设矛盾，因此该语言不是正则的。

f) 假设该语言 L 是正则的，且 L 的泵长度为 p。考察字符串 $s = 0^p1^p1^p0^p \in L$（即 $w = 0^p1^p$），由泵引理可知$|xy| \le p$，因此设 $y = 0^k$。$0^{p+k}1^p1^p0^p \notin L$，与假设矛盾，因此该语言不是正则的。

g) 假设该语言 L 是正则的，且 L 的泵长度为 p。考察字符串 $s = 0^p1^p \in L$（即 $w = 0^p$），由泵引理可知$|xy| \le p$，因此设 $y = 0^k$。由于 $0^{p+k}1^p \notin L$，与假设矛盾，因此该语言不是正则的。

h) 假设该语言 L 是正则的，且 L 的泵长度为 p。考察字符串 $s = 0^p1^p \in L$（即 $w = 0^p$），由泵引理可知$|xy| \le p$，因此设 $y = 0^k$。由于 $0^{p+k}1^p \notin L$，与假设矛盾，因此该语言不是正则的。

习题 4.1.3 证明下列语言都不是正则的：

a） 所有满足以下条件的串的集合：由 0 和 1 构成，开头的是 1，并且当我们把该串看

作是一个整数时该整数是一个素数。

b） 所有满足以下条件的 0^i1^j 形式的串的集合：i 和 j 的最大公约数是 1。

解答：

a) 假设有字符串 $w = xyz$ 属于正则语言 L，且 $|y| = l, |z| = m$。其对应的**素数**可以表示为 $q = 2^{l+m} \cdot x + 2^m \cdot y + z$，考察 y 打 q 圈后的字符串对应的数字如下：

$$p = 2^{ql+m} \cdot x + 2^m \cdot \sum_{j=0}^{q-1} 2^{jl} \cdot y + z$$

由费马小定理可得 $2^{q-1} \equiv 1 \pmod q$，因此：

$$2^{q-1} = kq + 1$$

$$(2^{q-1})^l = (kq+1)^l$$

由于易得 $(kq+1)^l$ 展开后仅有常数项 1，其余项均为 q 的倍数模 q 后为 0，有：

$$2^{(q-1)l} \equiv 1 \pmod q \tag{1}$$

$$2^{ql-l+l} \equiv 2^l \pmod q$$

$$2^{ql} - 1 \equiv 2^l - 1 \pmod q$$

即：

$$(2^{ql} - 1) = k'q + (2^l - 1)$$

又：

$$\frac{2^{ql}-1}{2^l-1} = \frac{k'q}{2^l-1} + 1 = 1 + 2^l + \cdots + 2^{(q-1)l}$$

由于等式右侧为整数，可得 $\frac{k'q}{2^l-1} \in \mathbb{Z}$。又因为 q 为素数，因此 q 与 2^l-1互质，从而有 $\frac{k'}{2^l-1} \in \mathbb{Z}$。因此有：

$$\frac{2^{ql}-1}{2^l-1} \equiv 1 \pmod q \tag{2}$$

下面考察 p 是否为素数：

$$\begin{aligned} p &= 2^{ql+m} \cdot x + 2^m \cdot \sum_{j=0}^{q-1} 2^{jl} \cdot y + z \\ &= (2^{ql+m} - 2^{l+m}) \cdot x + 2^m \cdot \left(\frac{2^{ql}-1}{2^l-1} - 1\right) \cdot y + q \end{aligned}$$

$$= 2^{l+m} \cdot \left(2^{(q-1)l} - 1\right) \cdot x \ + 2^m \cdot \left(\frac{2^{ql} - 1}{2^l - 1} - 1\right) \cdot y + \ q$$

由式(1)、(2)得以上三项均为 q 的倍数，即 $q|\,p$；因此，p 是合数不属于正则语言 L。综上所述，L 不是正则语言。

b) 假设正则语言 L 的泵长度为 p，设素数 p_0 是**大于 $\boldsymbol{p+1}$** 的素数。考察字符串 $w = 0^{p_0}1^{(p_0-1)!}$，易得 p_0 与 $(p_0-1)!$ 互质，从而 $w \in L$。假设$|y| = k$，则 w 打 p_0 圈得到：

$$w' = 0^{p_0\cdot(k+1)}1^{(p_0-1)!}$$

由于 $k \in [1,p]$，又 $p_0 > p+1$，因此 $k \in [1, p_0 - 1)$，即 $k+1 \in [2, p_0 - 1]$。因此，$p_0 \cdot (k+1)$ 和 $(p_0-1)!$ 有公因子 $k+1$，二者不互质。故 $w' \notin L$，L 不是正则语言。

习题 4.2.2 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 $L = \{a, aab, bba\}$，则 $L/a\ = \{\varepsilon, bb\}$，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。

解答：

对于正则语言 L，存在一个 DFA $M = (Q, \Sigma, \delta, q_0, F)$，使得 $L(M) = L$。由题意，对 L/a 构造 DFA $M' = (Q, \Sigma, \delta, q_0, F')$，其中：

$$F' = \{\, q \mid \delta(q, a) \in F, q \in Q\}$$

对于 L/a 中的任意字符串 w，$wa \in L \Leftrightarrow \hat{\delta}(q_0, wa) \in F \Leftrightarrow \delta\big(\hat{\delta}(q_0, w), a\big) \in F \Leftrightarrow \hat{\delta}(q_0, w) \in F'$，因此 $L(M') = L/a$，因此 L/a 是正则语言。

习题 4.2.7 如果 $w = a_1a_2 \cdots a_n$ 和 $x = b_1b_2 \cdots b_n$ 是同样长度的串，定义 $alt\,(w,x)$ 是把 w 和 x 交叉起来且以 w 开头所得到的串，即$a_1b_1a_2b_2 \cdots a_nb_n$。如果 L 和 M 是语言，定义 $alt(L,M)$ 是所有形式为 $alt\,(w,x)$ 的串的集合，其中 w 是 L 中的任意串，而 x 是 M 中与 w 等长的任意串。证明：如果 L 和 M 都是正则的，那么 $alt(L,M)$ 也是。

解答：

设存在 DFA $M_A = (Q_A, \Sigma_A, \delta_A, q_{0A}, F_A)$, $M_B = (Q_B, \Sigma_B, \delta_B, q_{0B}, F_B)$，使得对于语言$L$、$M$，

有 $L(M_A)=L$，$L(M_B)=M$。下面构造 $M_{alt}=(Q,\Sigma,\delta,q_0,F)$ 以识别 $alt(L,M)$。

其中：$Q=\{(q_i,\ q_j,\ s)\mid\ q_i\in q_A,\ q_j\in q_B,\ s\in\{0,1\}\}$。$s=0$ 表示当前读入了偶数个字符，即等待读入 L 中字符串的下一个字符；$s=1$ 表示当前读入了奇数个字符，即等待读入 M 中字符串的下一个字符。类似的有 $q_0=(q_{0A},\ q_{0B},\ 0)$。

$\Sigma=\Sigma_A\ \cup\ \Sigma_B$，即 M_{alt} 的字母表为M_A 与 M_B 的并。

$$\delta\left((q_i,q_j,s),a\right)=\begin{cases}(\delta_A(q_i,a),q_j,1) & ,s=0\\ (q_i,\delta_B(q_j,a),0) & ,s=1\end{cases}$$

$F=\{(F_i,F_j,0)\mid F_i\in F_A,F_j\in F_B\}$，即 M_{alt} 的终止状态要同时满足 F_A 与 F_B，且 $s=0$。

易得上述 M_{alt} 满足 $L(M_{alt})=alt(L,M)$。因此 $alt(L,M)$ 是正则语言。

习题 4.2.8 设 L 是一个语言，定义 $half(L)$ 是所有 L 中串的前一半构成的集合，即 $\{w\mid$对于某个满足 $|x|=|w|$的 x，wx 属于 $L\}$。例如，如果 $L=\{\varepsilon,0010,011,010110\}$，则 $half(L)=\{\varepsilon,00,010\}$。注意，长度为奇数的串对于 $half(L)$ 没有贡献。证明：如果 L 是正则的，那么 $half(L)$ 也是。

解答：

设存在 DFA $M_A=(Q_A,\Sigma_A,\delta_A,q_{0A},F_A)$，使得对于语言$L$，有 $L(M_A)=L$。下面构造 $M=(Q,\Sigma,\delta,q_0,F)$ 以识别 $half(L)$。

其中：Q 所代表的 M 中的状态有如 (q,S) 的形式。$q\in Q_A$，用于记录在 M 中输入某个字符串后对应 M_A 所在的状态；S 表示在 M_A 中接受了当前已经读入长度的字符串后，能够到达接受状态的所有状态。对于已读入长度的记忆，需要通过转移函数来实现。

转移函数满足 $\delta((q,S),a)=(\delta_A(q,a),T)$，其中 T 表示接受一个字符后能够到达 S 中任意状态的 M_A 中的状态的集合。

此外，$q_0=(q_{0A},F_A)$，对应在 A 中的起始状态，且接受的字符串长为 0，即集合对应 M_A 中的 F_A。终止状态 F 对应所有满足 $q'\in S$ 的状态 (q',S)，即代表当前读入字符串的长度与当前状态到接收状态的长度相等。

易得上述 M 满足 $L(M)=half(L)$。因此 $half(L)$ 是正则语言。

习题 4.2.9 我们把习题 4.2.8 推广到能够决定取走串中多大部分的一系列函数．如果 f 是一个整数函数，定义 $f(L)$ 为 $\{w|$对某个满足 $|x| = f(|w|)$的 x，wx 属于 $L\}$。例如，和运算 $half$ 对应的 f 是恒等函数 $f(n) = n$，因为 $half(L)$ 的定义中有 $|x| = |w|$．证明：如果 L 是正则的，那么对于以下的 f，$f(L)$ 也是正则的：

a) $f(n) = 2n$（也就是取走串的前三分之一）。

b) $f(n) = n^2$（也就是取走的长度是没取走部分长度的平方根）。

c) $f(n) = 2^n$（也就是取走的长度是剩下长度的对数）。

解答：

设存在 DFA $M_A = (Q_A, \Sigma_A, \delta_A, q_{0A}, F_A)$，使得对于语言$L$，有 $L(M_A) = L$。下面构造 $M = (Q, \Sigma, \delta, q_0, F)$ 以识别 $f(L)$。

其中：Q 所代表的 M 中的状态有如 (q, S) 的形式。$q \in Q_A$，用于记录在 M 中输入某个字符串 w 后对应 M_A 所在的状态；S 表示在 M_A 中接受了长度为 $f(|w|)$ 的字符串后，能够到达接受状态的所有状态，即 $S = \{p \in Q_A \mid \hat{\delta}_A(p, w') \in F_A, w' \in \Sigma^{f(|w|)}\}$。

转移函数满足 $\delta\big((q, S), a\big) = (\delta_A(q, a), T)$。下面解释集合 T 的构造方式：寻找某个字符串 w 满足 $\hat{\delta}_A(q_{0A}, w) = q$ 且 S 中所有状态都能接受某个长度为 $f(|w|)$的字符串到达接收状态（即$S = \{p \in Q_A \mid \hat{\delta}_A(p, w') \in F_A, w' \in \Sigma^{f(|w|)}\}$），则将 T 构造为$S = \{p \in Q_A \mid \hat{\delta}_A(p, w') \in F_A, w' \in \Sigma^{f(|w|+1)}\}$。

此外，$q_0 = (q_{0A}, F_A)$，对应在 A 中的起始状态，且接受的字符串长为 0，即集合对应 M_A 中的 F_A。终止状态 F 对应所有满足 $q' \in S$ 的状态 (q', S)，即代表对应当前读入字符串 w，有当前状态到接收状态的长度 $f(|w|)$。

易得上述 M 满足 $L(M) = f(L)$。故 a)、b)、c)中的 $f(L)$ 均是正则语言。

补充习题 1 给出如下的正则文法 G，求出对应的 DFA M，使得 $L(M) = L(G)$。

（1） $G_1 = (V, T, P_1, S)$

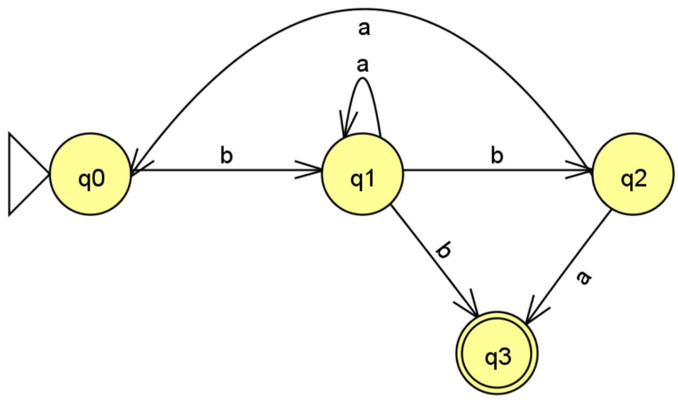
P_1: $S \to bB, B \to aB \mid bA \mid b, A \to a \mid aS$

（2） $G_2 = (V, T, P_2, S)$

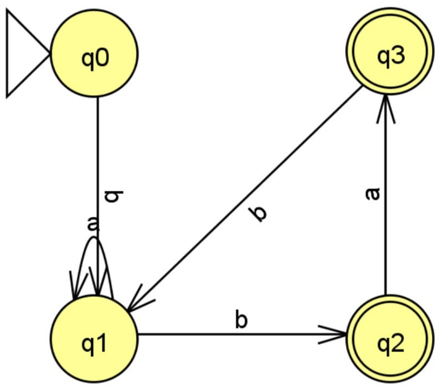
$P_2 \colon S \to aS \mid bB \mid a, B \to bA \mid aB \mid aS$

解答：

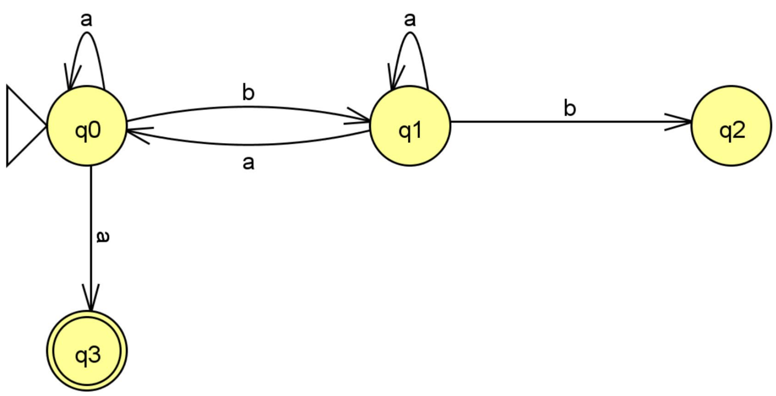
（1）构造 NFA 如下图：



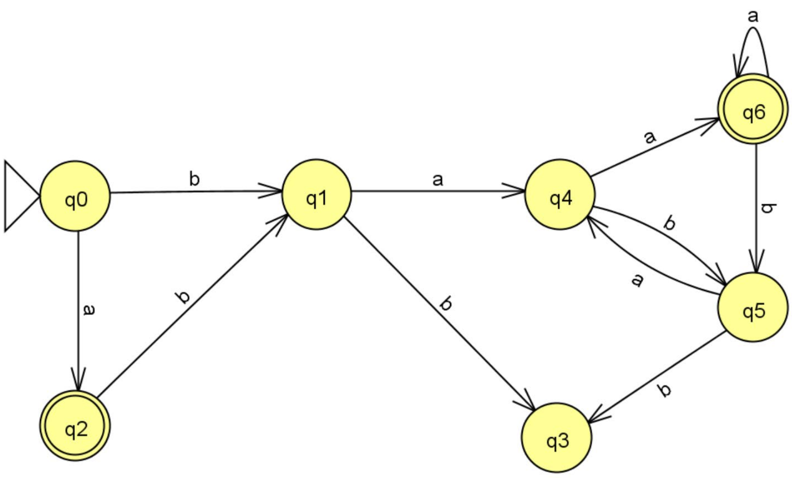
将其转换为 DFA 如下：



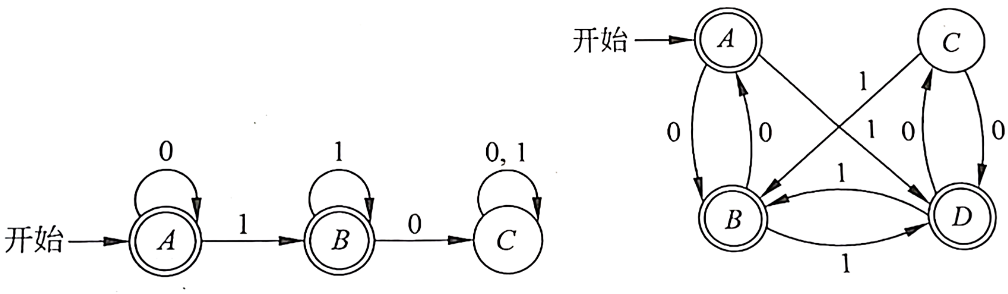
（2）构造 NFA 如下图：



将其转换为 DFA 如下：



补充习题 2 给出下图描述的两个 DFA M，分别求出对应的正则文法 G，使得 $L(G) = L(M)$。



a) DFA M_1　　b) DFA M_2

解答：

a） $G_1 = (V, T, P_1, S)$

P_1: $S \to 0S \mid 1A \mid \varepsilon$, $A \to 1A \mid 0B \mid \varepsilon$, $B \to 0B \mid 1B$

b） $G_2 = (V, T, P_2, S)$

P_2: $S \to 0A \mid 1C \mid \varepsilon$, $A \to 0S \mid 1C \mid \varepsilon$, $B \to 1A \mid 0C$, $C \to 1A \mid 0B \mid \varepsilon$

补充习题 3 Let $L_1 \subseteq \{0,1,2\}^*$ be a regular language,we can consider L_1 as a subset of integers under base 3, let L_2 be the corresponding set of L_1 over $\{0,1\}^*$ (i.e. under base 2), for example if $L_1 = \{11,12,121\}$, then $L_2 = \{100,101,10000\}$. Question: is L_2 a regular language?

Solution:

Whether L_2 is a regular language depends on L_1. Here's why:

(1) *If L_1 is a regular language whose strings are of a finite length, then so does L_2. Obviously, L_2 can be recognized by a* DFA, *making it a regular language (as exemplified by the example in the question).*

(2) *Consider L_1 represents integers that are mutiples of 3 under base 3, i.e., its regular expression is 10^+. Accordingly, L_2 represents integers that are mutiples of 3 under base 2. Assume that L_2 is a regular language, and then consider a string w_1 whose length is greater than the pumping length of L_2. According to the pumping lemma, it can be decomposed as $w_1 = xyz$, and thus*

$$2^{|y|+|z|} \cdot |x| + 2^{|z|} \cdot |y| + |z| = 3^i$$

Now let's cycle y once to get $w_2 = xyyz$, which represents another multiple of 3, $2^{2|y|+|z|} \cdot |x| + 2^{|z|} \cdot |y| + 2^{|y|+|z|} \cdot |y| + |z|$. If we substract the integers denoted by w_1 and w_2 (represent them by $[w_1]$ and $[w_2]$), we get

$$\begin{aligned}
[w_2] - [w_1] &= \left(2^{2|y|+|z|} \cdot |x| + 2^{|z|} \cdot |y| + 2^{|y|+|z|} \cdot |y| + |z|\right) \\
&\quad - \left(2^{|y|+|z|} \cdot |x| + 2^{|z|} \cdot |y| + |z|\right) \\
&= \left(\left(2^{2|y|+|z|} - 2^{|y|+|z|}\right) \cdot |x| + 2^{|y|+|z|} \cdot |y|\right) \\
&= 2^{|y|+|z|}\left(\left(2^{|y|} - 1\right) \cdot |x| + |y|\right)
\end{aligned}$$

$[w_2]$ is greater than $[w_1]$, so $[w_2] - [w_1]$ must have a factor $[w_1]$ (because they are both mutiples of 3). $\left(2^{|y|} - 1\right) \cdot |x| + |y| < 2^{|y|+|z|} \cdot |x| + 2^{|z|} \cdot |y| + |z| = [w_1]$, and $2^{|y|+|z|}$ has no factor 3, so $[w_2] - [w_1]$ doesn't have the factor $[w_1]$. Thus, w_2 doesn't belong to L_2, so L_2 is not a regular language.

In conclusion, L_3 *is a regular language cannot lead to the conclusion that* L_2 *is also a regular language.*