

第 2 章、正则语言习题解答 - 练习

2.1、图给出两台 DFA M_1 和 M_2 的状态图。回答下述有关问题。

- a. M_1 的起始状态是 q_1
- b. M_1 的接受状态集是 $\{q_2\}$
- c. M_2 的起始状态是 q_1
- d. M_2 的接受状态集是 $\{q_1, q_4\}$
- e. 对输入 aabb, M_1 经过的状态序列是 q_1, q_2, q_3, q_1, q_1
- f. M_1 接受字符串 aabb 吗？否
- g. M_2 接受字符串 吗？是

2.2、给出练习 2.1 中画出的机器 M_1 和 M_2 的形式描述。

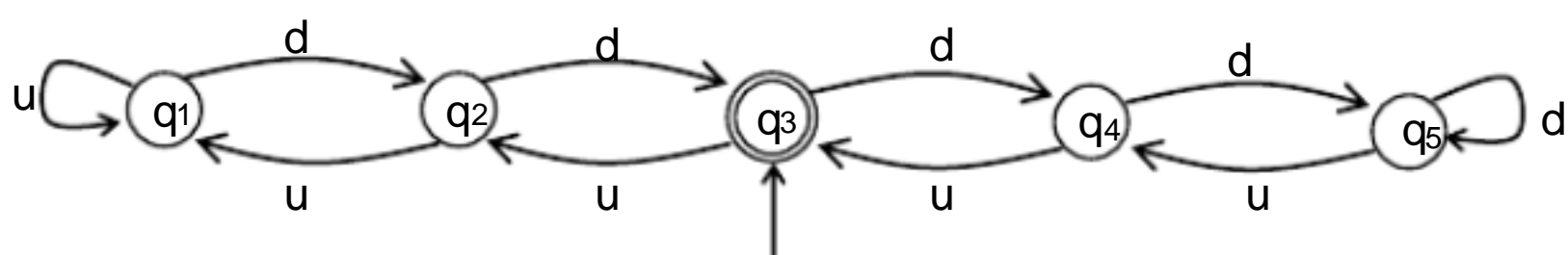
$M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ 其中 $Q_1 = \{q_1, q_2, q_3\}$; $\Sigma = \{a, b\}$; q_1 是起始状态; $F_1 = \{q_2\}$; δ_1 为:

	a	b
q_1	q_2	q_1
q_2	q_3	q_3
q_3	q_2	q_1

$M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ 其中 $Q_2 = \{q_1, q_2, q_3, q_4\}$; $\Sigma = \{a, b\}$; q_2 是起始状态; $F_2 = \{q_1, q_4\}$; δ_2 为:

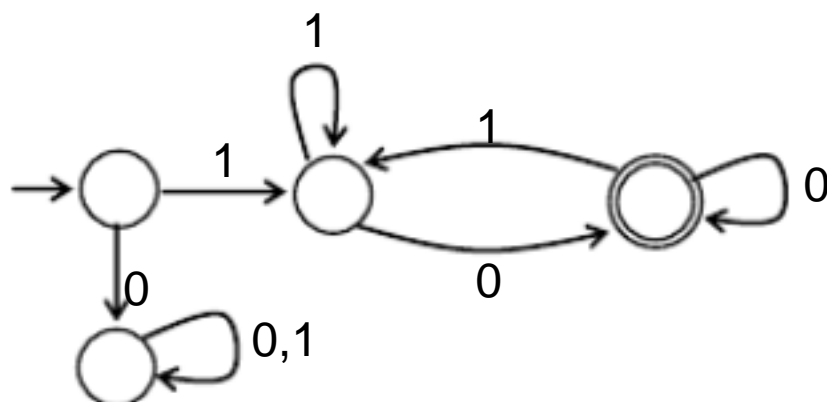
	a	b
q_1	q_1	q_2
q_2	q_3	q_4
q_3	q_2	q_1
q_4	q_3	q_4

2.3、DFA M 的形式描述为 $(\{q_1, q_2, q_3, q_4, q_5\}, \{u, d\}, q_3, \{q_3\})$, 其中 δ 在表 2-3 中给出。试画出此机器的状态图。

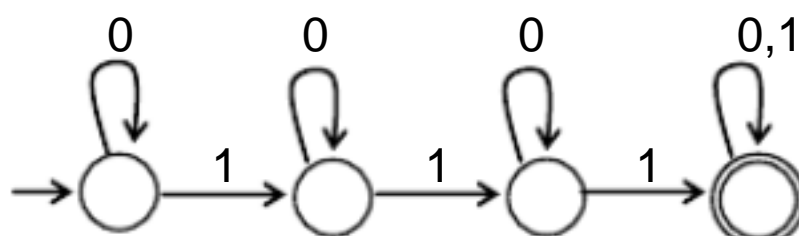


2.4、画出识别下述语言的 DFA 的状态图。

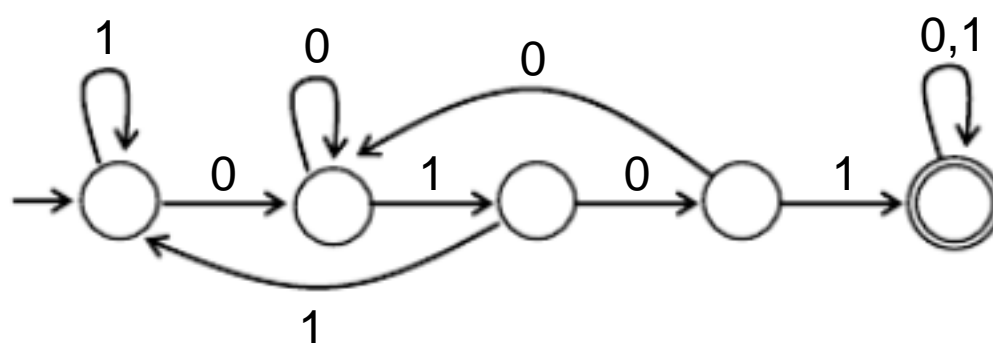
a) $\{w \mid w \text{ 从 } 1 \text{ 开始以 } 0 \text{ 结束}\}$



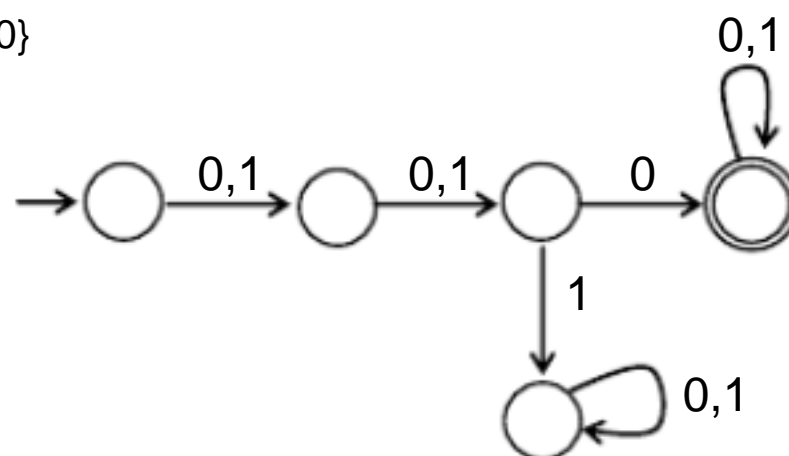
b) $\{w \mid w \text{ 至少有 } 3 \text{ 个 } 1\}$



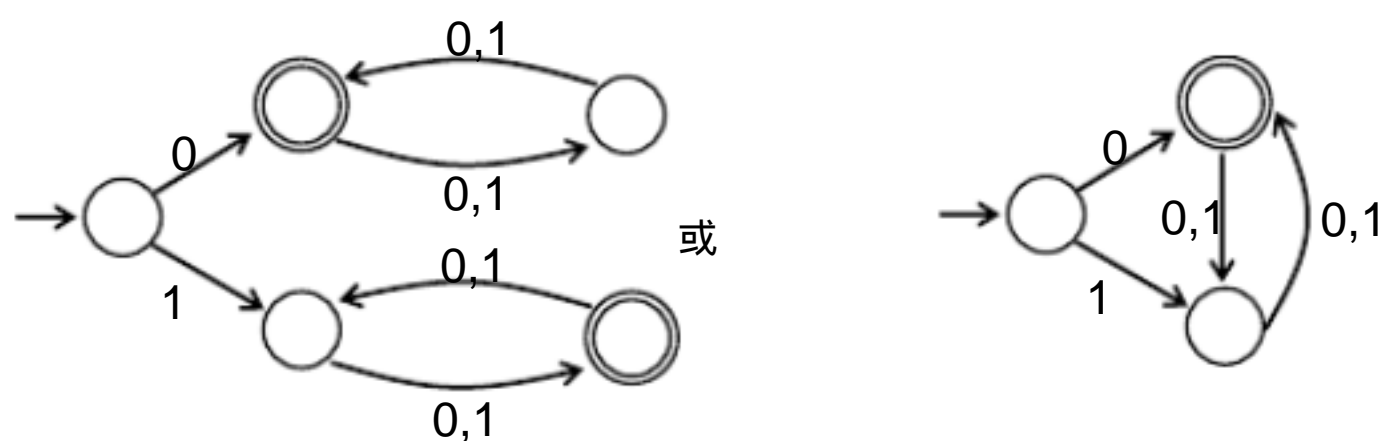
c) $\{w \mid w \text{ 含有子串 } 0101\}$



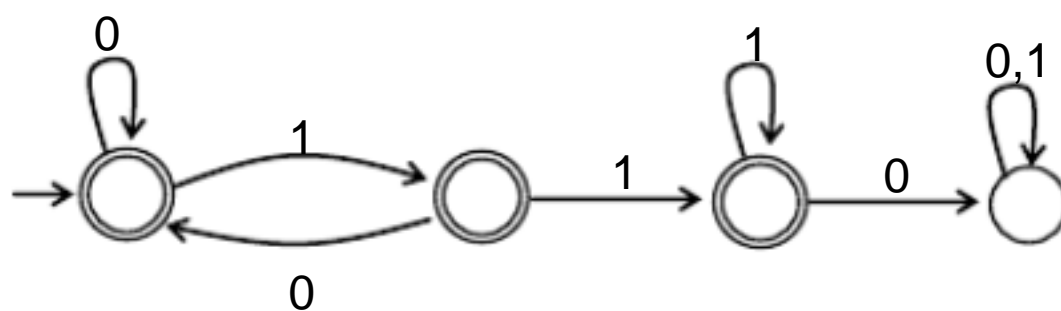
d) $\{w \mid w \text{ 的长度不小于 } 3, \text{ 且第三个符号为 } 0\}$



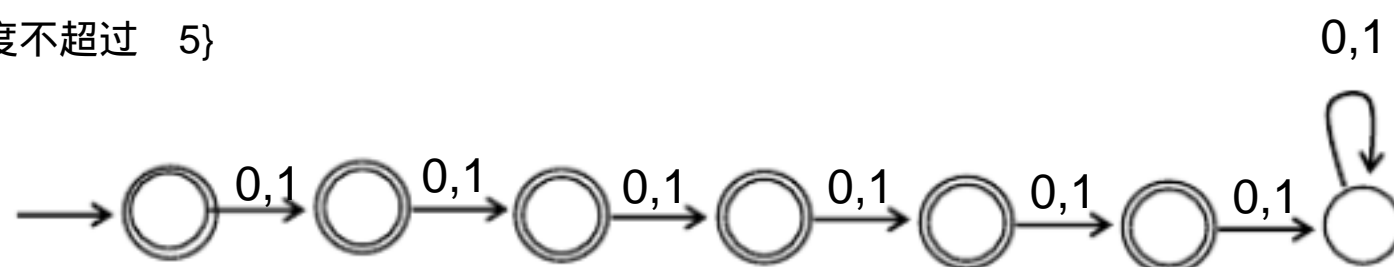
e) $\{w \mid w \text{ 从 } 0 \text{ 开始且为奇长度, 或从 } 1 \text{ 开始且为偶长度}\}$



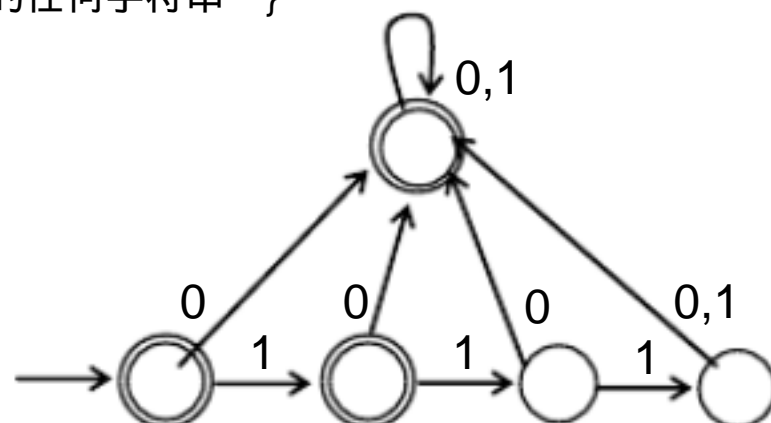
f) $\{w \mid w \text{ 不含子串 } 110\}$



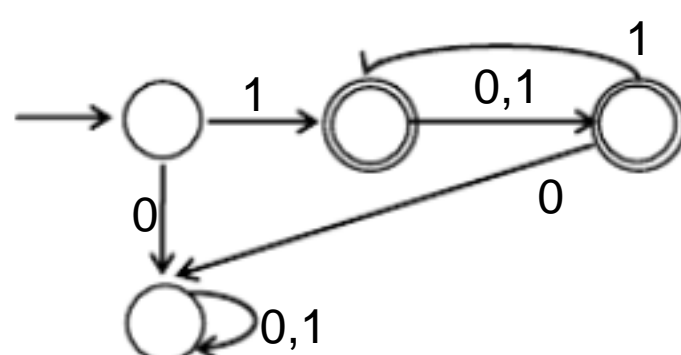
g) $\{w \mid w \text{ 的长度不超过 } 5\}$



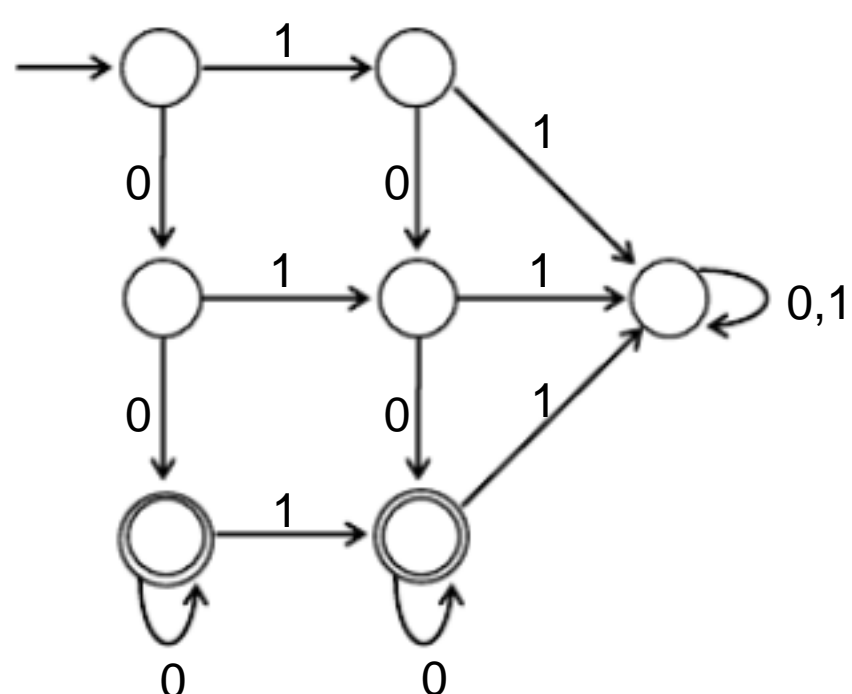
h) $\{w \mid w \text{ 是除 } 11 \text{ 和 } 111 \text{ 以外的任何字符串}\}$



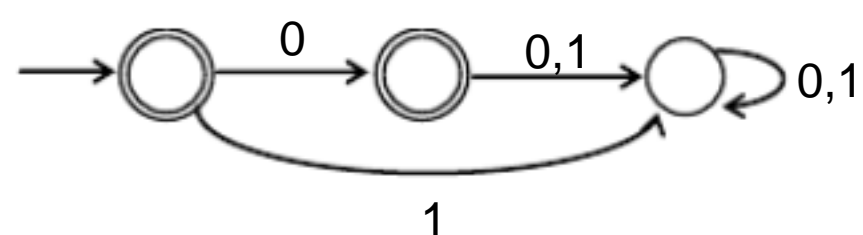
i) $\{w \mid w \text{ 的奇位置均为 } 1\}$



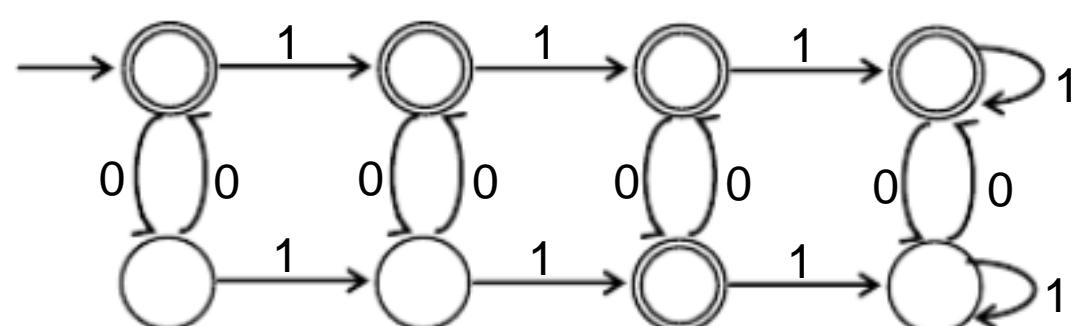
j) $\{w \mid w \text{ 至少含有 } 2 \text{ 个 } 0, \text{ 且至多含有 } 1 \text{ 个 } 1\}$



k) $\{\epsilon, 0\}$



l) $\{w \mid w \text{ 含有偶数个 } 0, \text{ 或恰好两个 } 1\}$, 和下一题 c) 作一比较。



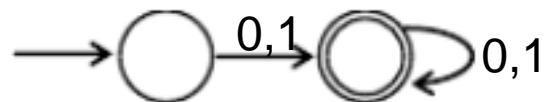
含有偶数个 0, 任意个 1。
注意：偶数包括 0。

含有奇数个 0, 恰好两个 1。

m) 空集

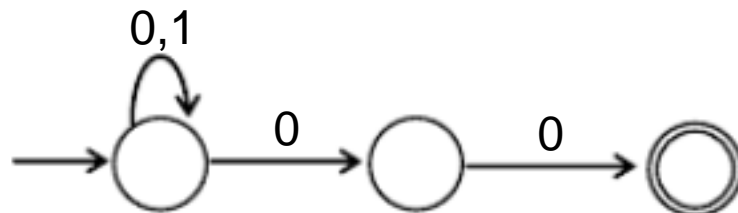


n) 除空串外的所有字符串

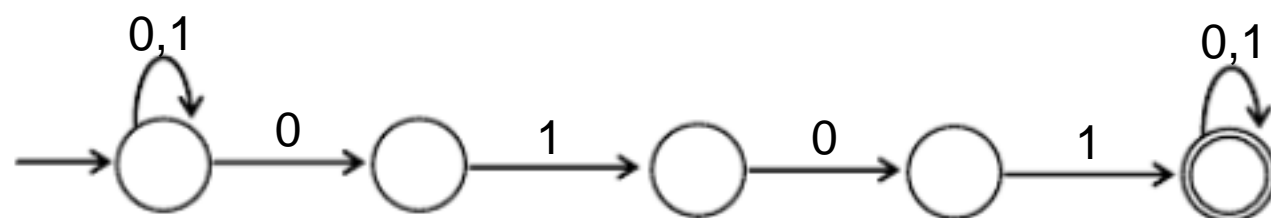


2.5、给出识别下述语言的 **NFA**，且要求符合规定的状态数。

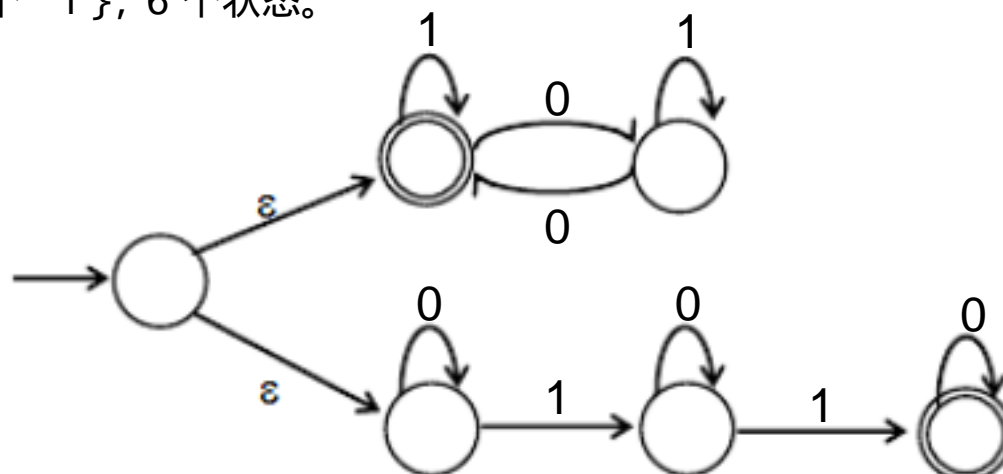
a. $\{w \mid w \text{ 以 } 00 \text{ 结束}\}$ ，三个状态



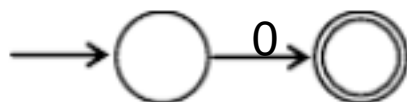
b. 语言 $\{w \mid w \text{ 含有子串 } 0101, \text{ 即对某个 } x \text{ 和 } y, w=x0101y\}$ ，5 个状态。



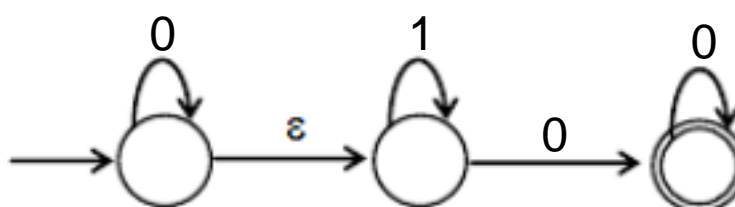
c. 语言 $\{w \mid w \text{ 含有偶数个 } 0 \text{ 或恰好两个 } 1\}$ ，6 个状态。



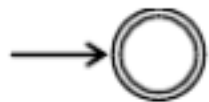
d. 语言 $\{0\}$ ，2 个状态。



e. 语言 0^*10^* ，3 个状态。



f. 语言 $\{\}$ ，1 个状态。



g. 语言 0^* ，1 个状态。



2.9、证明每一台 **NFA** 都能够转换成等价的只有一个接受状态的 **NFA**。

证明：设 $NFA M = \{Q, \Sigma, q_0, F\}$ ，其中

$$F = \{r_1, \dots, r_k\}$$

添加一个状态 E 。原来的接受状态 r_1, \dots, r_k 向 E 分别引箭头，然后将 r_1, \dots, r_k 变为非接受状态， E 变为接受状态。

因为添加箭头不影响 NFA 识别语言，所以命题成立。

2.10、

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

证明：

设 DFA $M = \{Q, \delta, q_0, F\}$

DFA $N = \{Q, \delta, q_0, Q-F\}$

假设 $w = w_1w_2\dots w_n$ 是字母表上任意字符串，因为 M 与 N 均为 DFA，所以必然存在 Q 中状态序列 r_0, r_1, \dots, r_n ，使得：

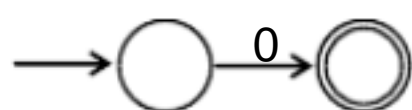
$$r_0 = q_0, \quad (r_i, \delta(r_i, w_{i+1})) = r_{i+1} \quad i = 0, \dots, n-1$$

- 1) 若 $r_n \in F$ ，则 $w \in B$ ；即 M 接受 w 。
- 2) 若 $r_n \notin F$ ，则 $w \notin B$ ；即 M 不接受 w ，即 $w \in B$ 的补集。但是 $r_n \in Q-F$ ，即 N 接受 w ，所以 N 识别 B 的补集。

即 B 的补集是正则语言。因此正则语言类在补运算下封闭。

- b. 举例说明：设 M 是一台识别语言 B 的 NFA，交换 M 的接受状态与非接受状态得到一台新的 NFA，这台新的 NFA 不一定识别 B 的补集。NFA 识别的语言类在补集下封闭吗？解释你的回答。

解：设 B 为 $\{0\}$ 。下图所示的 NFA M 可以识别它。



依题对它作变换，得到 N ：



则 N 识别的语言 $\{\}$ 不是 B 的补集。所以交换 M 的接受状态与非接受状态得到的新的 NFA 不一定识别 B 的补集。

但是由于 NFA 识别的语言类与 DFA 识别的语言类相同，即正则语言类。由 a 的证明，正则语言类在补运算封闭，可知，NFA 识别的语言类，即正则语言类在补运算下封闭。

若 NFA 识别语言 A ，必有等价的 DFA 识别 A ，从而由 a 知，可交换 DFA 的接受与非接受状态构造识别 A 的补集的 DFA，则必有等价的 NFA 识别 A 的补集。只是，该 NFA 未必由原 NFA 交换接受与非接受状态构造而成。

2.11 给出一个反例，说明下述构造不能证明定理 2.24，即正则语言类在星号运算下封闭。

设 $N_1 = (Q_1, \delta_1, q_1, F_1)$ 识别 A_1 。如下构造 $N = (Q, \delta, q, F)$ 。 N 应该识别 A_1^* 。

- a. N 的状态集是 N_1 的状态集。
- b. N 的起始状态与 N_1 的起始状态相同。
- c. $F = \{q_1\} \cup F_1$ 。

F 的接受状态是原来的接受状态加上它的起始状态。

- d. 定义 δ 如下：对每一个 q 属于 Q_1 和每一个 a 属于 Σ_ϵ 。

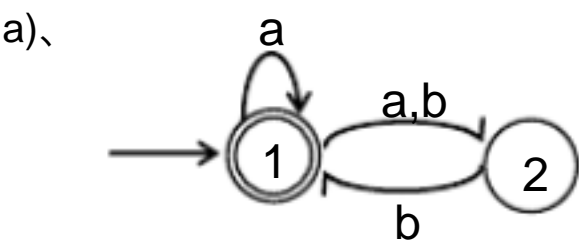
$$\delta(q,a) = \begin{cases} \delta_1(q,a), & \text{若 } q \notin F_1 \text{ 或 } a \neq \epsilon \\ \delta_1(q,a) \cup \{q\}, & \text{若 } q \in F_1 \text{ 且 } a = \epsilon \end{cases}$$

解：设 N_1 识别语言 $A = \{ \text{至少含有一个 } 1 \}$ ，其中输入字母表为 $\{0, 1\}$ ，可知 $A^* = \{ \text{或至少含有一个 } 1 \}$ 。



按上述规定构造 N 的状态图如上。可以看出 N 确实把 0 加进被识别的语言中，但是它也把字符 1 加进被识别的语言中。所以 $L(N) \neq A^*$ 。如此构造不能证明定理 2.24。

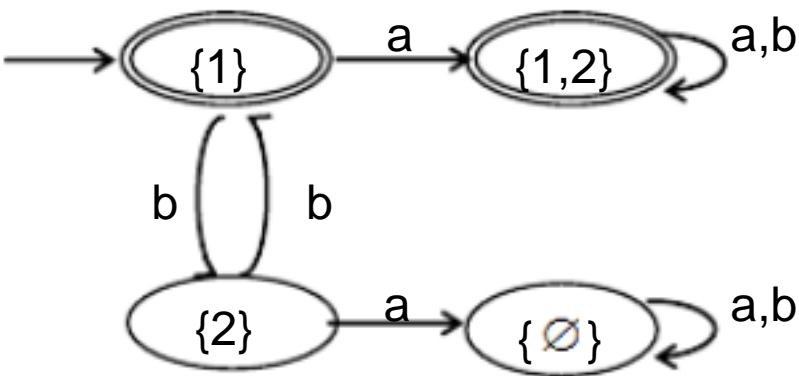
2.12 使用定理 2.19 中给出的构造，把下图中的两台非确定型有穷自动机转换成等价的确定型有穷自动机。



这一台 NFA 有两个状态 $1、2$ ，所以 DFA 有四个状态 $\{\emptyset, \{1\}, \{2\}, \{1,2\}\}$ 。

起始状态是 $\{1\}$ ，它等于从 1 出发沿着箭头能到达的所有状态加上 $\{1\}$ 本身。接受状态是所有包含接受状态的状态子集，即 $\{\{1\}, \{1,2\}\}$ 。

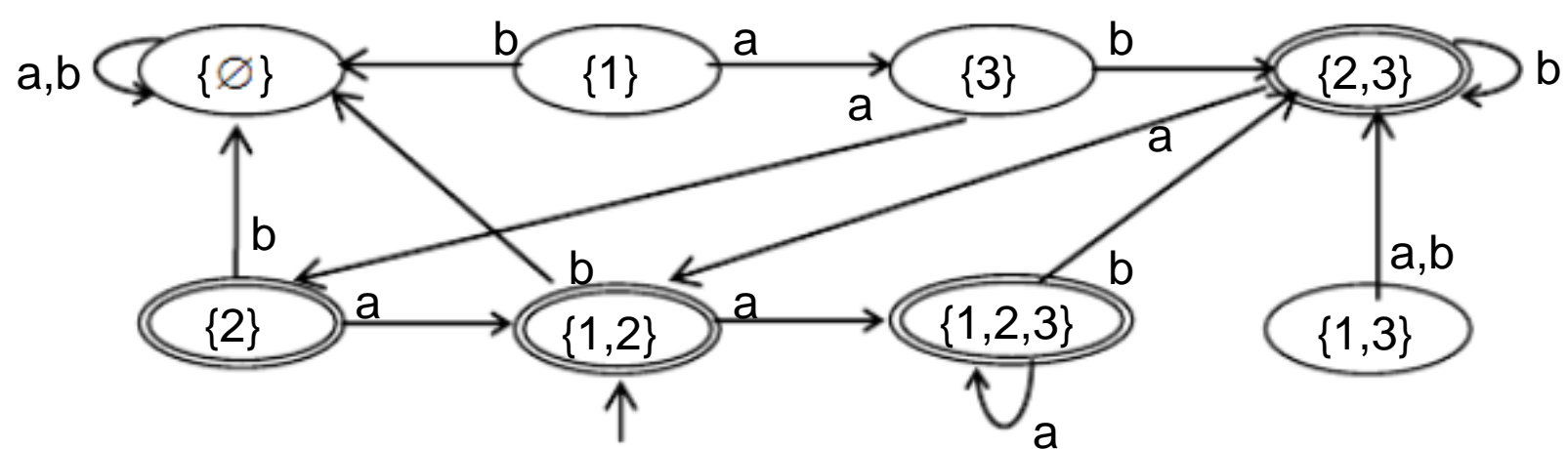
确定转移函数。每一个状态遇到输入 a 到一个地方，输入 b 到一个地方。



```

graph TD
    Start(( )) --> 1((1))
    1 -- ε --> 2(((2)))
    1 -- a --> 3((3))
    3 -- "a,b" --> 2
    3 -- b --> 3
  
```

D 的状态图如下：



The diagram shows a DFA with the following states and transitions:

- States: $\{\emptyset\}$, $\{2\}$, $\{1,2\}$, $\{3\}$, $\{1,2,3\}$, and $\{2,3\}$.
- Start state: $\{\emptyset\}$ (indicated by an incoming arrow from the left).
- Final state: $\{2,3\}$ (indicated by a double circle).
- Transitions:
 - $\{\emptyset\} \xrightarrow{a,b} \{\emptyset\}$ (self-loop)
 - $\{2\} \xrightarrow{b} \{\emptyset\}$
 - $\{2\} \xrightarrow{a} \{1,2\}$
 - $\{1,2\} \xrightarrow{b} \{\emptyset\}$
 - $\{1,2\} \xrightarrow{a} \{1,2,3\}$
 - $\{3\} \xrightarrow{a} \{\emptyset\}$
 - $\{3\} \xrightarrow{b} \{2,3\}$
 - $\{1,2,3\} \xrightarrow{a} \{1,2,3\}$ (self-loop)
 - $\{1,2,3\} \xrightarrow{b} \{2,3\}$
 - $\{2,3\} \xrightarrow{a} \{1,2,3\}$
 - $\{2,3\} \xrightarrow{b} \{2,3\}$ (self-loop)

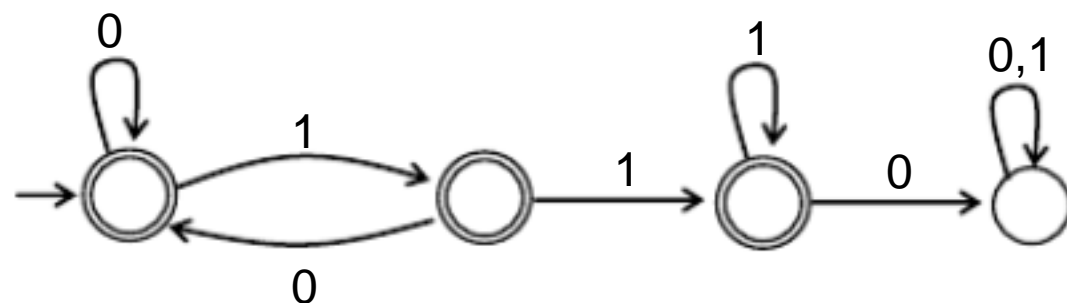
```

graph LR
    S(( )) -- b --> S12(((1,2)))
    S12 -- "a,b" --> S
    S12 -- a --> S123(((1,2,3)))
    S123 -- b --> S12
    S123 -- a --> S23(((2,3)))
    S23 -- b --> S123
    S23 -- b --> S23
    style S fill:#fff,stroke:#000,stroke-width:1px
    style S12 fill:#fff,stroke:#000,stroke-width:2px
    style S123 fill:#fff,stroke:#000,stroke-width:2px
    style S23 fill:#fff,stroke:#000,stroke-width:2px
  
```

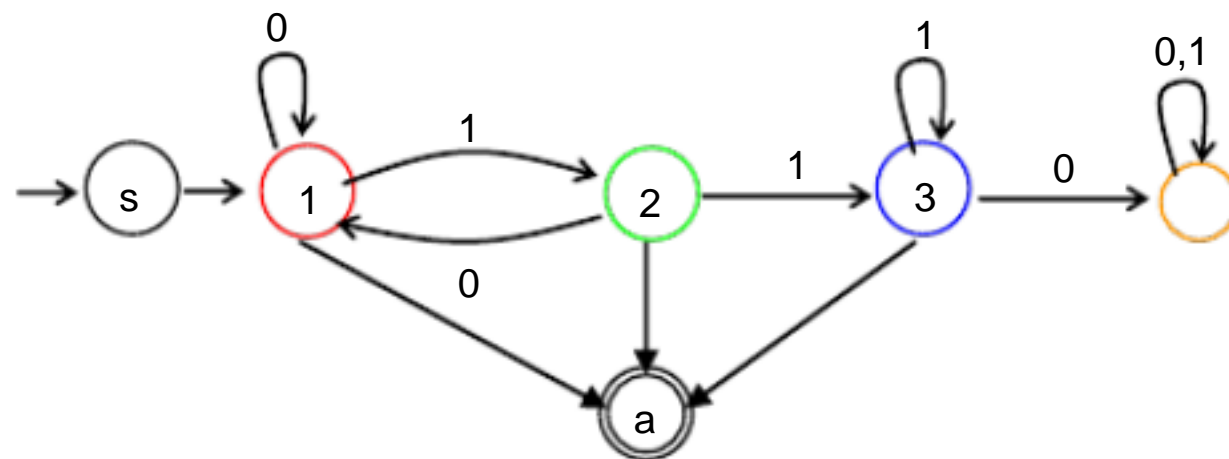
a. $\{w \mid w \text{ 从 } 1 \text{ 开始以 } 0 \text{ 结束}\} 1^* 0.$

- b. $\{w \mid w \text{ 至少有 3 个 } 1\} = 1^* 1^* 1^* 1^*$.
- c. $\{w \mid w \text{ 含有子串 } 0101\} = 0^* 0101^*$.
- d. $\{w \mid w \text{ 的长度不小于 3, 且第三个符号为 } 0\} = 0^*$.
- e. $\{w \mid w \text{ 从 } 0 \text{ 开始且为奇长度, 或从 } 1 \text{ 开始且为偶长度}\} = 0()^* 1()^*$.
- f. $\{w \mid w \text{ 不含子串 } 110\}$.

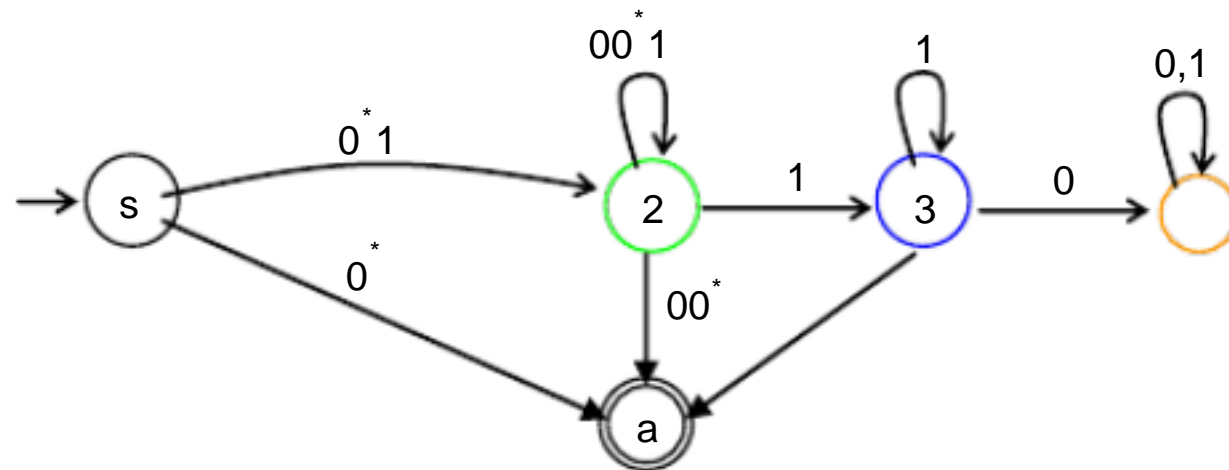
1) 识别该语言的确型有穷自动机如下，按照定理 2.28 证明中使用的算法进行转换：



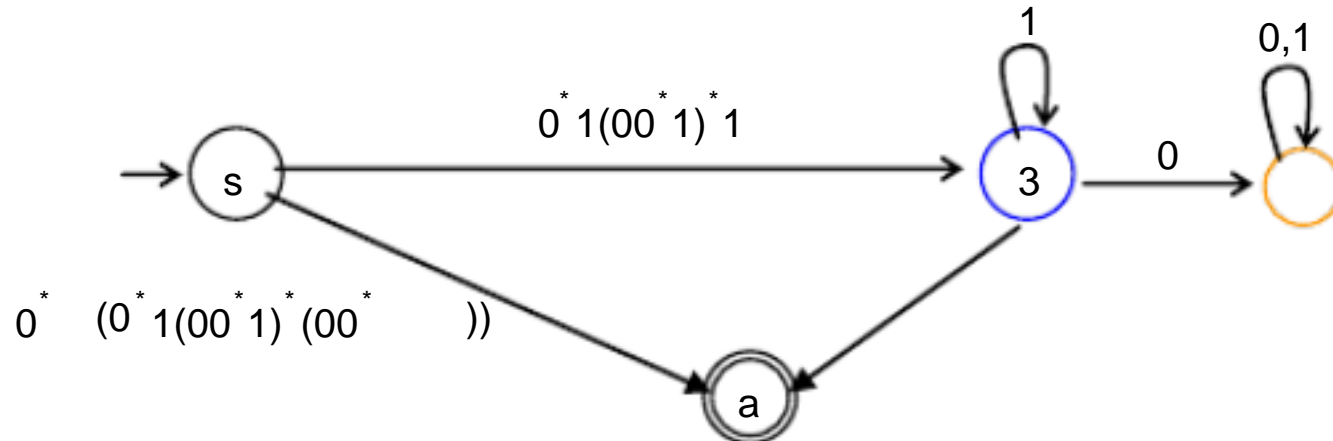
1、加新的开始状态和新的结束状态



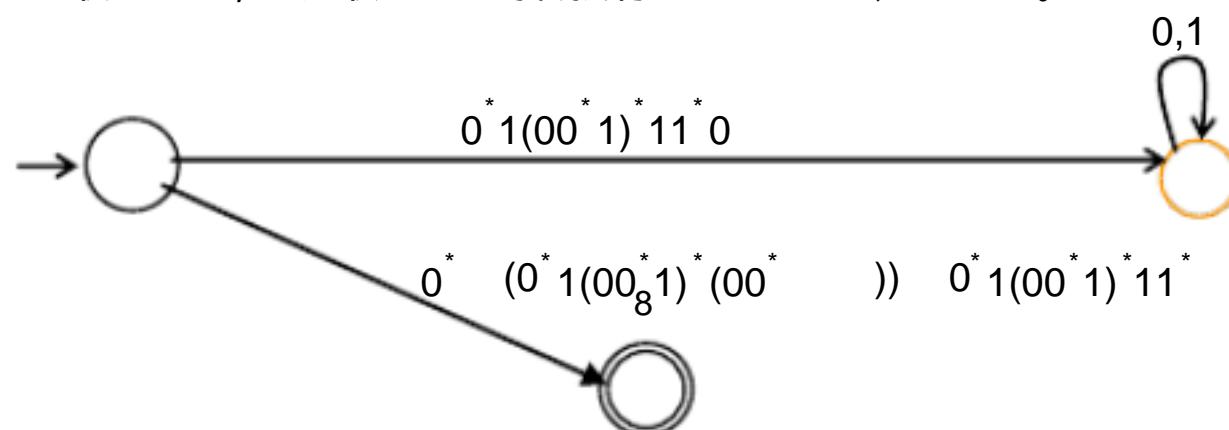
2、删除红色状态 1，经过状态 1 的转换有：s 1 a、s 1 2、2 1 a、2 1 2。



3、删除绿色状态 2，经过状态 2 的转换有：s 2 a、s 2 3。



4、删除蓝色状态 3，经过状态 3 的转换有：s 3 a、s 3 。



等价的正则表达式为： $0^* \{0^* 1(00^* 1)^* (00^* 11)^*\}$ $0^* 1(00^* 1)^* 11^* = 0^* \{0^* 1(00^* 1)^* (00^* 11)^*\}$

0^* ：只由 0 组成的字符串，或为空。

$0^* 1(00^* 1)^* (00^* 11)^*$ ：1 和 1 之间至少有一个 0，如果出现子串 11，那么子串 11 的后面一定是 1 或者字符串结束。

g. $\{w \mid w \text{ 的长度不超过 } 5\}$ ϵ

h. $\{w \mid w \text{ 是除 } 11 \text{ 和 } 111 \text{ 以外的任何字符串}\}$

1、说明：可能的值有

0^* ：以 0 开头的字符串。

1：一个字符 1。

11：

100^* ：以 110 开头的所有字符串

101^* ：以 110 开头的所有字符串

110^* ：以 110 开头的所有字符串

111：

1000^* ：以 111 开头的所有字符串

因为 100^* 和 101^* 可以合并为 10^* ，所以除 11 和 111 以外的任何字符串可以表示为：

$\epsilon 0^* 1 10^* 110^* 111^*$

2、说明： 0^* ：至少包含一个 0 的字符串。

1：一个字符 1。

11111^* ：至少包含四个 1 的全 1 字符串。

所以除 11 和 111 以外的任何字符串可以表示为： $\epsilon 0^* 1 11111^*$

i. $\{w \mid w \text{ 的奇位置均为 } 1\} 1 (1^*)^* 1 (1^*)^*$

j. $\{w \mid w \text{ 至少含有 } 2 \text{ 个 } 0, \text{ 且至多含有 } 1 \text{ 个 } 1\} 0^* (00^* 010^* 001^* 100)^* 0^*$

k. $\{, 0\}$ 0

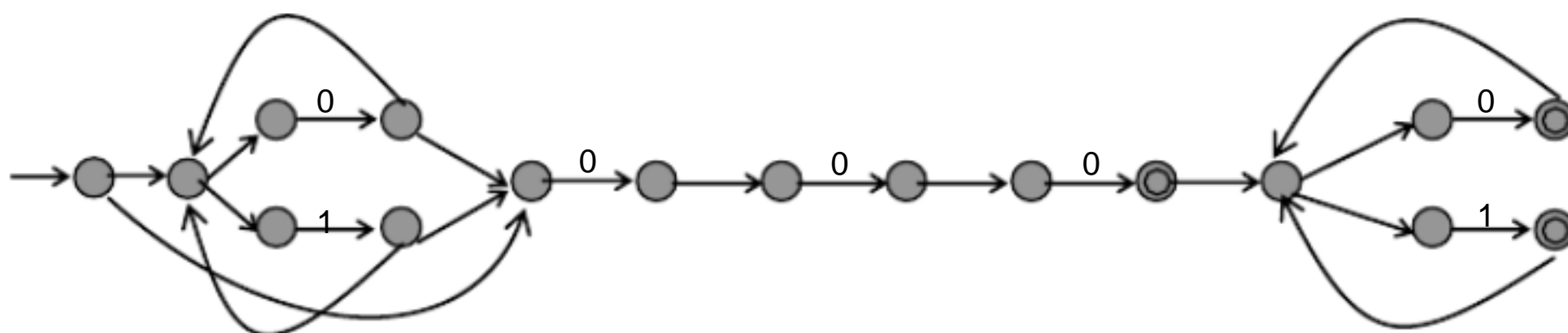
l. $\{w \mid w \text{ 含有偶数个 } 0, \text{ 或恰好两个 } 1\} (1^* 01^* 0)^* 1^* 0^* 10^* 10^*$

m. 空集 \emptyset

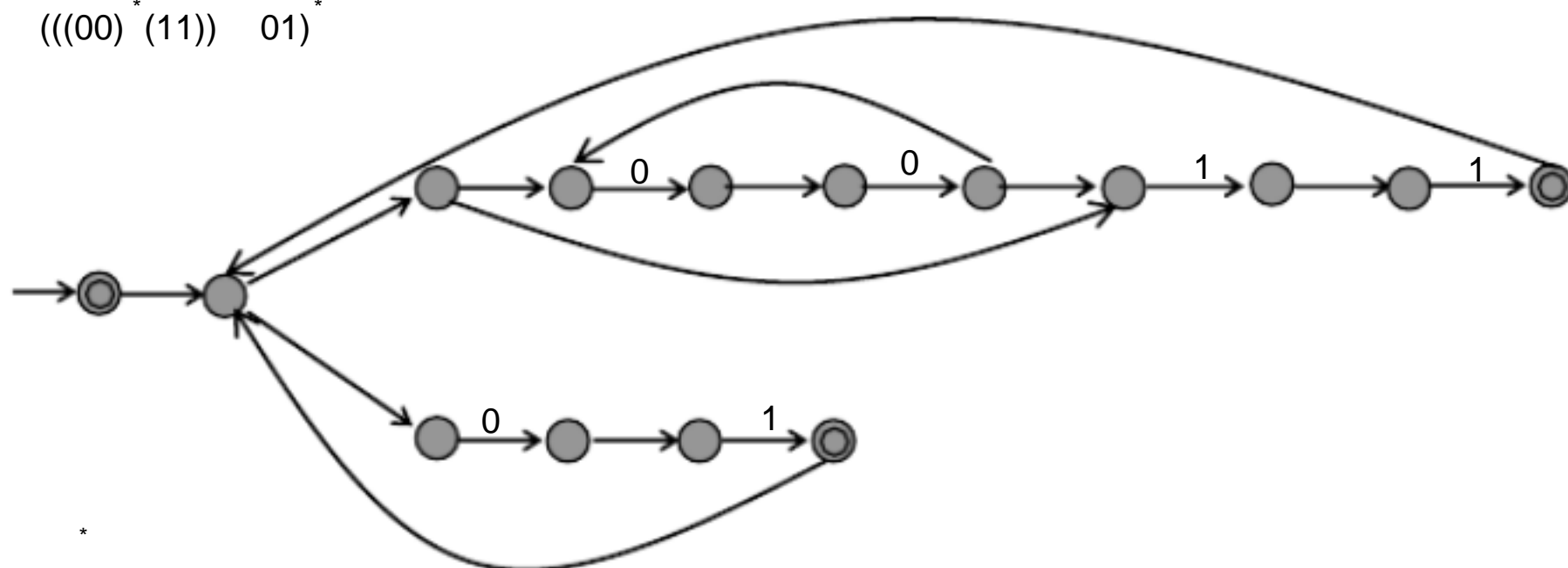
n. 除空串外的所有字符串 ϵ

2.14 使用引理 2.29 中叙述的过程，把下述正则表达式转换成非确定型自动机。

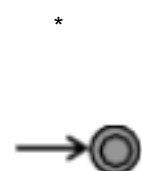
a. $(0\ 1)^* 000(0\ 1)^*$



b. $((00)^* (11)^* 01)^*$



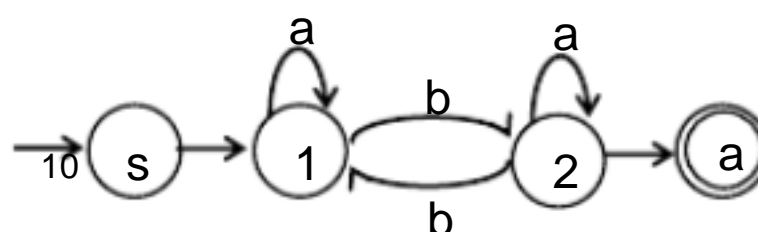
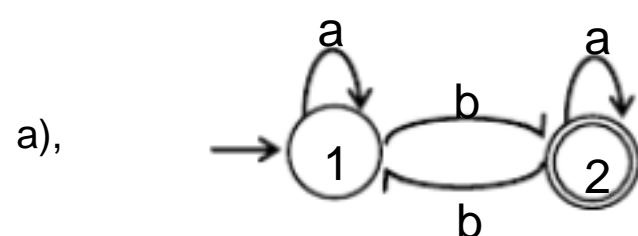
c.



2.15 对下述每一个语言，给出 4 个字符串，2 个是这个语言的成员，2 个不是这个语言的成员。这里假设字母表 $\Sigma = \{a, b\}$ 。

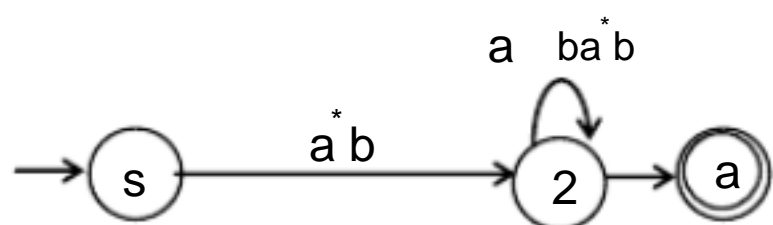
- | | | |
|--------------------|----------------|---------------------|
| a. $a^* b^*$ | 成员：ab, aab | 非成员：aba, ba |
| b. $a(ba)^* b$ | 成员：ab, abab | 非成员：abb, aa |
| c. $a^* b^*$ | 成员：aaa, bbb | 非成员：ab, ba |
| d. $(aaa)^*$ | 成员：aaa, aaaaaa | 非成员：a, aa |
| e. $a^* b^* a^*$ | 成员：aba, aaba | 非成员：aa, abb |
| f. $aba\ bab$ | 成员：aba, bab | 非成员：a, b |
| g. $(\epsilon a)b$ | 成员：b, ab | 非成员：a, bb |
| h. $(a\ ba\ bb)^*$ | 成员：a, bb | 非成员： ϵ , b |

2.16 使用引理 2.32 中叙述的过程，把图 2-38 中的有穷自动机转换成正则表达式。

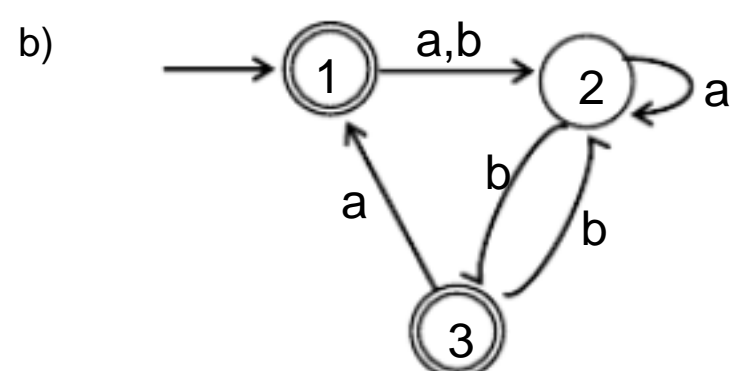
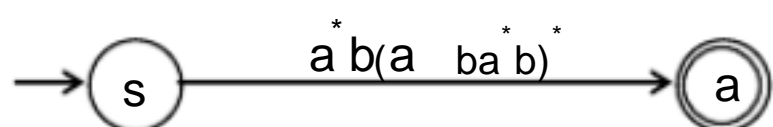


解：1、加新的开始状态和新的结束状态

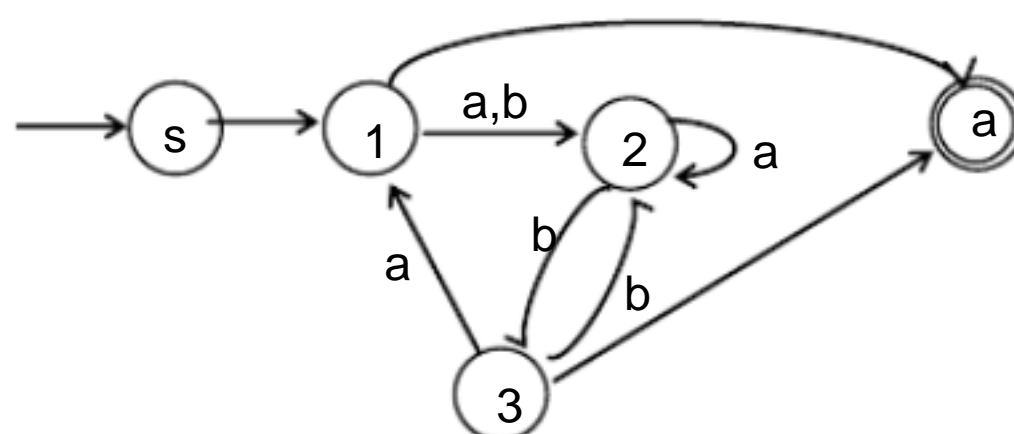
2、删除状态 1, 通过状态 1 的转换有 $s \rightarrow 1 \rightarrow 2$ 、 $2 \rightarrow 1 \rightarrow 2$



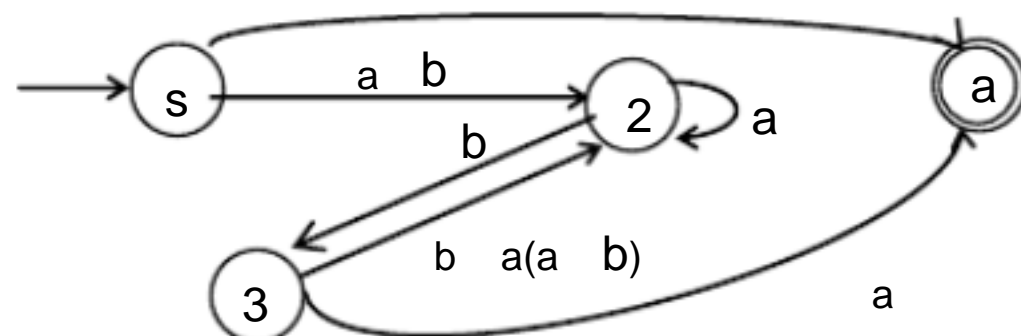
3、删除状态 2



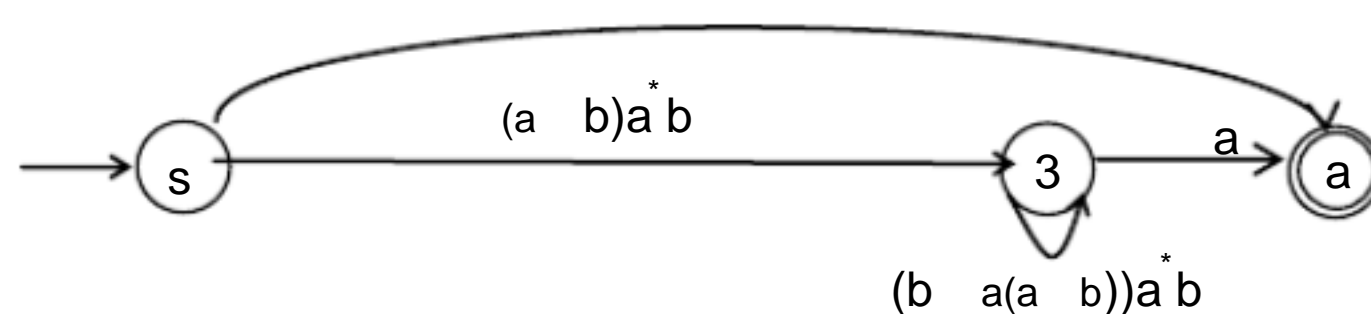
1、加新的开始状态和新的结束状态



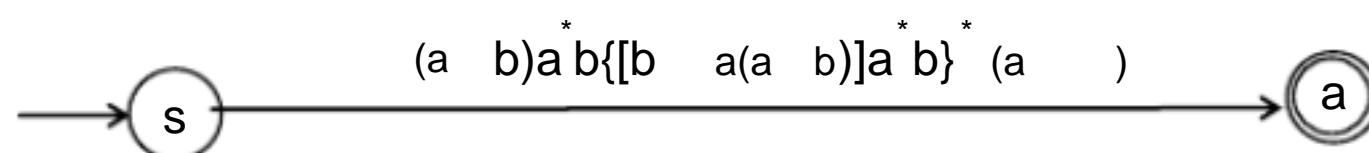
2、删除状态 1, 通过状态 1 的转换有 $s \rightarrow 1 \rightarrow 2$ 、 $s \rightarrow 1 \rightarrow a$ 、 $3 \rightarrow 1 \rightarrow a$ 、 $3 \rightarrow 1 \rightarrow 2$



3、删除状态 2, 通过状态 2 的转换有 $s \rightarrow 2 \rightarrow 3$ 、 $3 \rightarrow 2 \rightarrow 3$



3、删除状态 3



2.17 利用泵引理证明下述语言不是正则的。

a. $A_1 = \{0^n 1^n 2^n \mid n \geq 0\}$ 。

证明：假设 A_1 是正则的。设 p 是泵引理给出的关于 A_1 的泵长度。

令 $S = 0^p 1^p 2^p$ ，

S 是 A_1 的一个成员且 S 的长度大于 p ，所以泵引理保证 S 可被分成 3 段 $S = xyz$ 且满足泵引理的 3 个条件。根据条件 3， $|xy| \leq p$ ，所以 y 中只包含 0。在字符串 $xyyz$ 中，0 比 1、2 多，所以 $xyyz$ 不可能是 A_1 的成员。违反泵引理的条件 1，矛盾。

A_1 不是正则的。

b. $A_2 = \{www \mid w \in \{a,b\}^*\}$ 。

证明：假设 A_2 是正则的。设 p 是泵引理给出的关于 A_2 的泵长度。

令 $S = a^p b a^p b a^p b$ ，

S 是 A_2 的一个成员且 S 的长度大于 p ，所以泵引理保证 S 可被分成 3 段 $S = xyz$ 且满足泵引理的 3 个条件。根据条件 3， y 中只含 a ，所以 $xyyz$ 中第一个 a 的个数将比后两个 a 的个数多，故 $xyyz$ 不是 A_2 的成员。违反泵引理的条件 1，矛盾。

A_2 不是正则的。

c. $A_3 = \{a^{2^n} \mid n \geq 0\}$ 。(在这里， a^{2^n} 表示一串 2^n 个 a 。)

证明：假设 A_3 是正则的。设 p 是泵引理给出的关于 A_3 的泵长度。

令 $S = a^{2^p}$ ，

S 是 A_3 的一个成员且 S 的长度大于 p ，所以泵引理保证 S 可被分成 3 段 $S = xyz$ 且满足泵引理的 3 个条件。考虑一下 2^n 序列。

1, 2, 4, 8, 16,

注意到序列中相邻的两个数之间的间隔在增大，序列中大的数不可能彼此靠近。考虑字符串

xy^iz 与 $xy^{i+1}z$, 它们之间的长度相差 y 的长度。如果选择很大的 i , 则 xy^iz 与 $xy^{i+1}z$ 的长度靠得太近, 不可能都是 2 的幂。

设 $m=2^n$, 那么它和更大的 2 的幂之间的差为: $2^{n+1} - 2^n = 2^n = m$, 令 $|xy^iz|$ 为上述 m 。

如果 $|y| < |xy^iz| = m$, xy^iz 与 $xy^{i+1}z$ 就不可能都是 2 的幂。计算 i 值。

$$|y| \mid |xyz| = |S| = |a^{2^p}| = 2^p, \text{ 令 } i = 2^{p+1}$$

$$\text{有 } |y| \cdot 2^p < 2^{p+1} \mid |xy^{2^{p+1}}z| = |xy^iz| = m$$

A_3 不是正则的。

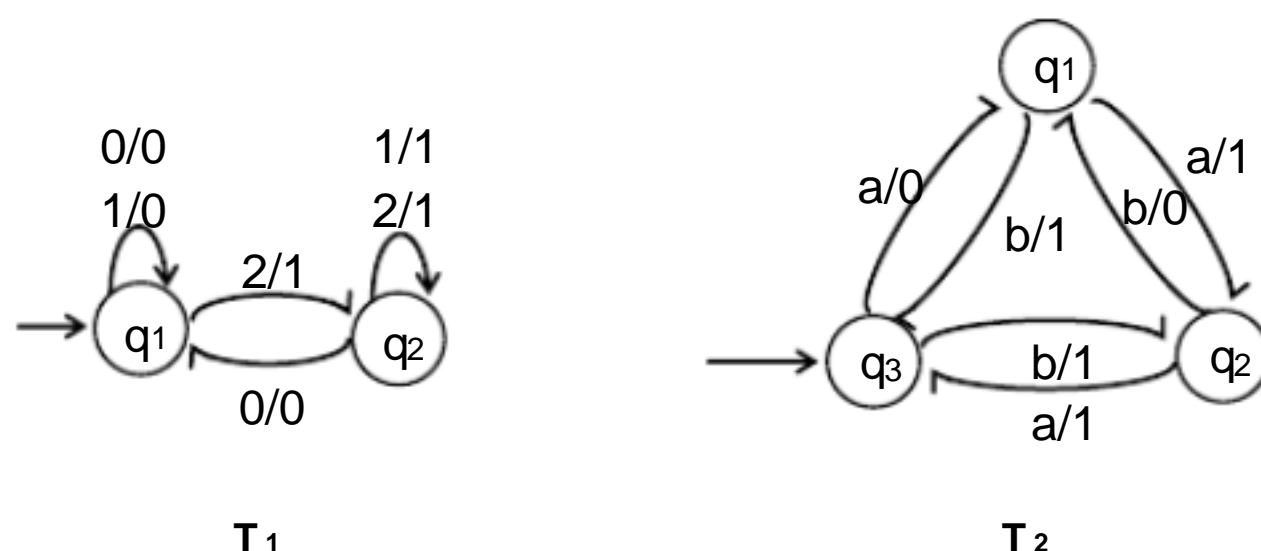
2.18 下面“证明” 0^*1^* 不是正则语言, 指出这个“证明”中的错误。 (因为 0^*1^* 是正则的, 所以一定错误。)

采用反证法证明。假设 0^*1^* 是正则的。令 p 是泵引理给出的关于 0^*1^* 的泵长度。取 S 为字符串 0^p1^p 。 S 是 0^*1^* 的一个成员, 但是例 2.38 已证明 S 不能被抽取。于是得到矛盾, 所以 0^*1^* 不是正则的。

解: 在例 2.38 中的语言是 $\{0^n1^n \mid n \geq 0\}$, 取 S 为字符串 0^p1^p , S 确实不能被抽取。

但针对语言 0^*1^* , S 是能被抽取的。将 S 分成三段 $S=xyz$, 由泵引理的条件 3, y 仅包含 0 , 而 xy^iz 属于语言 0^*1^* , 即 S 能被抽取, 故题设中的“证明”不正确。

2.19 有穷状态转换器 (FST) 是确定性有穷自动机的一种类型。 它的输出是一个字符串, 而不仅仅是接受或拒绝。图 2—39 是两台有穷状态状态转换器 T_1 和 T_2 的状态图。



FST 的每一个转移用两个符号标记, 一个指明该转移的输入符号, 另一个指明输出符号。两个符号之间用斜杠 “/” 把它们分开。在 T_1 中, 从 q_1 到 q_2 的转移有输入符号 2 和输出符号 1 。某些转移可能有多对输入-输出, 比如 T_1 中从 q_1 到它自身的转移。FST 在对输入串 w 计算时, 从起始状态开始,

一个接一个地取输入符号 $w_1 \dots w_n$ ，并且比照输入标记和符号序列 $w_1 \dots w_n = w$ 进行转移。每一次沿一个转移走一步，输出对应的输出符号。例如，对输入 **2212011**，机器 T_1 依次进入状态 $q_1, q_2, q_2, q_2, q_1, q_1, q_1$ 和输出 **1111000**。对输入 **abbb**， T_2 输出 **1011**。给出在下述每一小题中机器进入的状态序列和产生的输出。

- T_1 对输入串 011，输出：000，状态序列： q_1, q_1, q_1, q_1 。
- T_1 对输入串 211，输出：111，状态序列： q_1, q_2, q_2, q_2 。
- T_1 对输入串 0202，输出：0101，状态序列： q_1, q_1, q_2, q_1, q_2 。
- T_2 对输入串 b，输出：1，状态序列： q_1, q_3 。
- T_2 对输入串 bbab，输出：1111，状态序列： q_1, q_3, q_2, q_3, q_2 。
- T_2 对输入串 bbbbbb，输出：110110，状态序列： $q_1, q_3, q_2, q_1, q_3, q_2, q_1$ 。
- T_2 对输入串 ϵ 输出： ϵ 状态序列： q_1 。

2.20 给出有穷状态转换器的形式定义。

解：有穷状态转换器 FST 是一个五元组 $(Q, \Sigma, \Delta, \delta, q_0)$

- 1) Q 是一个有穷集合，叫做状态集
- 2) Σ 是一个有穷集合，叫做输入字母表
- 3) Δ 是一个有穷集合，叫做输出字母表
- 4) $\delta: Q \times \Sigma \rightarrow Q \times \Delta$ 是转移函数
- 5) q_0 是起始状态

FST 计算的形式定义：

$M = (Q, \Sigma, \Delta, \delta, q_0)$ 是一台有穷状态转换器， $w = w_1 w_2 \dots w_n$ 是输入字母表上的一个字符串。若存在

Q 中的状态序列： r_0, r_1, \dots, r_n 和输出字母表上的一个字符串 $s = s_1 \dots s_n$ ，满足下述条件：

- 1) $r_0 = q_0$;
- 2) $(r_i, w_{i+1}) = (r_{i+1}, s_{i+1}) \quad i = 0, 1, \dots, n-1$

则 M 在输入 w 时输出 s 。

2.21 利用你给练习 2.20 的答案，给出练习 2.19 中画出的机器 T_1 和 T_2 的形式描述。

解：有穷状态转换器 T_1 的形式描述为：

$$T_1 = (\{q_1, q_2\}, \{0, 1, 2\}, \{0, 1\}, \delta, q_1)$$

其中转移函数 δ 为：

	0	1	2
q ₁	q ₁ /0	q ₁ /0	q ₂ /1
q ₂	q ₁ /0	q ₂ /1	q ₂ /1

有穷状态转换器 T₂的形式描述为：

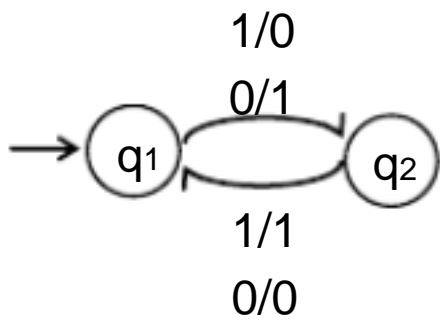
$$T_2=\{\{ q_1, q_2, q_3\}, \{ a, b\},\{0,1\}, \quad , q_1)$$

其中转移函数为：

	a	b
q ₁	q ₂ /1	q ₃ /1
q ₂	q ₃ /1	q ₁ /0
q ₃	q ₁ /0	q ₂ /1

mn2.22 给出一台具有下述行为的 **FST** 的状态图。它的输入、输出字母表都是 **{0,1}**。它输出的字符串与输入的字符串偶数位相同、奇数位相反。例如，对于输入 **0000111**，它应该输出 **1010010**。

解：



第 3 章、上下文无关语言习题解答 - 练习

3.1 回忆一下例 3.3 中给出的 CFG G_4 。为方便起见，用一个字母重新命名它的变元如下：

$$E \rightarrow E + T \mid T$$

$$T \rightarrow T \times E \mid F$$

$$F \rightarrow (E) \mid a$$

给出下述字符串的语法分析树和派生。

- a
- a+a
- a+a+a
- ((a))

答：

$$a. \quad E \Rightarrow T \Rightarrow F \Rightarrow a$$

$$b. \quad E \Rightarrow E + T \Rightarrow T + F \Rightarrow F + a \Rightarrow a + a$$

$$c. \quad E \Rightarrow E + T \Rightarrow E + T + F \Rightarrow T + F + a \Rightarrow F + a + a \Rightarrow a + a + a$$

$$d. \quad E \Rightarrow T \Rightarrow F \Rightarrow (E) \Rightarrow (T) \Rightarrow (F) \Rightarrow ((E)) \Rightarrow ((T)) \Rightarrow ((F)) \Rightarrow ((a))$$

3.2 a. 利用语言 $A = \{a^m b^n c^n \mid m, n \geq 0\}$ 和 $B = \{a^n b^n c^m \mid m, n \geq 0\}$ 以及例 3.20 (语言 $B = \{a^n b^n c^n \mid n \geq 0\}$ 不是上下文无关的)，证明上下文无关语言在交的运算下不封闭。

b. 利用 (a) 和 DeMorgan 律 (定理 1.10)，证明上下文无关语言在补运算下不封闭。

证明：

a. 先说明 A, B 均为上下文无关文法，对 A 构造 CFG C_1

$$S \rightarrow aS \mid T \mid \varepsilon$$

$$T \rightarrow bTc \mid \varepsilon \quad // \text{生成 } b^n c^n$$

对 B, 构造 CFG C_2

$$S \rightarrow Sc \mid R \mid \varepsilon$$

$$R \rightarrow aRb \mid \varepsilon \quad // \text{生成 } a^n b^n$$

由此知 A, B 均为上下文无关语言。

由例 3.20, $A \cap B = \{a^n b^n c^n \mid n \geq 0\}$ (假设 $m = n$) 不是上下文无关语言，所以上下文无关语言在交的运算下不封闭。

b. 用反证法。

假设 CFL 在补运算下封闭，则对于 (a) 中上下文无关语言 A, B ， \bar{A}, \bar{B} 也为 CFL。因为 CFL 对并运算封闭，所以 $\bar{A} \cup \bar{B}$ 也为 CFL，进而知道 $\overline{\bar{A} \cup \bar{B}}$ 为 CFL。由 DeMorgan 定律 $\overline{\bar{A} \cup \bar{B}} = A \cap B$ ，得出 $A \cap B$ 是 CFL。

这与 (a) 的结论矛盾，所以 CFL 对补运算不封闭。

3.3 设上下文无关文法 G ：

$R \rightarrow XRX|S$

$S \rightarrow aTb|bTa$

$T \rightarrow XTX|X$

$X \rightarrow a|b$

回答下述问题：

a. G 的变元和终结符是什么？起始变元是哪个？

答：变元是： R, X, S, T ；起始变元是 R 。终结符是： a, b ，

b. 给出 $L(G)$ 中的三个字符串。 答： ab, ba, aab 。

c. 给出不在 $L(G)$ 中的三个字符串。 答： a, b, ϵ 。

d. 是真是假： $T \Rightarrow aba$ 。 答：假

e. 是真是假： $T \Rightarrow^* aba$ 。 答：真

f. 是真是假： $T \Rightarrow T$ 。 答：假

g. 是真是假： $T \Rightarrow^* T$ 。 答：假

h. 是真是假： $XXX \Rightarrow^* aba$ 。 答：真

i. 是真是假： $X \Rightarrow^* aba$ 。 答：假

j. 是真是假： $T \Rightarrow^* XX$ 。 答：真

k. 是真是假： $T \Rightarrow^* XXX$ 。 答：真

l. 是真是假： $X \Rightarrow^* \epsilon$ 。 答：假

m. 用普通的语言描述 $L(G)$ ：

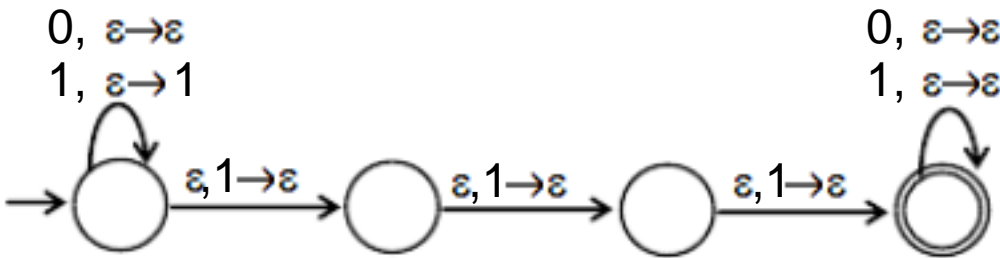
3.4 和 3.5 给出产生下述语言的上下文无关文法和 PDA，其中字母表 $\Sigma = \{0, 1\}$ 。

a. $\{w \mid w \text{ 至少含有 } 3 \text{ 个 } 1\}$

S A1A1A1A

A 0A|1A| ϵ

读输入中的符号。每读一个 1，把一个 1 推入栈，每读 1 个 0，不读栈也不写栈。同时非确定性地转移，并把 1 个 1 弹出栈。如果能转移三次，共弹出三个 1，则接受这个输入，并继续读输入符号直至结束。否则拒绝这个输入。

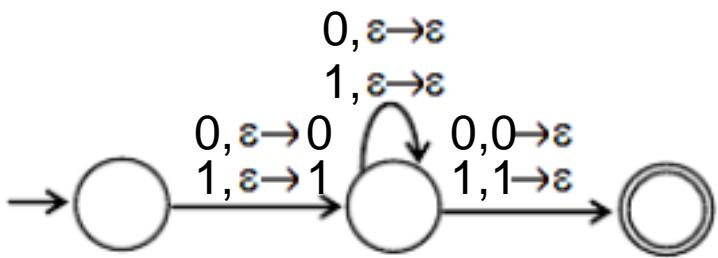


b. $\{w \mid w \text{ 以相同的符号开始和结束, } w \text{ 的长大于 } 1\}$

S 0A0|1A1

A 0A|1A| ϵ

读输入中的第一个符号并将其推入栈。继续读后面的符号，同时非确定性地猜想输入符号和栈符号是否相同，如果相同则转移到接受状态，如果此时输入结束，则接受这个输入，否则继续输入。

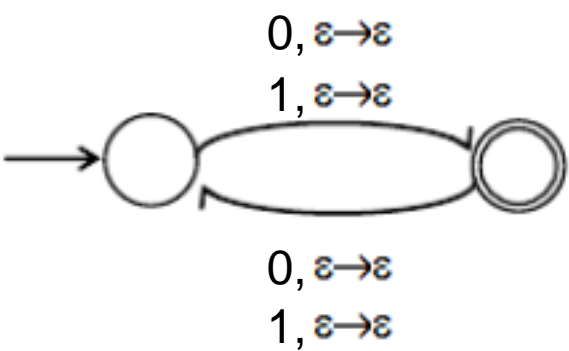


c. $\{w \mid w \text{ 的长度为奇数 }\}$

S ASA|0|1

A 0|1

读输入中的 1 个符号，转移到接受状态，再读一个符号，转移到非接受状态，如此循环。可见接受长度为奇数的字符串。

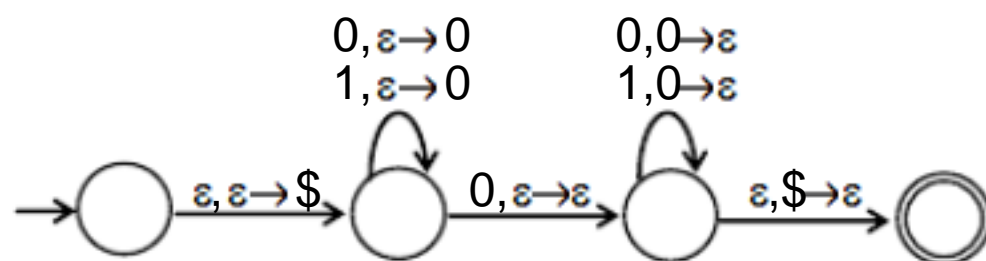


d. $\{w \mid w \text{ 的长度为奇数且正中间的符号为 } 0\}$

S ASA|0

A 0|1

读输入中的 1 个符号，推入 1 个 0。每当读到 0 时就非确定性地猜想已经到达字符串的中点，然后变成读输入中的 1 个符号，就把栈中的 0 弹出。当输入结束的同时栈被排空，则接受，否则拒绝。

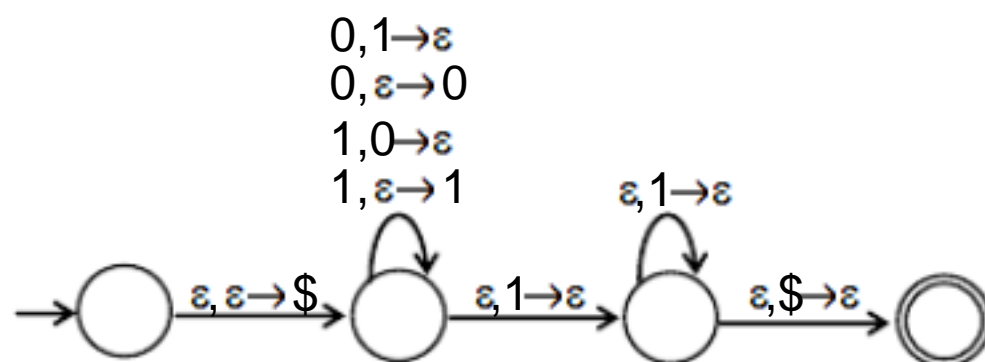


e. $\{w \mid w \text{ 中 } 1 \text{ 比 } 0 \text{ 多}\}$

答：S T1T | T1S // T1T 可以产生 1 比 0 多 1 个的所有字符串。

// T1S 可以产生 1 比 0 多 2 个以上的所有字符串。

T 0T1T | 1T0T | // T 可以产生 0 和 1 数目相等的所有字符串。



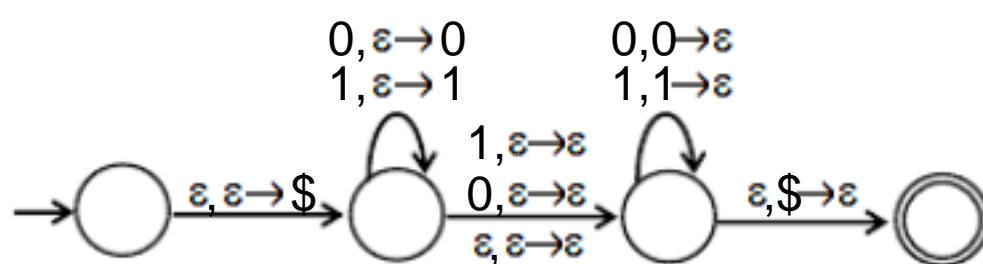
如果输入 0 时，栈顶元素是 1，则弹出 1；否则将 0 推入堆栈。

如果输入 1 时，栈顶元素是 0，则弹出 0；否则将 1 推入堆栈。

非确定地猜想栈顶元素是 1，且栈中都是 1，如果是，则接受；否则拒绝。

f. $\{w \mid w = w^R, \text{ 即 } w \text{ 是一个回文, 回文是顺读和倒读都一样的字符串}\}$

S 0S0 | 1S1 | 1 | 0 |

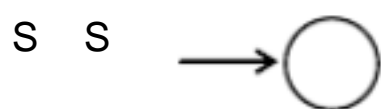


如果 W 是回文，那么它的中点有三种可能：

- 1) 字符个数是奇数，中点的字符是 1。
- 2) 字符个数是奇数，中点的字符是 0。
- 3) 字符个数是偶数，中点的字符是 ε。

开始时，把读到的字符推入栈中，在每一步非确定性地猜想已经到达字符串的中点。然后变成把读到的每一个字符弹出栈，检查在输入中和在栈顶读到的字符是否一样。如果它们总是一样的，并且当输入结束时栈是空的，则接受；否则拒绝。

g. 空集



3.6 给出产生下述语言的上下文无关文法：

a. 字母表 $\{a,b\}$ 上 a 的个数是 b 的个数的两倍的所有字符串组成的集合。

答： $S \rightarrow bSaSaS \mid aSbSaS \mid aSaSbS \mid \epsilon$

b. 语言 $\{a^n b^n \mid n \geq 0\}$ 的补集。见问题 3.25 中的 CFG:

答：分析问题： $\{a^n b^n \mid n \geq 0\}$ 语言的 CFG 为： $S \rightarrow aSb \mid \epsilon$ 。违反条件的情况可能有两种：

1. 一种是在连续的 a 中间插入了字符 b ，或者在连续的 b 中间插入了字符 a 。
2. a 和 b 的数目不相等。

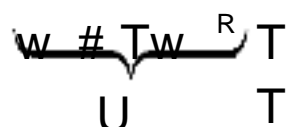
所以可以设计文法如下：

$S \rightarrow aSb \mid bT \mid Ta$ // 只能生成错序的或者 a 和 b 个数不相等的字符串。

$T \rightarrow aT \mid bT \mid \epsilon$ // 生成所有由 a, b 组成的字符串。

c. $\{w \# x \mid w, x \in \{0,1\}^* \text{ 且 } w^R \text{ 是 } x \text{ 的子串}\}$ 。

答： 分析问题：根据题义，语言 $w \# x$ 可以分解成为：



其中 T 是所有由 $0, 1$ 组成的字符串。

所以可以设计文法如下：

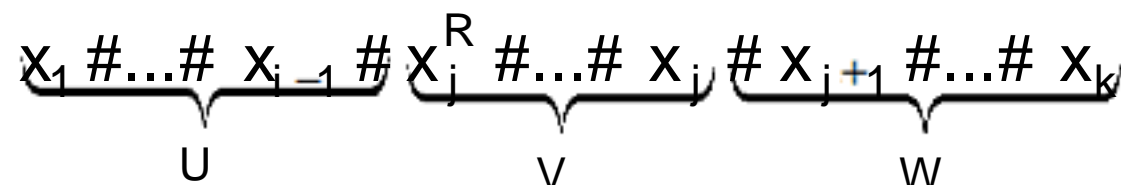
$S \rightarrow UT$

$U \rightarrow 0U0 \mid 1U1 \mid \#T$ // 生成 $w \# Tw^R$

$T \rightarrow 0T \mid 1T \mid \epsilon$ // 生成所有由 $0, 1$ 组成的字符串

d. $\{x_1 \# x_2 \# \dots \# x_k \mid k \geq 1, \text{ 每一个 } x_i \in \{a,b\}^*, \text{ 且存在 } i \text{ 和 } j \text{ 使得 } x_i = x_j^R\}$ 。

答： 分析问题：根据题义，语言 $x_1 \# x_2 \# \dots \# x_k$ 可以分解成为：



所以可以设计文法如下：

$S \rightarrow UVW$

$U \rightarrow A \# U \mid \epsilon$

$W \quad \#AW| \varepsilon$

$A \quad aA|bA| \varepsilon \quad // \text{ 生成所有由 } a, b \text{ 组成的字符串 } x_i$

$V \quad aVa|bVb|\#U$

3.7 略。

3.8 证明在 3.1 节开始部分给出的文法 G_2 中，字符串 **the girl touches the boy with the flower** 有两个不同的最左派生，叙述这句话的两个不同意思。

G_2 如下：

< 句子 > < 名词短语 > < 动词短语 >

< 名词短语 > < 复合名词 > | < 复合名词 > < 介词短语 >

< 动词短语 > < 复合动词 > | < 复合动词 > < 介词短语 >

< 介词短语 > < 介词 > < 复合名词 >

< 复合名词 > < 冠词 > < 名词 >

< 复合动词 > < 动词 > | < 动词 > < 名词短语 >

< 冠词 > a_|the_

< 名词 > boy_|girl_|flower_

< 动词 > touch_|likes_|Sees_

< 介词 > with_

答：

1. 第一种最左派生

< 句子 > \Rightarrow < 名词短语 > < 动词短语 > \Rightarrow < 复合名词 > < 动词短语 > \Rightarrow < 冠词 > < 名词 > < 动词短语 >
 \Rightarrow a_ < 名词 > < 动词短语 > \Rightarrow a_girl_ < 动词短语 > \Rightarrow a_girl_ < 复合动词 >
 \Rightarrow a_girl_ < 动词 > < 名词短语 > \Rightarrow a_girl_touches_ < 名词短语 >
 \Rightarrow a_girl_touches_ < 复合名词 > < 介词短语 > \Rightarrow a_girl_touches_ < 冠词 > < 名词 > < 介词短语 >
 \Rightarrow a_girl_touches_the_ < 名词 > < 介词名词 > \Rightarrow a_girl_touches_the_boy_ < 介词短语 >
 \Rightarrow a_girl_touches_the_boy_ < 介词 > < 复合名词 > \Rightarrow a_girl_touches_the_boy_with_ < 复合名词 >
 \Rightarrow a_girl_touches_the_boy_with_ < 冠词 > < 名词 > \Rightarrow a_girl_touches_the_boy_with_the_ < 名词 >
 \Rightarrow a_girl_touches_the_boy_with_the_flower

含义是 ：女孩碰这个带着花的男孩

2. 第二种最左派生

< 句子 > \Rightarrow < 名词短语 > < 动词短语 > \Rightarrow < 复合名词 > < 动词短语 > \Rightarrow < 冠词 > < 名词 > < 动词短语 >

$\Rightarrow a_<\text{名词}><\text{动词短语}>\Rightarrow a_girl_<\text{动词短语}>\Rightarrow a_girl_<\text{复合动词}><\text{介词短语}>$
 $\Rightarrow a_girl_<\text{动词}><\text{名词短语}><\text{介词短语}>\Rightarrow a_girl_touches_<\text{名词短语}><\text{介词短语}>$
 $\Rightarrow a_girl_touches_<\text{冠词}><\text{名词}><\text{介词短语}>\Rightarrow a_girl_touches_the_<\text{名词}><\text{介词短语}>$
 $\Rightarrow a_girl_touches_the_boy_<\text{介词短语}>\Rightarrow a_girl_touches_the_boy_<\text{介词}><\text{复合名词}>$
 $\Rightarrow a_girl_touches_the_boy_with_<\text{复合名词}>\Rightarrow a_girl_touches_the_boy_with_<\text{冠词}><\text{名词}>$
 $\Rightarrow a_girl_touches_the_boy_with_the_<\text{名词}>$
 $\Rightarrow a_girl_touches_the_boy_with_the_flower$

含义是： 女孩用花碰这个男孩

3.9 给出产生语言 $A=\{a^i b^j c^k \mid i,j,k \geq 0 \text{ 且或者 } i=j \text{ 或者 } j=k\}$ 的上下文无关文法。你给出的文法是歧义的吗？为什么？

解：下面是产生 A 的一个 CFG:

$S \rightarrow UV \mid AB$
 $U \rightarrow aUb \mid \epsilon \quad // \text{ 产生 } a^i b^i, i=j$
 $V \rightarrow cV \mid \epsilon \quad // \text{ 产生 } c^k$
 $A \rightarrow aA \mid \epsilon \quad // \text{ 产生 } a^i$
 $B \rightarrow bBc \mid \epsilon \quad // \text{ 产生 } b^j c^j, j=k$

这个 CFG 是歧义的，因为字符串 abc 有如下两种不同的最左派生：

$S \Rightarrow UV \Rightarrow aUbV \Rightarrow abV \Rightarrow abcV \Rightarrow abc$
 $S \Rightarrow AB \Rightarrow aAB \Rightarrow aB \Rightarrow abBc \Rightarrow abc$

3.10 给出识别 3.9 中语言 A 的下推自动机的非形式描述。

解：其非形式描述为：

此 PDA 有两个非确定性的分支：

1. 读输入中的符号 a ，把一个 a 推入栈。同时非确定性地读 b ，每读一个 b 从栈中弹出一个 a ，若栈中没有 a 则拒绝。当栈为空时进入接受状态。继续读输入符号，每读一个 c ，不改变栈内容和状态，若读入 a 或 b 则拒绝。
2. 读输入中的符号 a ，不改变栈内容。当读到 b 时，将一个 b 推入栈，同时非确定性地读 c ，每读一个 c 从栈中弹出一个 b ，若栈中没有 b 则拒绝。当栈为空时进入接受状态。

如果输入串结束时，一个非确定性分支在接受状态则接受，否则拒绝。

3.13 设有上下文无关文法 G :

$S \rightarrow TT|U$

$T \rightarrow 0T|T0| \#$

$U \rightarrow 0U00| \#$

- 用普通的语言描述 $L(G)$ 。
- 证明 $L(G)$ 不是正则的。

答：

$S \rightarrow TT|U$

$T \rightarrow 0T|T0| \#$ // 产生所有由 0 和一个 # 组成的字符串

$U \rightarrow 0U00| \#$ // 产生由 0 和 # 组成的字符串，且符号 # 右边的 0 的个数是左边的 0 的个数的二倍。

- $L(G) = \{0^i \# 0^j \# 0^k \mid i, j, k \geq 0\} \cup \{0^i \# 0^{2i} \mid i \geq 1\}$ 。
- 假设是正则的。令 p 是泵引理给出的泵长度，取 s 为字符串 $0^p \# 0^{2p}$ 。由于 s 是 $L(G)$ 的一个成员且长度大于 p ，泵引理保证 s 可以分成 3 段 $s=xyz$ ，使得对于任意的 $i \geq 0$ ， $xy^i z$ 在 $L(G)$ 中。
根据泵引理的条件 3： $|xy| \leq p$ ，则 y 一定只由 0 组成，从而字符串 $xyyx = 0^{p+1} \# 0^{2p}$ ，所以 $xyyx \notin L(G)$ 。矛盾，故 $L(G)$ 不是正则的。

3.14 用定理 3.6 中给出的过程，把下述 CFG 转换成等价的乔姆斯基范式文法。

$A \rightarrow BAB|B| \epsilon$

$B \rightarrow 00| \epsilon$

答：

- 添加新起始变元 S_0 ,

$S_0 \rightarrow A$

$A \rightarrow BAB|B| \epsilon$

$B \rightarrow 00| \epsilon$

- 删除 ϵ 规则 $B \rightarrow \epsilon$

$S_0 \rightarrow A$

$A \rightarrow BAB|AB|BA|A|B| \epsilon$

$B \rightarrow 00$

3) 删除 ε 规则 $A \rightarrow \varepsilon$,

$S_0 \rightarrow A | \varepsilon$ //乔姆斯基范式文法允许规则 $S \rightarrow \varepsilon$

$A \rightarrow BAB | AB | BA \blacksquare BB$

$B \rightarrow 00$

4) 删除单一规则 $A \rightarrow B$

$S_0 \rightarrow \blacksquare | \varepsilon$

$A \rightarrow BAB | AB | BA | 00 | BB$

$B \rightarrow 00$

5) 删除单一规则 $S_0 \rightarrow A$,

$S_0 \rightarrow BAB | AB | BA | 00 | BB | \varepsilon$

$A \rightarrow BAB | AB | BA | 00 | BB$

$B \rightarrow 00$

6) 添加新的变元和规则

$S_0 \rightarrow VB | AB | BA | UU | BB | \varepsilon$

$A \rightarrow VB | AB | BA | UU | BB$

$B \rightarrow UU$

$V \rightarrow BA$

$U \rightarrow 0$

3.19 证明：设 G 是一个 **Chomsky** 范式 **CFG**，则对任一长度 $n \geq 1$ 的字符串 $w \in L(G)$ ， w 的任何派生恰好有 $2n-1$ 步。

证明：对字符串长度 n 使用数学归纳法。

归纳基础：当 $n=1$ 时，对于 Chomsky 范式 CFG，长度为 1 的字符串必由 1 步派生，此时命题为真。

归纳步骤：假设命题对长度不超过 $k-1$ 的字符串成立，要证明命题对长度为 $k+1$ 的字符串成立。

对于长度为 $k+1$ 的字符串 $s=s_1s_2\ldots s_{k+1}$ ，存在 S_0 的一个直接派生 $S_0 \Rightarrow AB$ ，使得 A 产生子串 $s_1s_2\ldots s_p$ ， B 产生子串 $s_{p+1}s_{p+2}\ldots s_{k+1}$ ，其中 $1 \leq p \leq k+1$ 。由归纳假设， A 产生子串 $s_1s_2\ldots s_p$ 需要 $2p-1$ 步派生，而 B 产生子串 $s_{p+1}s_{p+2}\ldots s_{k+1}$ 需要 $2(k+1-p)-1$ 步派生。因此， S_0 产生字符串 s 需要 $1+(2p-1)+[2(k+1-p)-1] = 2k+1 = 2(k+1)-1$ 步派生。即当 $n=k+1$ 时命题亦真。

证毕。

第 4 章、丘奇 - 图灵论题习题解答 - 练习

4.1、此练习和 TM M_2 有关，例 4.4 给出了它的描述和状态图，它判定语言 $A = \{0^{2^n} \mid n \geq 0\}$ 。在下列每个输入串上，给出 M_2 所进入的格局序列：

- a. 0
- b. 00
- c. 000
- d. 000000

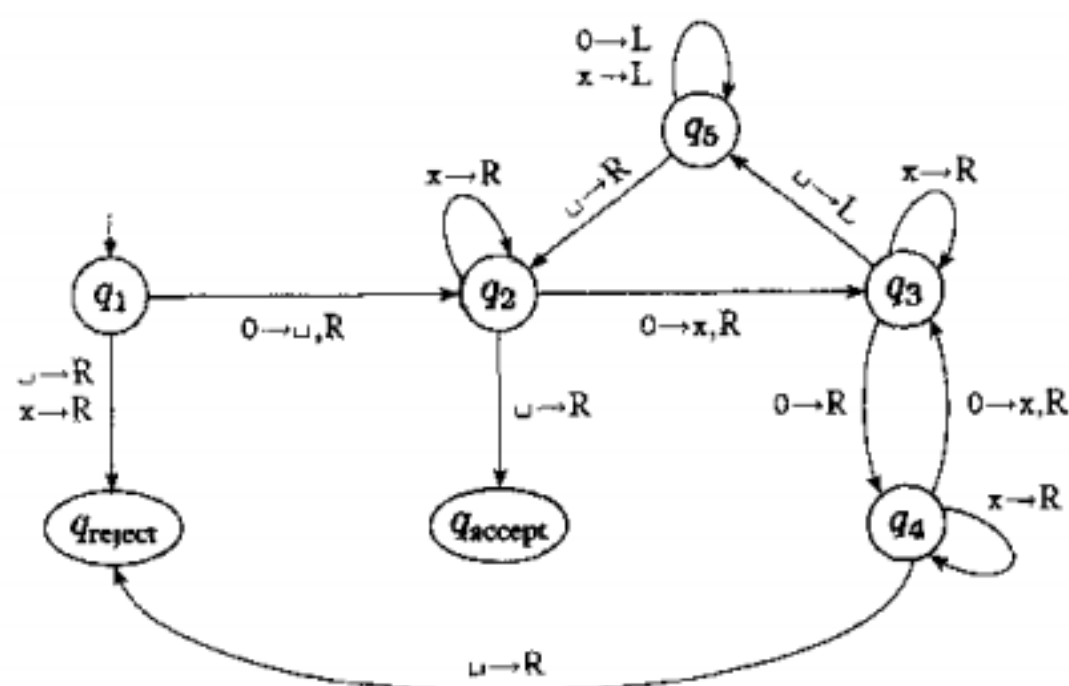


图 4-4 M_2 的状态图

答：

- a. $q_1 0$ $\sqcup q_2 \sqcup$ $\sqcup q_{\text{accept}}$
- b. $q_1 00$ $\sqcup q_2 0 \sqcup$ $\sqcup x q_3 \sqcup$ $\sqcup q_5 x \sqcup$ $q_5 \sqcup x \sqcup$ $\sqcup q_2 x \sqcup$
 $\sqcup x q_2 \sqcup$ $\sqcup x \sqcup q_{\text{accept}}$
- c. $q_1 000$ $\sqcup q_2 00 \sqcup$ $\sqcup x q_3 0 \sqcup$ $\sqcup x 0 q_4 \sqcup$ $\sqcup x 0 q_{\text{reject}}$
- d. $q_1 000000$ $\sqcup q_2 00000$ $\sqcup x q_3 0000$ $\sqcup x 0 q_4 000$ $\sqcup x 0 x q_3 00$
 $\sqcup x 0 x 0 q_4 0$ $\sqcup x 0 x 0 x q_3 \sqcup$ $\sqcup x 0 x 0 q_5 x$ $\sqcup x 0 x q_5 0 x$ $\sqcup x 0 q_5 x 0 x$
 $\sqcup x q_5 0 x 0 x$ $\sqcup q_5 x 0 x 0 x$ $q_5 \sqcup x 0 x 0 x$ $\sqcup q_2 x 0 x 0 x$ $\sqcup x q_2 0 x 0 x$
 $\sqcup x x q_3 x 0 x$ $\sqcup x x x q_3 0 x$ $\sqcup x x x 0 q_4 x$ $\sqcup x x x 0 x q_4 \sqcup$
 $\sqcup x x x 0 x \sqcup q_{\text{reject}}$

4.2、此练习和 TM M_1 有

关，例 4.5 给出了它的描

述和状态图，它判定语言

$B = \{w\#w \mid w \in \{0,1\}^*\}$ 。在下

列每个输入串上，给出

M_1 所进入的格局序列：

- 11
- 1#1
- 1##1
- 10#11
- 10#10

答：

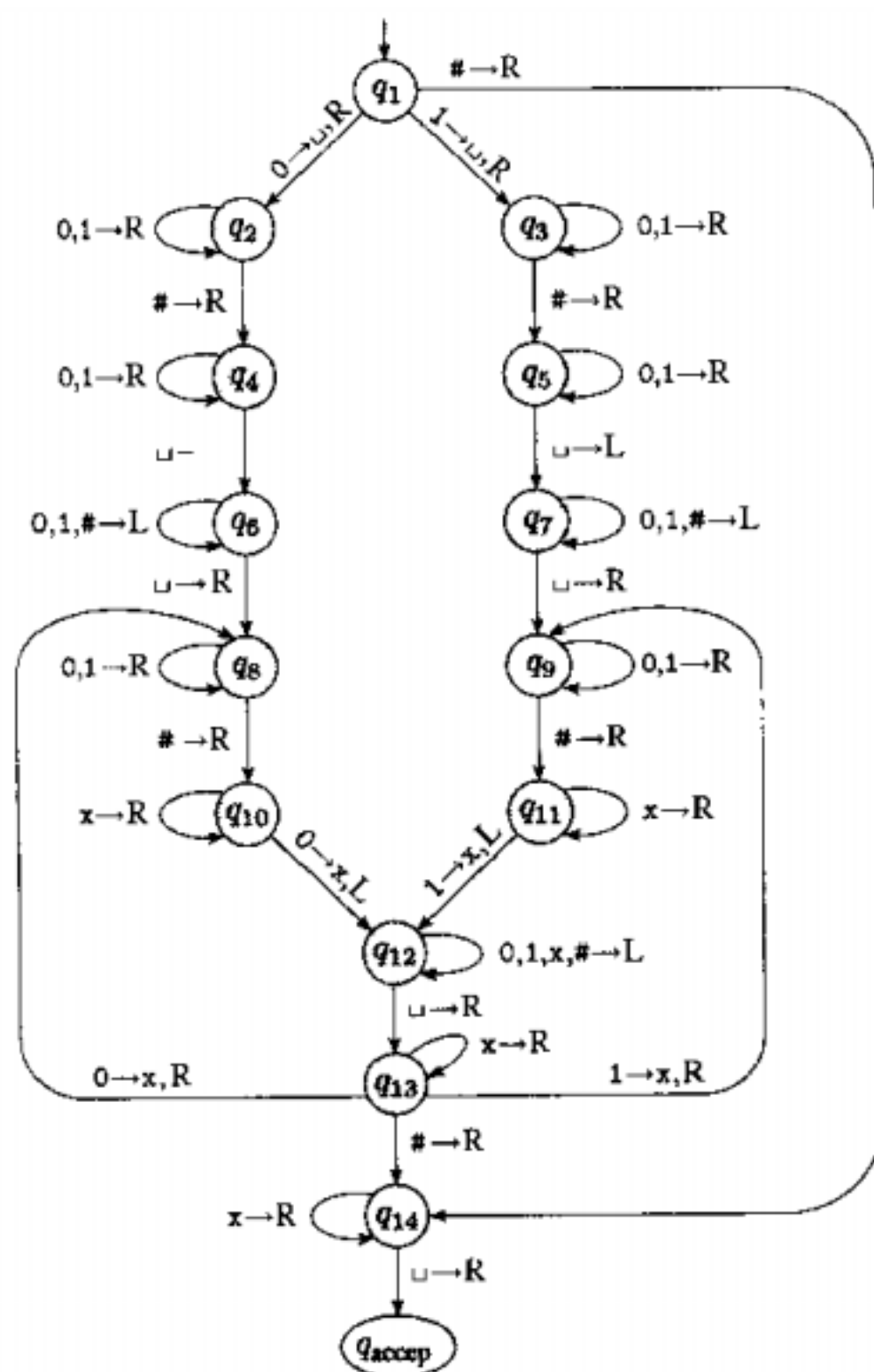


图 4-5 图灵机 M_1 的状态图

- $q_{11}1$ 、 $\sqcup q_31$ 、 $\sqcup 1q_3 \sqcup$ 、 $\sqcup 1q_{\text{reject}} \sqcup$
- $q_11\#1$ 、 $\sqcup q_3\#1$ 、 $\sqcup \#q_51$ 、 $\sqcup \#1q_5 \sqcup$ 、 $\sqcup \#q_71$ 、 $\sqcup q_7\#1$
 $q_7 \sqcup \#1$ 、 $\sqcup q_9\#1$ 、 $\sqcup \#q_{11}1$ 、 $\sqcup q_{12}\#x$ 、 $q_{12} \sqcup \#x$ 、 $\sqcup q_{13}\#x$
 $\sqcup \#q_{14}x$ 、 $\sqcup \#xq_{14} \sqcup$ 、 $\sqcup \#x \sqcup q_{\text{accept}}$
- $q_11##1$ 、 $\sqcup q_3##1$ 、 $\sqcup \#q_5\#1$ 、 $\sqcup \#q_{\text{reject}}\#1$
- $q_110\#11$ 、 $\sqcup q_30\#11$ 、 $\sqcup 0q_3\#11$ 、 $\sqcup 0\#q_511$ 、 $\sqcup 0\#1q_51$ 、 $\sqcup 0\#11q_5 \sqcup$
 $\sqcup 0\#1q_71$ 、 $\sqcup 0\#q_711$ 、 $\sqcup 0q_7\#11$ 、 $\sqcup q_70\#11$ 、 $q_7 \sqcup 0\#11$ 、 $\sqcup q_90\#11$
 $\sqcup 0q_9\#11$ 、 $\sqcup 0\#q_{11}11$ 、 $\sqcup 0q_{12}\#x1$ 、 $\sqcup q_{12}0\#x1$ 、 $q_{12} \sqcup 0\#x1$ 、 $\sqcup q_{13}0\#x1$
 $\sqcup xq_8\#x1$ 、 $\sqcup x\#q_{10}x1$ 、 $\sqcup x\#xq_{10}1$ 、 $\sqcup x\#xq_{\text{reject}}1$
- $q_110\#10$ 、 $\sqcup q_30\#10$ 、 $\sqcup 0q_3\#10$ 、 $\sqcup 0\#q_510$ 、 $\sqcup 0\#1q_50$ 、 $\sqcup 0\#10q_5 \sqcup$
 $\sqcup 0\#1q_70$ 、 $\sqcup 0\#q_710$ 、 $\sqcup 0q_7\#10$ 、 $\sqcup q_70\#10$ 、 $q_7 \sqcup 0\#10$ 、 $\sqcup q_90\#10$

$\sqcup 0q_9\#10$ 、 $\sqcup 0\#q_{11}10$ 、 $\sqcup 0q_{12}\#x0$ 、 $\sqcup q_{12}0\#x0$ 、 $q_{12}\sqcup 0\#x0$ 、 $\sqcup q_{13}0\#x0$
 $\sqcup xq_8\#x0$ 、 $\sqcup x\#q_{10}x0$ 、 $\sqcup x\#xq_{10}0$ 、 $\sqcup x\#q_{12}xx$ 、 $\sqcup xq_{12}\#xx$ 、 $\sqcup q_{12}x\#xx$
 $q_{12}\sqcup x\#xx$ 、 $\sqcup q_{13}x\#xx$ 、 $\sqcup xq_{13}\#xx$ 、 $\sqcup x\#q_{14}xx$ 、 $\sqcup x\#xq_{14}x$ 、 $\sqcup x\#xxq_{14}\sqcup$
 $\sqcup x\#xx\sqcup q_{\text{accept}}$

4.3 修改定理 4.10 以得到推论 4.12 的证明，即证明一个语言是可判定的当且仅当有非确定型 **TM** 判定它。（可以假设关于树的下列定理成立：如果一个树中的每个结点只有有限多个子结点，且此树的每一个分枝只有有限多个结点，则此树本身只有有限个结点）。

证明：

定理 4.10 为：每个非确定型图灵机都有一个与之等价的确定型图灵机。

修改定理 4.10 为：每个非确定性判定器都有与之等价的确定型判定器。

现在设 N 是一个非确定性判定器， N 对所有输入，所有分枝都停机。构造一个与之等价的确定型判定器 D 。 D 有三个带：一个输入带，一个模拟带，一个地址带。 D 按“宽度优先”策略搜索 N 的不确定计算分枝树。

$M =$ “输入 w ，

- 1) 开始时，输入带包含输入 w ，模拟带和地址带都是空的。
- 2) 把输入带的内容复制到模拟带上。
- 3) 在模拟带上模拟 N 在输入 w 上的非确定性计算树的某个分枝。
 - a) 在 N 的每一步动作之前，查询地址带上的下一个数字，以决定在 N 的转移函数所允许的选择中作何选择。
 - b) 如果地址带上下一个数字是空白，则放弃这个分枝，转到第四步。
 - c) 如果这个非确定性的选择是无效的，则放弃这个分枝，转到第四步。
 - d) 如果遇到拒绝格局，则放弃这个分枝，转到第四步。
 - e) 如果遇到接受格局，则接受这个输入。
- 4) 按照字典顺序取下一个地址串。根据假设非确定性计算树只有有限的结点，所以地址串有一个上界 A_{\max} 。
 - a) 如果下一个地址串已经超过 A_{\max} ，则说明已经搜索了所有分枝，那么拒绝这个输入。
 - b) 否则用下一个地址串来替代原有的串。转到第二步，以模拟 N 的计算的下一个分枝。”

如果有一个非确定性判定器判定语言，根据上述证明，必有与之等价的确定型判定器判定该语言，则该语言是可判定的。

如果一个语言是可判定的，则有一个确定性判定器判定该语言。确定性判定器自然是非确定型

判定器，所以必有非确定型判定器判定它。

4.4 给出枚举器的形式定义。可将其看作一种双带图灵机，用它的第二个带子作为打印机。包括它所枚举的语言的定义。

解：枚举器 $E=(Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$ ，其中转移函数 δ 为：

$$\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\} \times \Gamma \times \{R\}$$

$$\delta(q, a)=(r, b, s_1, c, R)$$

表示若 E 处于状态 q ，且在工作带上读到 a ，则状态转移到 r ，当前格改写为 b 并按 s_1 所指示的方向移动读写头，打印带上写下 c 并向右移动读写头。

另外 E 的起始格局只能是 $q_0 \circ$

$$L(E)=\{w|w \text{ 是枚举器 } E \text{ 打印出的串}\}$$

4.5 检查图灵机的形式定义，回答下列问题并解释你的推测：

- a. 图灵机能在它的带子上写下空白符吗？
- b. 带字母表 Γ 和输入字母表 Σ 能相同吗？
- c. 图灵机的读写头能在连续的两步中处于同一个位置吗？
- d. 图灵机能只包含一个状态吗？

解：

- a. 能。因为空白符属于带字母表 Γ ；
- b. 不能。因为空白符不属于输入字母表 Σ ；
- c. 能。当读写头处于左端点时，如果转移是向左转移，因为不准机器从带的左端点移出，所以下一个格局读写头仍在左端点。
- d. 不能。因为 $q_{\text{accept}} \neq q_{\text{reject}}$ ，至少应有两个状态。

4.6 定理 4.13 证明了一个语言是图灵可识别的当且仅当有枚举器枚举它。为什么不能用下列更简单的算法作为证明？像以前一样， s_1, s_2, \dots 是 Σ^* 中所有的串。

$E=$ “忽略输入，

- 1) 对于 $i=1, 2, 3, \dots$ 重复下列步骤。
- 2) 在 s_i 上运行 M 。
- 3) 如果它接受，则打印出 s_i 。”

答：因为 M 不一定是判定器，可能会在运行某个 s_i 时不停机，则 $L(M)$ 中按字典序的序数大于 s_i 的

字符串不可能被枚举出来。

这个思想和定理 4.10 中的“宽度优先”的策略类似。

4.7 下面描述的不是一个合法的图灵机，解释为什么。

M_{bad} = “输入是变元 x_1, x_2, \dots, x_k 上的一个多项式 p :

- 1) 让 x_1, x_2, \dots, x_k 取所有可能的整数值。
- 2) 对所有这些值求 p 的值。
- 3) 只要某个取值使得 p 为 0，则接受，否则拒绝”

答：因为图灵机的一个本质要求是一步只能做有限的工作。参考希尔伯特问题一节中图灵机 M_1 的定义。

4.8 下面的语言都是字母表 $\{0, 1\}$ 上的语言，以实现水平的描述给出判定这些语言的图灵机：

- a. $\{w | w \text{ 包含相同个数的 } 0 \text{ 和 } 1\}$
- b. $\{w | w \text{ 所包含的 } 0 \text{ 的个数是 } 1 \text{ 的个数的二倍}\}$
- c. $\{w | w \text{ 所包含的 } 0 \text{ 的个数不是 } 1 \text{ 的个数的二倍}\}$

答：构造具有 3 条带的图灵机。第一条带子是输入带，第二条带子存放输入串所包含的 0，第三条带子存放输入串所包含的 1。

对于问题 a.

M_1 = “对于输入字符串 w :

- 1) 扫描输入，读到 0 就把 0 写入第 2 条带，读到 1 就把 1 写入第 3 条带，直到读到空白符为止。
- 2) 第 2、3 条带的读写头回到最左端。
- 3) 读第 2、3 条带，每次读一个字符，如果同时读出空白符，则接收，否则拒绝。”

对于问题 b:

M_2 = “对于输入字符串 w :

- 1) 扫描输入，读到 0 就把 0 写入第 2 条带，读到 1 就把 1 写入第 3 条带，直到读到空白符为止。
- 2) 第 2、3 条带的读写头回到最左端。
- 3) 每次读带 3 的一个字符就读带 2 的两个字符。
- 4) 如果从带 3 上读出的字符是 1，从带 2 上读出的两个字符都是 0，转第 3 步。否则，拒绝。
如果从带 3 上读出的字符是空白，从带 2 上读出的第一个字符也是空白，就接收，否则拒绝。”

对于问题 c:

$M_3 =$ “对于输入字符串 w :

- 1) 运行 M_2 。
- 2) 如果 M_2 拒绝, 则接受; 如果 M_2 接受, 则拒绝。”

4.18 设多项式 $c_1x^n + c_2x^{n-1} + \dots + c_nx + c_{n+1}$ 有根 $x=x_0$, c_{\max} 是 c_i 的最大绝对值。证明

$$|x_0| < (n+1) \frac{c_{\max}}{|c_1|}$$

证明:

- 1) 当 $|x_0| \leq 1$ 时, 由条件 $c_{\max} \geq |c_1|$ 知不等式显然成立。
- 3) 当 $|x_0| > 1$ 时, 由

$$\begin{aligned} c_1x_0^n + c_2x_0^{n-1} + \dots + c_nx_0 + c_{n+1} &= 0 \\ c_1x_0^n &= -(c_2x_0^{n-1} + \dots + c_nx_0 + c_{n+1}) \\ c_1x_0 &= -\frac{(c_2x_0^{n-1} + \dots + c_nx_0 + c_{n+1})}{x_0^{n-1}} \\ c_1x &= -(c_2 + \dots + c_n \frac{1}{x_0^{n-2}} + c_{n+1} \frac{1}{x_0^{n-1}}) \end{aligned}$$

取绝对值

$$\begin{aligned} |c_1||x_0| &= |c_2 + \dots + c_n \frac{1}{x_0^{n-2}} + c_{n+1} \frac{1}{x_0^{n-1}}| \\ &\leq |c_2| + \dots + |c_n| \frac{1}{|x_0|^{n-2}} + |c_{n+1}| \frac{1}{|x_0|^{n-1}} \\ &\leq |c_2| + \dots + |c_n| + |c_{n+1}| \end{aligned}$$

最后得到

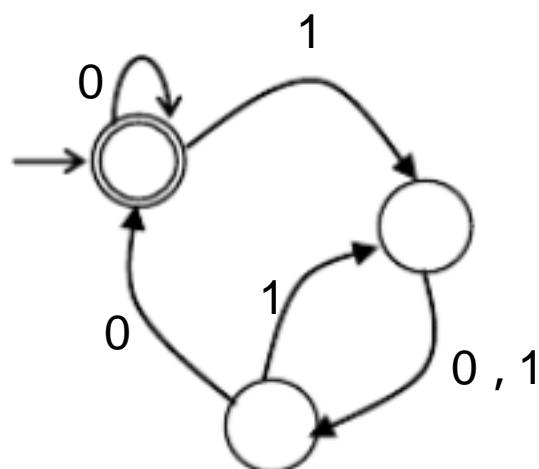
$$|c_1||x_0| \leq nc_{\max} < (n+1)c_{\max}$$

$$|x_0| < (n+1) \frac{c_{\max}}{|c_1|}$$

第 5 章、可判定性习题解答

练习

5.1 对于下图所示的 DFA M ，回答下列问题，并说明理由。



- a. $\langle M, 0100 \rangle \in A_{DFA}$? 是， M 接受 0100。
- b. $\langle M, 011 \rangle \in A_{DFA}$? 否， M 不接受 011。
- c. $\langle M \rangle \in A_{DFA}$? 否，不是 A_{DFA} 的有效输入编码。
- d. $\langle M, 0100 \rangle \in A_{REX}$? 是，DFA 和正则表达式等价。
- e. $\langle M \rangle \in E_{DFA}$? 否， M 接受串 0100。
- f. $\langle M, M \rangle \in EQ_{DFA}$? 是， $L(M) = L(M)$

5.2 考虑 DFA 和正则表达式是否等价的问题。将这个问题表示为一个语言并证明它可判定。

解：设 $EQ_{D-R} = \{ \langle A, B \rangle \mid A \text{ 是 DFA, } B \text{ 是正则表达式, } L(A) = L(B) \}$ 。

构造如下 TM F ，

$F =$ “对于输入 $\langle A, B \rangle$ ， A 是 DFA， B 是正则表达式，

- 1) 将正则表达式 B 转化为等价的 DFA C 。
- 2) 判定 $\langle A, C \rangle \in EQ_{DFA}$ (EQ_{DFA} 可判定)。
- 3) 若接受，则接受；若拒绝，则拒绝。”

F 判定 EQ_{D-R} 。

5.3 设 $ALL_{DFA} = \{ \langle A \rangle \mid A \text{ 是一个识别 } \Sigma^* \text{ 的 DFA} \}$ 。证明 ALL_{DFA} 可判定。

证明：设计一个判定 ALL_{DFA} 的 TM M 即可。

$M =$ “对于输入 $\langle A \rangle$ ，其中 A 是一个 DFA:

- 1) 构造 DFA B ,使得 $L(B)=\Sigma^*$ 。
- 2) 判定 $\langle A , B \rangle \in EQ_{DFA}$ (EQ_{DFA} 可判定)。
- 3) 若接受，则接受；若拒绝，则拒绝。 ”

5.4 $A_{CFG}=\{\langle G \rangle \mid G \text{ 是一个派生 } \varepsilon \text{ 的 CFG}\}$ 。证明 A_{CFG} 可判定。

证明：设计一个判定 A_{CFG} 的 TM M 即可。

M= “ 输入 $\langle G \rangle$, G 是一个 CFG ,

- 1) 判定 $\langle G , \quad \rangle \in A_{CFG}$ (A_{CFG} 可判定)。
- 2) 若接受，则接受；若拒绝，则拒绝。 ”

5.5 设 $INFINITE_{DFA}=\{\langle A \rangle \mid A \text{ 是一个 DFA , 且 } L(A) \text{ 是一个无限语言 } \}$ 。证明 $INFINITE_{DFA}$ 是可判定的。

证明：设计一个判定 $INFINITE_{DFA}$ 的 TM M 即可。

M= “ 对于输入 $\langle A \rangle$, 其中 A 是一个 DFA :

- 1) 按照引理 2.32 证明中的构造方法 , 把 DFA A 转换成等价的正则表达式。
- 2) 扫描正则表达式 , 如果包含星号运算符 $*$, 则接受 ; 否则拒绝。 ”。

5.6 设 X 是集合 {1 , 2 , 3 , 4 , 5} , Y 是集合 {6 , 7 , 8 , 9 , 10}。以表 5-4 描述函数

$f : X \rightarrow Y , g : X \rightarrow Y$ 。

N	f(n)	g(n)
1	6	10
2	7	9
3	6	8
4	7	7
5	6	6

- a. f 不是一对一的。因为 $1 \neq 3$, 但是 $f(1) = f(3) = 6$ 。 g 是一对一的。
- b. f 不是到上的。因为 $8 \notin Y$, 不存在 $a \in X$, 使得 $f(a) = 8$ 。 g 是到上的。
- c. f 不是对应的。 g 是对应的。

5.7 设 B 是 $\{0, 1\}$ 上所有无限序列的集合。用对角化方法证明 B 是不可数的。

证明：为证明 B 是不可数的，必须证明在 B 和 N 之间不存在对应。下面用反证法证之。假设在 B 和 N 之间存在对应 f ，现在的任务是证明它没有应有的性质。因为它是一个对应，必须能将 N 的所有元素与 B 的所有元素进行配对。如果能找到 B 中的一个 x ，它和 N 中的任何元素都不能配对，则找到了矛盾。

为此，实际构造出这样一个 x 。方法如下：在选择它的每一位数字时，都使得： x 不同于某个无限序列，且此无限序列已与 N 中的一个元素配对。这样就能保证 x 不同于任何已配对的无限序列。

用一个例子来说明这个思路。假设对应 f 存在，且设 $f(1) = 010101 \dots$ ， $f(2) = 101010 \dots$ ， $f(3) = \dots$ 等等。则 f 将 1 和 $010101 \dots$ 配对，将 2 和 $101010 \dots$ 配对，依此类推。

要保证对每个 n 都有 $x \neq f(n)$ 。为保证 $x \neq f(1)$ ，只要保证 x 的第一位数字不同于 $f(1) = 010101 \dots$ 的第一位数字，即不是数字 0，令它为 1。为保证 $x \neq f(2)$ ，只要保证 x 的第二位数字不同于 $f(2) = 101010 \dots$ 的第一位数字，即不是数字 0，令它为 1。以这种方法继续下去，就能够得到 x 的所有数字。不难知道，对任意 n ， x 都不是 $f(n)$ ，因为 x 与 $f(n)$ 在第 n 位上不同。

5.8 设 $T = \{(i, j, k) | i, j, k \in N\}$ 。证明 T 是可数的。

证明：先列出 T 的所有元素；然后将此序列中的第一个元素与 N 中的 1 配对，将第二个元素与 2 配对，依此类推。

设 $i = j = k = 1$ ， T 的元素为 1 个

$(1, 1, 1)$

设 $i = j = k = 2$ ， T 的元素为 8 个

$(1, 1, 1) \setminus (1, 1, 2) \setminus (1, 2, 1) \setminus (1, 2, 2)$

$(2, 1, 1) \setminus (2, 1, 2) \setminus (2, 2, 1) \setminus (2, 2, 2)$

设 $i = j = k = 3$ ， T 的元素为 27 个

.....。

按此顺序排列元素。第一种情况只包含一个元素 $(1, 1, 1)$ ，第二种情况包含 8 个元素，忽略重复的元素。所以此序列的前八个元素是 $(1, 1, 1) \setminus (1, 1, 2) \setminus (1, 2, 1) \setminus (1, 2, 2) \setminus (2, 1, 1) \setminus (2, 1, 2) \setminus (2, 2, 1) \setminus (2, 2, 2)$ 。以这种方法继续下去，就得到 T 的所有元素的一个序列。

5.9 回忆 **5.10** 定义中定义“集合有相同规模”的方法。证明“有相同规模”是一个等价关系。

证明：根据定义：称 A 和 B 有相同的规模，如果存在一对一且到上的函数 $f: A \rightarrow B$ 。既是一对一又

是到上的函数称为对应。在对应中， A 的每个元素映射到 B 的唯一一个元素，且 B 的每个元素都有 A 的唯一一个元素映射到它。

设“有相同规模”为二元关系 R 。

- 1) R 是自反的，即对每一个集合 A ，显然 A 和 A 有相同的规模，即 ARA 。
- 2) R 是对称的，即对每一个集合 A 和 B ，如果 ARB ，即 A 和 B 有相同的规模，显然 B 和 A 也有相同的规模，即 BRA 。
- 3) R 是传递的，即对每一个集合 A ， B 和 C ，如果 ARB 且 BRC ，即 A 和 B 有相同的规模， B 和 C 有相同的规模，显然 A 和 C 也有相同的规模，即 ARC 。

9. 正则语言类在连结运算下封闭。

证明：设 $N_1=(Q_1, \delta_1, q_1, F_1)$ 识别 A_1 ， $N_2=(Q_2, \delta_2, q_2, F_2)$ 识别 A_2 ，构造识别 $A_1 \cdot A_2$ 的 N ，这里 $N=(Q, \delta, q_1, F_2)$ 。

- 1) $Q = Q_1 \cup Q_2$ 。
- 2) N 的起始状态是 N_1 的起始状态 q_1 。
- 3) N 的接受状态集是 N_2 的接受状态集 F_2 。
- 4) 定义 δ 如下：对每一个 $q \in Q$ 和每一个 $a \in \Sigma$ ，令

$$\delta(q, a) = \begin{cases} \delta_1(q, a), & \text{若 } q \in Q_1 \text{ 且 } q \notin F_1 \\ \delta_1(q, a), & \text{若 } q \in F_1 \text{ 且 } a \neq \varepsilon \\ \delta_1(q, a) \cup \{q_2\}, & \text{若 } q \in F_1 \text{ 且 } a = \varepsilon \\ \delta_2(q, a), & \text{若 } q \in Q_2 \end{cases}$$