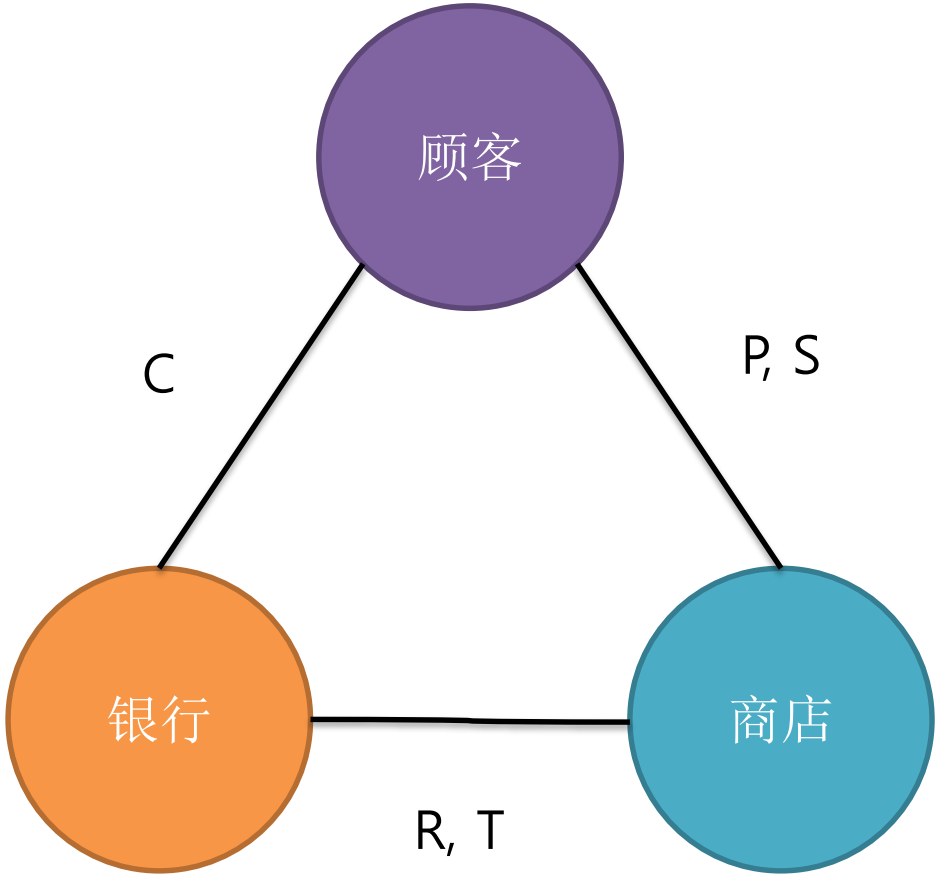
第二章（DFA）2024

本章内容

- 在对于DFA有感性认识的前提下，本章采用形式化方式介绍DFA的完整内容，并过渡到非确定型有穷自动机NFA以及有ε-转移的非确定有穷自动机ε-NFA，证明它们三者在枚举语言的能力上等价。
- 先从DFA开始，包括DFA定义及表示、DFA如何判定输入串为接受还是拒绝、扩展DFA转移函数以表示DFA的连续转移，并用于表示DFA的语言。
- 类似的叙述方式用于NFA和ε-NFA的介绍，并引入ε-闭包和ε-闭集概念，以及将NFA转换为DFA的子集构造法。
- 概括粗粒度知识点：语言识别器；判定性质；等价性质。

一个简单购物系统的DFA建模示例（F2.1）



P付款：顾客把电子货币发给商店，商店收到。

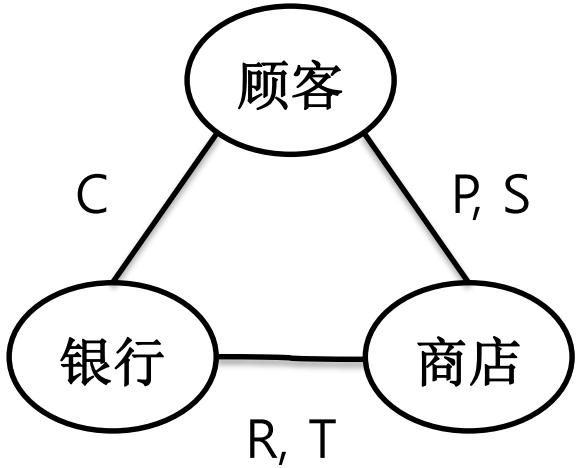
S送货：商店送货给顾客，顾客收到。

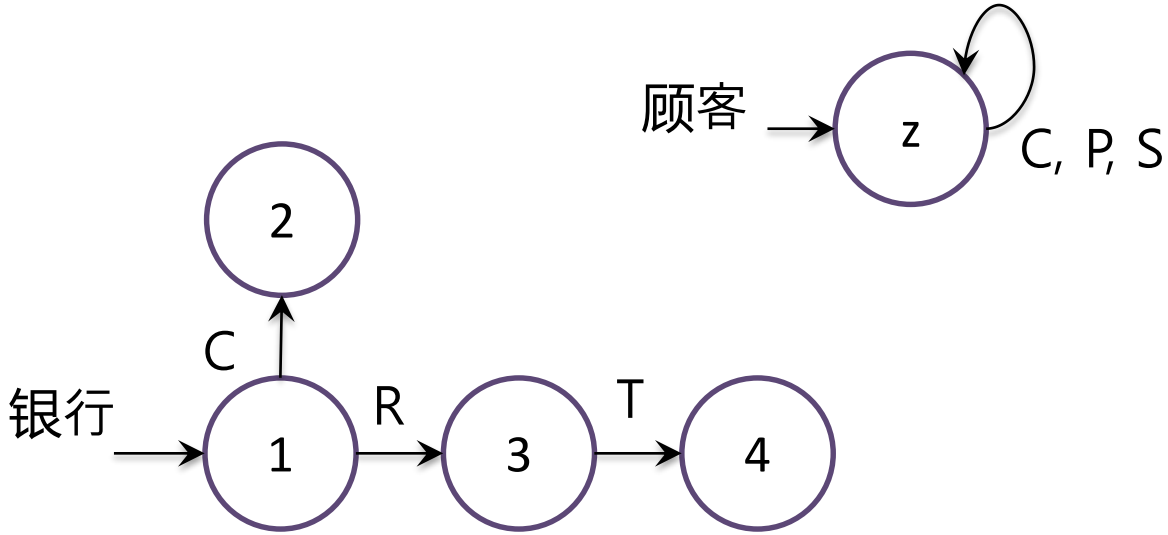
C取消：顾客让银行取消电子货币，银行取消。

R兑换：商店将电子货币发给银行，银行收到并予以认证。

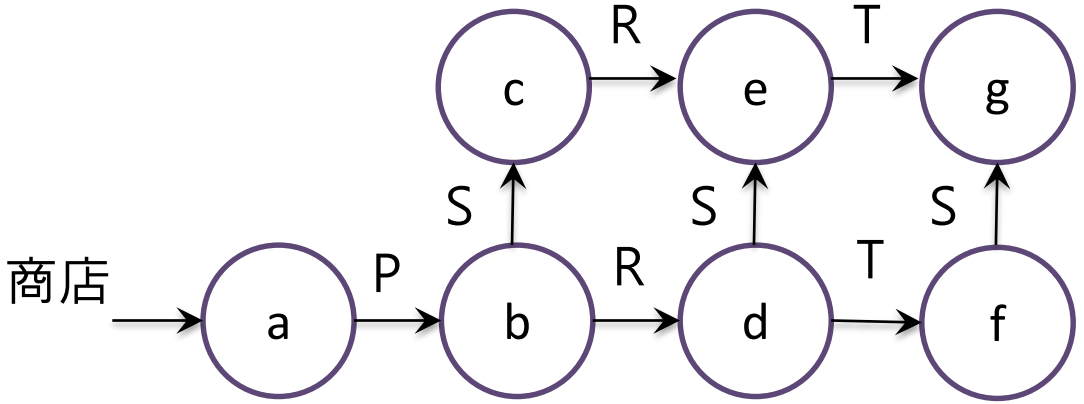
T转帐：银行签发给商店电子货币，商店收到。

顾客、商店、银行的自动机模型

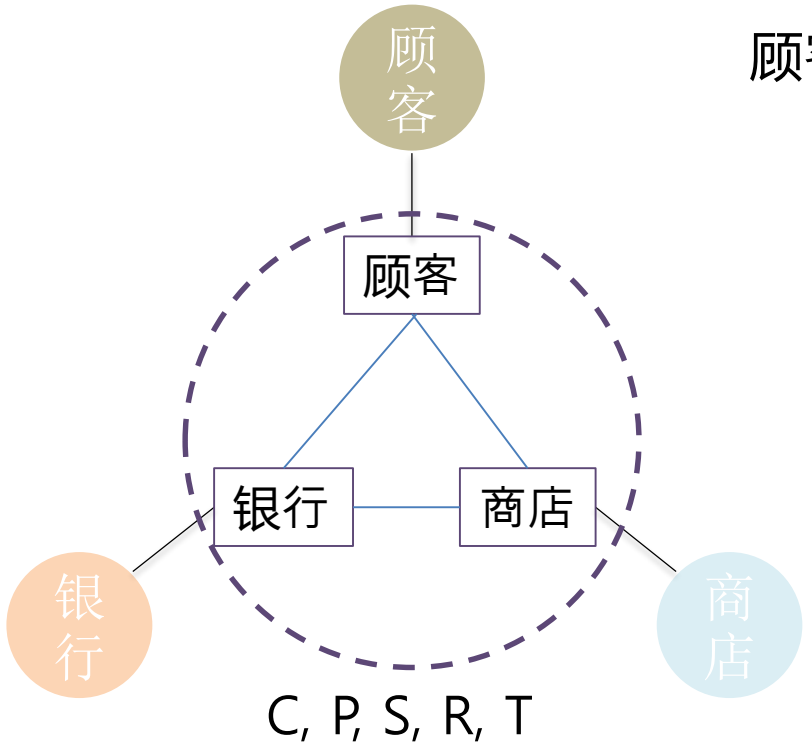


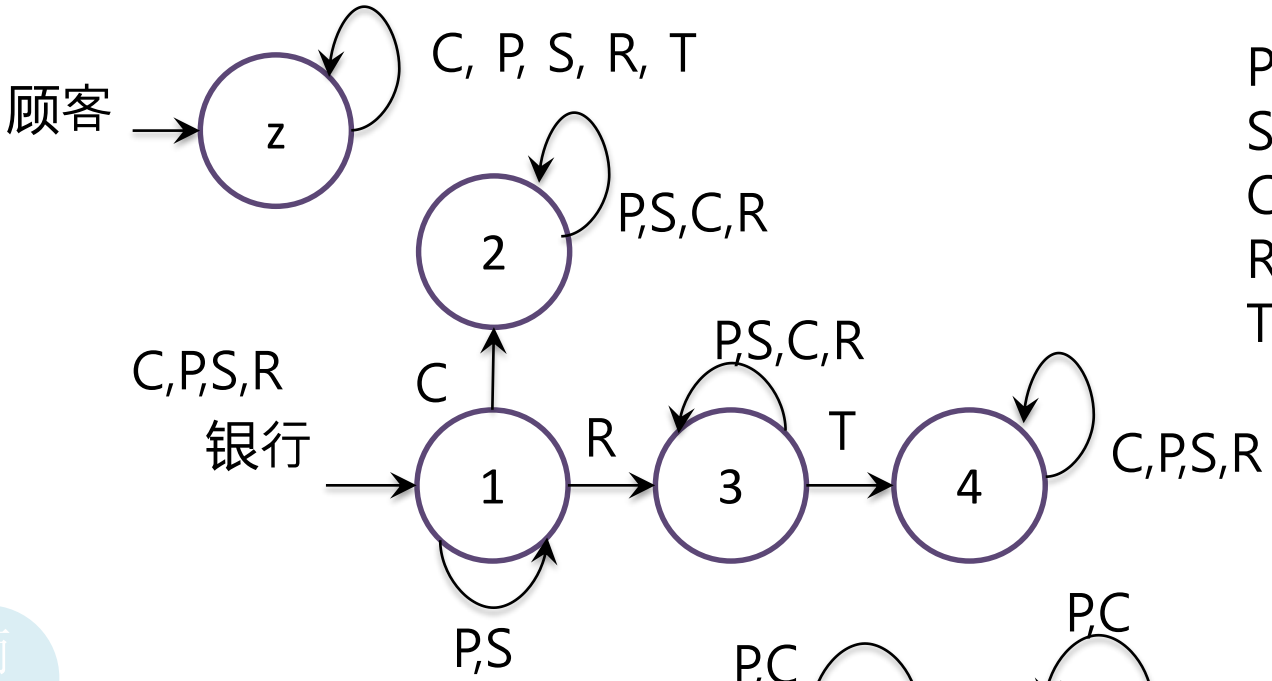


P付款：顾客把电子货币发给商店，商店收到。
S送货：商店送货给顾客，顾客收到。
C取消：顾客让银行取消电子货币，银行取消。
R兑换：商店将电子货币发给银行，银行收到并予以认证。
T转帐：银行签发给商店电子货币，商店收到。

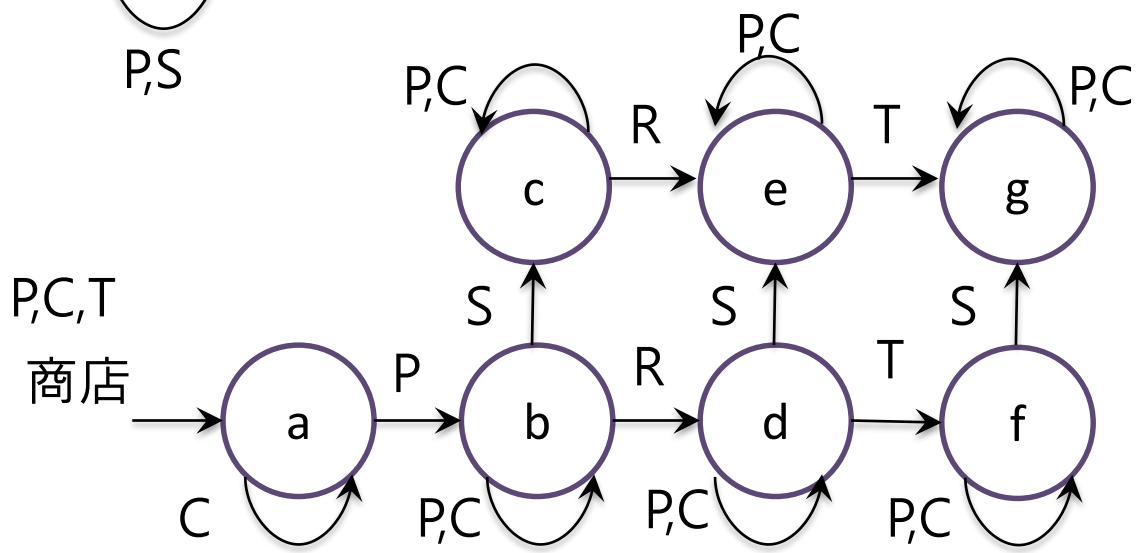


一个系统：各自动机对于无关交互保持迁移不变



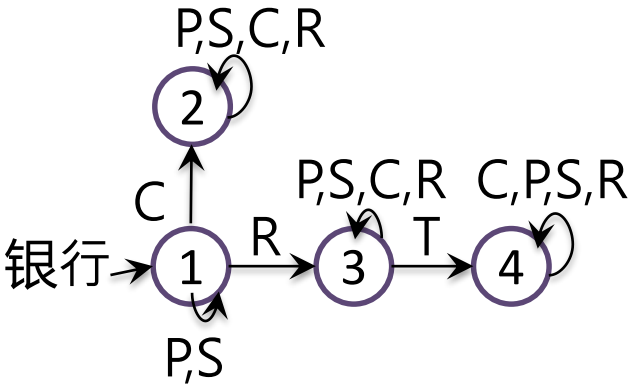


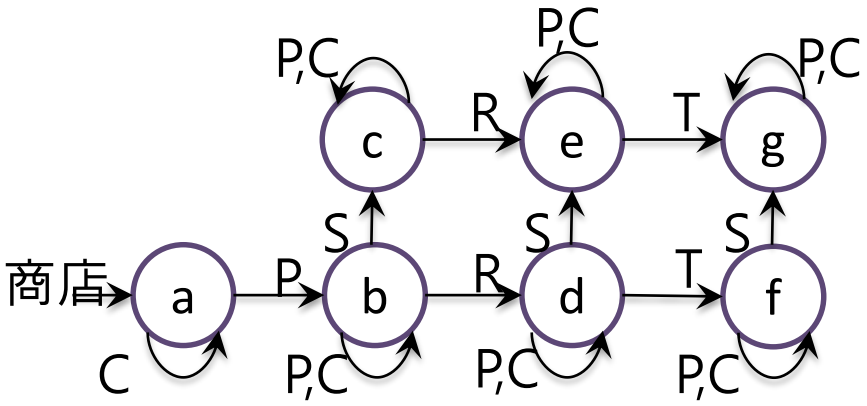
P付款【顾客】
S送货【商店】
C取消【顾客】
R兑换【商店】
T转帐【银行】

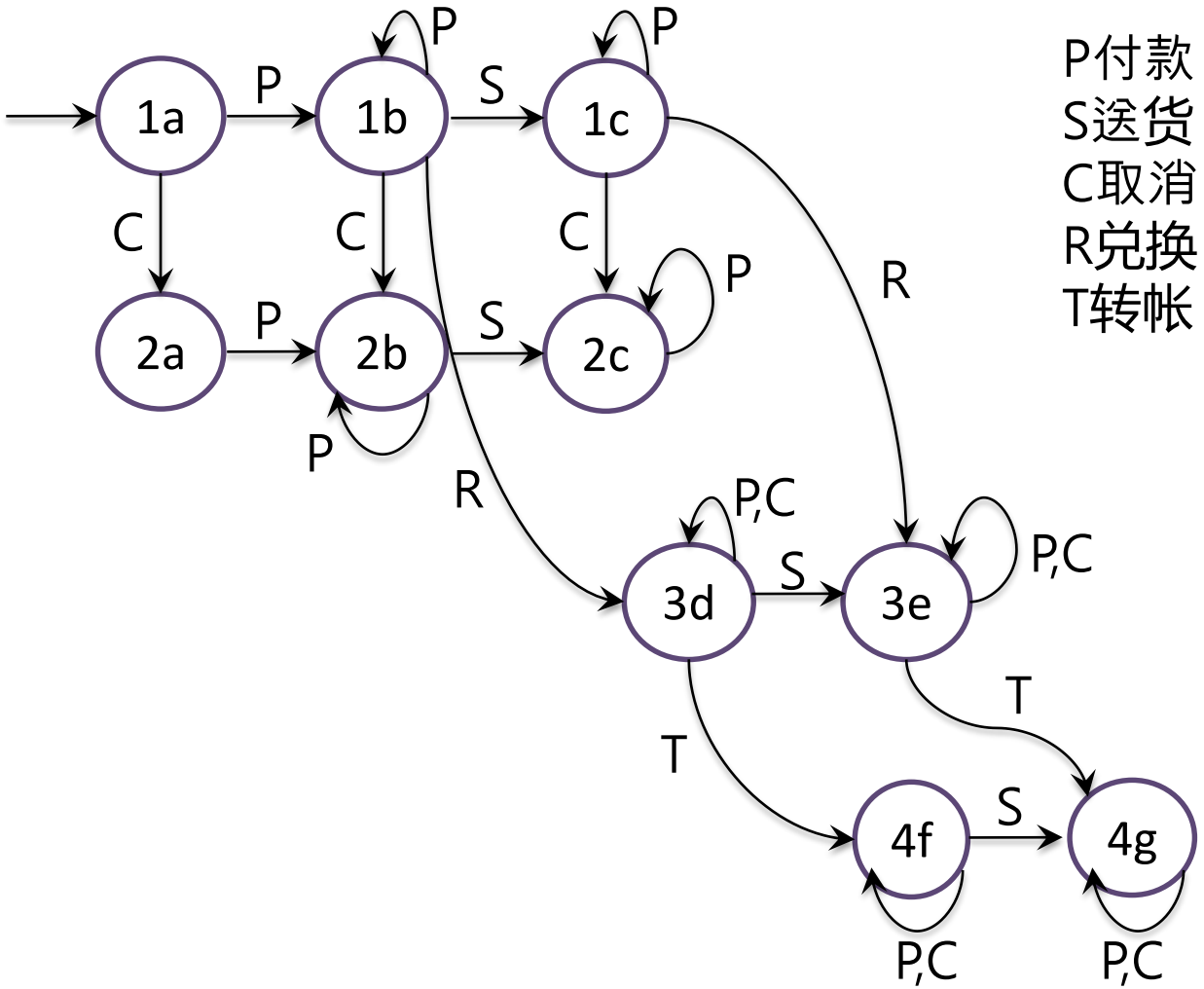


三个各自的状态对于由另外两方发起的交互都要有迁移弧射出；银行收到R后才T；客户随意C和P；商店收到P后会S也会R

乘积自动机



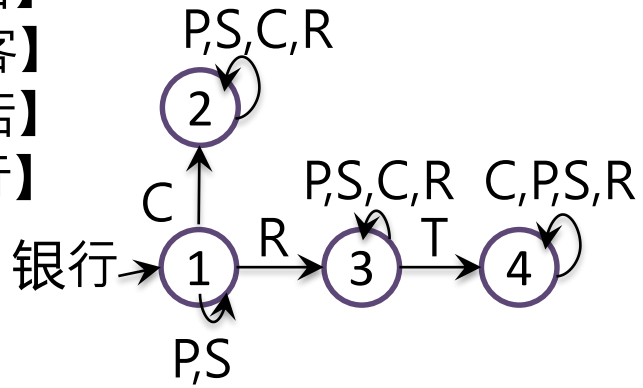


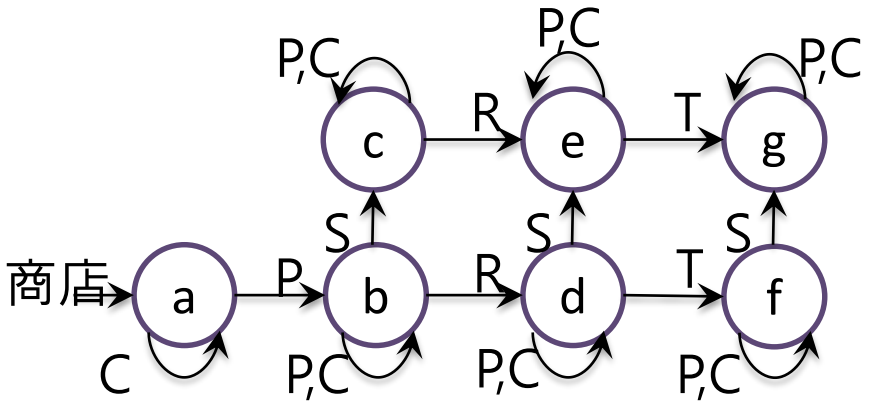


P付款【顾客】
S送货【商店】
C取消【顾客】
R兑换【商店】
T转帐【银行】

乘积自动机

P付款【顾客】
S送货【商店】
C取消【顾客】
R兑换【商店】
T转帐【银行】





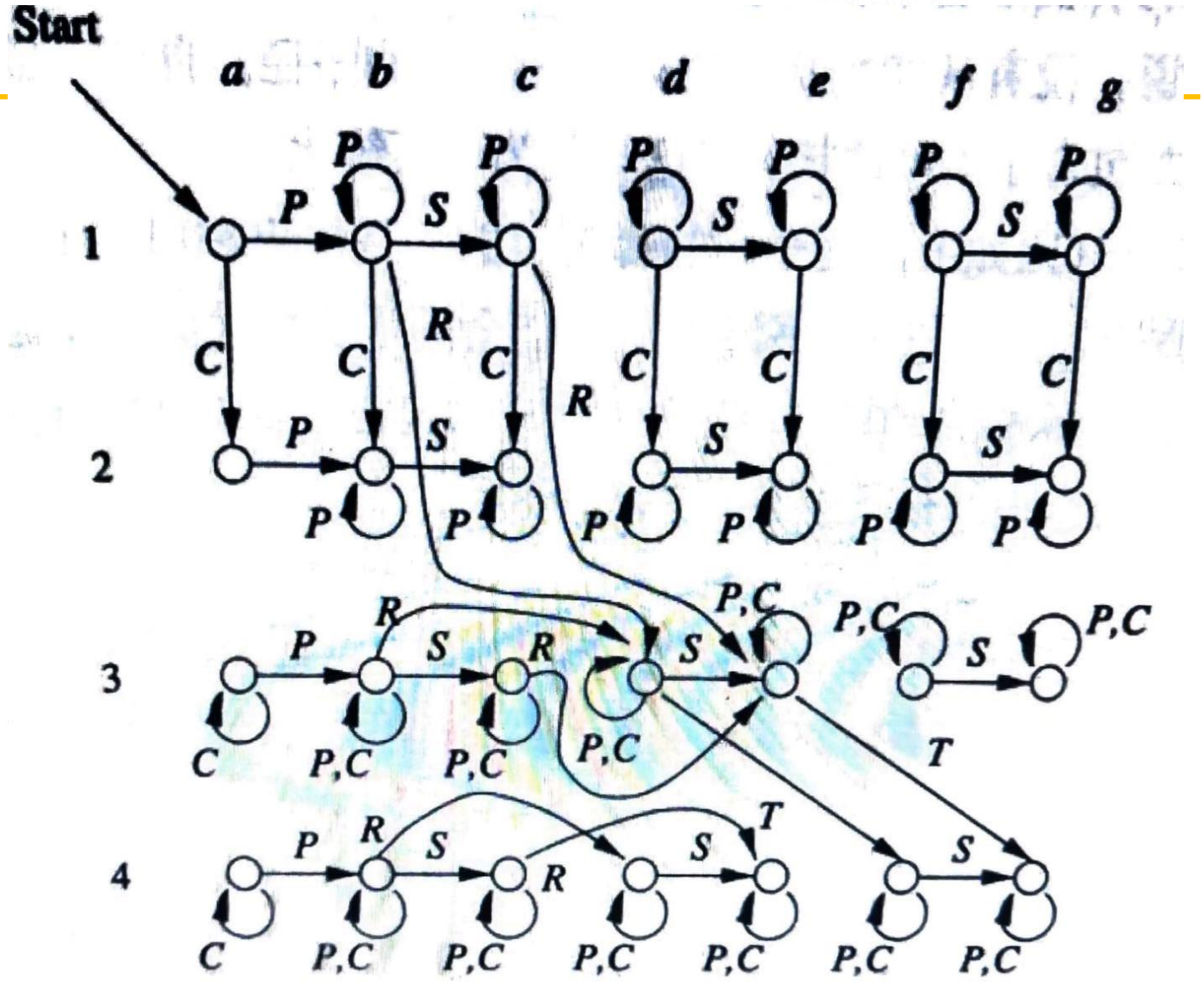
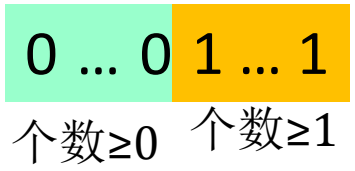


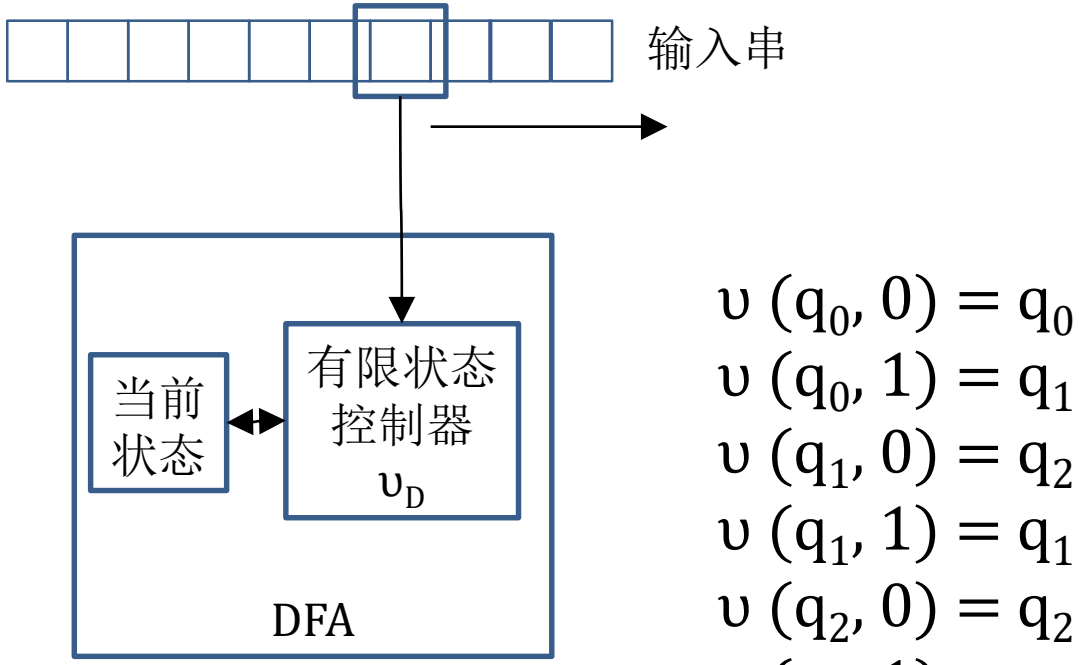
图2-3 商店和银行的乘积自动机

2.1 确定型有穷自动机DFA

- DFA $A = (Q, \Sigma, \upsilon, q_0, F)$
- 有穷个状态的集合Q
- 字母表Σ，有穷个符号的集合
- 状态转移函数$\upsilon: Q \times \Sigma \to Q$
- $q_0 \in Q$ 初始状态
- $F \subseteq Q$ 接受状态



q_0 初始状态　q_1 接受状态　q_2



$$\upsilon(q_0, 0) = q_0$$
$$\upsilon(q_0, 1) = q_1$$
$$\upsilon(q_1, 0) = q_2$$
$$\upsilon(q_1, 1) = q_1$$
$$\upsilon(q_2, 0) = q_2$$
$$\upsilon(q_2, 1) = q_2$$

$$DFA\ (\{q_0, q_1, q_2\}, \{0, 1\}, \upsilon, q_0, \{q_1\})$$

DFA表示：代数、转移图、转移表

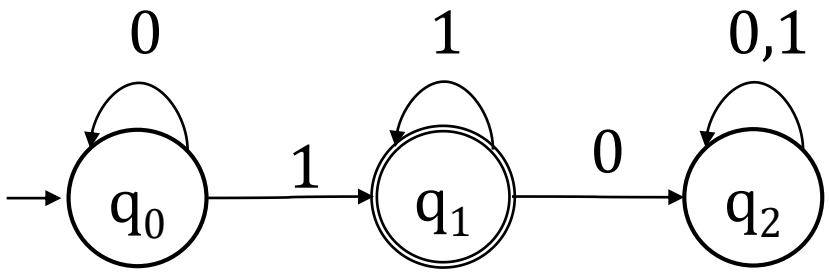
DFA $A = (\{q_0, q_1, q_2\}, \{0, 1\}, \upsilon, q_0, \{q_1\})$

$\upsilon = ((q_0, 0), q_0), ((q_0, 1), q_1), ((q_1, 0), q_2), ((q_1, 1), q_1), ((q_2, 0), q_2), ((q_2, 0), q_2))$

DFA $A = (Q, \Sigma, \upsilon, q_0, F)$，其中，

$Q = \{q_0, q_1, q_2\}$，$\Sigma = \{0, 1\}$，

$\upsilon = \ldots$，$q_0 = q_0$，$F = \{q_1\}$



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

$\mathcal{G}(A) = (Q, E, \Sigma)$

$E = \{(p, q, a) \mid p, q \in Q, a \in \Sigma, q = \upsilon(p, a)\}$

$\mathbb{T}[ind(q), 0]$

$\mathbb{T}[0, ind(a)]$

$\mathbb{T}[ind(q), ind(a)]$

三种DFA表示都是等价的

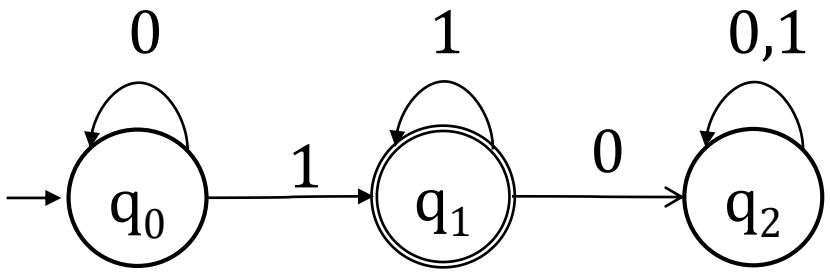
代数 A	*转移图* $\mathcal{G}$(A)	*转移表* $\mathbb{T}$
Q元素q	顶点(圆圈)	$\mathbb{T}[ind(q), 0]$
Σ元素a	权(弧标记)	$\mathbb{T}[0, ind(a)]$
υ元素	边(转移弧)	$\mathbb{T}[1..\|Q\|, 1..\|\Sigma\|]$元素
q_0	箭头标记顶点	箭头标记$\mathbb{T}[1, 0]$
F元素	双圈顶点	*标记$\mathbb{T}[1..\|Q\|, 0]$元素

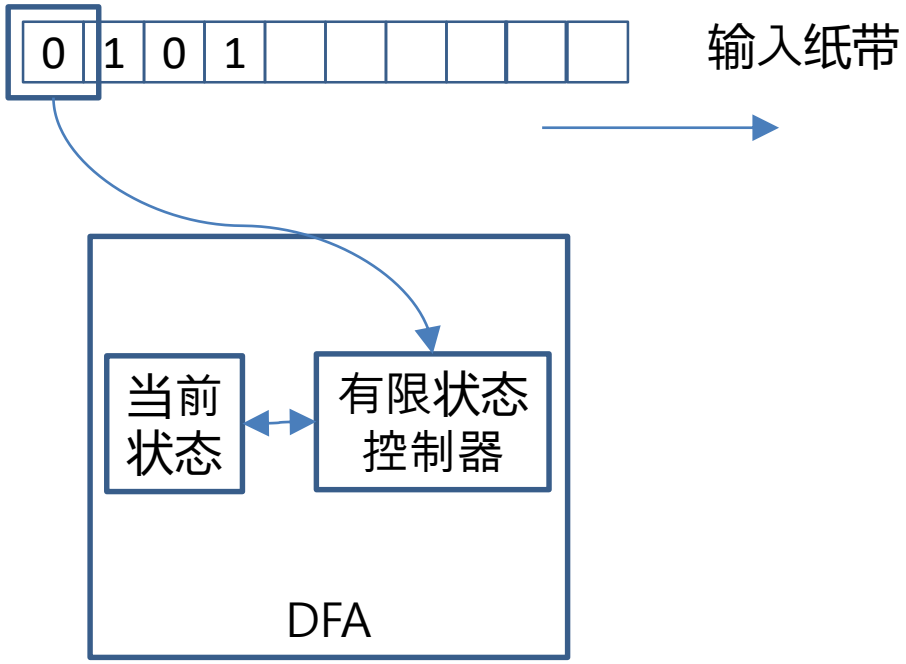
代数系统：集合Q、Σ、运算υ、封闭性

DFA的判定性质: DFA接受、拒绝输入串

- 0、1、00、01、000、001、010、0101

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





(当前状态, 当前输入符号) -> 当前状态

用算法描述DFA的判定性质(运行过程)

输入：DFA $A=(Q, \Sigma, \upsilon, q_0, F)$

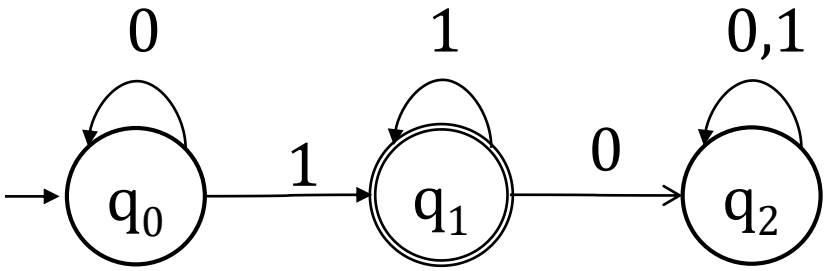
输入：给定输入串$w \in \Sigma^*$

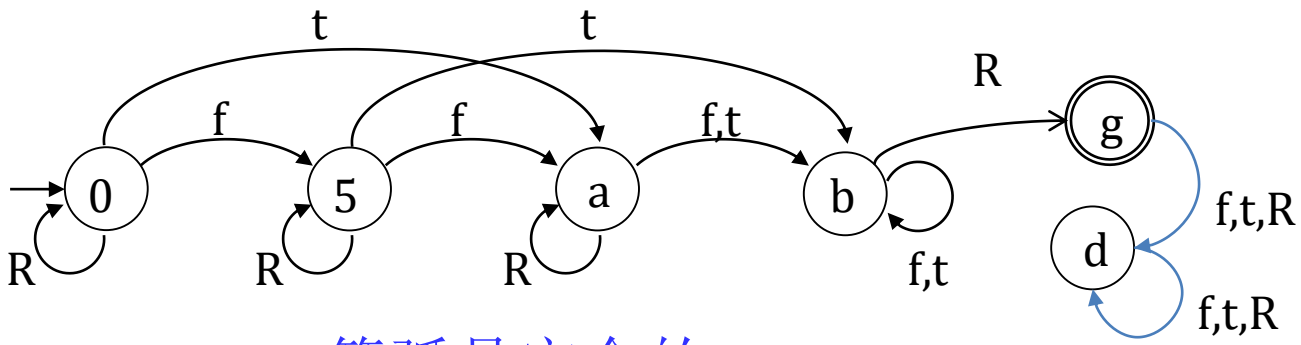
输出：dfa(w)为真接受否则拒绝

```
int dfa(w) {
q=q0;                    //当前状态
x=w;                     //剩余串
while (x != ε) {
        x=ay;            //当前输入符号a
        q= υ(q, a);      //状态转移
        x=y;             //修改剩余串
        }
return q∈F;              //为真表示接受
}
```

习惯约定：w, x, y, … 是串， a, b, c,… 是符号。

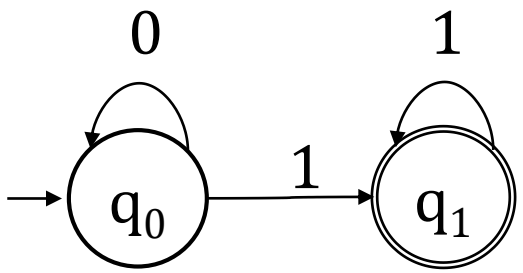
简化的DFA（DFA的一般形式）

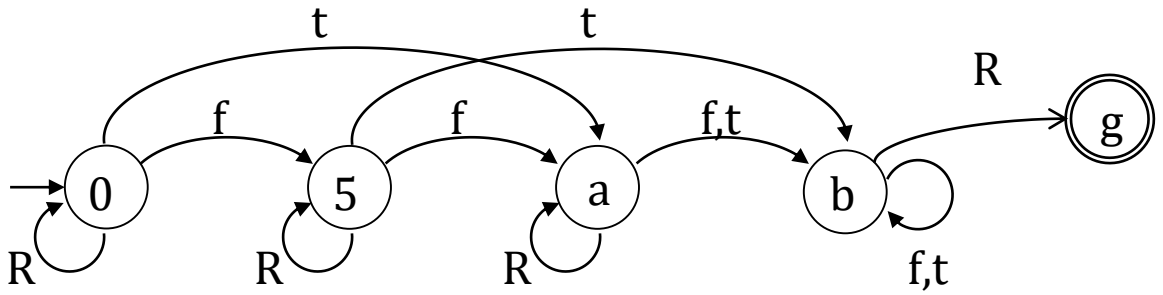




箭弧是完全的

即，υ是映射





箭弧允许有缺失

即，υ是关系

DFA的通用判定算法

- 算法：DFA的判定性质

输入：DFA $A=(Q, \Sigma, \upsilon, q_0, F)$

输入：给定输入串$w \in \Sigma^*$

输出：dfa(w)为真接受否则拒绝

$DFA\ A = (Q, \Sigma, \upsilon, q_0, F)$，

完全的：$\forall q \in Q \cdot \forall a \in \Sigma \cdot \upsilon(q, a) \in Q$，

简化的：$\forall q \in Q \cdot \forall a \in \Sigma \cdot \upsilon(q, a) \in Q \cup \{\perp\}$，

为了描述方便，υ 的值为$\perp$表示无定义。

```
int dfa(w) {
q=q0;
x=w;
while (x != ε) {
        x=ay ;
        //自动机突然死亡以示拒绝
        if((q=υ(q, a))∉Q)return 0;
        x=y;
        }
return q∈F;
}
```

形式描述DFA接受串w

- DFA $A=(Q, \Sigma, \upsilon, q_0, F)$ 接受 w 如果对于$r_0, r_1, ..., r_n \in Q$ 满足:
  - $r_0 = q_0$
  - $\upsilon(r_i, w_{i+1}) = r_{i+1}$ 其中 $i = 0, ..., n-1$
  - $r_n \in F$
- 为方便描述，使用路径术语：
  - 路径起点（始端）r_0，终点（末端）r_n，路径标记w
  - 路径隐含$\upsilon(r_i, w_{i+1}) = r_{i+1}$ 其中 $i = 0, ..., n-1$
- 何时不接受（拒绝）？
  - 不存在末端为接受状态的路径（路径长度为|w|)
  - 或自动机突然死亡（路径长度<|w|且以w的一个前缀为标记)

扩展转移函数

- 为了方便表示连续的转移，扩展转移函数υ 为 $\tilde{\upsilon}$
- 归纳定义$\tilde{\upsilon}$：

基础： $\tilde{\upsilon}(q, \varepsilon) = q$

归纳： $\tilde{\upsilon}(q, wa) = \upsilon(\tilde{\upsilon}(q, w), a)$

- $\tilde{\upsilon}(q, w)$表示DFA中存在唯一一条始端为q末端为$\tilde{\upsilon}(q, w)$的标记为w的路径。
- 对于简化DFA， $\tilde{\upsilon}(q, w)$的值又可能为$\perp$。

对一个 DFA，

给定状态q，

给定输入串$w=a_1a_2...a_n$，

$\tilde{\upsilon}$ 是这样计算的：

从q开始依次选择标记为$a_1, a_2, ..., a_n$的弧走过一条路径，路径末端就是$\tilde{\upsilon}(q, w)$。

扩展转移函数举例

$\tilde{\upsilon}(q_0, 011)$
$= \upsilon(\tilde{\upsilon}(q_0, 01), 1)$
$= \upsilon(\upsilon(\tilde{\upsilon}(q_0, 0), 1), 1)$
$= \upsilon(\upsilon(\upsilon(\tilde{\upsilon}(q_0, \varepsilon), 0), 1), 1)$
$= \upsilon(\upsilon(\upsilon(q_0, 0), 1), 1)$
$= \upsilon(\upsilon(q_0, 1), 1)$
$= \upsilon(q_1, 1)$
$= q_1$

	0	1
$\rightarrow{}^*q_0$	q_0	q_1
$^*q_1$	q_2	q_1
q_2	q_2	q_2

$\tilde{\upsilon}(q_0, 101)$
$= \upsilon(\tilde{\upsilon}(q_0,10), 1)$
$= \upsilon(\upsilon(\tilde{\upsilon}(q_0, 1), 0), 1)$
$= \upsilon(\upsilon(q_1, 0), 1)$
$= \upsilon(q_2, 1)$
$= q_2$

	0	1
$\rightarrow{}^*q_0$	q_0	q_1
$^*q_1$	$\perp$	q_1

$\tilde{\upsilon}(q_0, 101) =$
$= \upsilon(\tilde{\upsilon}(q_0, 10), 1)$
$= \upsilon(\upsilon(\tilde{\upsilon}(q_0, 1), 0), 1)$
$= \upsilon(\upsilon(q_1, 0), 1)$
$= \upsilon(\perp, 1)$
$= \perp$

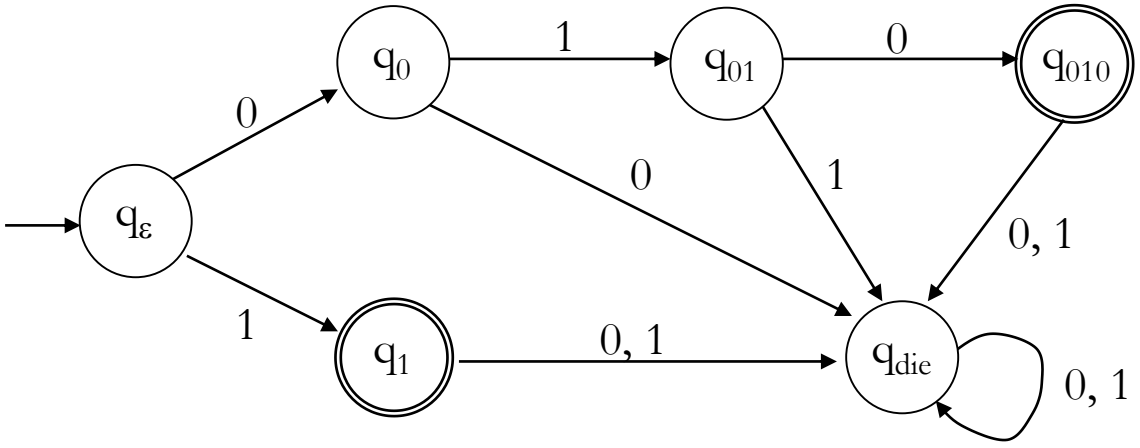
DFA状态的可达性

- 对于$q \in Q$，不存在始端为q_0末端为q的路径，则q是从初始状态不可达的。如果DFA有这样的状态q，则称其具有可达性情形①。
- 对于$q \in Q$，$q_1 \in F$，不存在始端为q末端为q_1的路径，则q是不能到达接受状态的。如果DFA有这样的状态q，则称其具有可达性情形②。
- 对于$q \in Q$，$q_1 \in F$，同时存在始端为q_0末端为q的路径、始端为q末端为q_1的路径。如果DFA有这样的状态q，则称其具有可达性情形③。

①$\forall w \in \Sigma^* \cdot q \neq \tilde{\upsilon}(q0, w)$

②$\forall p \in F, w \in \Sigma^* \cdot p \neq \tilde{\upsilon}(q, w)$

③$\exists p \in F, \exists x,y \in \Sigma^* \cdot q = \tilde{\upsilon}(q0, x) \wedge p = \tilde{\upsilon}(q, y)$



DFA定义语言

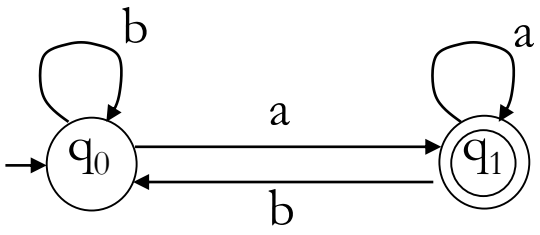
- DFA $A = (Q, \Sigma, \upsilon, q_0, F)$
- $L(A) = \{w \in \Sigma^* \mid \tilde{\upsilon}(q_0, w) \in F\}$

- DFA定义的语言是正则语言
- 语言是正则的如果它是某个DFA的语言

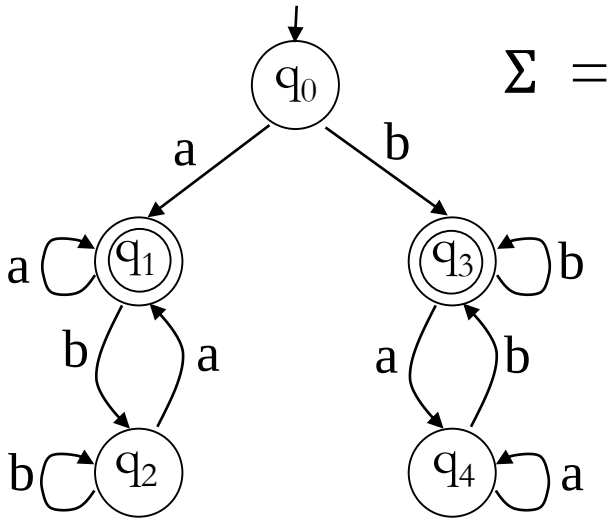
DFA的语言示例

- DFA $M = (Q, \Sigma, \upsilon, q_0, F)$
- $L(M) = \{w \in \Sigma^* \mid \tilde{\upsilon}(q_0, w) \in F\}$

$\Sigma = \{a, b\}$



$\Sigma = \{a, b\}$

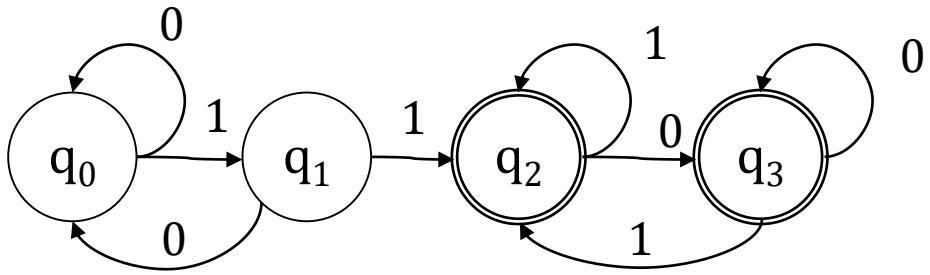


{0,1}上的以11为前缀的串

	0	1
→q0	⊥	q1
q1	⊥	q2
*q2	q2	q2

{0,1}上的含有子串11的串

	0	1
→q_0	q0	q_1
q_1	q_0	q_2
*q_2	q_3	q_2
*q_3	q_3	q_2

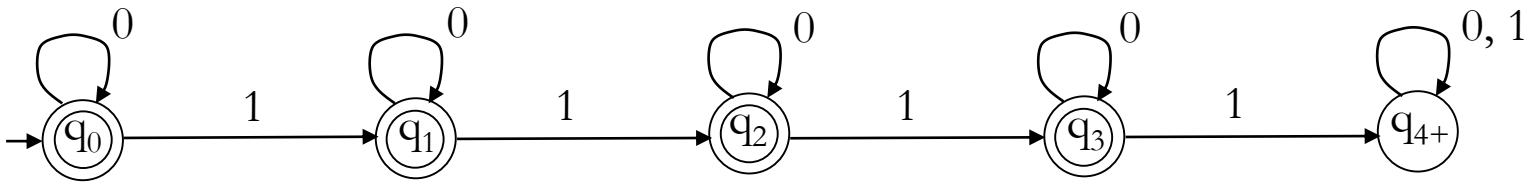


图(a)

	0	1
→q_0	q_0	q_1
q_1	q_0	q_2
*q_2	q_2	q_2

{0,1}上最多有三个1的串

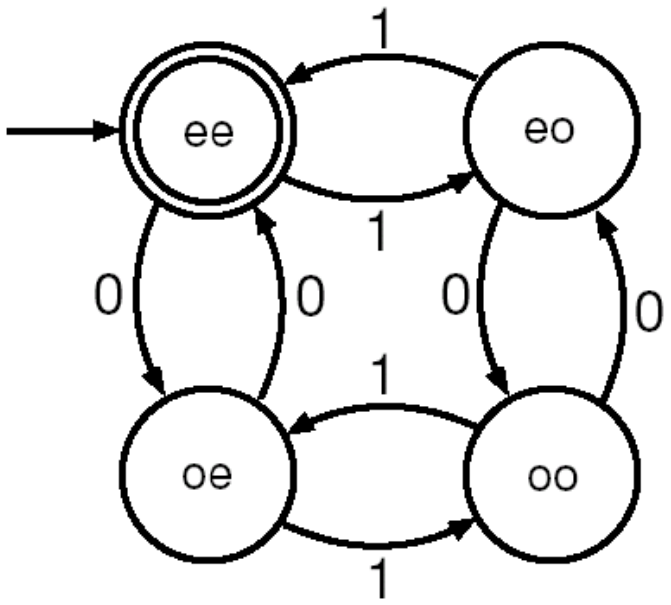
- 构造字母表{0, 1}上的DFA 接受所有最多含有三个1的串



正则语言的例子

- {0,1}上的，含有偶数个0和偶数个1 的串的全集，是正则语言。

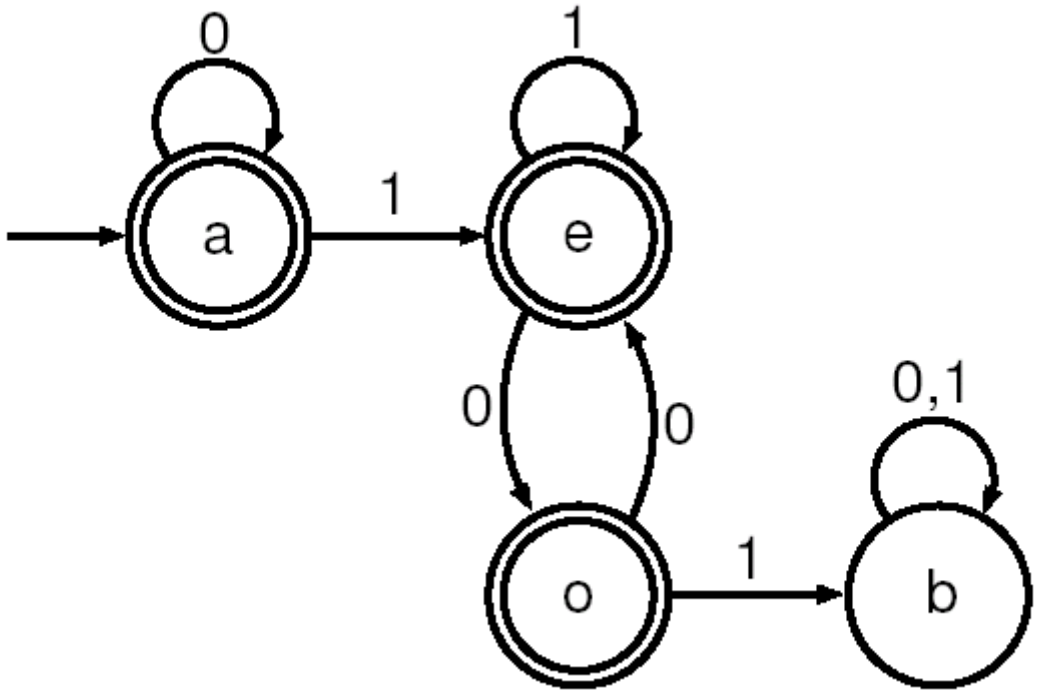
思路：输入串已消耗掉的部分只有四种情况，分别用状态来记住，继续读入符号将继续发生状态转移，当输入串消耗完以后，所达到的状态就决定了原输入串是什么情况，比如偶数个0和偶数个1



正则语言的例子

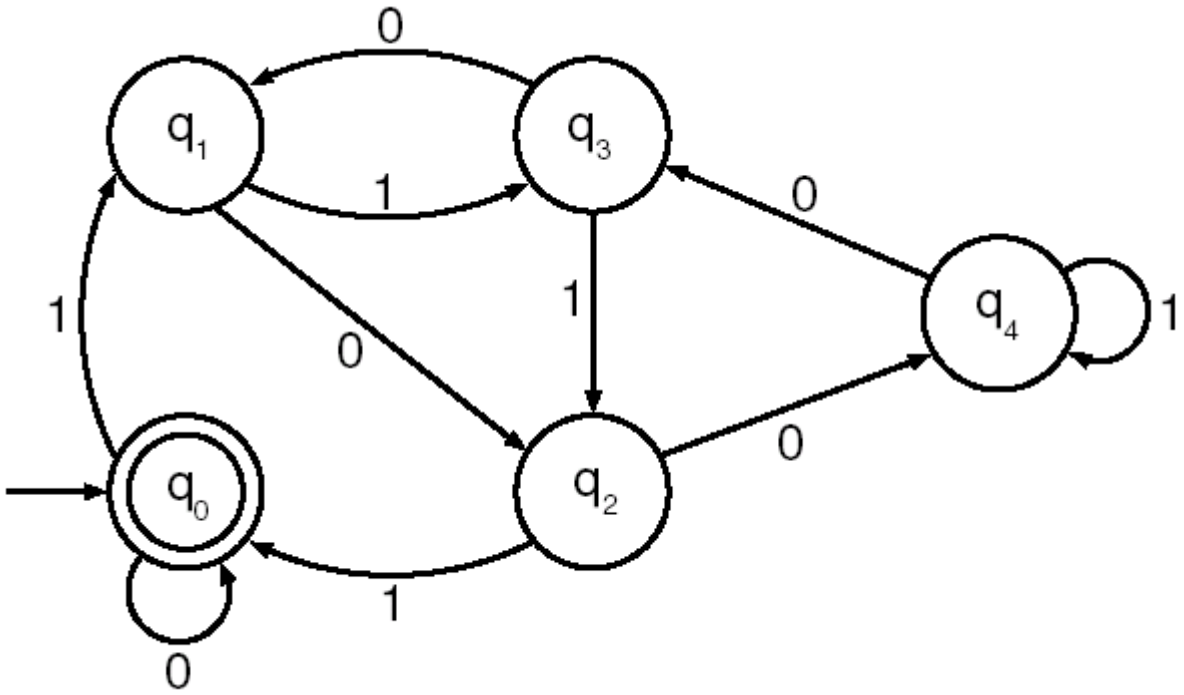
- {0,1}上不包含一对1且中间被奇数个0隔开的任何串。

  思路：排除包含模式10...01的串，该模式中的0为奇数个



正则语言的例子

- 令L={w ∈{0,1}*|w 看作二进制数的话能被5整除}



分析：一个数除以5，其余数只能是0，…，4五种，因此我们以q_0，…，q_4分别表示这五种状态。因为接受能被5整除的数，故状态0既为初始状态，又为终结状态。接着，考虑二进制数在其串后增添0或1时，状态的转化情况。在二进制串后添1位，即可理解为将先前的串值乘以二再加上所添的数值。那么，串尾添数后新的数值模5的余数便可以计算出来。即可以得到添0或1后的新的状态。

归纳{0,1}上语言的DFA设计题目

- 前缀有这个、没这个、同时有这两个、或者有这个或者那个；
- 子串同上；
- 后缀同上；
- 汇总性：有n个1、没有n个1、有n个1和m个0、没有n个1同时也没有m个0；
- 计算性：被5整除、是素数、是奇数、是偶数

- 整数、定点数

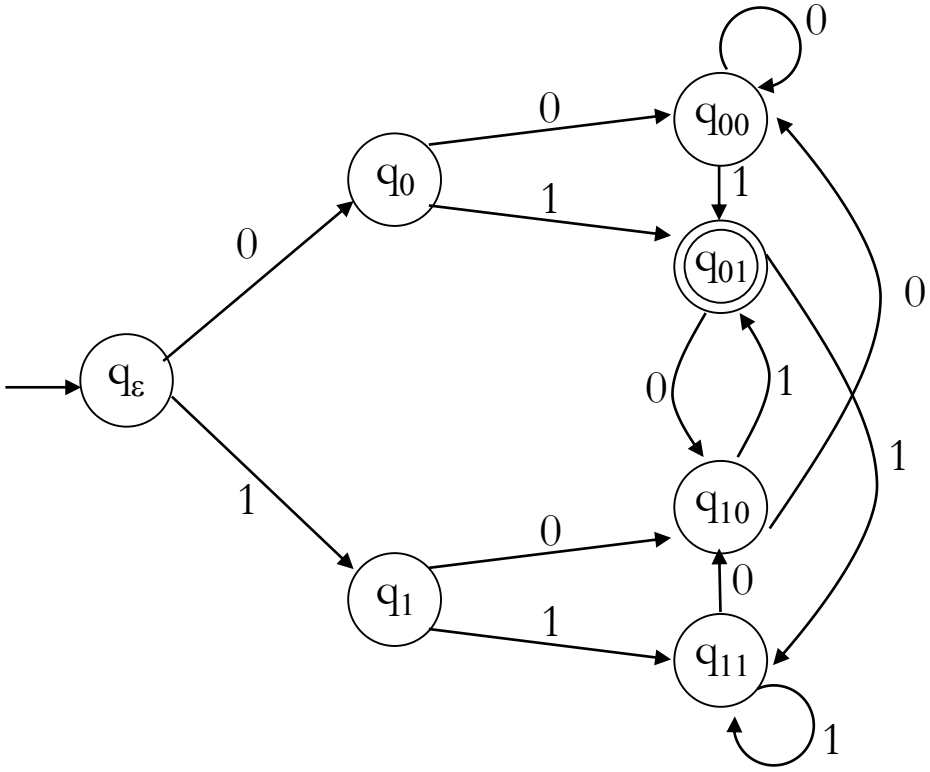
- 对题目要求：典型且简单

用状态作记忆

- {0, 1}上以01结尾的串

用四个状态记住输入串的前缀的最后两位（已消耗串的最后两位）。

q_{00}
q_{01}
q_{10}
q_{11}



用状态作记忆局限性

- {0, 1}上以101结尾的串

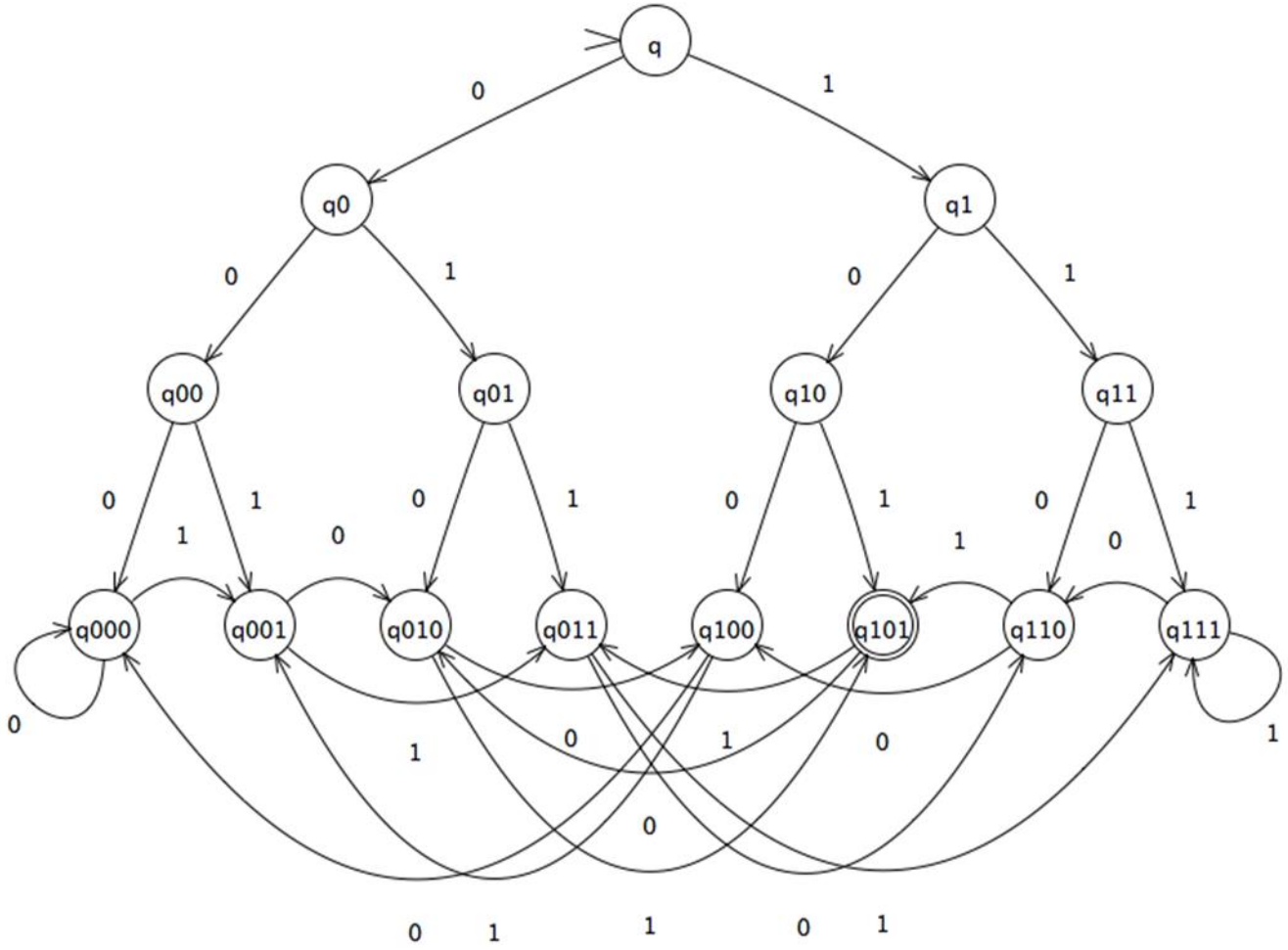
用八个状态记住已消耗串的最后三位。

q_{000}

q_{001}

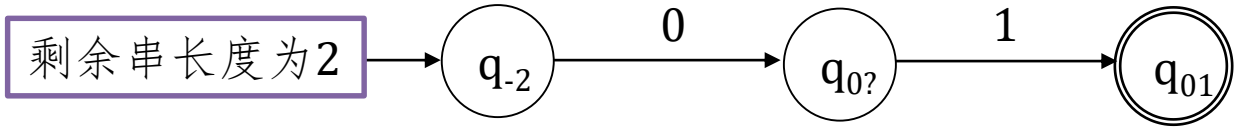
…

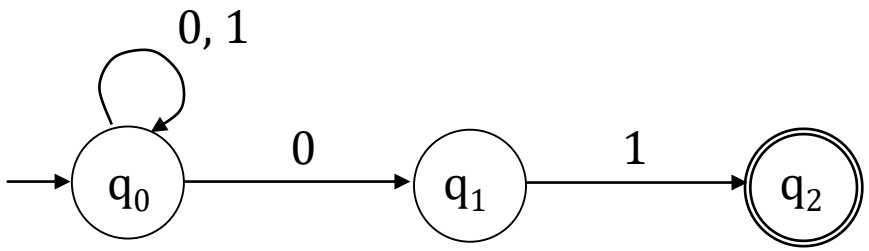
q_{111}



引入不确定性来避免状态爆炸

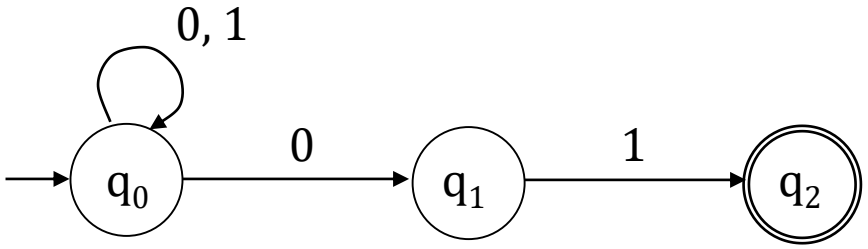
- 猜测

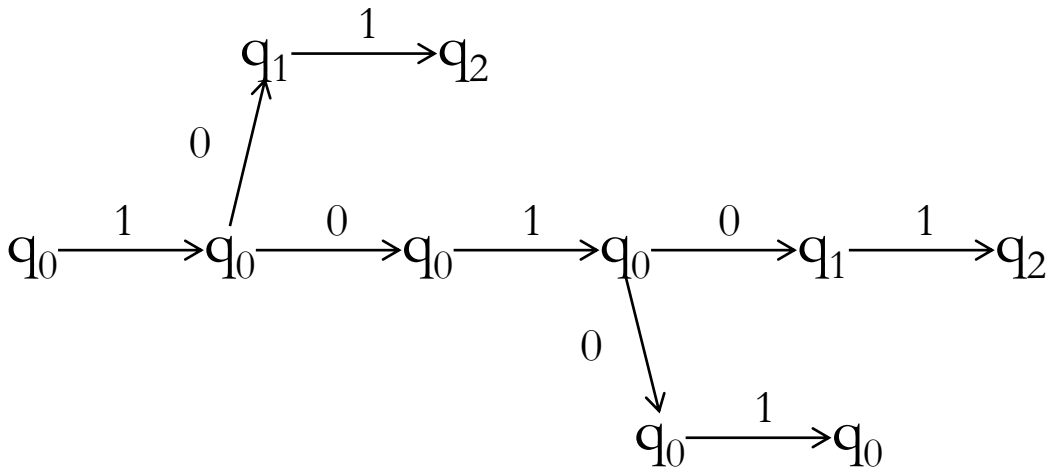




穷举与猜测

- 输入串10101





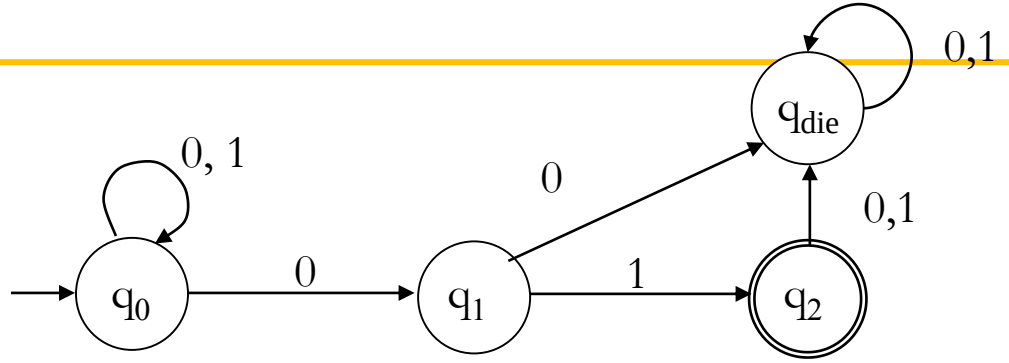
死亡

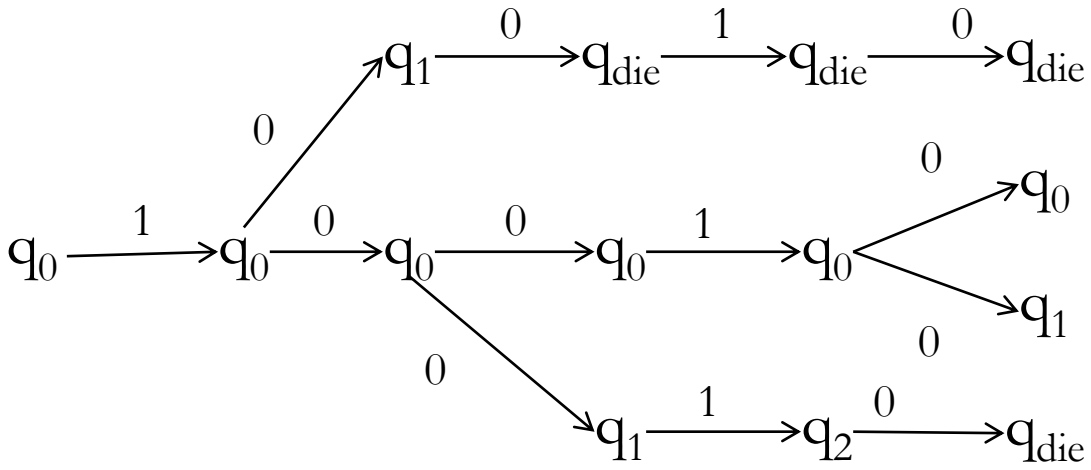
接受

拒绝

穷举与猜测

- 输入串10010





陷入

拒绝

拒绝

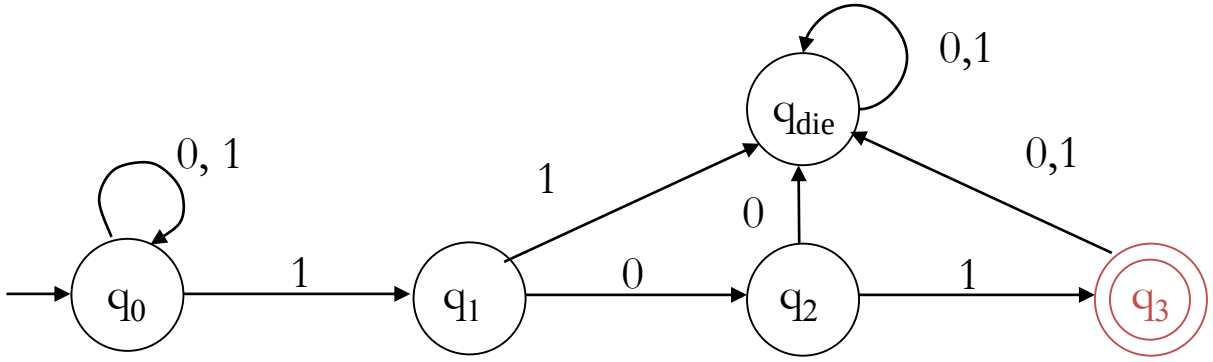
线索穷举

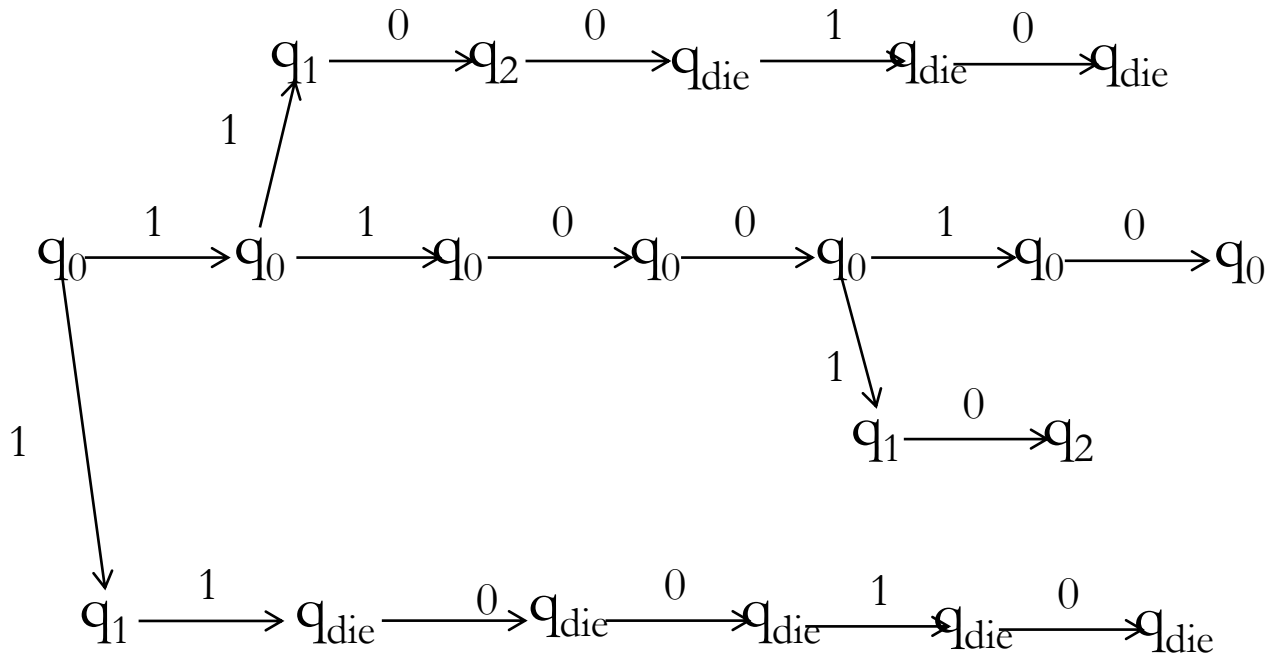
- 对于输入串，线索是自动机的一条以初始状态为起点的状态转移路径，该路径正好消耗掉输入串的某个前缀。线索穷举是有穷自动机的运行方式
  - DFA运行时只有一条线索（路径）。
  - 非确定有穷自动机（NFA）运行时有有穷个线索。
- 对于给定的输入串w，只要有一个线索成功，则w被自动机接受，否则不被接受
  - 成功的线索：始端为初始状态，标记为w，末端为接受状态。
- 在前例中，
  - 以101结尾的简化型NFA对于输入串110101有5个线索，其中有一个成功。
  - 以101结尾的完全型NFA对于输入串110010有4个线索，均不成功。
  - 以101结尾的完全型NFA对于输入串110101有几个个线索？成功者为何？

NFA的线索穷举成为一个树

- w为输入串，w的第i（i≥0）个前缀就是长度为i的前缀，记为i-前缀
- w=hello-world
- w前缀：ε，h，he，hel，hell，hello，hello-，hello-w，...，hello-world
- w有|w|+1个前缀
- 给定输入串的线索穷举树▶
- 根为初始状态，层数为0
- 从根到每一个叶子的路径都是一条线索（每个线索都在树中恰有一条从根到叶子的路径对应）
- 第i层（i>0）的状态集合=Ti
- 以根为起点，无论沿着那条线索，都消耗掉输入串的i-前缀并转移到Ti中的状态
- Ti就是活动状态集，也是w的i-前缀的当前状态集。

线索穷举树举例

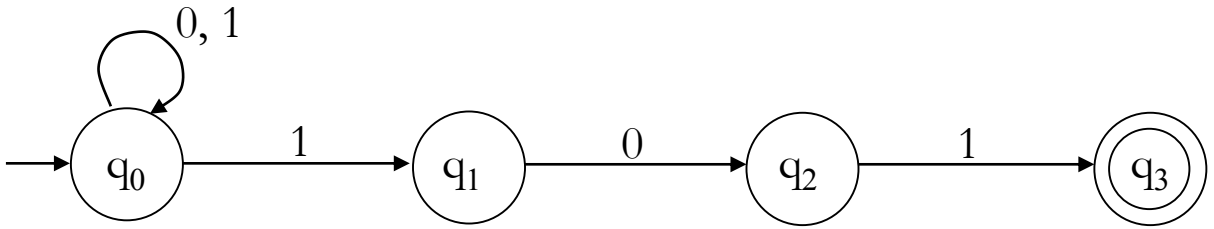


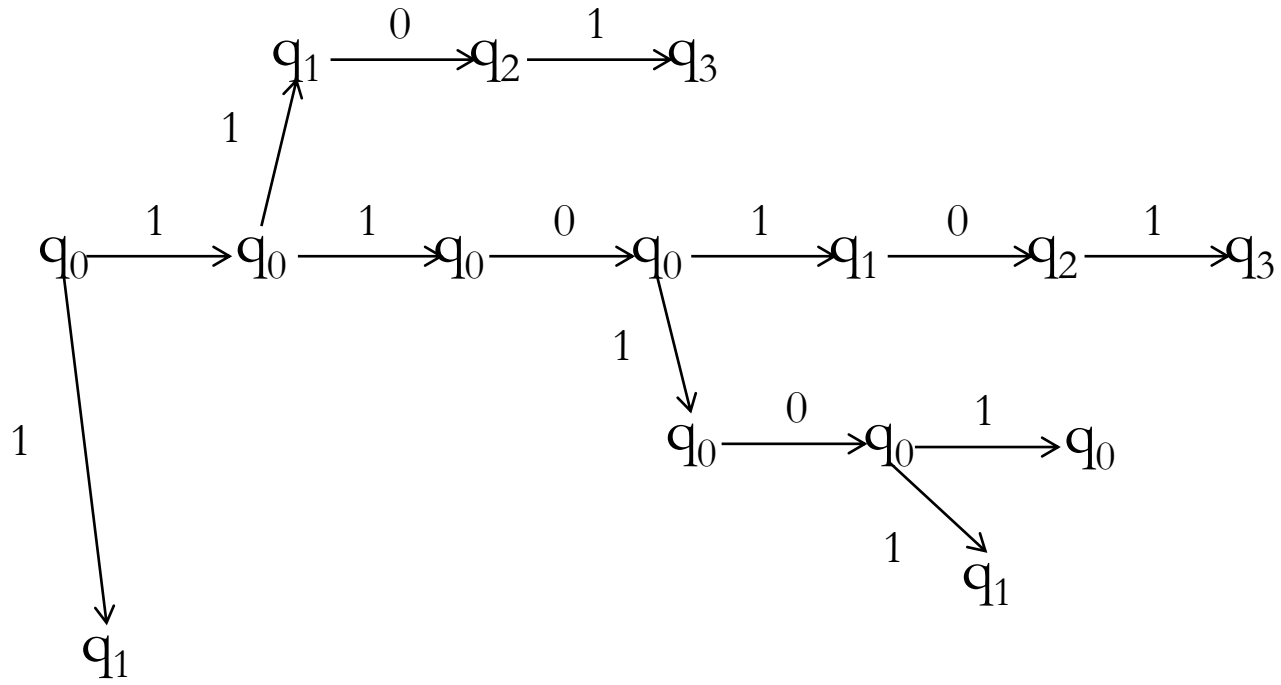


输入串: 110010

i	T	i-前缀
0	$\{q_0\}$	ε
1	$\{q_0,q_1\}$	1
2	$\{q_0,q_1,q_{die}\}$	11
3	$\{q_0,q_2,q_{die}\}$	110
4	$\{q_0,q_{die}\}$	1100
5	$\{q_0,q_1,q_{die}\}$	11001
6	$\{q_0,q_2,q_{die}\}$	110010

线索穷举树举例





输入串: 110101

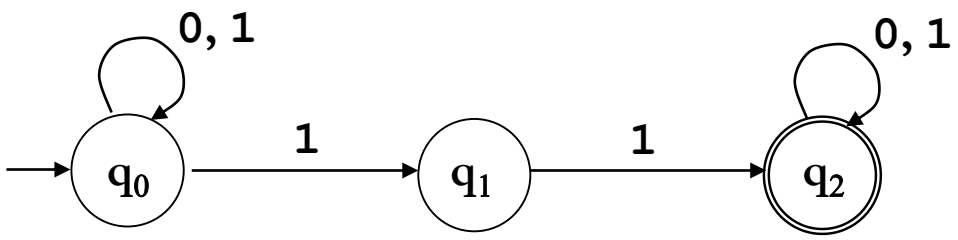
i	T	i-前缀
0	$\{q_0\}$	ε
1	$\{q_0,q_1\}$	1
2	$\{q_0,q_1\}$	11
3	$\{q_0,q_2\}$	110
4	$\{q_0,q_1,q3\}$	1101
5	$\{q_0,q2\}$	11010
6	$\{q_0,q_1,q3\}$	110101

NFA例

- 构造字母表{0, 1}上的NFA 接受含有子串 11的串。

10011010, 11001, 11101　　接受

ε, 000, 010101　　拒绝

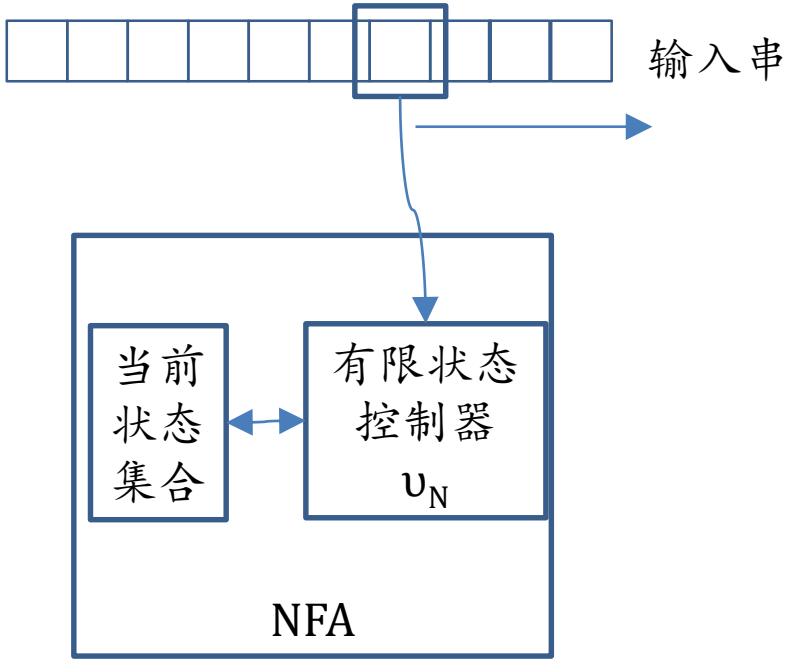


NFA与DFA有什么不同?

- DFA：从同一个状态射出的带有相同标记的箭弧最多有一条。
- NFA：从同一个状态射出的带有相同标记的箭弧可以有多条（NFA）。
- ε-NFA：有ε-转移的NFA。

- ε-转移：状态间的ε-转移不消耗输入符号。

- 本章2.3小节介绍NFA
- 本章2.4小节介绍ε-NFA
- 之后，二者不再区分，都是NFA



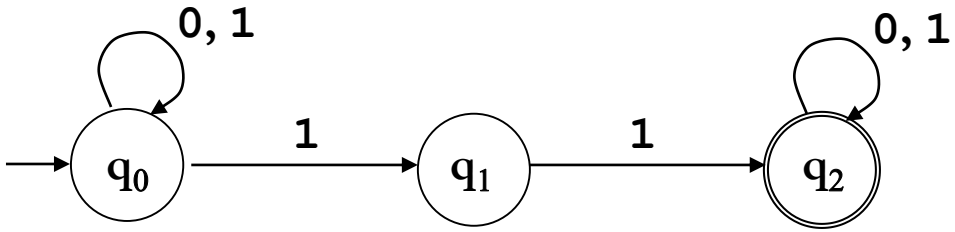
表示NFA：代数、转移图、转移表

- NFA $A=(Q, \Sigma, \upsilon, q_0, F)$
  - 状态集 $Q=\{q_0, q_1, q_2\}$
  - 字母表 $\Sigma=\{0, 1\}$
  - 转移函数υ: $Q\times\Sigma\rightarrow 2^Q$
  - 初始状态$q_0\in Q$
  - 接受状态集合 $F=\{q_2\}\subseteq Q$

- $\upsilon=\{((q_0,0),\{q_0\}), ((q_0,1),\{q_0,q_1\}),$
  $((q_1,0),\{\}),((q_1,1),\{q_2\}),$
  $((q_2,1),\{q_2\}),((q_2,1),\{q_2\})\}$
- NFA $A=(\{q_0,q_1,q_2\}, \{0,1\}, \upsilon, q_0, \{q_2\})$

两种定义：转移函数υ是映射； υ允许$\perp$值

用$\upsilon(q, a)=\perp$表示$\upsilon(q, a)$无定义。



	0	1
$\rightarrow q_0$	$\{q_0\}$	$\{q_0, q_1\}$
q_1	φ	$\{q_2\}$
$*q_2$	$\{q_2\}$	$\{q_2\}$

NFA的判定性质

- 输入：NFA $A=(Q, \Sigma, \upsilon, q_0, F)$，$w \in \Sigma^*$
- 输出：接受、拒绝w

```
int nfa(w) {
T = {q0};                    //活动状态集初始化
x = w;                       //剩余串初始化
while (x != ε) {
        x = ay;              //求出当前输入符号a
        T = Uq∈T · υ(q, a);  //更新活动状态集
        x = y;}              //更新剩余串
return T∩F != φ;             //活动状态集中若有接受状态则接受
}
```

NFA的扩展转移函数

- 为了方便表示读入一个串所发生的连续转移，扩展转移函数υ为υ̃，υ̃ 被称为扩展转移函数。

- 归纳方式定义为：
- 基础： $\tilde{\upsilon}(q, \varepsilon) = \{q\}$
- 归纳： $\tilde{\upsilon}(q, wa) = \bigcup_{p \in \tilde{\upsilon}(q, w)} \upsilon(p, a)$

q 是NFA的状态，
对于串$w=a_1a_2...a_n$，
$T_0=\{q\}$，
$T_i=\bigcup_{p \in T_{i-1}} \upsilon(p, a_i)$, $1 \le i \le n$，
那么$\tilde{\upsilon}(q, w)=T_n$

- <u>$\tilde{\upsilon}(q, w)$：对于始端为q，标记为w的所有路径，由它们的末端组成的集合。</u>

例2-12的示例

$\tilde{\upsilon}(q_0, 01011) = \cup p \in \tilde{\upsilon}(q_0, 0101) \cdot \upsilon(p, 1)$

$= \cup p \in [\ \cup p \in \tilde{\upsilon}(q_0, 010) \cdot \upsilon(p, 1)\ ] \cdot \upsilon(p, 1)$

$= \cup p \in [\ \cup p \in [\ \cup p \in \tilde{\upsilon}(q_0, 01) \cdot \upsilon(p, 0)\ ] \cdot \upsilon(p, 1)\ ] \cdot \upsilon(p, 1)$

$= \cup p \in [\ \cup p \in [\ \cup p \in [\ \cup p \in \tilde{\upsilon}(q_0, 0) \cdot \upsilon(p, 1)\ ] \cdot \upsilon(p, 0)\ ] \cdot \upsilon(p, 1)\ ] \cdot \upsilon(p, 1)$

$= \cup p \in [\ \cup p \in [\ \cup p \in [\ \cup p \in [\ \cup p \in \tilde{\upsilon}(q_0, \varepsilon) \cdot \upsilon(p, 0)\ ] \cdot \upsilon(p, 1)\ ] \cdot \upsilon(p, 0)\ ] \cdot \upsilon(p, 1)\ ] \cdot \upsilon(p, 1)$

$= \cup p \in [\ \cup p \in [\ \cup p \in [\ \cup p \in [\ \cup p \in \{q_0\} \cdot \upsilon(p, 0)\ ] \cdot \upsilon(p, 1)\ ] \cdot \upsilon(p, 0)\ ] \cdot \upsilon(p, 1)\ ] \cdot \upsilon(p, 1)$

$= \cup p \in [\ \cup p \in [\ \cup p \in [\ \cup p \in \{q_0, q_1\} \cdot \upsilon(p, 1)\ ] \cdot \upsilon(p, 0)\ ] \cdot \upsilon(p, 1)\ ] \cdot \upsilon(p, 1)$

$= \cup p \in [\ \cup p \in [\ \cup p \in \{q_0, q_2\} \cdot \upsilon(p, 0)\ ] \cdot \upsilon(p, 1)\ ] \cdot \upsilon(p, 1)$

$= \cup p \in [\ \cup p \in \{q_0, q_1\} \cdot \upsilon(p, 1)\ ] \cdot \upsilon(p, 1)$

$= \cup p \in \{q_0, q_2\} \cdot \upsilon(p, 1)$

$= \{q_0\}$

$\tilde{\upsilon}(q_0, \varepsilon) = \{q_0\}$

$\tilde{\upsilon}(q_0, 0) = \cup p \in \tilde{\upsilon}(q_0, \varepsilon) \cdot \upsilon(p, 0) = \{q_0, q_1\}$

$\tilde{\upsilon}(q_0, 01) = \cup p \in \tilde{\upsilon}(q_0, 0) \cdot \upsilon(p, 1) = \upsilon(q_0, 1) \cup \upsilon(q_1, 1) = \{q_0, q_2\}$

$\tilde{\upsilon}(q_0, 010) = \cup p \in \tilde{\upsilon}(q_0, 01) \cdot \upsilon(p, 0) = \upsilon(q_0, 0) \cup \upsilon(q_2, 0) = \{q_0, q_1\}$

$\tilde{\upsilon}(q_0, 0101) = \cup p \in \tilde{\upsilon}(q_0, 010) \cdot \upsilon(p, 1) = \upsilon(q_0, 1) \cup \upsilon(q_1, 1) = \{q_0, q_2\}$

$\tilde{\upsilon}(q_0, 01011) = \cup p \in \tilde{\upsilon}(q_0, 0101) \cdot \upsilon(p, 1) = \upsilon(q_0, 1) \cup \upsilon(q_2, 1) = \{q_0\}$

NFA的语言

- NFA $A=(Q, \Sigma, \upsilon, q_0, F)$的语言，
- $L(A) = \{w \in \Sigma^* \mid \tilde{\upsilon}(q_0, w) \cap F \neq \varphi\}$。
- 这个定义也显示，只要有一个成功线索，就表示接受该输入串。
- 证明例2.12接受{0,1}上后缀为01的串组成的语言
  - 命题一：$w \in \Sigma^*, q_0 \in \tilde{\upsilon}(q_0, w)$；
  - 命题二：$q_1 \in \tilde{\upsilon}(q_0, w)$，当且仅当w以0结尾；
  - 命题三：$q_2 \in \tilde{\upsilon}(q_0, w)$，当且仅当w以01结尾。

小结

- DFA：定义及表示、判定性算法、扩展转移函数、
  语言与DFA构建。
- NFA：定义及表示，判定性算法、扩展转移函数、线索穷举，活动状态集

- 习题
- p50: 2.1~2.4