B0911005Y-01: Introduction to Theory of Computation **2023 Spring**

Homework 1 — March 14

Lecturer: Mingji Xia *Completed by:* 吉骏雄

第 1.1 次作业：1.6 f,i,1.14, 1.31. (1.35 选做)

1.6 画出识别下述语言的 DFA 状态图, 在所有问题中字母表均为 $\{0,1\}$.

f. $\{\omega \mid \omega$不含子串110$\}$

证明

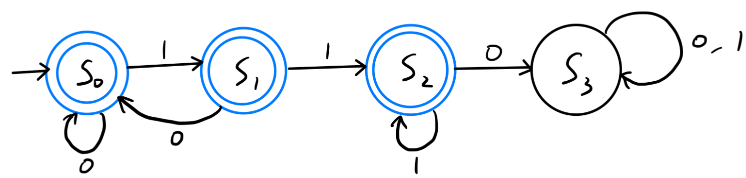

图 1.1. 1.6.1

蓝色双圆圈的状态表示接受状态. 下均同此表示. □

i. $\{\omega \mid \omega$的奇数位置均为1$\}$

证明

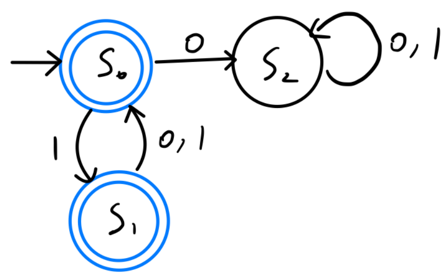

图 1.2. 1.6.2

如上图所示. □

1.14

a. 证明: 若 M 是一台识别语言 B 的 DFA, 交换 M 的接受状态与非接受状态得到一台新的 DFA, 则这台新 DFA 识别 B 的补集. 因而, 正则语言类在补运算下封闭.

证明 M 这台 DFA 在接收一串字符后, 按照定义, 会停止在一个确定的位置. 一个字符串 ω 要么会让 M 停止在接受状态, 这时 $\omega \in B$, 要么会让 M 停止在非接受状态, 这时 $\omega \notin B$. 如果让接受状态与非接受状态交换, 记 B' 为新的 DFA (记作 M') 所识别的语言, 那么原本被接受的字符串 ω 就会不被新的 DFA 接受, 于是 $\omega \notin B'$, 原本不被接受的字符串现在被 B' 接受, 因而 $\omega \in B'$. 于是根据定义, $\forall \omega \in B(\omega \notin B') \wedge \forall \omega \notin B(\omega \in B')$, 所以 B' 是 B 的补集, 即这台新 DFA 识别 B 的补集, B' 也是正则语言. 由于 M 的任意性, 我们可以得到正则语言类在补运算下封闭的结论. □

b. 举例说明: 若 M 是一台识别语言 C 的 NFA, 交换 M 的接受状态与非接受状态, 得到一台新 NFA, 这台 NFA 不一定识别 C 的补集. NFA 识别的语言类在补运算下封闭吗? 并解释回答.

证明 新 NFA 不识别 C 的补集的例子如图 1.3:

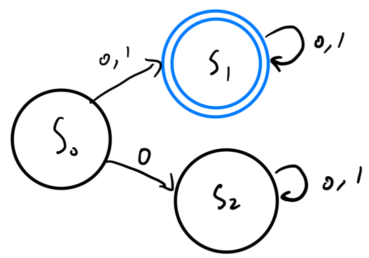

图 1.3. 1.14

考虑字符串 0, 无论是该 NFA 还是其接受状态与非接受状态交换后得到的新 NFA, 0 均为被接受的输入, 因此这台特殊的 NFA 不识别 C 的补集.

但是, NFA 识别的语言类在补运算下封闭. 题目只是指出, NFA 的接受状态与非接受状态对换后, 新 NFA 的识别语言并不是原识别语言 C 的补集, 并不是说原识别语言 C 的补集就不是正则语言. 我们知道, 每一台 NFA 都等价于某一台 DFA, 这样我们可以找到一台 DFA, 并将之接受状态与非接受状态交换, 得到识别语言是 C 的补集的一台状态机 M'; 再构造一个 NFA, 使得 NFA 的状态图中, 所有状态和边都和 M' 相同. 这样, 这台 NFA 便满足题目要求, NFA 识别的语言类在补运算下封闭. □

1.31 对语言 A 和语言 B, 设 A 和 B 的完全间隔交叉 (perfect shuffle) 为:

$$\{\omega \mid \omega = a_1 b_1 \cdots a_k b_k,\ \text{其中} a_1 \cdots a_k \in A \text{ 并且} b_1 \cdots b_k \in B,\ \text{任一} a_i, b_i \in \Sigma\}$$

证明: 正则语言类在完全间隔交叉下封闭.

证明 第一步是依据某个已有 DFA 构造一个新 DFA, 只读取奇/偶数位的输入, 并且接收相同的字符串 (对奇/偶数位). 我们只讨论" 只读取奇数位输入的新 DFA 构造", 因为偶数位的原理相同.

对于一个名为 M 的 DFA, 分别用 $S_0, S_1, \ldots, S_n$ 来表示它的 $n+1$ 个不同状态. 我们现构造一个新的 DFA, 名为 M', 它拥有 $2n+2$ 个不同的状态, 分别是 $S_{\mathrm{a}0}, S_{\mathrm{a}1}, \ldots, S_{\mathrm{a}n}, S_{\mathrm{b}0}, S_{\mathrm{b}1}, \ldots, S_{\mathrm{b}n}$. M' 的状态转移依照如下定义给定:

1. 如果 M 中有 $S_i \times \alpha \xrightarrow{\delta} S_j$, $S_i, S_j \in Q$, $\alpha \in \Sigma$, 那么在 M' 中构造一条状态转移: $S_{\mathrm{a}i} \times \alpha \xrightarrow{\delta} S_{\mathrm{b}j}$, $S_{\mathrm{a}i}, S_{\mathrm{b}j} \in Q'$, $\alpha \in \Sigma$, 其中 Q, Q' 分别为 M, M' 的状态集合. 这个定义负责读取奇数位的输入.

2. $\forall i \in \{0, 1, \ldots, n\}$, $\forall \alpha \in \Sigma$, 构造转移: $S_{\mathrm{b}i} \times \alpha \xrightarrow{\delta} S_{\mathrm{a}i}$, $S_{\mathrm{b}i}, S_{\mathrm{a}i} \in Q'$. 这个定义负责忽略偶数位的输入.

在此之后, 我们还需要定义新的接收状态. 如果 M 的接受状态集合为 $S_{r_0}, S_{r_1}, \ldots, S_{r_n}$, 那么 M' 的接受状态集合就应该被定义为 $S_{\mathrm{a}r_0}, S_{\mathrm{a}r_1}, \ldots, S_{\mathrm{a}r_n}$. 根据推理可知, 状态转移图是一个二部图, 因此, a 表示的是定义了原本下标中数字对应的新的两个状态中, 表明已读取数字为偶数的一个状态. 这样 M' 只会接受长度为偶数的字符串.

只读取偶数位置输入的构造同理, 要求也是必须输入字符串偶数长才能接受.

既然这样可以构造出两种 DFA, 那么这两种构造后的接受语言也都是正则语言, 因此对正则语言类, 此种构造是封闭的.

第二步是证明引理: 正则语言类在交下是封闭的. 首先定义两个语言的交为: $A \cap B = \{x \mid x \in A \text{ 且} x \in B\}$. 使用和证明并运算封闭的方法类似, 只不过定义 $F = \{(r_1, r_2) \mid r_1 \in F_1 \text{ 且} r_2 \in F_2\}$, 之后引理显然可证.

第三步是把 A 和 B 对应的状态机分别取奇数位有效和偶数位有效的变换, 然后交在一起. 这两步都封闭的话, 两步运算的符合还是封闭的, 于是得到的语言仍然是正则语言. 根据定义, 我们能够给出得到的语言的定义: 奇数位相连是 A 对应的接受语言中的字符串, 偶数位相连是 B 对应的接受语言中的字符串, 并且此字符串必须是偶数为长度. 这与题目对完全间隔交叉的定义如出一辙, 即 $\{\omega \mid \omega = a_1 b_1 \cdots a_k b_k, \text{ 其中} a_1 \cdots a_k \in A \text{ 并且} b_1 \cdots b_k \in B, \text{ 任一} a_i, b_i \in \Sigma\}$. 因此, 正则语言类在完全间隔交叉下封闭, 题目得证.

□

1.35 设 $A/B = \{\omega \mid \text{ 对于某} x \in B, \omega x \in A\}$. 证明如果 A 是正则的, B 是任意语言, 那么 A/B 是正则的.

证明 记正则语言 A 对应的某一个 DFA 为 $M = \{Q, \Sigma, \delta : Q \times \Sigma \to Q, q_0 \in Q, F \subseteq Q\}$. 记函数 $\vartheta : Q \times \Sigma \to 2^Q$ 满足:

$$\forall q \in Q,\ \forall \sigma \in \Sigma,\ \vartheta(q, \sigma) = \{q' \mid \delta(q', \sigma) = q\}$$

即 ϑ 是一个在同一字母下将所有转移反向的函数. 然后扩充定义这个函数对字符串的操作, 即从前向后依次对字符串中的每一个字母进行映射操作; 对集合的操作, 即分别对集合中的每一个字母/字符串进行映射操作. 总之, 定义此函数使我们可以回溯 DFA 的转移步骤.

定义对语言的反转为 $L^R = \{\omega^R \mid \forall \omega \in L\}$. 集合 $C := \vartheta(F, B^R)$, 这表示从该 DFA 的每一个接受状态开始, 对语言 B 的所有字符串的反转进行回溯, 得到的状态的集合. 满足 $\forall q \in C, \exists \omega \in B, \exists q' \in F\ (\delta(q, \omega) = q')$. 容易证明, 这个定义等价于:

$$C = \{q \mid q = \delta(q_0, A/B)\}$$

所以我们只需要把由 A 定义出的 DFA 中, 接受状态集 F 替换成 C, 就可以得到一个对应接收语言为 A/B 的 DFA, 所以 A/B 也是正则的. □

第 1.2 次作业: 1.16 (b) 1.21 (b) 1.28 (a)

1.16 使用定理 1.19 给出的构造, 把下图的两台非确定型有穷自动机转换成等价的确定型有穷自动机.

b. 如图 1.4

证明 如图 1.5

□

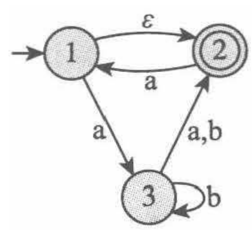

图 1.4. 1.16 (b) 题目图片

	ε	a	b
1	{2}	{3}	∅
2	∅	{1}	∅
3	∅	{2}	{2,3}

	a	b
∅	∅	∅
{1}	{1,3}	∅
{2}	{1}	∅
{3}	{2}	{2,3}
{1,2}	{1,3}	∅
{1,3}	{1,2,3}	{2,3}
{2,3}	{1,2}	{2,3}
{1,2,3}	{1,2,3}	{2,3}

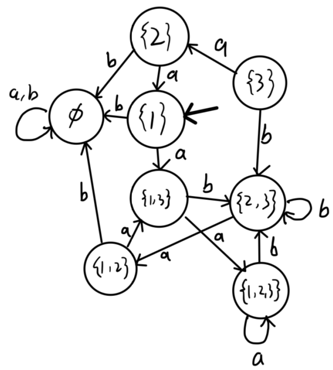

图 1.5. 1.16 (b)

1.21 使用引理 1.32 中描述的过程, 把下述有穷自动机转换成正则表达式.

b. 如图 1.6

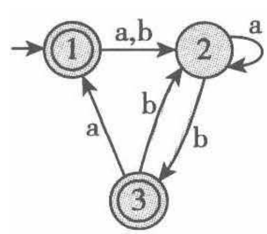

图 1.6. 1.21 (b) 题目图片

证明 如图 1.7

□

1.28 使用定理 1.28 给出的过程将下述正则表达式转换成 NFA. 在所有问题中 $\Sigma = \{a, b\}$.

b $a^+ \cup (ab)^+$

证明 如图 1.8

□

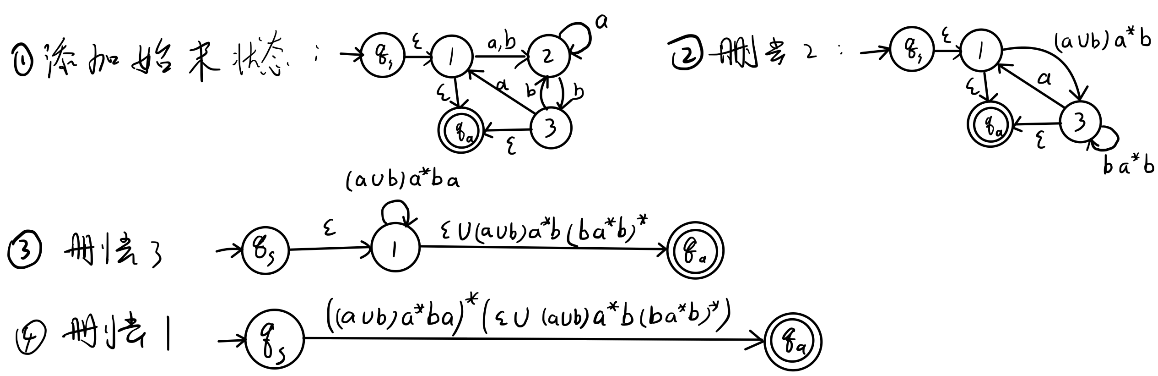

图 **1.7.** 1.21 (b)

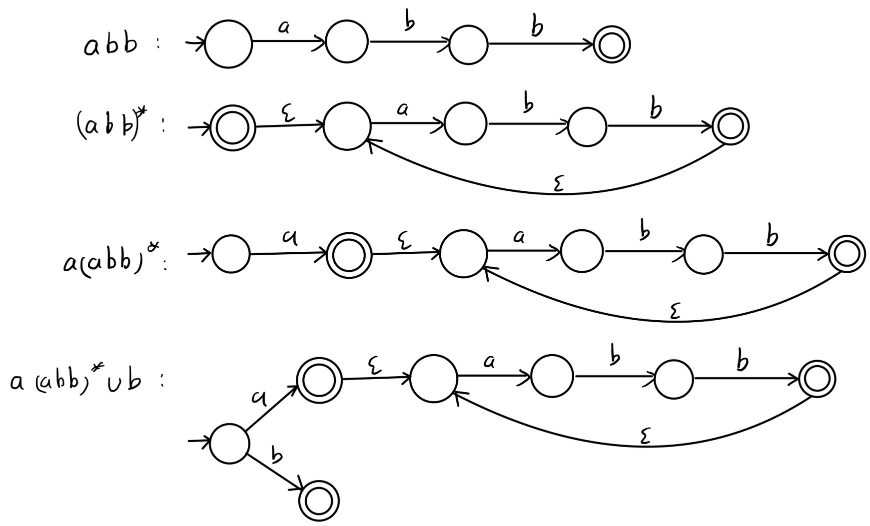

图 **1.8.** 1.28 (b)

第 1.3 次作业: 1.47, 1.48, 补充题 3 个, 其中 2 个选做

1.47 设 x 和 y 是两个字符串, L 是一个语言. 如果存在字符串 z, 使得 xz 和 yz 中恰好有一个是 L 的成员, 则称 x 和 y 是用 L 可区分的; 否则, 对每一个字符串 z, xz 和 yz 要么都是、要么都不是 L 的成员, 则称 x 和 y 是用 L 不可区分的. 如果 x 和 y 是用 L 不可区分的, 记作 $x \equiv_L y$. 证明 $\equiv_L$ 是一个等价关系.

证明 等价关系的性质为: 对称性, 自反性和传递性, 下分别证明 $\equiv_L$ 拥有这三个性质.

对称性: 如果 $x \equiv_L y$, 那么 x 和 y 是用 L 不可区分的, 这样对每一个字符串 z, xz 和 yz 要么都是、要么都不是 L 的成员. 这件事显然可以反过来, 于是 y 和 x 是用 L 不可区分的, 因此 $y \equiv_L x$ 显然成立, 对称性成立.

自反性: 显然, 对每一个字符串 z, xz 和 xz 作为完全相同的字符串, 要么都是、要么都不是 L 的成员. 因此 $x \equiv_L x$, 自反性成立.

传递性: 如果 $x \equiv_L y$ 且 $y \equiv_L w$, 那么对每一个字符串 z, $[(xz \in L \wedge yz \in L) \vee (xz \notin L \wedge yz \notin L)] \wedge [(yz \in L \wedge wz \in L) \vee (yz \notin L \wedge wz \notin L)]$. 经过逻辑化简, 我们可以得到 $(xz \in L \wedge yz \in L \wedge wz \in L) \vee (xz \notin L \wedge yz \notin L \wedge wz \notin L)$, 这可以推出 $(xz \in L \wedge wz \in L) \vee (xz \notin L \wedge wz \notin L)$. 这就是说, 对每一个字符串 z, xz 和 wz 要么都是、要么都不是 L 的成员, 所以 $x \equiv_L w$, 传递性成立. □

1.48 Myhill-Nerode 定理. 参见问题 1.47, 设 L 是一个语言, X 是一个字符串集合. 如果 X 中的任意两个不同的字符串都是用 L 可区分的, 则称 X 是用 L 两两可区分的. 定义 L 的指数为用 L 两两可区分的集合中的元素个数的最大值. L 的指数可能是有穷的或无穷的.

a. 证明: 如果 L 被一台有 k 个状态的 DFA 识别, 则 L 的指数不超过 k.

证明 那么 L 为一个正则语言. 记这台 DFA 为 $M = (Q, \Sigma, \delta, q_0, F)$. 我们考虑一个映射 f, 表示一个字符串被该 DFA 执行得到的结果状态, 有归纳定义如下: $f(\varepsilon) = q_0; \forall \omega \in \Sigma^*, \forall a \in \Sigma, f(\omega a) = \delta(f(\omega), a)$.

这时我们定义另一个等价关系 (易证) $\forall x, y \in \Sigma^*, x \simeq y \iff f(x) = f(y)$. 这样看来, f 给出了一个蕴含了等价关系 $\equiv_L$ 的等价关系, 因为 $x \simeq y \to x \equiv_L y$. 这是因为, 后者的定义其实是 $\forall z \in \Sigma^*, (f(xz) \in L \wedge f(yz) \in L) \vee (f(xz) \notin L \wedge f(yz) \notin L)$, 而 $f(x) = f(y) \to f(xz) = f(yz) \to (f(xz) \in L \wedge f(yz) \in L) \vee (f(xz) \notin L \wedge f(yz) \notin L)$. 考虑到 L 在 $\simeq$ 下的等价类取决于字符串运算后终点的位置, 也就是停止的状态, 而这种状态最多有 k 个, 所以 L 的指数 (也就是在 $\equiv_L$ 下的等价类个数) 不可能超过 k. □

b. 证明: 如果 L 的指数是一个有穷数 k, 则它被一台有 k 个状态的 DFA 识别.

证明 L 的指数是一个有穷数 k, 意味着任意字符串 $z \in \Sigma^*$, 一共有至多 k 个 $\equiv_L$ 下的等价类. 下面给出一种 DFA 的构造, 使得其接收语言 L. 其中用到定义: $[x]_L$ 指 x 在 $\equiv_L$ 下所在的等价类.

定义 DFA: $M = (Q, \Sigma, \delta, q_0, F)$, 满足:

- $Q = \{[x]_L \mid x \in \Sigma^*\}$
- $\delta([x]_L, a) = [xa]_L$
- $q_0 = [\varepsilon]_L$
- $F = \{[x]_L \mid x \in L\}$

为了证明 M 的接受语言就是 L, 我们需要证明两件事: 一是 L 中的所有字符串都能够被 M 接受, 二是不在 L 中的所有字符串都不能够被 M 接受.

1. $\forall y \in L$, 将 y 放入自动机, 显然, 我们进入的状态就是 $[y]_L$, $y \in L$ (可以由递归的定义接受字符串, 证明这件事), 而这正是一个被接受的状态.

2. $\forall y \notin L$, 将 y 放入自动机, 进入的状态就是 $[y]_L$, $y \notin L$. 另 $\forall x \in L$, 存在一个空串 ε 使得 $x\varepsilon \in L \wedge y\varepsilon \notin L$, 因此 x 与 y 是用 L 不可区分的, 因此 $y \notin \{[x]_L \mid x \in L\} = F$, 即非语言 L 中的字符串都是不被 M 接受的.

根据以上两面的论述, 我们得知 M 的接受语言确是 L; 而 M 的状态数就是 L 的指数 k, 证毕. □

c. 由此得到: L 是正则的当且仅当它有有穷的指数. 而且, 它的指数是识别它的最小 DFA 的大小.

证明 根据 a 得出, L 是正则的 (被一台 DFA 识别), 仅当它有有穷的指数, 且它的指数不超过 DFA 的状态数.

根据 b 得出, L 是正则的, 当它有有穷的指数, 且它的指数是一台识别它的 DFA 的状态数.

a 和 b 一同给出: L 是正则的当且仅当它有有穷的指数.

b 给出了 a 中 L 的指数可以是 DFA 状态数的构造, 因此我们可以得出, L 的指数是识别它的最小 DFA 的大小. □

补充题 1 证明 $\{0^m1^n \mid m > n\}$ 不是正则语言.

证明 泵引理: 对于任意的 (无穷) 正则语言 A, 存在正整数 p (称为泵长度), 使得对于任意 $\omega \in A$，如果 $|\omega|p$, 则 ω 可被分成 3 段，即 $\omega = xyz$, 且 $|y|\geq 1$, $|xy|\leq p$, 且对于任意 $r \geq 0$, $xy^rz \in A$.

仔细观察泵引理的证明过程, 发现是使用 $i \in \{0, 1, 2, \ldots, p\}$ 这前 p 个状态证明的, 只要换成 $i \in \{S-p-1, S-p, S-p+1, \ldots, S-1\}$ (S 表示状态数) 这后 p 个 (除去最后一个状态), 可以得到与最后的状态有关的” 泵引理”, 即将条件中的 $|xy|\leq p$ 改为 $|yz|\leq p$.

使用这个新的” 泵引理”, 并假设这个语言是正则的, 那么存在 p 使得泵引理的条件满足. 记此语言为 L, 我们可以取 $\omega \in L$ 满足 $n > p + 3$, 这样 $\omega = xyz$ 中的 x 一定全部由 1 组成. 取 $r = m + 1$, 显然 $xy^rz = 0^m1^{n+m|y|}$, 其中 $n + m|y|> m$, 与 $xy^rz \in L$ 矛盾! 故 L 不是正则语言. □

(选做) 补充题 2 证明 $\{0^m1^n \mid m \neq n\}$ 不是正则语言. (懒得想了)

(选做) 补充题 3: 证明 $\{1^n \mid n\text{是素数}\}$ 不是正则语言.

证明 考虑泵引理, 假设题中语言 (记为 L) 是正则语言, 则 $\exists p \in \mathbb{Z}^+$ 使得 $\omega \in L \wedge |\omega|\geq p$, $\omega = xyz$, $|y|\geq 1$, $|xy|\leq p$, $\forall r \geq 0\,(xy^rz \in L)$.

寻找反例: $r = |\omega|-|y|$, 那么:

$$|xy^rz|= |x|+r|y|+|z|= |y|(|\omega|-|y|) + |\omega|-|y|= (|y|+1)(|\omega|-|y|)$$

显然这样的字符串长度不为质数, 导出矛盾! 因此不能满足泵引理, L 不是正则语言. □