复习

- 文法的定义
- Chomsky文法体系分类

$G=(N,T,P,S)$

$P: \alpha \rightarrow \beta$

0	无限制	递归可枚举语言	图灵机
1.	$\lvert\alpha\rvert \le \lvert\beta\rvert$	上下文有关文法	线性有界自动机
2.	$A \rightarrow \beta$	上下文无关文法	下推自动机
3.	$A \rightarrow wB \mid w$ $A \rightarrow Bw \mid w$	正则文法	有限自动机

第三章 有限自动机与右线性文法

本章主要内容

- 确定有限自动机
- 非确定有限自动机
- 确定与非确定有限自动机的等价性
- 右线性文法和有限自动机的等价性
- 右线性文法的性质(泵浦定理)
- 使用归纳法进行证明的方法

一、有限状态系统的概念

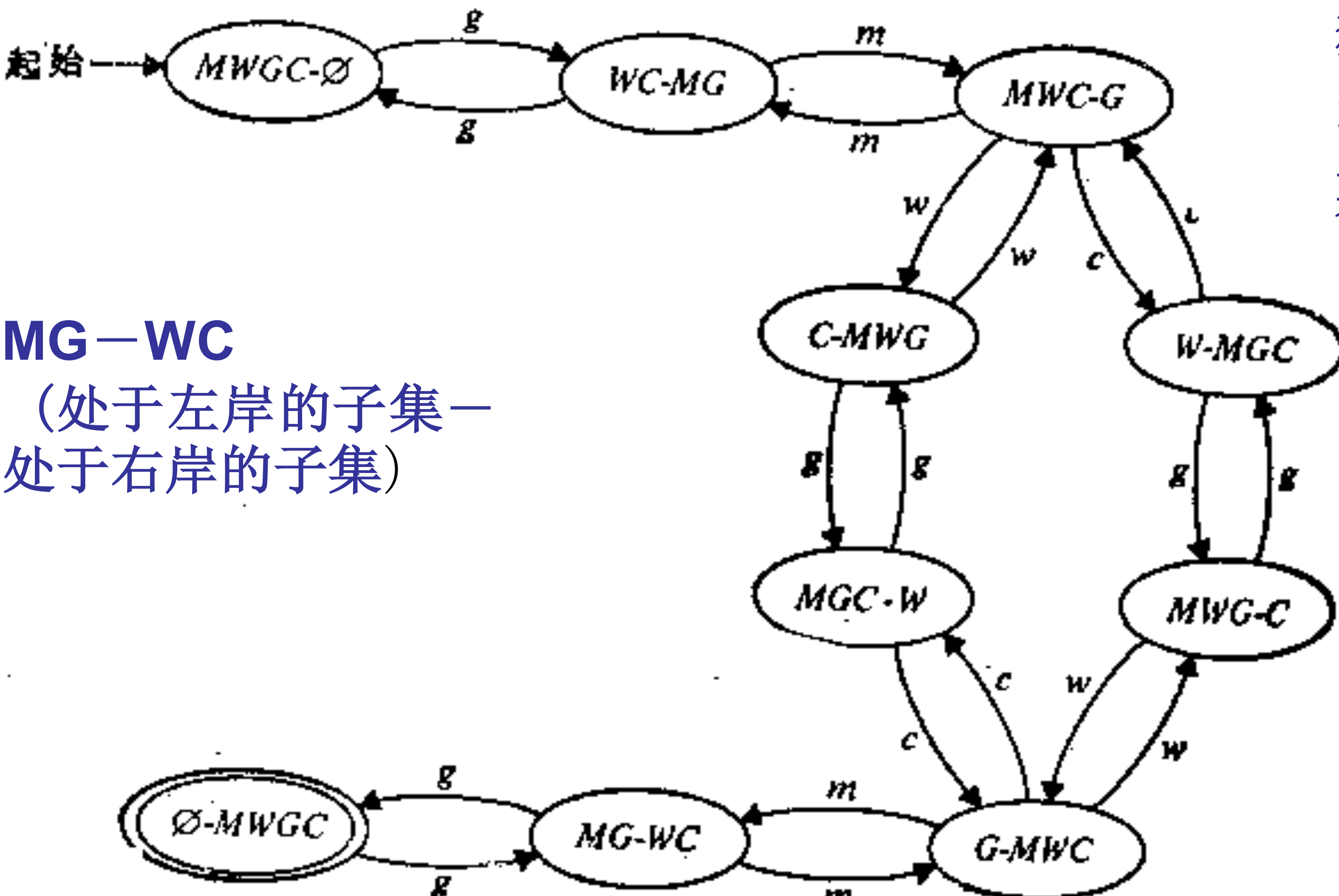
- 状态:状态是可以将事物区分开的一种标识。
- 具有离散状态的系统:如数字电路(0,1), 十字路口的红绿灯。离散状态系统的状态数是有限的.
- 具有连续状态的系统:比如水库的水位,室内温度等可以连续变化,即有无穷个状态.
- 有限状态系统必然是离散状态系统（而且状态数有限）,因为只有有限个状态.

- 实例

　　一个人带着一头狼，一头羊，以及一棵青菜，处于河的左岸。有一条小船,每次只能携带人和其余的三者之一。人和他的伴随品都希望渡到河的右岸，而每摆渡一次，人仅能带其中之一。然而如果人留下狼和羊不论在左岸还是在右岸，狼肯定会吃掉羊。类似地，如果单独留下羊和菜，羊也肯定会吃掉菜。如何才能既渡过河而羊和菜又不被吃掉呢?

将过河问题模型化：

人(M)
狼(W)
羊(G)
菜(C)



MG－WC
（处于左岸的子集－处于右岸的子集）

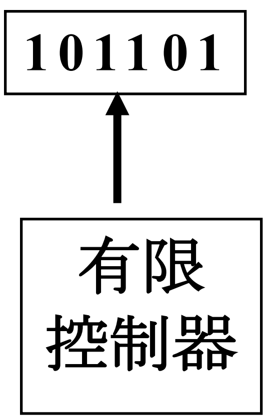
二、有限自动机的概念

有限自动机的概念

- 具有离散 输入 输出系统的一种数学模型 (可以没有输出， 比较特殊的也可以没有输入).
- 有限的状态
- 状态+输入→状态转移
- 每次转换的后继状态都唯一 → DFA
- 每次转换的后继状态不唯一 → NFA
  - DFA Deterministic finite automaton
  - NFA Non Deterministic finite automaton

FA 的模型

FA可以理解成一个控制器,它读一条输入带上的字符。



(1) 控制器包括有限状态;

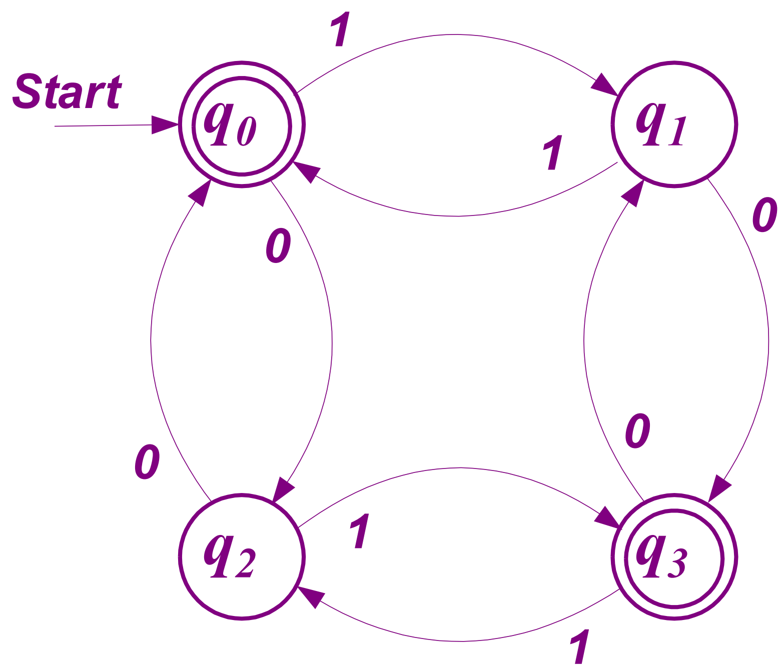
(2) 从左到右依次读取字符;

(3) 状态+激励 → 状态迁移

(根据当前所处状态和输入字符进行状态转移)

有限自动机的五要素

- ✧ 有限状态集
- ✧ 有限输入符号集
- ✧ 转移函数
- ✧ 一个开始状态
- ✧ 一个终态集合



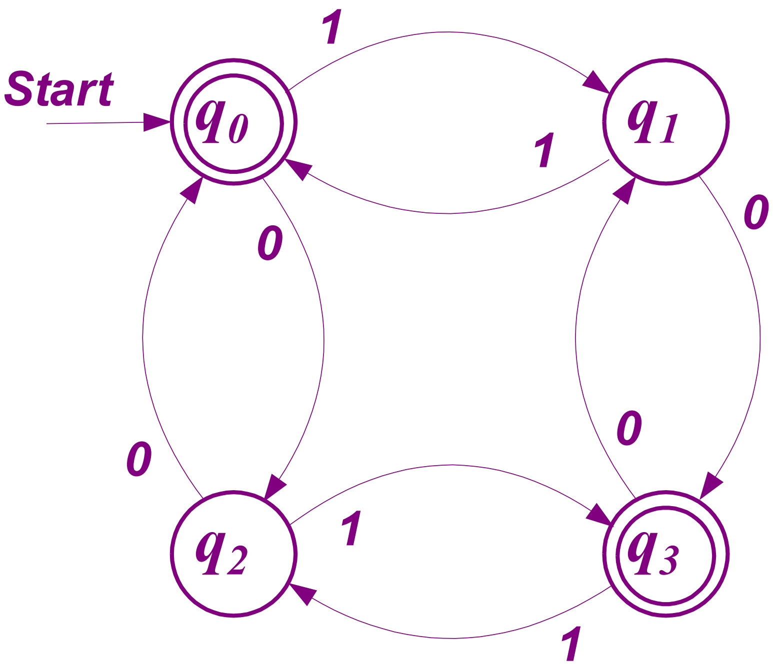
三、DFA的形式定义

定义：DFA是一个五元组 $M=(Q,T,\delta,q_0,F)$

- **Q: 有限的状态集合**
- **T: 有限的输入字母表**
- **δ: 转换函数(状态转移集合): $Q\times T \rightarrow Q$**
- **q_0: 初始状态， $q_0 \in Q$**
- **F: 终止状态集，$F \subseteq Q$**

- **表示方法:**

转移图表示的 DFA



- $Q = \{q_0, q_1, q_2, q_3\}$
- $T = \{0, 1\}$
- $\delta(q_0, 0) = q_2,\ \delta(q_0, 1) = q_1$
  $\delta(q_1, 0) = q_3,\ \delta(q_1, 1) = q_0$
  $\delta(q_2, 0) = q_0,\ \delta(q_2, 1) = q_3$
  $\delta(q_3, 0) = q_1,\ \delta(q_3, 1) = q_2$
- q_0
- $F = \{q_0, q_3\}$

转移表表示的 DFA

	0	1
$\rightarrow *q_0$	q_2	q_1
q_1	q_3	q_0
q_2	q_0	q_3
$*q_3$	q_1	q_2

- $Q = \{q_0, q_1, q_2, q_3\}$
- $T = \{0, 1\}$
- $\delta(q_0, 0) = q_2,\ \delta(q_0, 1) = q_1$
  $\delta(q_1, 0) = q_3,\ \delta(q_1, 1) = q_0$
  $\delta(q_2, 0) = q_0,\ \delta(q_2, 1) = q_3$
  $\delta(q_3, 0) = q_1,\ \delta(q_3, 1) = q_2$
- q_0
- $F = \{q_0, q_3\}$

四、扩展转移函数适合于输入字符串

δ’函数：接收一个字符串的状态转移函数。

$$\delta': Q \times T^* \to Q$$

✧ 对任何$q \in Q$，定义：

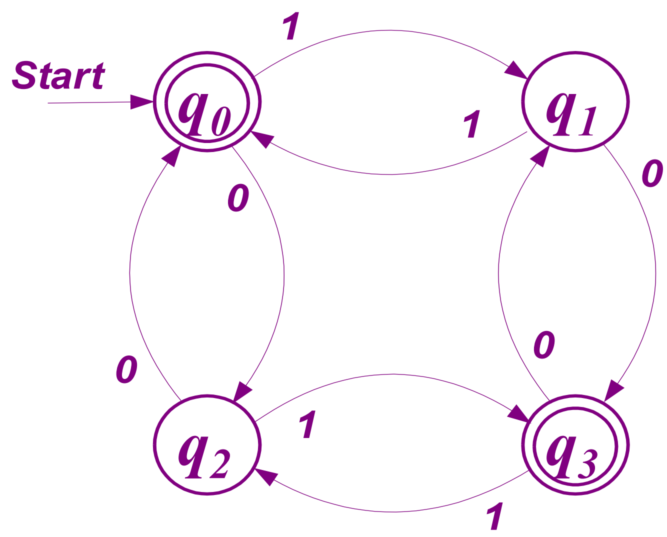
1. $\delta'(q, \varepsilon) = q$
2. 若ω是一个字符串, a是一个字符

   定义: $\delta'(q,\omega a)=\delta(\delta'(q,\omega),a)$

对于DFA:$\delta'(q,a)=\delta(\delta'(q,\varepsilon),a)=\delta(q,a)$，即对于单个字符时δ和δ'是相等的。为了方便，以后在不引起混淆时用δ代替δ'

扩展转移函数适合于输入字符串

	0	1
$\rightarrow *q_0$	q_2	q_1
q_1	q_3	q_0
q_2	q_0	q_3
$*q_3$	q_1	q_2



✧ 举例

$$\delta'(q_0, \varepsilon) = q_0$$

$$\delta'(q_0, 0) = \delta(q_0, 0) = q_2$$

$$\delta'(q_0, 00) = \delta(q_2, 0) = q_0$$

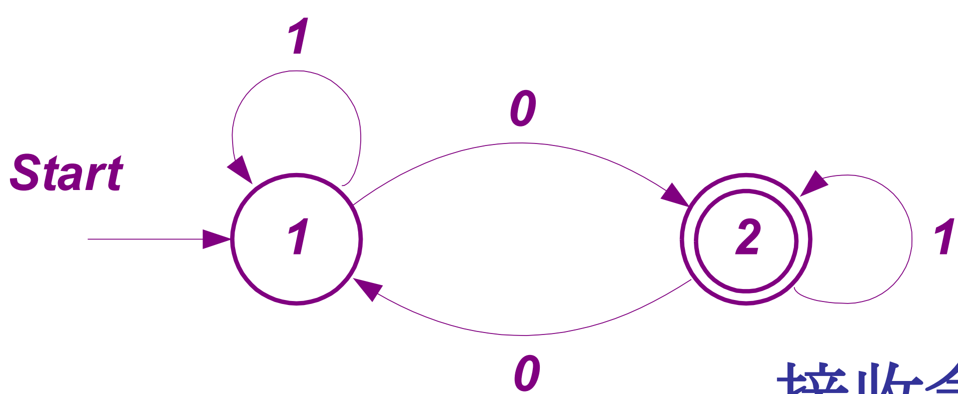
$$\delta'(q_0, 001) = \delta(q_0, 1) = q_1$$

$$\delta'(q_0, 0010) = \delta(q_1, 0) = q_3$$

DFA接受的语言

- **被DFA接收的字符串:** 输入结束后使DFA的状态到达终止状态。否则该字符串不能被D FA接收.
- **DFA接收的语言:** 被DFA接收的字符串的集合.

  $$L(M) = \{\ \omega \mid \delta'(q_0, \omega) \in F\ \}$$
- 例：T = {0, 1}

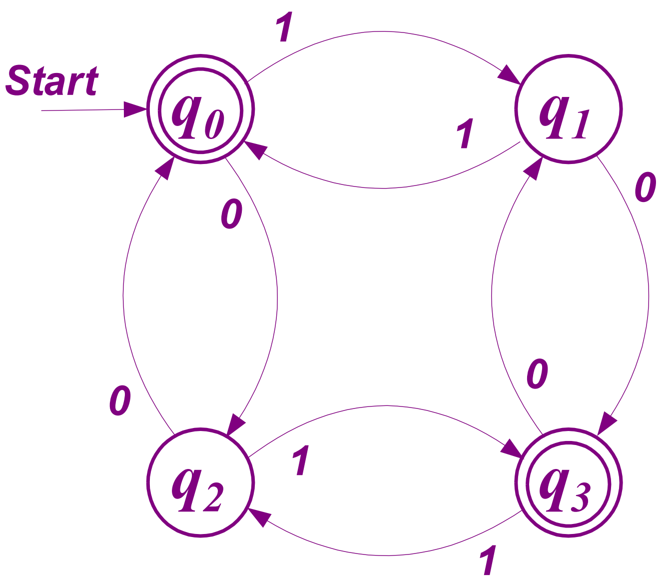


接收含有奇数个0的任意串

五、格局

- 为描述有限自动机的工作过程，对于它在某一时刻的工作状态，可用两个信息表明：当前状态q，待输入字符串ω。两者构成一个瞬时描述，用（q, ω）表示，称为格局。
- 初始格局：（q_0, ω)
- 终止格局：(q,ε), q∈F

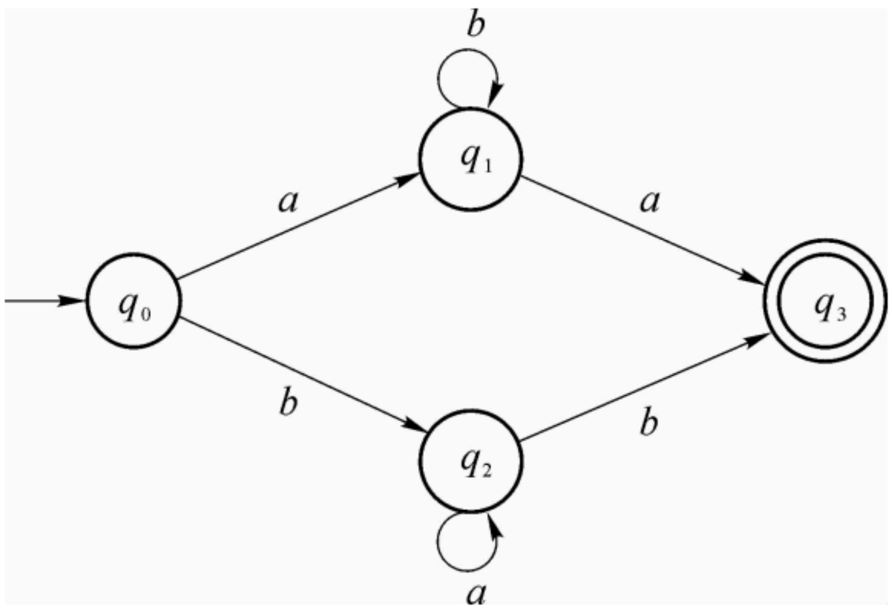
格局示例



- 如图，接受001010的格局
  （q_0,001010） ├(q_2,01010) ├(q_0,1010) ├(q_1,010) ├(q_3,10) ├(q_2,0) ├(q_0,ε)
- 格局数量是无限的。
- 有限状态自动机是无记忆的。例如接受0010101111和接受01011111时，都可以进入格局(q_0,1111)。

课堂作业

- 写出下面自动机的完整定义以及状态转移表
- 采用格局方式写出该自动机识别abba的过程

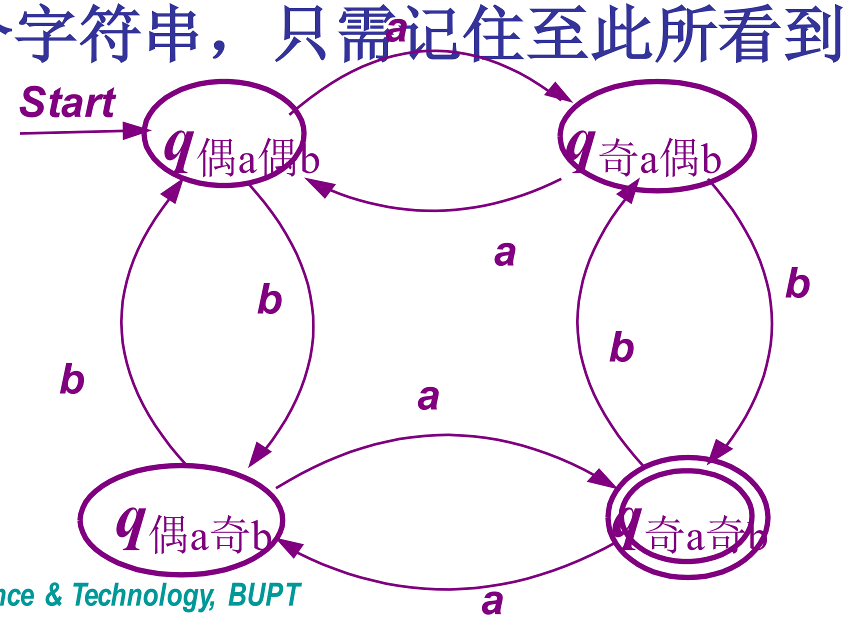


设计有限自动机

- 自动机的设计是一个创造过程，没有简单的算法或过程。
- 技巧：假设自己是机器，思考如何去实现机器的任务。
- 为判断到目前为止所看到的字符串是否满足某个语言，须估算出读一个字符串时需要记住哪些关键的东西。

例1：构造自动机，识别所有由奇数个a和奇数个b组成的字符串。

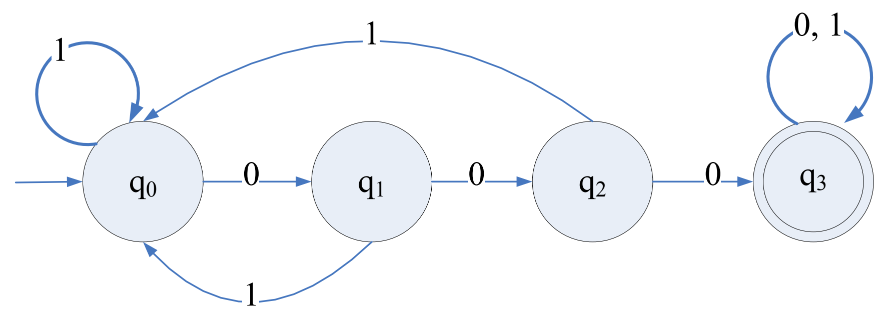
关键：不需要记住所看到的整个字符串，只需记住至此所看到的a、b个数是偶数还是奇数。



- 例2：构造有限自动机M，识别{0,1}上的语言
  - $L=\{x000y \mid x,y\in\{0,1\}^*\}$
- 分析：该语言特点是每个串都包含连续3个0的子串，自动机的任务是识别/检查是否存在子串000。

  由于字符是逐一读入，当读入一个0时，就需记住（状态q_1），

  若接着读入的还是0，则需记住已读入连续的2个0了（状态q_2），若接着读入的还是0，则需记住已读入连续的3个0了（状态q_3）.



例3：构造有限自动机M，识别{0,1,2}上的语言,每个字符串代表的数字能整除3。

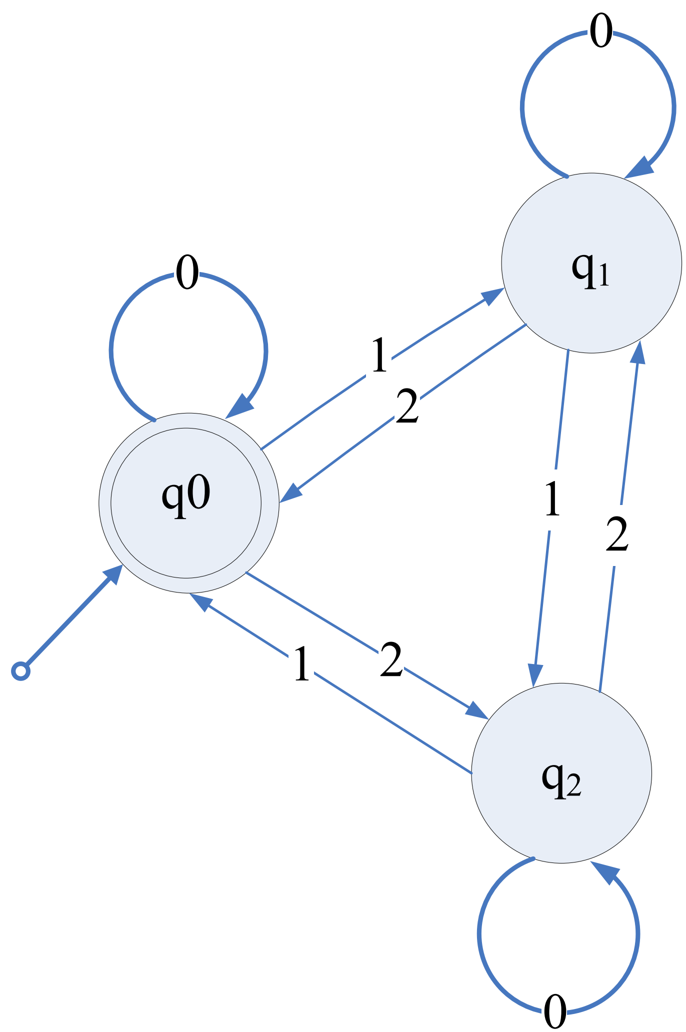
分析：如果一个十进制数的所有位的数字之和能整除3，则该十进制数就能整除3。

一个十进制数除以3，余数只能为0、1、2。---设计为状态。

状态q_0表示已读入的数字和除3余0,

状态q_1表示已读入的数字和除3余1,

状态q_2表示已读入的数字和除3余2,

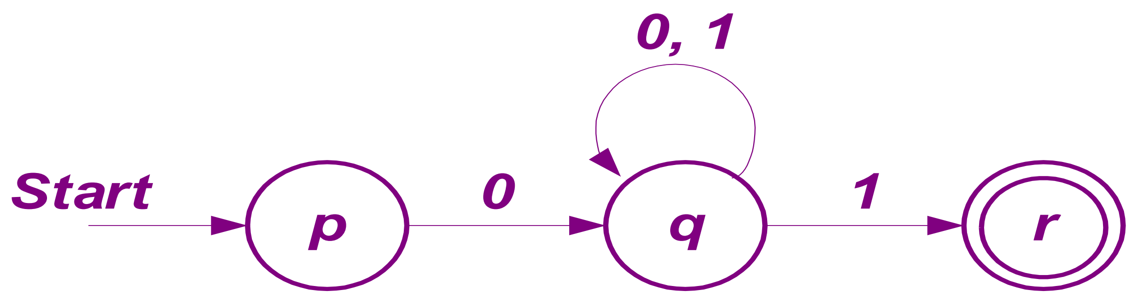


第二节 不确定的有限自动机(NFA)

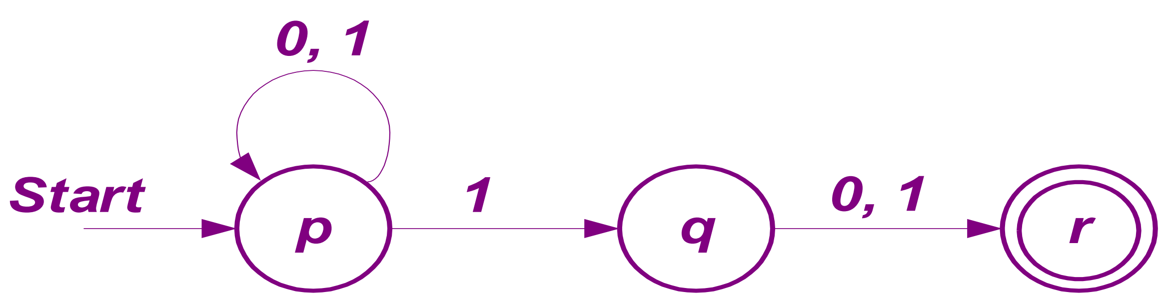
修改DFA的模型,使之在某个状态, 对应一个输入,可以有多个转移, 到达不 同的状态, 即：具有同时处于几个状态的能力。 则称为不确定的有限自动机。

例 ：

(1)



(2)



一、不确定有限自动机的形式定义

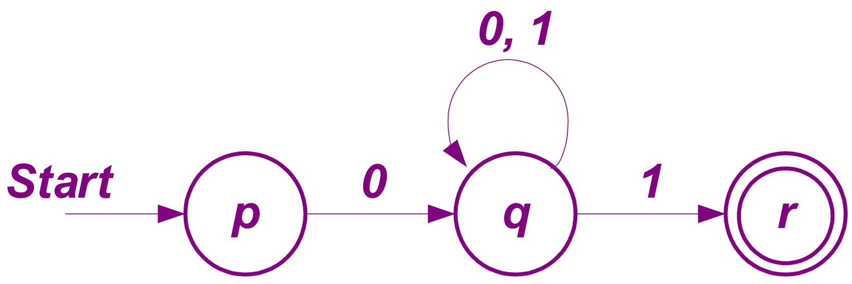
- NFA是一个五元组,M=(Q,T,δ,q0,F).

  其中δ是$Q \times T \rightarrow 2^Q$的函数,其余与DFA相同.

- 如果接收一个字符串后NFA进入一个状态集,而此集合中包含一个以上F中的状态, 则称NFA接收该字符串.

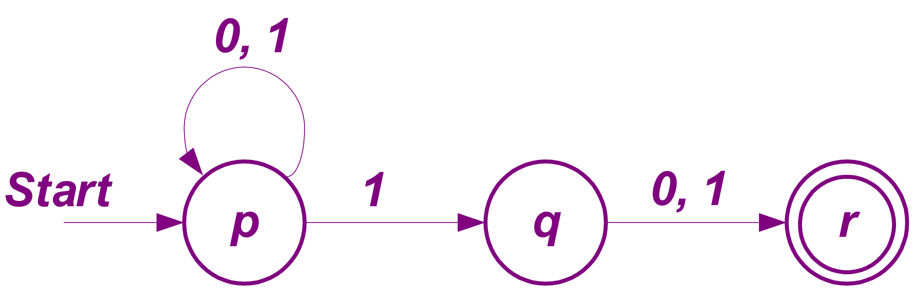
转移图和转移表表示的NFA

(1)



	0	1
$\rightarrow p$	$\{q\}$	ϕ
q	$\{q\}$	$\{q, r\}$
$*r$	ϕ	ϕ

(2)



	0	1
$\rightarrow p$	$\{p\}$	$\{p, q\}$
q	$\{r\}$	$\{r\}$
$*r$	ϕ	ϕ

注意：转移表中的每一项都是一个集合。含空集Φ

二、NFA的状态转移函数

✧ 与 DFA 唯一不同之处　$\delta: Q \times T \to 2^Q$

同样, δ可扩展为δ' （$\delta': Q \times T^* \to 2^Q$)

1.$\delta'(q, \varepsilon) = \{q\}$　　含义: 不允许无输入的状态变化.

2.$\delta'(q,\omega a)=\{p \mid$存在$r\in\delta'(q,\omega)\wedge p\in\delta(r,a)\}$

- 含义:δ'(q,ωa)对应的状态集合是δ'(q,ω)对应的每个状态下再接收字符a以后可能到达的状态集合的并集. 即

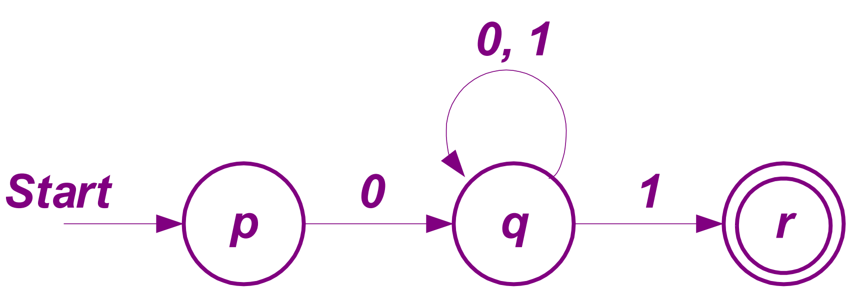
若 $\delta'(q, \omega) = \{r_1, r_2, \dots, r_k\}$, 则

$\delta'(q, \omega a) = \cup\ \delta(r_i, a)$

其中 $\omega \in T^*$, $a \in T$, $r_i \in Q$

扩展转移函数适合于输入字符串

	0	1
$\rightarrow p$	$\{q\}$	ϕ
q	$\{q\}$	$\{q, r\}$
$*r$	ϕ	ϕ



✧ 举例

$$\delta'(p, \varepsilon) = \{p\}$$

$$\delta'(p, 0) = \{q\}$$

$$\delta'(p, 01) = \{q, r\}$$

$$\delta'(p, 010) = \{q\}$$

$$\delta'(p, 0100) = \{q\}$$

$$\delta'(p, 01001) = \{q, r\}$$

NFA 接受的语言

- 设一个 NFA $A = (Q, T, \delta, q_0, F)$
- 定义 A 的语言：

$$L(A) = \{ \omega \mid \delta'(q_0, \omega) \cap F \neq \phi \}$$

第三节 NFA与DFA的等价性

- **DFA**是**NFA**的特例，所以**NFA**必然能接收**DFA**能接收的语言. 因此证明等价性只要能够证明一个**NFA**所能接收的语言必能被另一个**DFA**所接收。

1. 定理: 设一个**NFA**接受语言**L**, 那么必然存在一个**DFA**接受**L**。

2. 证明:

   - 策略:对于任意一个**NFA**,构造一个接收它所能接收语言的**DFA**，这个**DFA**的状态对应了**NFA**的状态集合。

从 NFA 构造等价的 DFA（子集构造法）

✧ 设 L 是某个 NFA $N = (Q_N, \mathrm{T}, \delta_N, q_0, F_N)$ 的语言，则存在一个 DFA M，满足 $L(M) = L(N) = L$.

✧ **证明:** 定义 $M = (Q_D, \mathrm{T}, \delta_D, \{q_0\}, F_D)$，其中

- $Q_D = \{ S \mid S \subseteq Q_N \} = 2^{Q}$
- 对 $S \in Q_D$ 和 $a \in \mathrm{T}$，$\delta_D(S, a) = \bigcup_{q \in S} \delta_N(q,a)$.
- $F_D = \{S \mid S \subseteq Q_N \wedge S \cap F_N \neq \phi \}$

需要证明：对任何 $\omega \in \mathrm{T}^*$,

$$\delta'_D(\{q_0\}, \omega) = \delta'_N(q_0, \omega).$$

归纳于 $|w|$ 可证上述命题. □

子集构造法举例

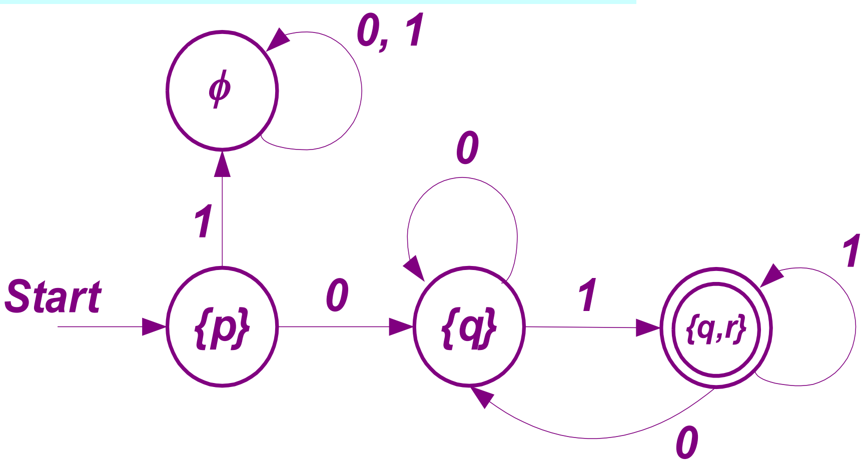
1、初始的NFA

	0	1
$\rightarrow p$	$\{q\}$	ϕ
q	$\{q\}$	$\{q, r\}$
$* r$	ϕ	ϕ

2、子集构造，计算状态可达

	0	1
ϕ	ϕ	ϕ
$\rightarrow \{p\}$	$\{q\}$	ϕ
$\{q\}$	$\{q\}$	$\{q, r\}$
$* \{r\}$	ϕ	ϕ
$\{p, q\}$	$\{q\}$	$\{q, r\}$
$*\{p, r\}$	$\{q\}$	ϕ
$*\{q, r\}$	$\{q\}$	$\{q, r\}$
$*\{p, q, r\}$	$\{q\}$	$\{q, r\}$

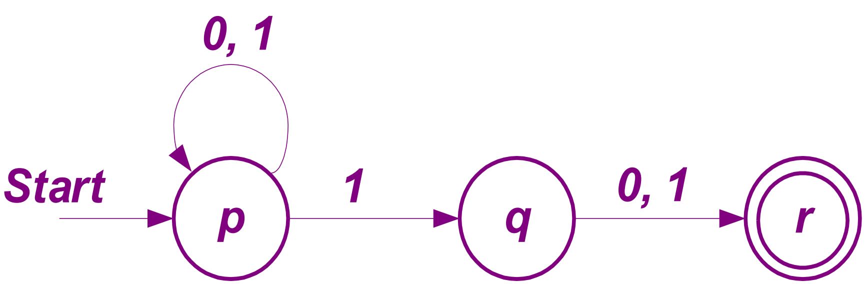
3、经筛选后的DFA

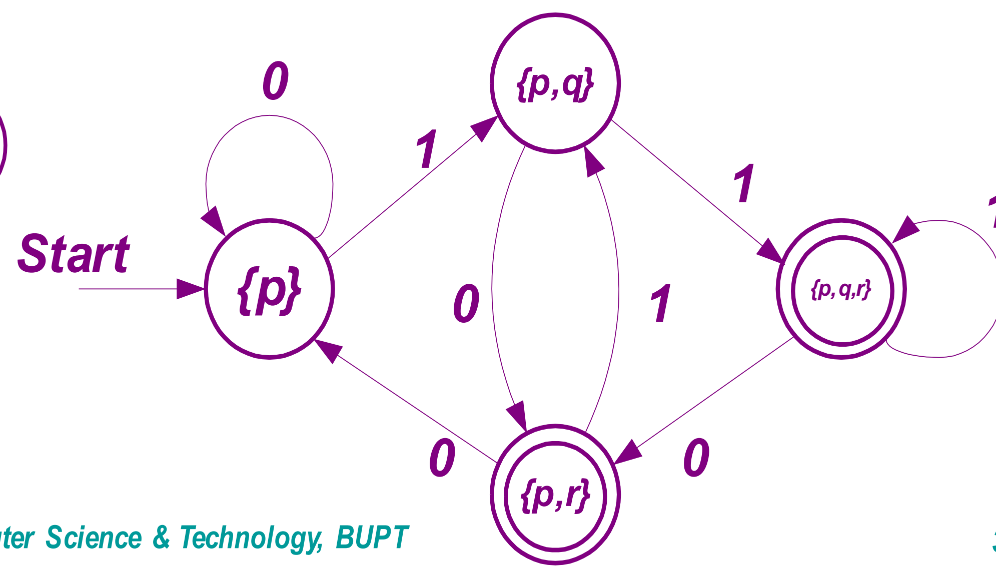


子集构造法举例

	0	1
$\rightarrow p$	$\{p\}$	$\{p, q\}$
q	$\{r\}$	$\{r\}$
$*r$	ϕ	ϕ

	0	1
$\rightarrow \{p\}$	$\{p\}$	$\{p, q\}$
$\{p, q\}$	$\{p, r\}$	$\{p, q, r\}$
$*\{p, r\}$	$\{p\}$	$\{p, q\}$
$*\{p, q, r\}$	$\{p, r\}$	$\{p, q, r\}$





证明:从 NFA 构造等价的 DFA

- 设 $N = (Q_N, T, \delta_N, q_0, F_N)$ 是一个 NFA，通过子集构造法得到相应的 DFA $D = (Q_D, T, \delta_D, [q_0], F_D)$，则

  对任何 $\omega \in T^*$， $\delta'_D(\{q_0\}, \omega) = \delta'_N(q_0, \omega)$.

- 证明:归纳于 $|\omega|$

  1 设 $|\omega| = 0$, 即 $\omega = \varepsilon$.

  由定义知 $\delta'_D(\{q_0\}, \varepsilon) = \delta'_N(q_0, \varepsilon) = \{q_0\}$.

  2 设 $|\omega| = n+1$, 并 $\omega = xa$, $a \in T$. 注意到 $|x| = n$.

  假设 $\delta'_D(\{q_0\}, x) = \delta'_N(q_0, x) = \{p_1, p_2, \ldots, p_k\}$.

  则 $\delta'_D(\{q_0\}, \omega) = \delta_D(\delta'_D(\{q_0\}, x), a)$

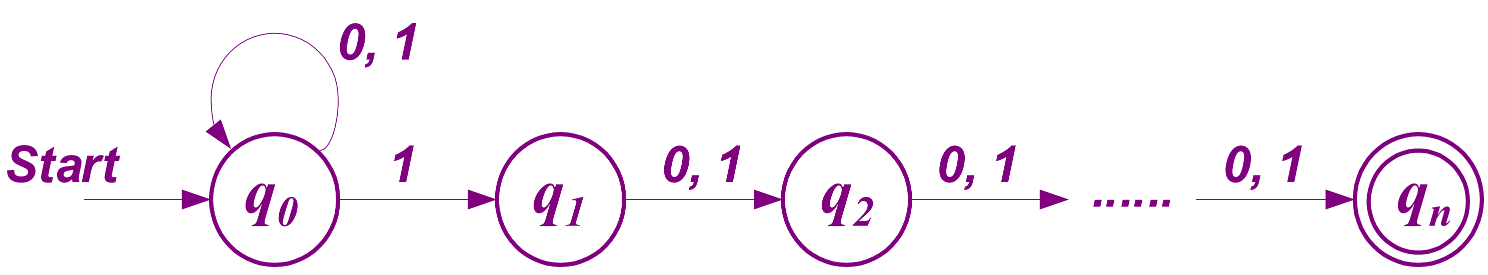
  $= \delta_D(\{p_1, p_2, \ldots, p_k\}, a) = \bigcup_{i=1}^{k} \delta_N(p_i, a)$.

  $= \delta'_N(q_0, \omega)$

子集构造法得到的状态数

- 实践中，通过子集构造法得到的 DFA 的状态数目与原NFA 的状态数目大体相当.
- 在较坏的情况下，上述 DFA 的状态数目接近于所有子集的数目.
- 举例

  由如下 NFA 构造的 DFA 的状态数目为2^n



课堂作业

- 设计有限状态自动机识别{a,b}上的字符串$\{w|xabay, x,y\in\{a,b\}^*\}$，并写出该自动机识别aabab的过程。

作业： 第三章 习题10, 14

10. 设字母表 $T=\{a,b\}$，找出接受下列语言的 DFA：

(1) 含有 3 个连续 b 的所有字符串的集合；

(2) 以 aa 为首的所有字符串集合；

(3) 以 aa 结尾的所有字符串集合；

(4) $L=\{a^n b^m a^k \mid n,m,k \geqslant 0\}$。

14. 构造 DFA M_1 等效于 NFA M，NFA M 如下：

(1) $M=(\{q_0,q_1,q_2,q_3\},(a,b),\delta,q_0,\{q_3\})$，其中 δ 如下：

$$
\begin{array}{ll}
\delta(q_0,a)=\{q_0,q_1\} & \delta(q_0,b)=\{q_0\} \\
\delta(q_1,a)=\{q_2\} & \delta(q_1,b)=\{q_2\} \\
\delta(q_2,a)=\{q_3\} & \delta(q_2,b)=\varnothing \\
\delta(q_3,a)=\{q_3\} & \delta(q_3,b)=\{q_3\}
\end{array}
$$

(2) $M=(\{q_0,q_1,q_2,q_3\},(a,b),\delta,q_0,\{q_1,q_3\})$，其中 δ 如下：

$$
\begin{array}{ll}
\delta(q_0,a)=\{q_1,q_3\} & \delta(q_0,b)=\{q_1\} \\
\delta(q_1,a)=\{q_2\} & \delta(q_1,b)=\{q_1,q_2\} \\
\delta(q_2,a)=\{q_3\} & \delta(q_2,b)=\{q_0\} \\
\delta(q_3,a)=\varnothing & \delta(q_3,b)=\{q_0\}
\end{array}
$$