# 第五讲

## 3.9 (part 2) 右线性语言的封闭性

## 3.10　双向和有输出的有限自动机

# 复习

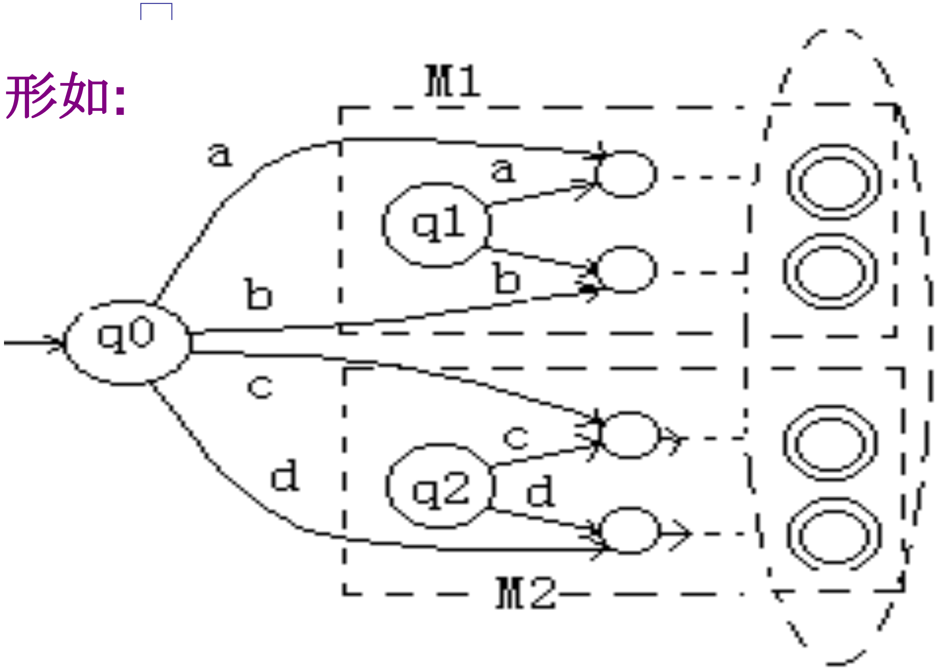
- $L=\{a^nb|n>=1\}$ 是几型语言？识别该语言的文法及其对应自动机
- $L=\{a^nb^n|n>=1\}$是几型语言？识别该语言的文法，判断该语言是否是正则语言

# 右线性语言的封闭性

上节从文法产生的角度证明了右线性语言及其并，积，闭包是正则集；本节用有限自动机接受的语言来证明。(书P103~)

1. 设$L_1$和$L_2$是右线性语言，证明$L_1 \cup L_2$为右线性语言

M形如:




构造NFA $M=(Q, T, \delta, q_0, F)$

$Q=Q_1 \cup Q_2 \cup \{q_0\}$

$q_0$是新的起始状态

$$F=\begin{cases} F_1 \cup F_2 & 当\varepsilon \notin L_1，\varepsilon \notin L_2 \\ F_1 \cup F_2 \cup \{q_0\} & 否则 \end{cases}$$

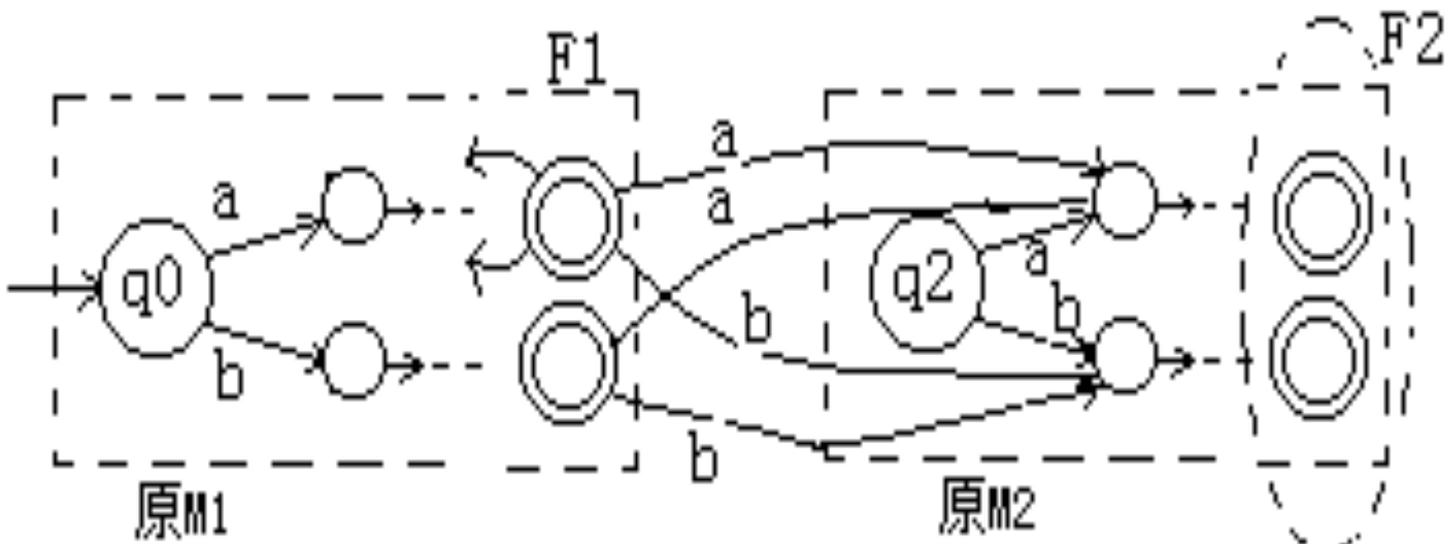
# 右线性语言的封闭性

**2. 设 $L_1$ 和 $L_2$ 是右线性语言，证明 $L_1L_2$ 为右线性语言 (书P104~)**

构造NFA M＝(Q，T，δ，$q_0$，F)，　其中 $Q=Q_1\cup Q_2$　　$q_0=q_1$

$$F=\begin{cases} F_2 & \text{当} q_2\notin F_2 \\ F_1\cup F_2 & \text{当} q_2\in F_2 \end{cases}$$

**M形如:**



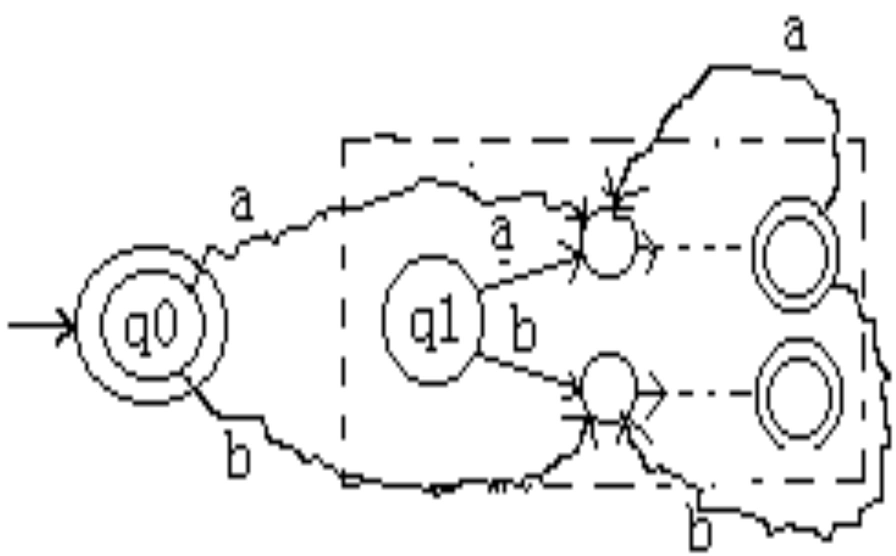
# 右线性语言的封闭性

3. 设$L_1$是右线性语言，证明$L_1$*是右线性语言(书P106~)

构造NFA M＝(Q，T，δ，$q_0$，F)，  $Q = Q_1 \cup \{q_0\}$

$q_0$是新的起始状态，  $F = F_1 \cup \{q_0\}$

$L_1^*$形如




实质上相当于用一个ε转换到q1

# 右线性语言的封闭性

4. 设 $L_1$ 是右线性语言，证明 $\overline{L_1}$ 是右线性语言

证明：构造接受 $\overline{L_1}$ 的 $M=(Q, T, \delta, q_0, F)$

其中 $Q = Q_1 \cup \{\gamma\}$　　$\gamma$ 是一个新状态

$T \supseteq T_1$　　$q_0 = q_1$　　$F = (Q_1 - F_1) \cup \{\gamma\}$

(即将 $M_1$ 的终止状态变为 $M$ 的非终止状态)

$\delta$ 定义为：当 $a \in T_1$，则 $\delta(q, a) = \delta_1(q, a)$

— 保留原有函数

当 $a \in T - T_1$，则 $\delta(q, a) = \gamma$

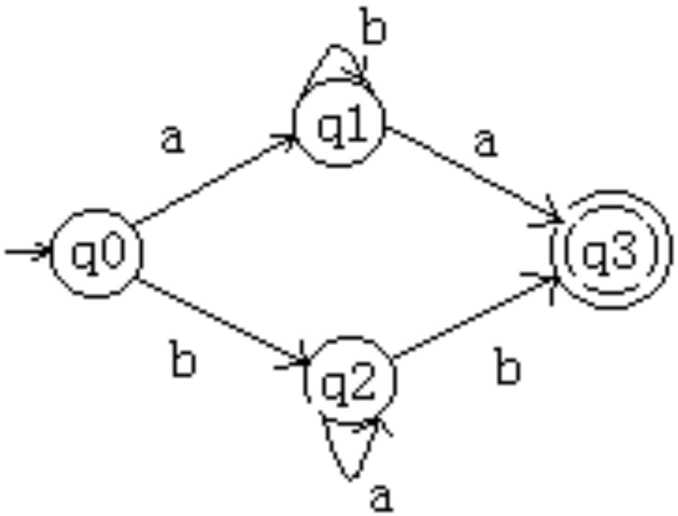
— 遇到原来没有的字符全转至 $\gamma$.

对任意 $a \in T$，$\delta(\gamma, a) = \gamma$

# 右线性语言的封闭性

例：（书P76）

对下图的$M_1$，求$\overline{L(M_1)}$

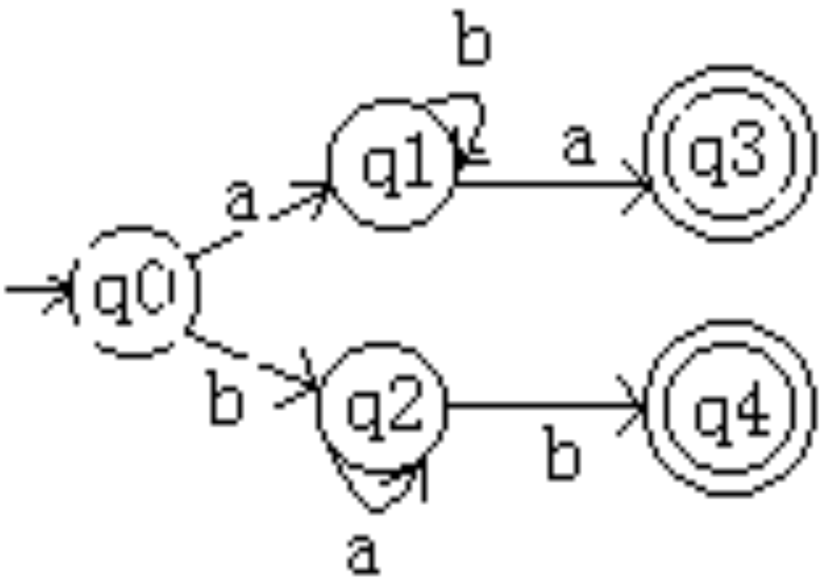


# 右线性语言的封闭性

例:

设DFA $M_1 = (\{q_0, q_4\}, \{a, b\}, \delta_1, q_0, \{q_3, q_4\})$

对 $T = \{a, b, c\}$，求 $\overline{L(M_1)}$



# 右线性语言的封闭性

**5. 证明$L_1 \cap L_2$是封闭的**

证明：

$$\because L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$$

$\therefore$ 得证

**6. 证明右线性语言对于置换是封闭的.**

**(略 -- 自学)**

# 补充：构造自动机的“交”

**通过构造证明，说明正则语言在交运算下封闭。**

设 两个DFA $M_1=(Q1, T, \delta_1, q_1, F_1)$

$$M_2=(Q2, T, \delta_2, q_2, F_2)$$

构造新的 DFA M,

$M=(Q_1 \times Q_2, T, \delta, [q1, q2], F_1 \times F_2)$

**对一切$p1 \in Q_1$，$p_2 \in Q_2$，$a \in T$，**

$\delta([p_1, p_2], a) = [\delta_1(p_1, a), \delta_2(p_2, a)]$

则$L(M) = L(M_1) \cap L(M_2)$

# 有关正则语言的几个判定性质

- 判定正则语言是否为空
- 判定正则语言中是否包含特定的字符串
- 判定两个正则语言是否相等

# 判定正则语言是否为空

- 以 DFA 表示正则语言
  - 判定算法 测试从初态是否可达某一终态. 先求所有可达状态的集合，若其中包含终态，则该正规语言非空，否则为空语言。

    可由如下步骤递归地计算可达状态集合:

    基础: 初态是可达的;

    归纳: 设状态 $q$ 是可达的，若对于某个输入符号或 $\varepsilon$, $q$ 可转移到 $p$，则 $p$ 也是可达的;

  - 算法复杂度 设有限自动机的状态数目为 $n$，上述判定算法的复杂度为 $O(n^2)$.

# 判定正则语言中是否包含特定的字符串

- 以 DFA 表示正则语言
  - 判定算法 从初态开始，处理输入字符串 w，如果可以结束于某一终态，则该正规语言中包含 w，否则不包含 w。
  - 算法复杂度 设输入字符串 w 的长度 $|w|=n$，上述判定算法的复杂度为 $O(n)$.
- 以 NFA （或ε–NFA）表示正则语言 可以将其转化为等价的 DFA，然后执行上述过程；也可以直接模拟其处理字符串的过程，判定算法的复杂度为 $O(n2^s)$，其中 n 为字符串的长度，s 为 NFA （或ε–NFA）的状态数目.
- 以正则表达式表示正则语言 将其转化为等价的 ε–NFA，然后执行上述过程.

# 判定两个正则语言是否相等

- **判定算法** 可由采取如下步骤：
  1. 先将两个正则语言的表达形式都转化为DFA，问题转化为两个DFA是否是等价的；
  2. 适当重命名，使两个DFA没有重名的状态；
  3. 将两个DFA相并，构造一个新的DFA，原来的终态仍是终态，转移边不发生任何变化；
  4. 对新构造的DFA运用填表算法，如果原来DFA的两个初态不可区别，则这两个正则语言相等，否则不相等.

- **算法复杂度** 以上算法的复杂度即填表算法的复杂度，其上限为$O(n^4)$；可以适当设计填表算法的数据结构，使其复杂度降为 $O(n^2)$.

# 双向有限自动机 (2DFA)

定义:

- 读入一个字符之后，读头既可以左移一格，也可以右移一格，或者不移动的有限自动机，为双向有限自动机.
- 确定的双向有限自动机: 每读入一字符，必须向左或右移动，不考虑不移动的情况.

# 2DFA的形式定义

2DFA M＝(Q，T，δ，$q_0$，F)

δ是从Q×T→Q×{L，R}的映射.

即 δ(q，a)=(p，R) 或 δ(q，a)=(p，L)

（在状态q，读入a，进入状态p，读头向右(向左)移一格）

用格局描述:

设有$\omega_1 q \omega_2$

$\omega_1$ --- 已输入串　　q --- 当前状态　　$\omega_2$ --- 待输入串

δ(q，$a_{m+1}$)=(p，R)的格局表示:

$$a_1 a_2 \ldots a_m \, q \, a_{m+1} \ldots a_n \vdash a_1 a_2 \ldots a_{m+1} \, p \, a_{m+2} \ldots a_n$$

δ(q，$a_{m+1}$)=(p，L)的格局表示:

$$a_1 a_2 \ldots a_m \, q \, a_{m+1} \ldots a_n \vdash a_1 a_2 \ldots a_{m-1} \, p \, a_m \, a_{m+1} \ldots a_n$$

# 2DFA

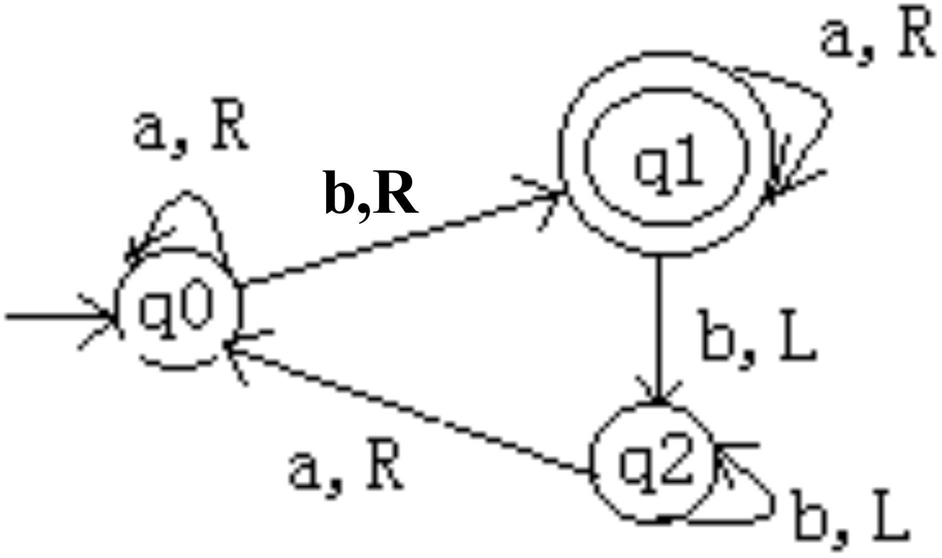
**2DFA接受的字符串集合是:**

**$L(M)=\{\omega \mid q_0\omega \vdash^* \omega q,\ q \in F\}$**

**例: 书P78 例1.**

**其状态图为**

**写出babaa的识别过程**



# 有输出的有限自动机

有输出的有限自动机是有限自动机的一个类型. 这类自动机在有字符输入时， 不仅存在状态转换， 同时引起字符输出.

根据输入字符， 自动机状态， 输出字符三者之间关系， 可有两类有输出的自动机:

- 米兰机(**Mealy**): 输出字符与输入字符及状态有关.
- 摩尔机(**Moore**): 输出字符仅与状态有关.

最大优点: 节省状态!

# 米兰机

米兰机形式定义: $M=(Q, T, R, \delta, g, q_0)$

其中 Q 有限状态集合

T 有限输入字母表

R 有限输出字母表

$\delta: Q\times T\rightarrow Q$ 转换函数

$g: Q\times T\rightarrow R$ 输出函数

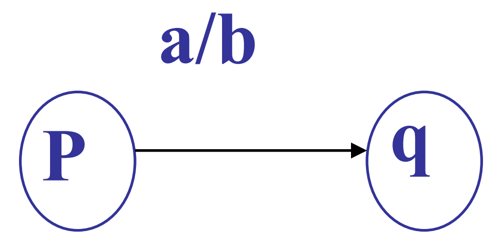
$q_0$: 初始状态 $q_0\in Q$

δ和g函数共同描述了米兰机的工作状况.

# 米兰机

米兰机图形表示:

$$\begin{cases} \delta(p, a) = q \\ g(p, a) = b \end{cases} \Rightarrow$$

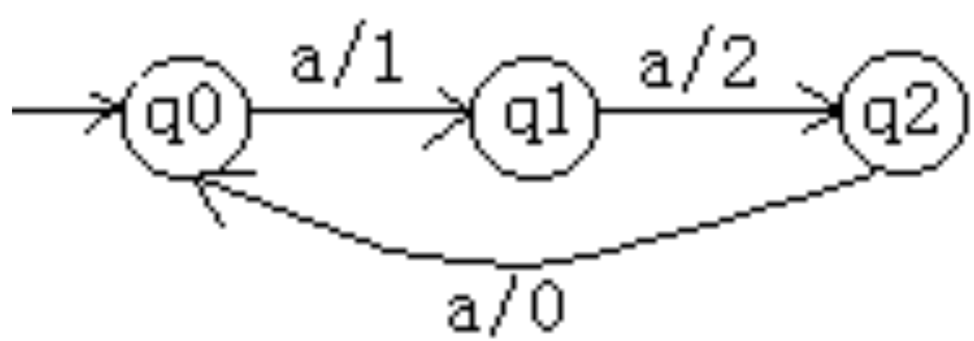



**例:** (P79 例2) 设计米兰机，其输出是输入字符个数的模3数

**解:** 输出字母表R={0, 1, 2}. 设输入字母表T={a},

M的状态数应有3个，记录已输入字符个数的模3数.

Q={$q_0$, $q_1$, $q_2$} 分别表示输入字符数模3 = 0, 1, 2



# 米兰机

**例:** （P80 例3）

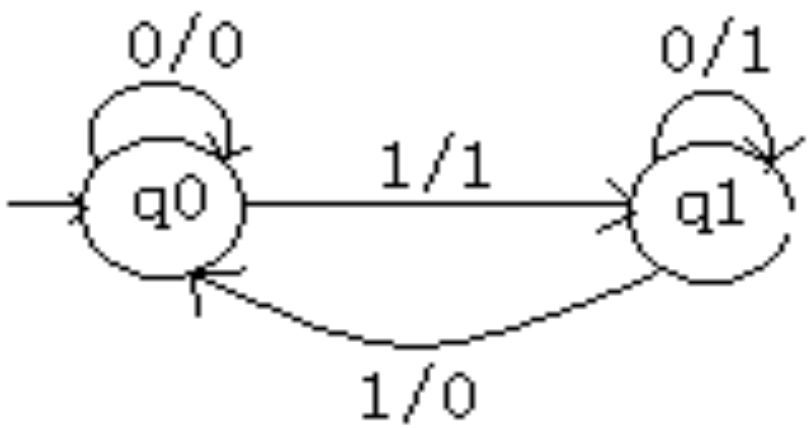
设计米兰机， 其输入$\in$\{0，1\}*，

当输入串有奇数个1时，输出1.

当输入串有偶数个1时，输出0.

**解:** 需二个状态$\{q_0, q_1\}$

$q_0$ 表示输入串有偶数个1， $q_1$ 表示输入串有奇数个1



# 课堂练习

**练习:** 设计米兰机，输入是0，1组成的串，要求输出串对输入串延迟两个时间单位.

解: $M=(Q, T, R, \delta, g, q_0)$, $T=\{0, 1\}$ $R=\{0, 1\}$

**分析**: 可能的状态---即一个输入在输出前可能处于的状态

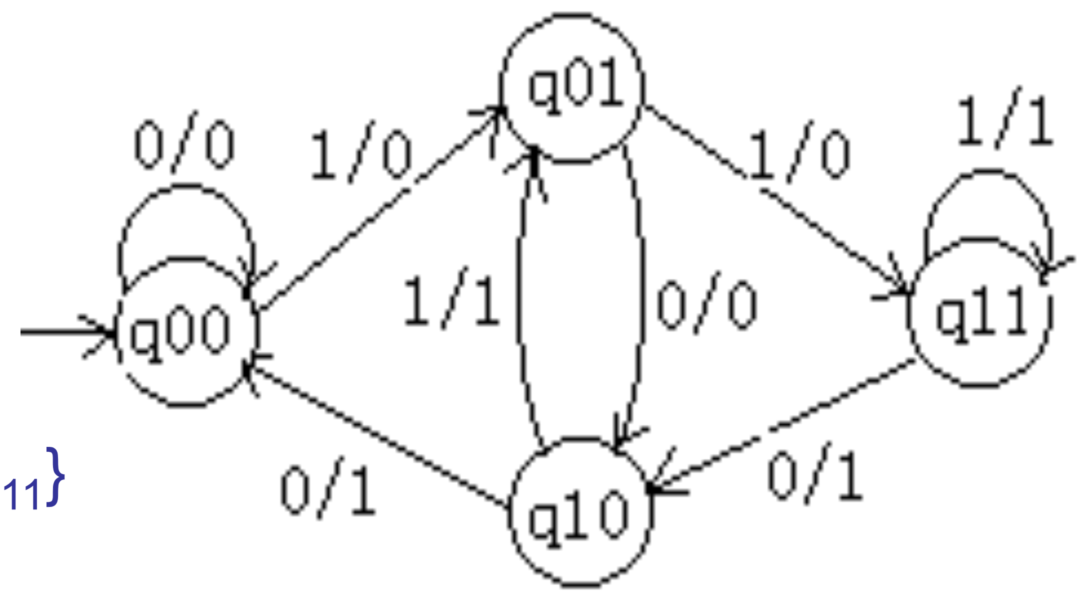
Q: 00 $q_{00}$

01 $q_{01}$

10 $q_{10}$

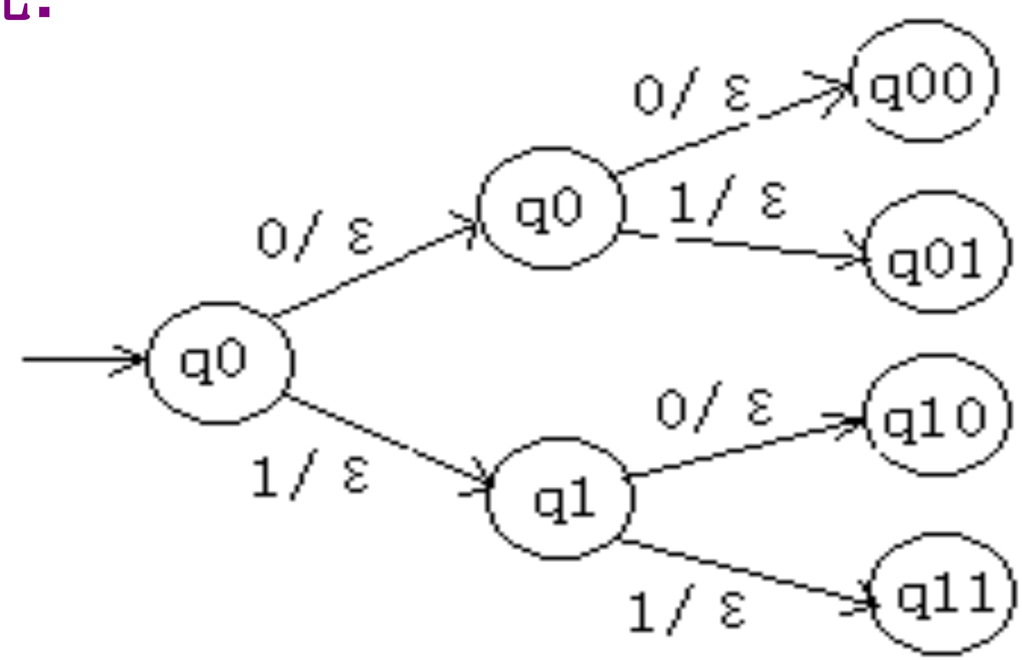
11 $q_{11}$

$Q=\{q_{00}, q_{01}, q_{10}, q_{11}\}$



# 课堂练习

## 初始情况:




刚开始工作时输入前两个字符，输出为ε

# 课堂练习

设语言L由0，1组成，且字符串的最后两个字符相同.

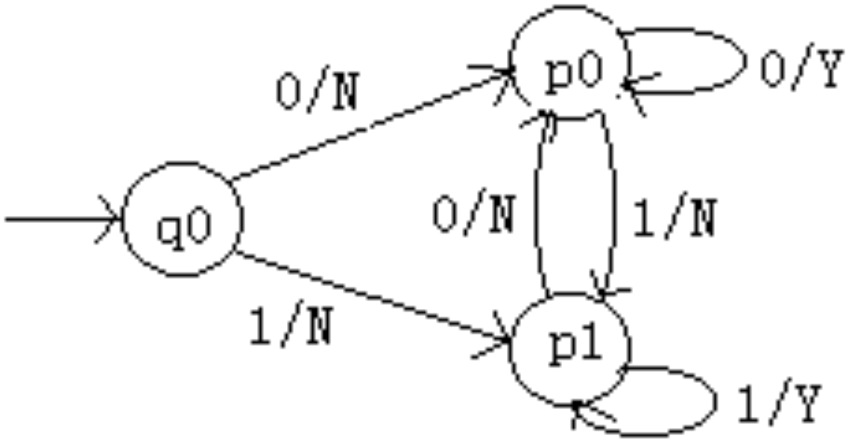
构造米兰机M ，输出Y/N表示输入串是否属于L.

**解**: 设状态集Q为

初始状态$q_0$

状态$p_0$ 表示输入串最后字符为0

状态$p_1$ 表示输入串最后字符为1



# 摩尔机

摩尔机的输出只与到达的状态有关

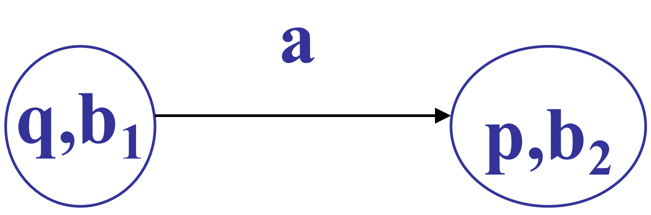
形式定义: $M=(Q, T, R, \delta, g, q_0)$

$g: Q \rightarrow R$

$\delta: Q \times T \rightarrow Q$

图形表示:

$$\begin{cases} \delta(q, a) = p \\ g(p) = b_2 \end{cases} \Longrightarrow$$



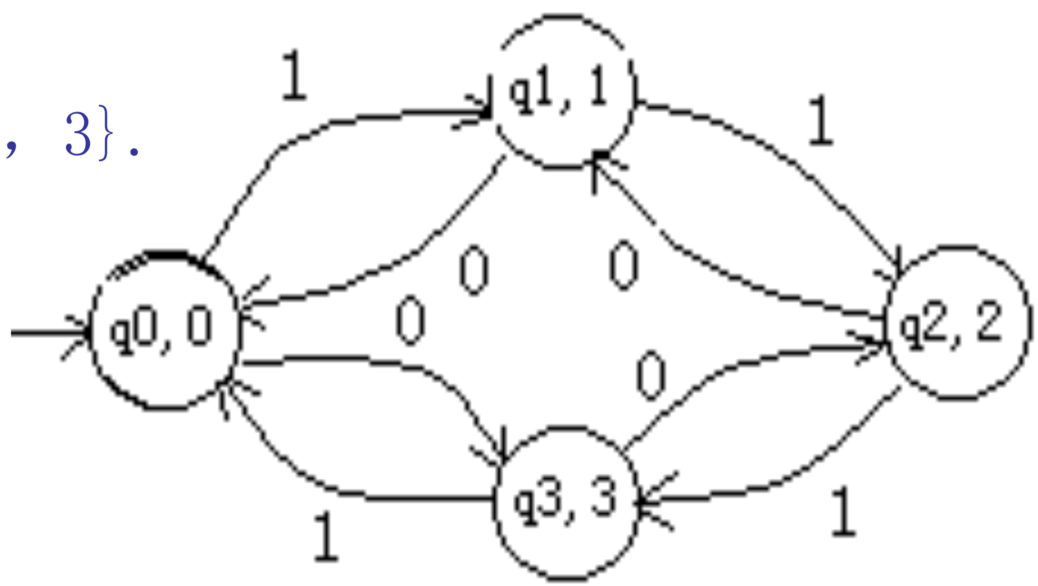
# 摩尔机

**例:** (书P81 例4)

设计自动机M，其输入串∈{0，1}*，输出是$(n_1 - n_0) \bmod 4$，$n_0$是ω中含0的个数，$n_1$是ω中含1的个数。

**解:** 需四个状态，

取输出字母表 R={0，1，2，3}.




四个状态 $q_0$， $q_1$， $q_2$， $q_3$
分别输出0，1，2，3

# 课堂练习

设计摩尔机，接受0，1组成的串，串的首符为1，串中出现且只出现一个0。若接受输出Y，否则输出N。

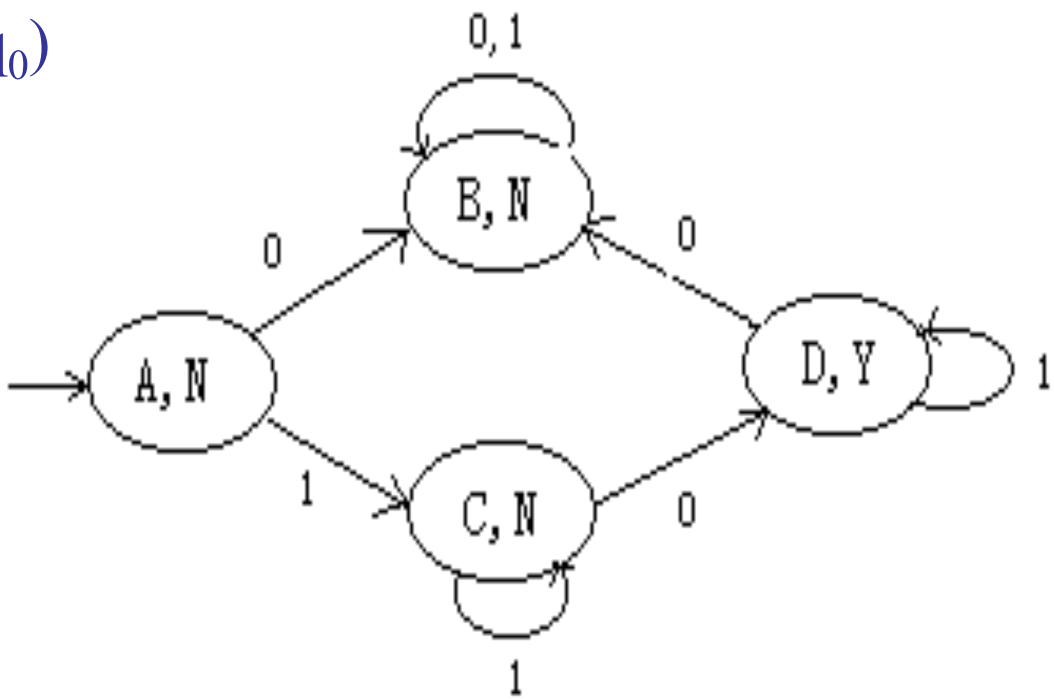
**解**: L的串应为$11^*01^*$

M=(Q，T，R，δ， g， $q_0$)

T={0，1}，    R={Y，N}

设计状态

**Q:**
- A 初始状态
- B 拒绝接受
- C 可能接受
- D 接受



# 米兰机和摩尔机的变换

- 已知摩尔机可构造等价的米兰机

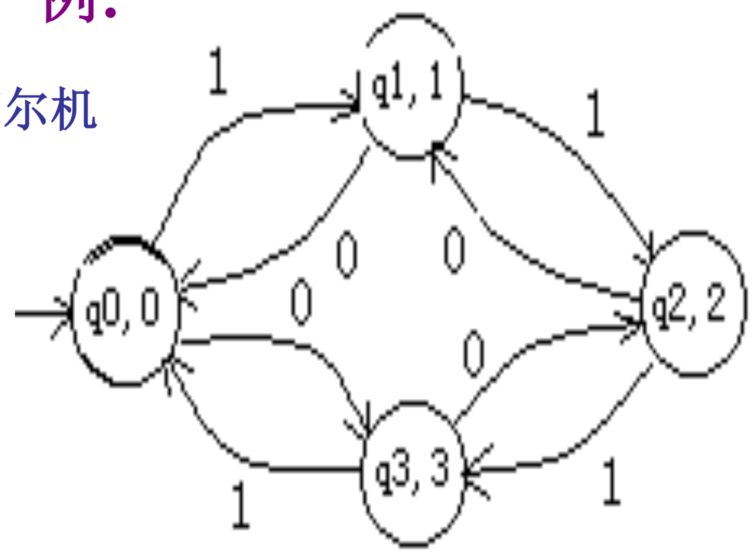
设 摩尔机M＝$(Q, T, R, \delta, g, q_0)$　米兰机M’＝$(Q, T, R, \delta, g', q_0)$

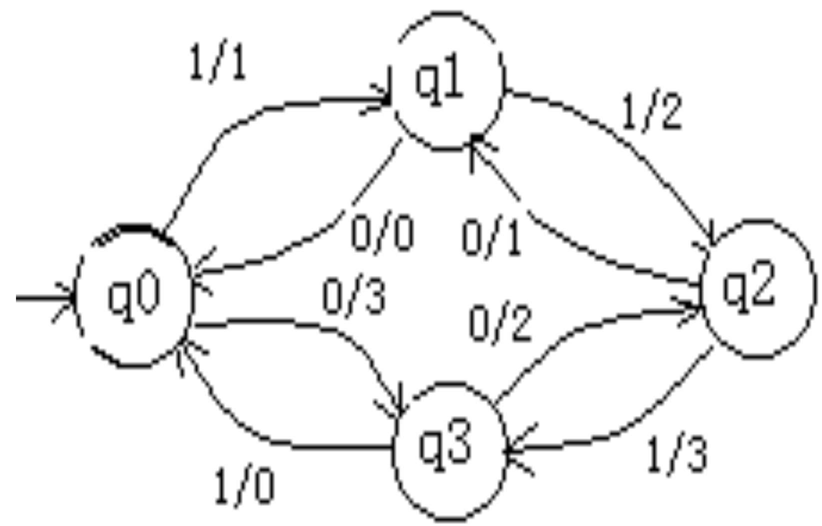
如果M中有$\delta(q，a)=p$，$g(p) = b$

则M’中有$g'(q，a) = b = g(\delta(q，a))$

**例:**

摩尔机




米兰机

# 米兰机和摩尔机的变换

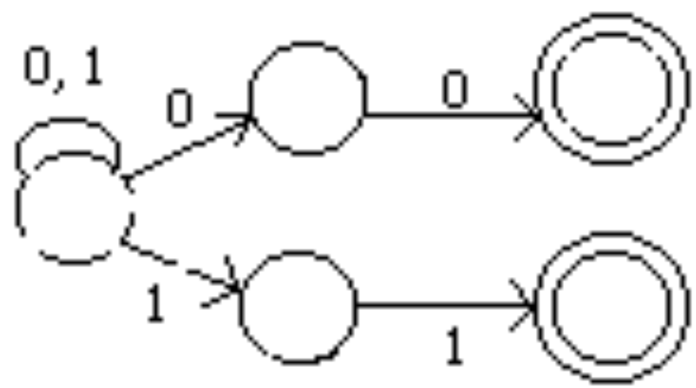
- 由米兰机也可构造等价的摩尔机——（略）
- 小结

双向自动机，米兰机， 摩尔机的共同优点是描述问题清晰，且节省状态.

**例如：** 对$L = (0+1)^*(00+11)$ (前面米兰机的例子)

米兰机只需三个状态

而用DFA则不会少于5个状态



作业：

Chap3 习题 17(1,2,3) 18, 19 题