计算理论

教材:

[S] 唐常杰等译, Sipser著, 计算理论导引(第3版), 机械工业.

参考资料:

[L] Lewis等著, 计算理论基础, 清华大学.

计算理论
第一部分 计算模型

第1章 有限自动机

第3章 图灵机

第二部分 可计算性

第4章 存在没有算法的问题

第三部分 计算复杂性

第7章 P, NP 与NP完全性

计算机的基本能力和局限性是什么?

第1章 有限自动机

常用证明方法

数学归纳法

反证法

鸽巢原理

数学归纳法

命题: 所有的马都是一种颜色.

证明: 我们只需要证明任意n 匹马只有一种颜色.

(i) (初始步) 当 n=1时, 只有一匹马, 马的颜色只有一种.

(ii) (递推步) 假设对正整数k , k 匹马只有一种颜色. 考察 k+1 匹马. 不妨把这些马编号为 1, 2, ..., k, k+1 . 由归纳假设, 去掉编号为2的马后, (1, 3, 4, ..., k+1) 这k匹马只有一种颜色, 于是编号为3, 4, ..., k+1的马与编号为 1的马同色.

同样地由归纳假设, 去掉编号为 k+1 的马后, (1, 2, 3, ..., k)这k 匹马只有一种颜色, 所以编号为2 的马与编号为 1的马同色.

综上, 编号为2, 3, ..., k, k+1的马都与编号为1 的马同色, 因此 k+1匹马的颜色相同. 证毕.

字符串与语言

字母表: 任意一个有限集. 常用记号Σ, Γ.

符号: 字母表中的元素

字符串: 字母表中符号组成的有限序列

如asdf, 通俗地说即单词

串的长度|·|, 例: |abcde|=5

串的连接*, 例: (abc)*(de)=abcde

串的反转R, 例: $(abcde)^R$=edcba

空词: 记为ε, 长度为0

语言: 给定字母表上一些字符串的集合

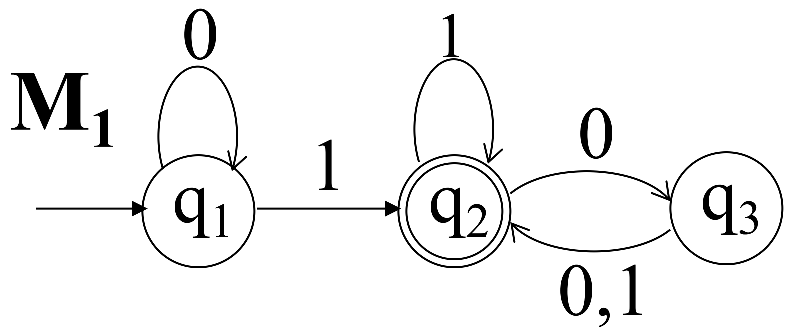
取字母表 Σ = {0,1}, Σ上的语言举例:

A={0,00,0000}, B={0,00,01,000,001,…}

确定型有限(穷)自动机的形式定义

定义: 有限自动机是一个5元组(Q,Σ,δ,s,F),

1) Q是有限集, 称为状态集;
2) Σ是有限集, 称为字母表;
3) $\delta: Q\times \Sigma\rightarrow Q$是转移函数;
4) $s\in Q$是起始状态;
5) $F\subseteq Q$是接受状态集;



- 状态图等价于形式定义

$Q=\{q_1,q_2,q_3\}$, 状态集

$\Sigma=\{0,1\}$, 字母表

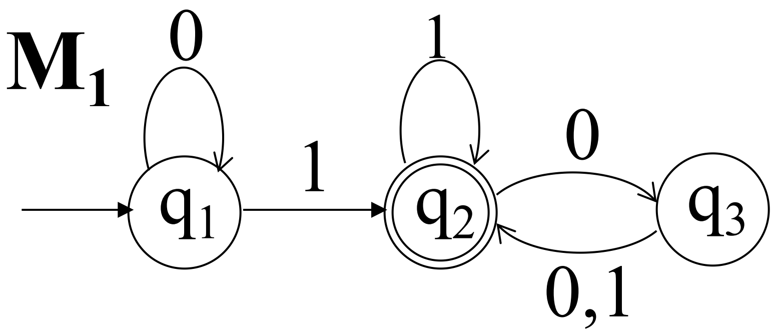
$s=q_1$, 起始状态

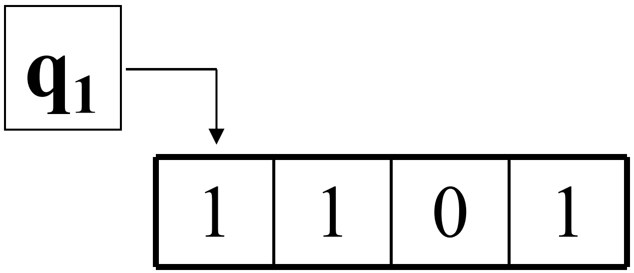
$F=\{q_2\}$接受状态集

δ	0	1
q_1	q_1	q_2
q_2	q_3	q_2
q_3	q_2	q_2

有限自动机

读写头不能改写, 且只能右移





状态: q_1, q_2, q_3　　　　起始状态q_1

接受状态q_2　　　　转移: 箭头

运行: 从起始状态开始沿转移箭头进行.

输出: 输入读完处于接受状态则接受, 否则拒绝.

接受: 1, 11, 100, 101, 1101, …

拒绝: ε, 0, 10, 110, 1010, …

DFA计算的形式定义

设$M=(Q,\Sigma,\delta,s,F)$是一个DFA,

$w=w_1w_2\ldots w_n$是字母表Σ上的一个字符串.

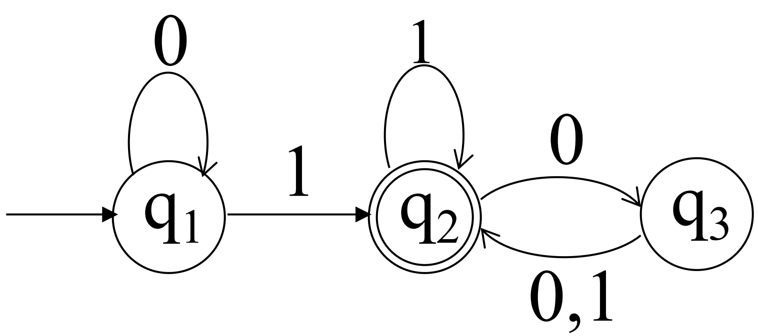
若存在Q中的状态序列$r_0,r_1,\ldots,r_n$, 满足

1) $r_0 = s$;

2) $\delta(r_i, w_{i+1}) = r_{i+1}$;

3) $r_n \in F$

则M接受w.



$$s \xrightarrow{w_1} r_1 \xrightarrow{w_2} r_2 \overset{\cdots}{\dashrightarrow} r_{n-1} \xrightarrow{w_n} r_n$$

有限自动机的语言:正则语言

对有限自动机M, 若 $A = \{ w \in \Sigma^* \mid M接受w \}$,

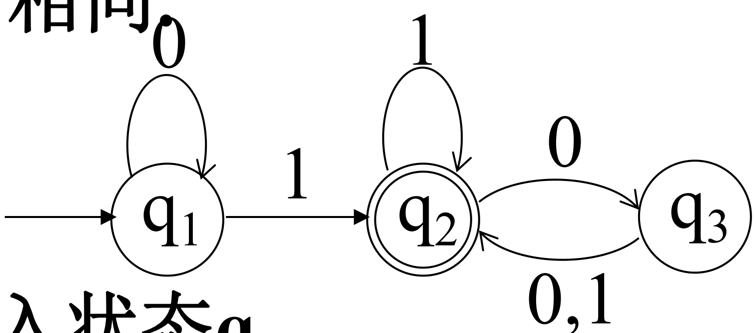
即A是有限自动机**M**的**语言**, 记为L(M)=A, 也称M**识别**A.

注: M的语言唯一. M不识别任何其它语言.

若存在DFA识别语言A, 则称A是**正则语言**.

称两个有限自动机**等价**若它们语言相同.

$\mathbf{M_1}$



注: 在任何状态, 读到1后一定会进入状态q_2.

$L(M_1)$={w | w是0,1串, 至少含一个1,

且最后一个1后面含有偶数个0 }

注: 任何其它语言都不是M_1的语言.

有限自动机的设计(难点)

- 自己即自动机
- 寻找需要记录的关键信息

设计识别下列语言的DFA:

$\{ w \in \{0,1\}^* \mid w$从1开始, 以0结束 $\}$

$\{ w \in \{0,1\}^* \mid w$含有子串1010 $\}$

$\{ w \in \{0,1\}^* \mid w$的倒数第2个符号是1 $\}$

$\{ 0^k \mid k$是2或3的倍数 $\}$

有限自动机的设计

{ w∈{0,1}* | w从1开始, 以0结束 }

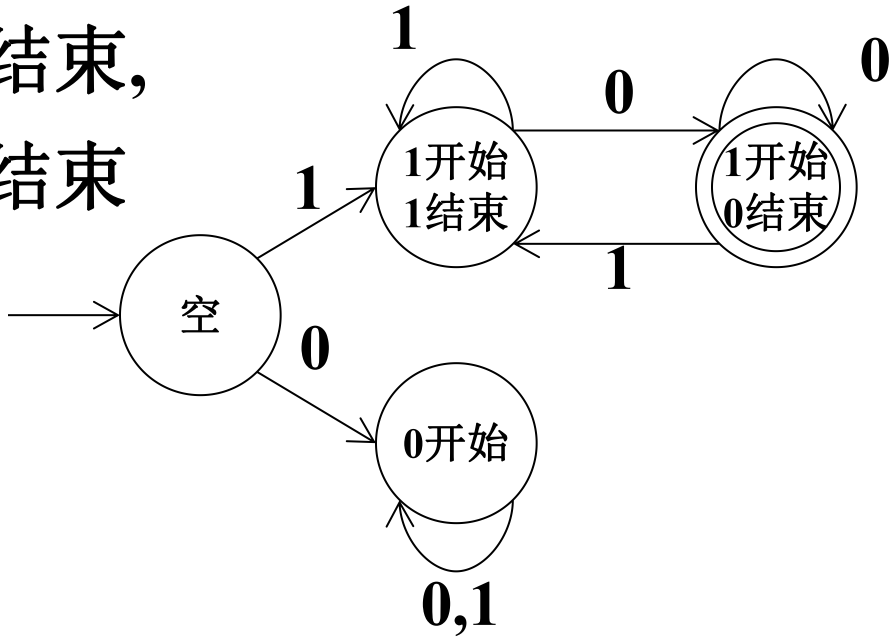
Σ={0,1}, 根据关键信息设计状态,

空,

以0开始,

以1开始以0结束,

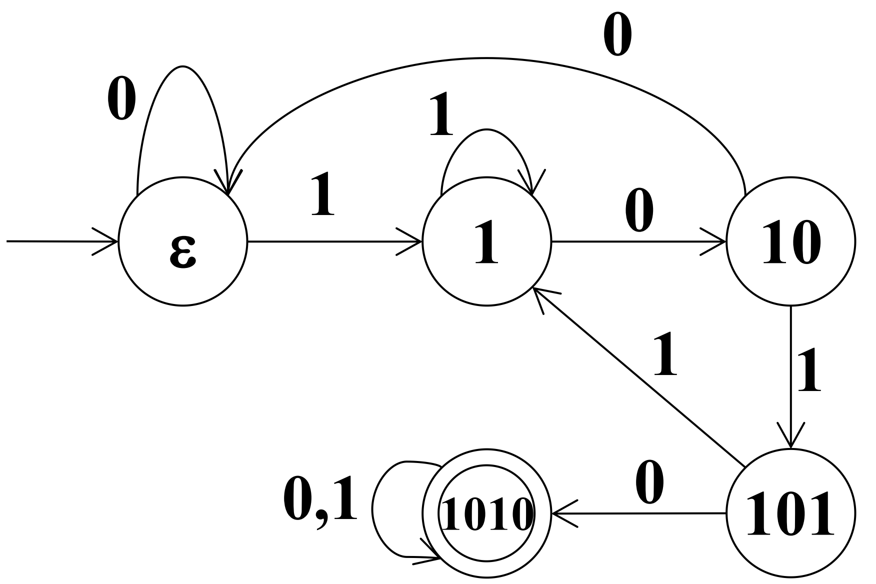
以1开始以1结束



有限自动机的设计

$\{\, w \in \{0,1\}^* \mid w$含有子串$1010 \,\}$

$\Sigma=\{0,1\}$，关键信息: ε, 1, 10, 101, 1010



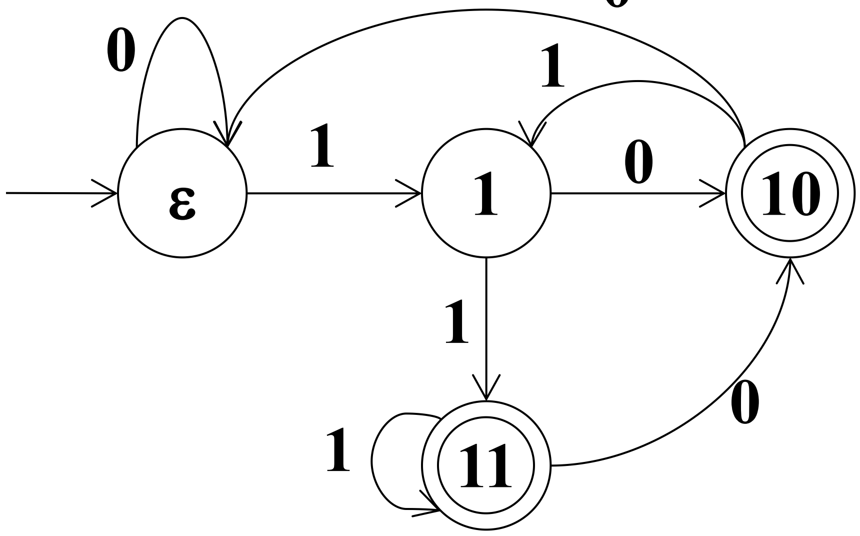
有限自动机的设计

$\{ w \in \{0,1\}^* \mid w$倒数第2个符号是1 $\}$

只需关注最后两个符号

$\Sigma=\{0,1\}$, 关键信息: ε, 0, 00, 1, 01, 10, 11

关键信息改进: ε, 1, 10, 11

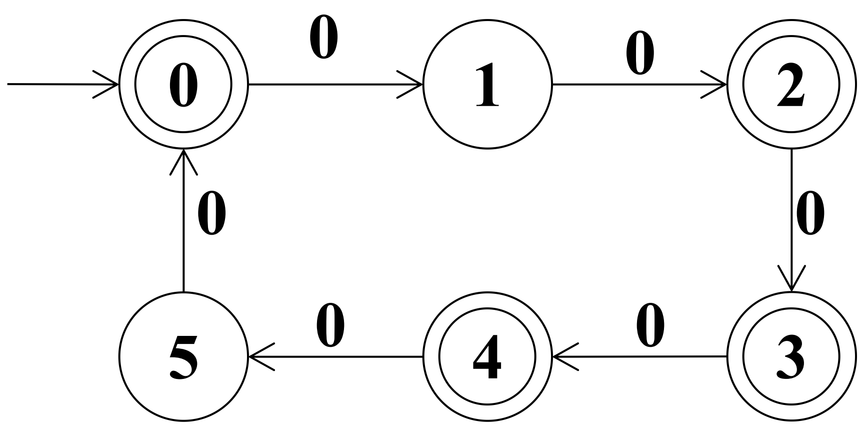


有限自动机的设计

{ 0^k | k是2或3的倍数 }

$\Sigma=\{0\}$, 关键信息: $\varepsilon, 0^1, 0^2, 0^3, 0^4, 0^5$.

记为: 0,1,2,3,4,5

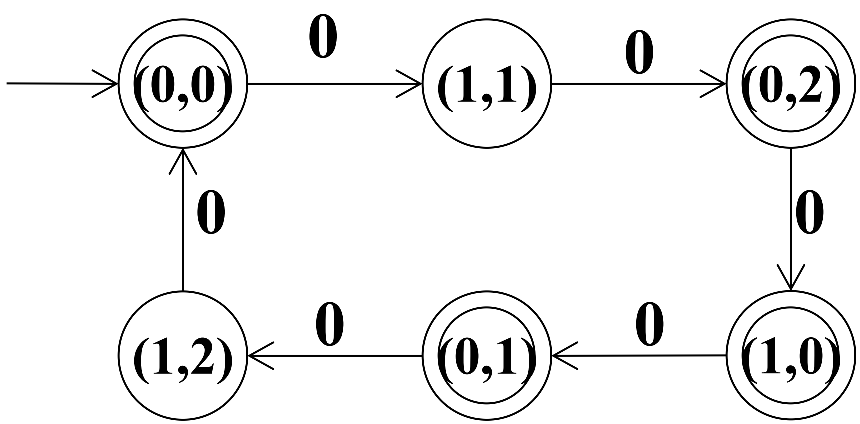


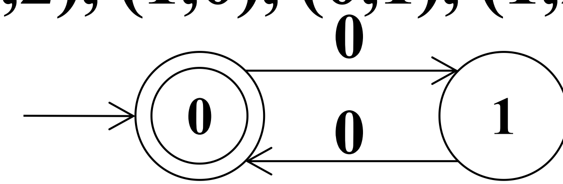
有限自动机的设计

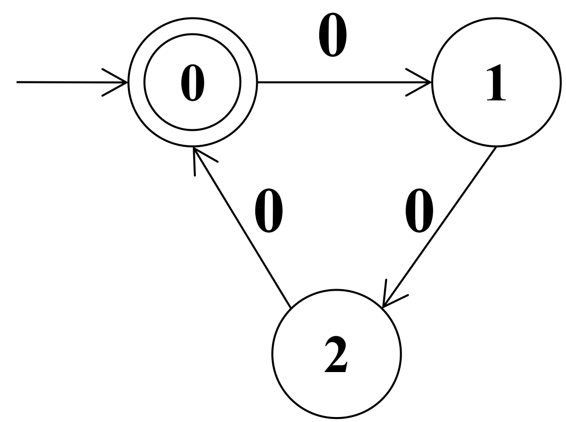
$\{ 0^k \mid k$是2或3的倍数 $\}$

$\Sigma=\{0\}$, 关键信息: $\varepsilon, 0^1, 0^2, 0^3, 0^4, 0^5$,

记为: 0,1,2,3,4,5 或 (0,0), (1,1), (0,2), (1,0), (0,1), (1,2)







$\{0^k|k$是2或3的倍数$\} = \{0^k|k$是2倍数$\} \cup \{0^k|k$是3的倍数$\}$

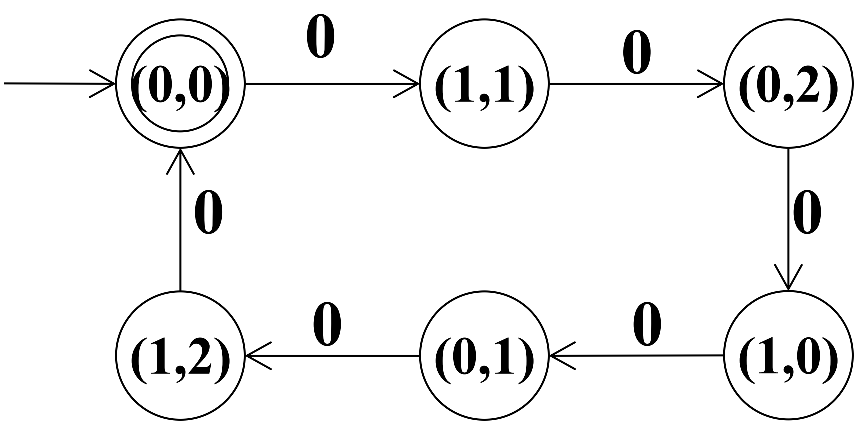
$\{0^k|k$是2和3的倍数$\} = \{0^k|k$是2倍数$\} \cap \{0^k|k$是3的倍数$\}$?

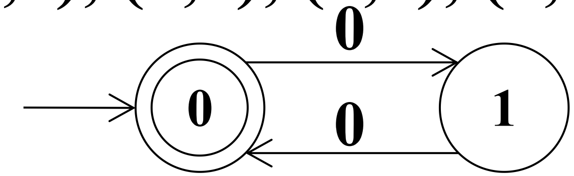
有限自动机的设计

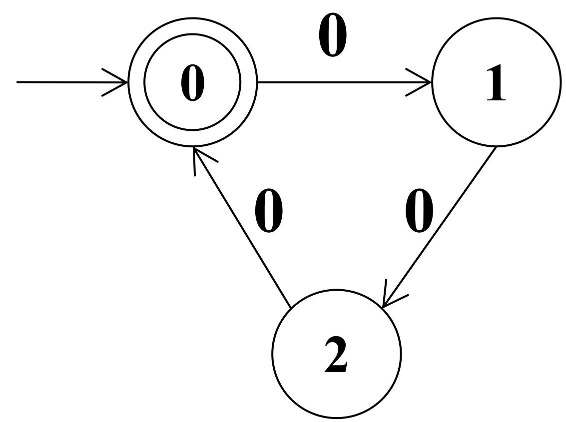
$\{ 0^k \mid k$是2和3的倍数$\}$

$\Sigma=\{0\}$, 关键信息: $\varepsilon, 0^1, 0^2, 0^3, 0^4, 0^5,$

记为: 0,1,2,3,4,5 或 (0,0), (1,1), (0,2), (1,0), (0,1), (1,2)







$\{0^k|k$是2和3的倍数$\} = \{0^k|k$是2倍数$\} \cap \{0^k|k$是3的倍数$\}$

正则语言的并是正则语言

定理: 设A,B都是Σ上的正则语言, 则$A\cup B$也是正则语言.

证明: 设$M_1=(Q_1,\Sigma,\delta_1,s_1,F_1)$和$M_2=(Q_2,\Sigma,\delta_2,s_2,F_2)$是DFA,

且 $L(M_1)=A$, $L(M_2)=B$,

令$Q=Q_1\times Q_2$, $s=(s_1,s_2)$, $F = F_1\times Q_2 \cup Q_1\times F_2$,

δ: $Q\times\Sigma\rightarrow Q$, $\forall a\in\Sigma$, $r_1\in Q_1$, $r_2\in Q_2$,

$$\delta(\ (r_1,r_2)\ ,\ a\ ) = (\ \delta_1(r_1,a), \delta(r_2,a)\ ),$$

即对$i=1,2$, 第i个分量按M_i的转移函数变化.

令$M=(Q,\Sigma,\delta,s,F)$, 则 $\forall x\ (\ x\in L(M) \leftrightarrow x\in A \cup B\ )$

即 $L(M) = A \cup B$. 证毕

正则语言的交是正则语言

定理: 设A,B都是Σ上的正则语言, 则$A\cap B$也是正则语言.

证明: 设$M_1=(Q_1,\Sigma,\delta_1,s_1,F_1)$和$M_2=(Q_2,\Sigma,\delta_2,s_2,F_2)$是DFA,

且 $L(M_1)=A$, $L(M_2)=B$,

令$Q=Q_1\times Q_2$, $s=(s_1,s_2)$, $F=F_1\times F_2$,

δ: $Q\times\Sigma\to Q$, $\forall a\in\Sigma$, $r_1\in Q_1$, $r_2\in Q_2$,

$$\delta(\,(r_1,r_2)\,,\,a\,)=(\,\delta_1(r_1,a),\,\delta(r_2,a)\,),$$

即对i=1,2, 第i个分量按M_i的转移函数变化.

令$M=(Q,\Sigma,\delta,s,F)$, 则 $\forall x\,(\,x\in L(M)\leftrightarrow x\in A\cap B\,)$

即 $L(M)=A\cap B$. 证毕

证明特点: 构造性证明

正则语言与正则运算

如果语言A被一DFA识别,则称A是正则语言

算术中, 对象是数, 操作是运算, 如+×.

计算理论中, 对象是语言, 操作是语言的运算.

定义: 设A和B是两个语言,定义正则运算

并,连接,星号如下:

- 并: $A \cup B = \{x | x \in A \text{或} x \in B\}$
- 连接: $A \circ B = \{xy | x \in A \text{且} y \in B\}$
- 星号: $A^* = \{x_1 x_2 \ldots x_k | k \geq 0 \text{且每个} x_i \in A\}$

正则运算举例

设字母表Σ由标准的26个字母组成

A={good,bad}, B={boy,girl}, 则

$A \cup B$={ good, bad, boy, girl }

$A \circ B$={ goodboy, goodgirl, badboy, badgirl }

A^*={ε, good, bad, goodgood, goodbad, … }

问题:

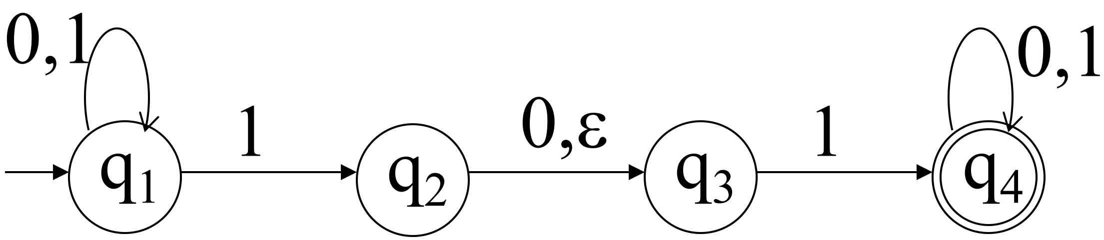
1. 正则语言对于正则运算是否封闭?
2. 如何判断一个语言是正则语言?

非确定型机器(难点)

前面因为δ: Q× Σ→Q是一个函数, 所以

- 每步存在唯一的方式进入下一状态
- 称为确定型有限自动机(DFA)

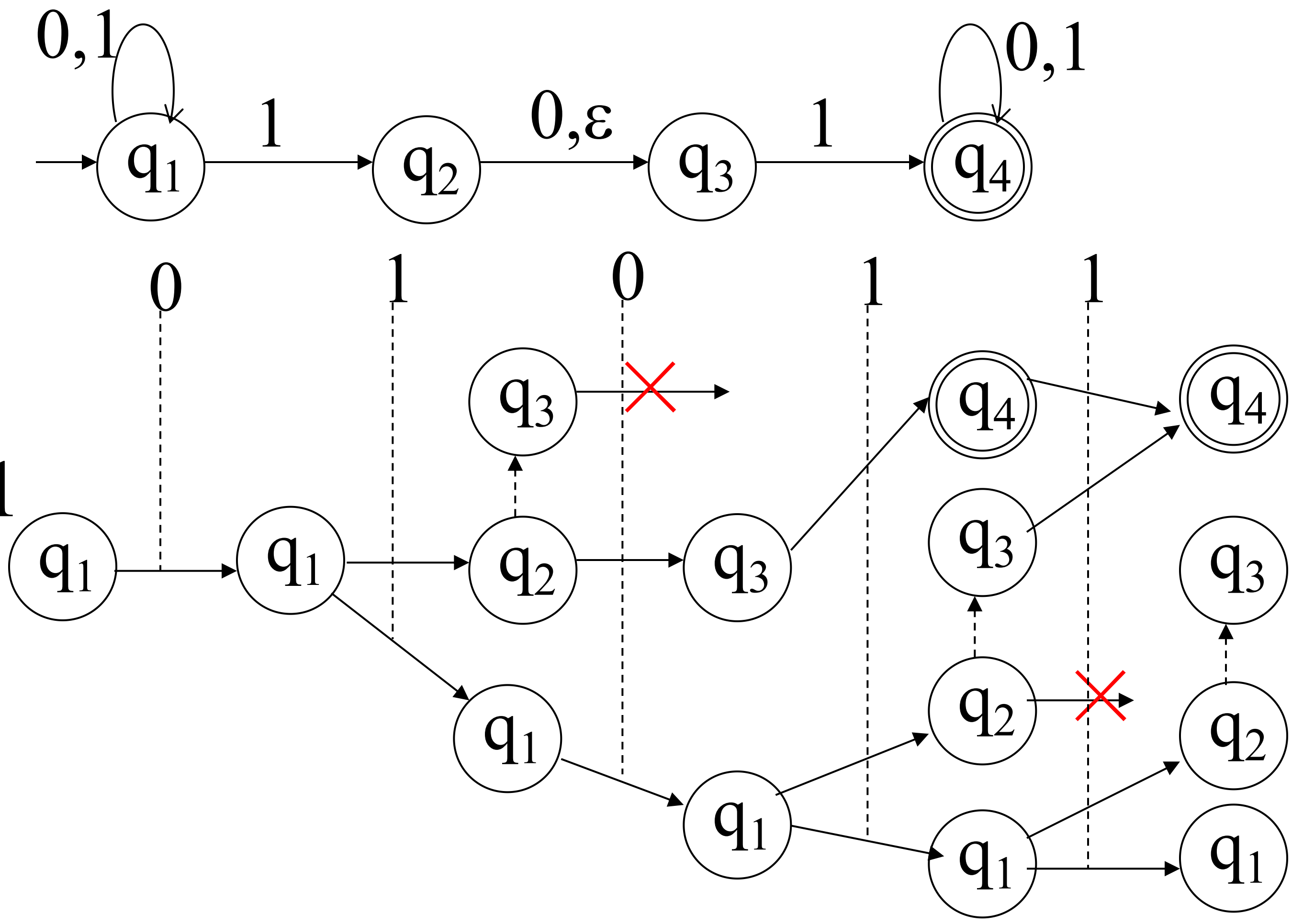
现在引入非确定型有限自动机(NFA)



- 每步可以0至多种方式进入下一步
- 转移箭头上的符号可以是空串ε,
  表示不读任何输入就可以转移过去

非确定型计算

输入
01011



注: 若起始状态有射出的ε箭头?

NFA的计算方式

Step 1.设读到符号s, 对(每个副本)机器状态q,

若q有多个射出s箭头,则机器分裂成多个副本.

状态相同的副本视为同一副本.

Step 2. 对每个副本的状态,若其上有射出的ε箭头,

则不读任何输入, 机器分裂出相应副本.

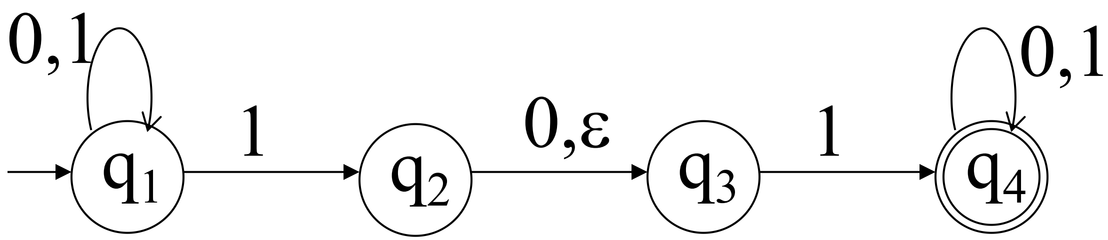
Step 3. 读下一个输入符号, 转step1.

若无输入符号, 计算结束, 并且,

若此时有一个副本处于接受状态,则接受,

否则拒绝.

NFA的形式定义



状态图
与
形式定义
包含
相同信息

定义: NFA是一个5元组(Q,Σ,δ,s,F),

1) Q是状态集;

2) Σ是字母表;

3) $\delta: Q\times \Sigma_\varepsilon \to P(Q)$是转移函数;

4) $s\in Q$是起始状态;

5) $F\subseteq Q$是接受状态集;

其中 $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$

试写出该状态图对应的形式定义

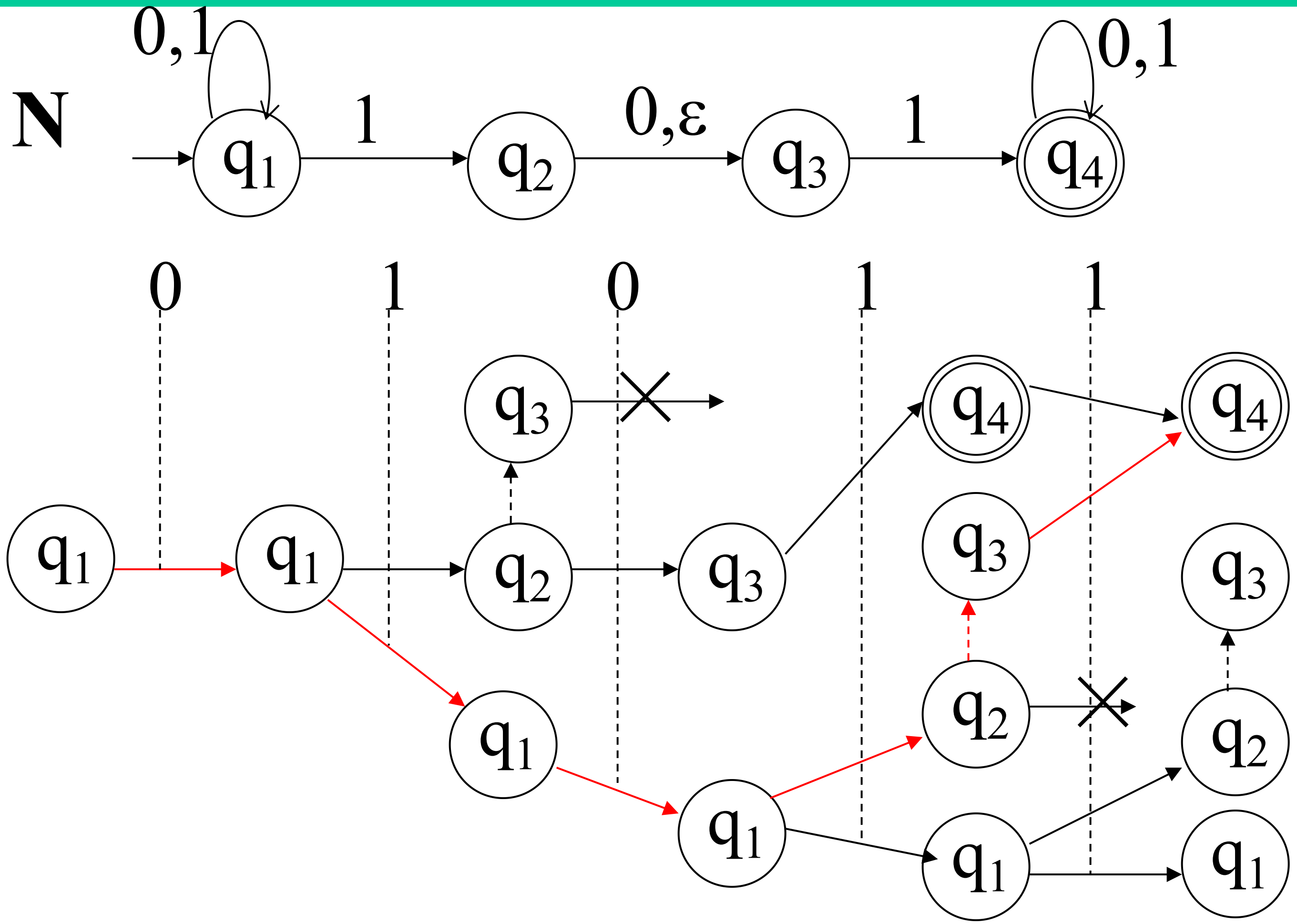
$\delta(q_1,1) = \{q_1,q_2\}$

$\delta(q_2,\varepsilon) = \{q_3\}$

$\delta(q_2,1) = \varnothing$

$\delta(q_1,\varepsilon) = \varnothing$

如何定义NFA的计算



NFA计算的形式定义

设$N=(Q,\Sigma,\delta,q_0,F)$是一台NFA, w是Σ上字符串

称 N接受w,

若 w能写作$w=w_1w_2\ldots w_n$, $w_i\in\Sigma_\varepsilon$,且

存在Q中的状态序列$r_0,r_1,\ldots,r_n$, 满足

1) $r_0 = q_0$;

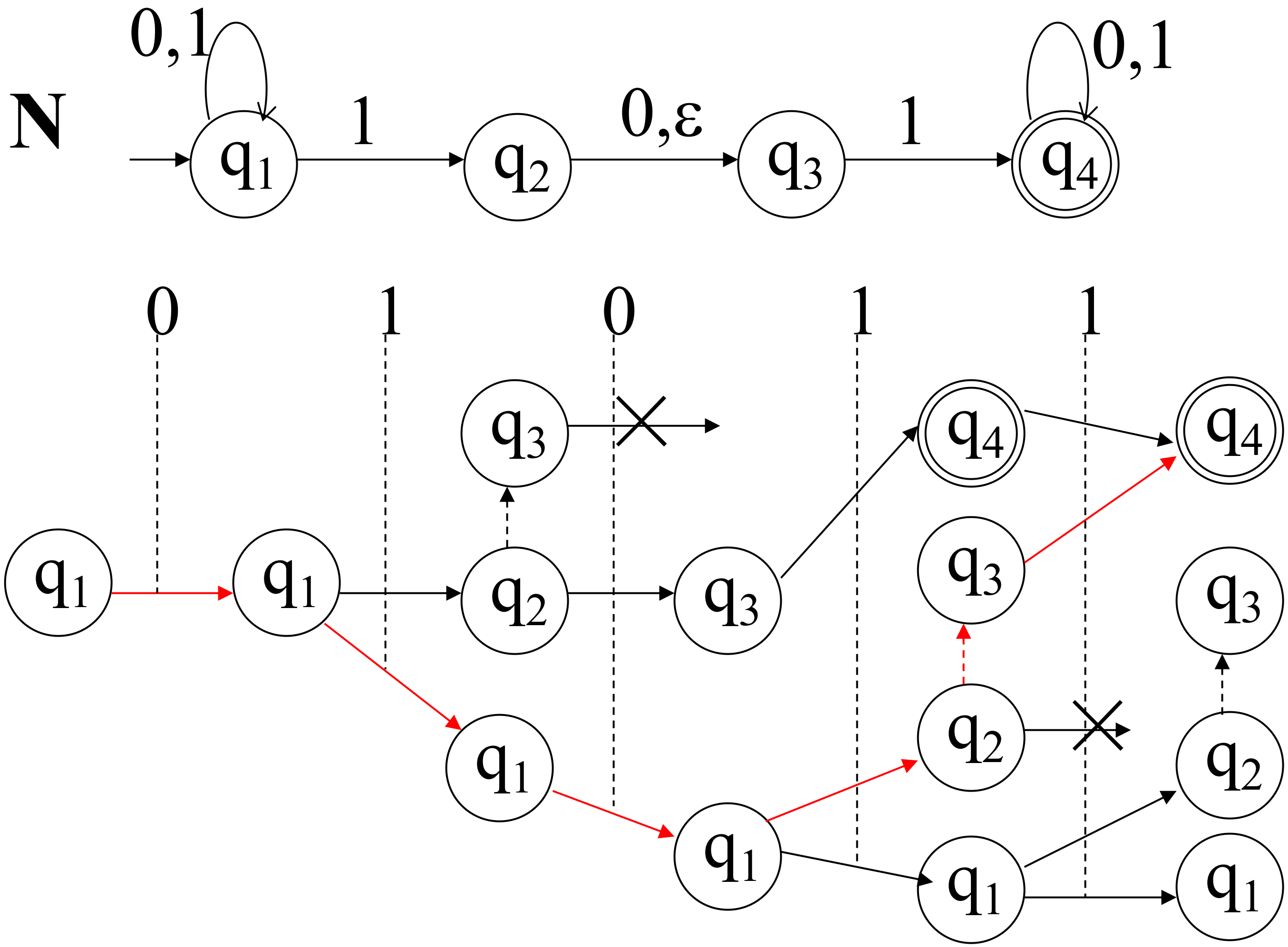
2) $r_{i+1}\in\delta(r_i, w_{i+1})$;

3) $r_n \in F$

$$r_0 \xrightarrow{w_1} r_1 \xrightarrow{w_2} r_2 \overset{\cdots}{\dashrightarrow} r_{n-1} \xrightarrow{w_n} r_n$$

对于输入, NFA计算的路径可能不唯一.

NFA计算形式定义举例



NFA的设计(难点)

- 自己即自动机
- 寻找需要记录的关键信息

设计识别{0,1}上以下语言的NFA:

$\{ w \in \{0,1\}^* \mid w$从1开始, 以0结束 $\}$

$\{ w \in \{0,1\}^* \mid w$含有子串1010 $\}$

$\{ w \in \{0,1\}^* \mid w$是倒数第2位是1 $\}$

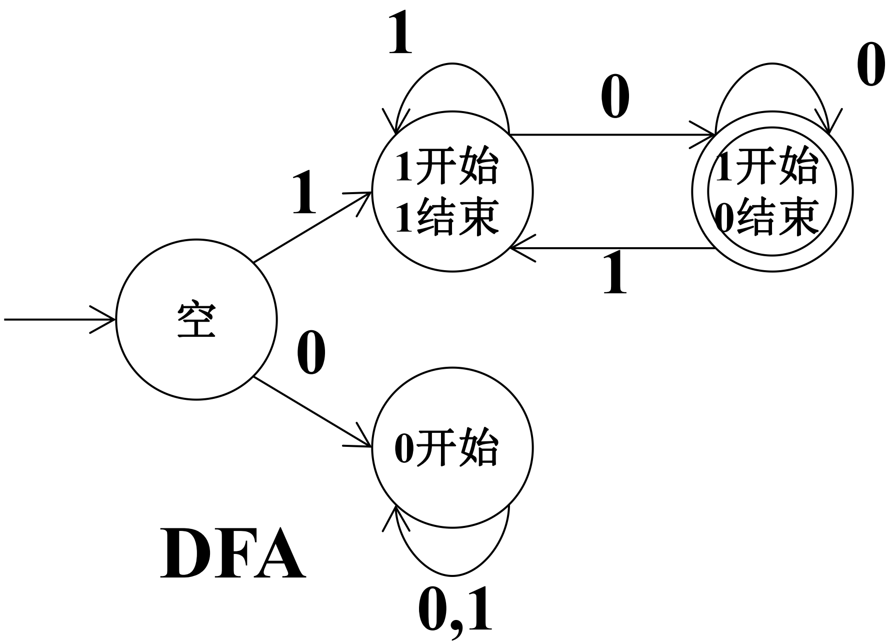
$\{ 0^k \mid k$是2或3的倍数 $\}$

NFA的设计

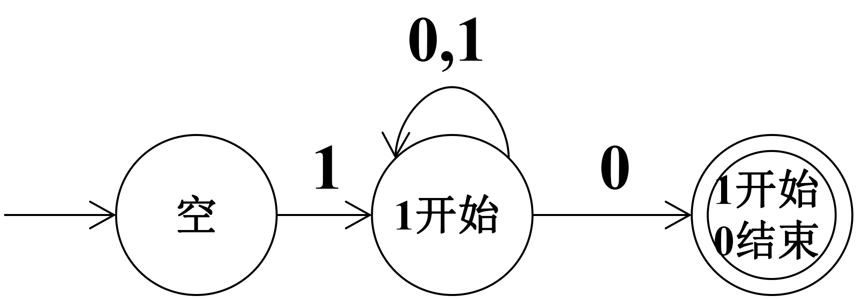
{ w∈{0,1}* | w从1开始, 以0结束 }

Σ={0,1}, 根据关键信息设计状态,

空, 以0开始, 以1开始以1结束, 以1开始以0结束



DFA

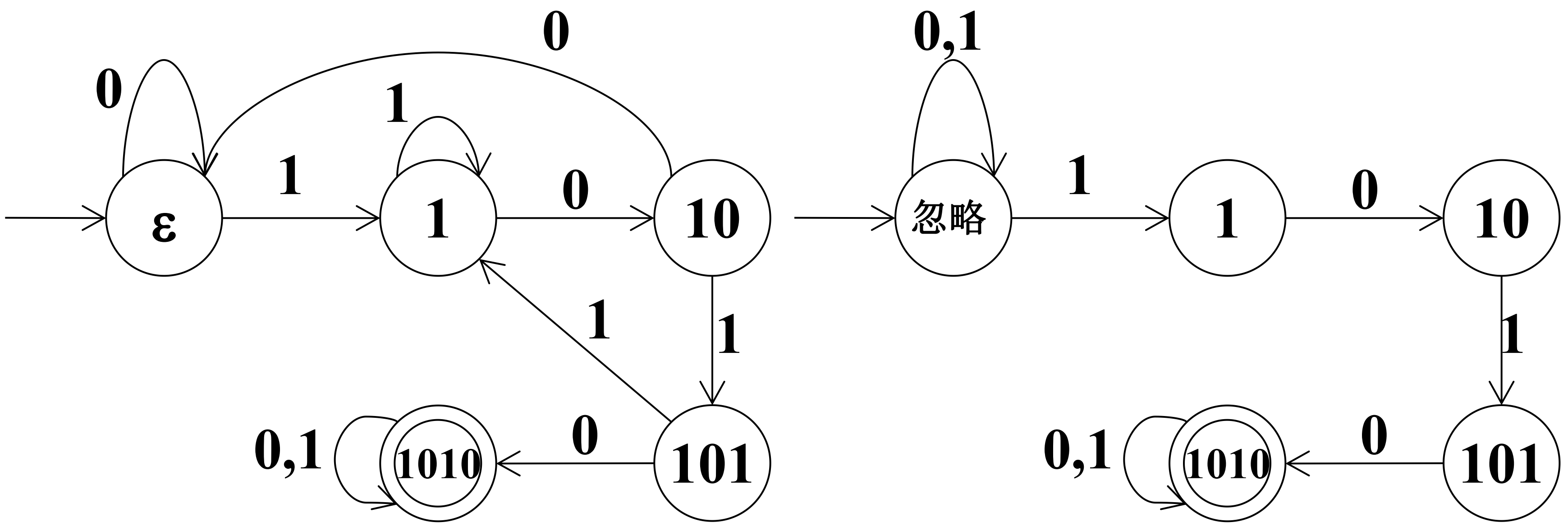


NFA

NFA的设计

{ w∈{0,1}* | w含有子串1010 }

Σ={0,1}，关键信息：忽略(ε), 1, 10, 101, 1010



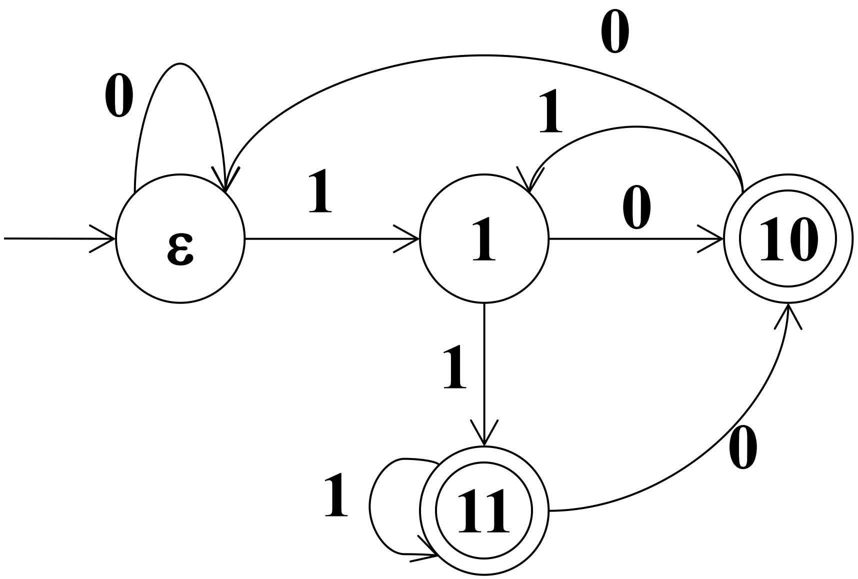
DFA

NFA

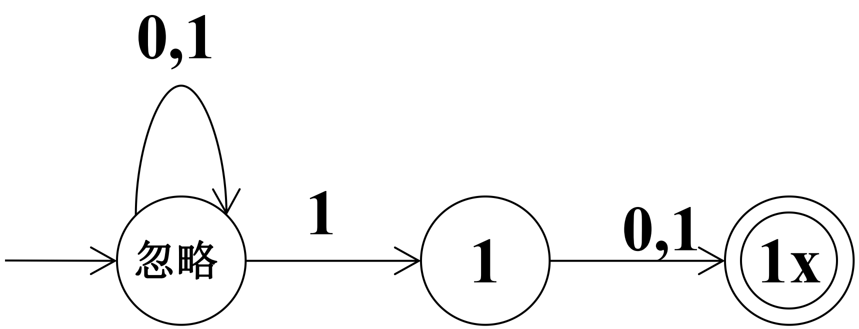
NFA的设计

{ w∈{0,1}* | w倒数第2个符号是1 }

Σ={0,1}, 关键信息: 忽略(ε), 1, 1x,



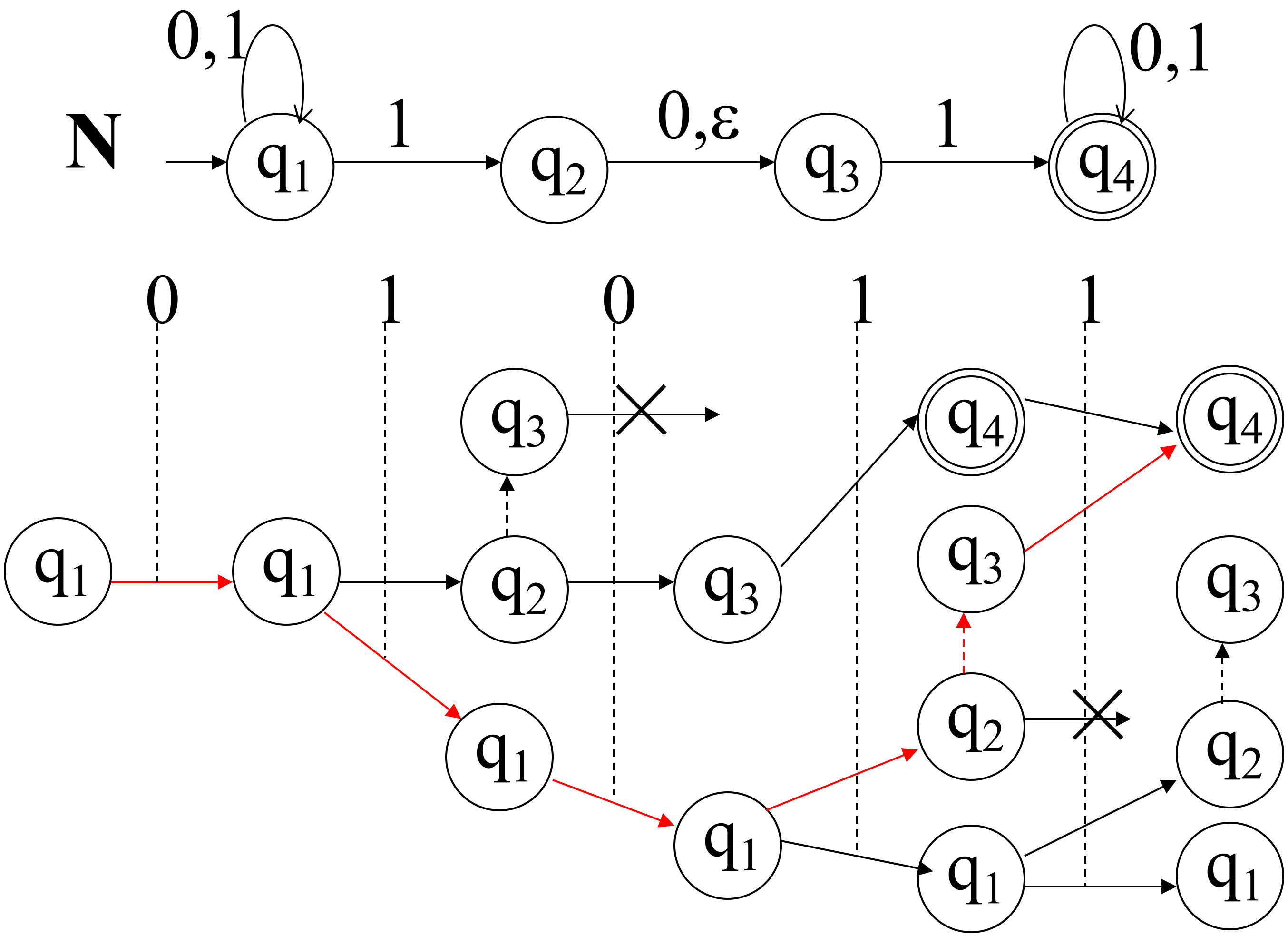
DFA



NFA

NFA与DFA等价

定理: 每个NFA都有一台等价的DFA.



构造DFA
关键信息
每个时刻
所有
副本状态
的集合

每个NFA都有等价的DFA

NFA的确定化:子集法

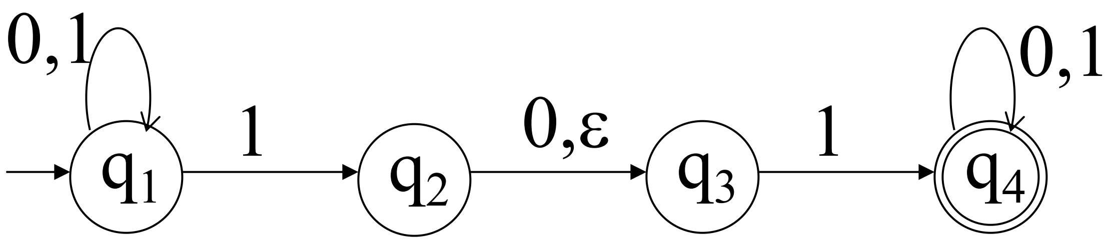
(1)首先将从 NFA N的起始状态S出发经过任意条ε弧所能到达的状态组成的集合作为确定化后的DFA M的起始状态S′。

(2)从S′出发，经过对任意输入符号a∈∑的状态转移所能到达的状态（包括读入输入符号a之后所有可能的ε转移所能到达的状态）所组成的集合作为M的新状态。

（3）如此重复，直到不再有新的状态出现为止。

（4）在所产生的状态中，含有原NFA接受态的子集作为DFA的接受态。

每个NFA都有等价的DFA



以原状态的子集
为新机器的状态

编号	δ	0	1
1	$\{q_1\}$ 1	$\{q_1\}$	$\{q_1, q_2, q_3\}$2
2	$\{q_1, q_2, q_3\}$	$\{q_1, q_3\}$3	$\{q_1, q_2, q_3, q_4\}$4
3	$\{q_1, q_3\}$	$\{q_1\}$	$\{q_1, q_2, q_3, q_4\}$
4*	$\{q_1, q_2, q_3, q_4\}$	$\{q_1, q_3, q_4\}$5	$\{q_1, q_2, q_3, q_4\}$
5*	$\{q_1, q_3, q_4\}$	$\{q_1, q_4\}$6	$\{q_1, q_2, q_3, q_4\}$
6*	$\{q_1, q_4\}$	$\{q_1, q_4\}$	$\{q_1, q_2, q_3, q_4\}$

每个NFA都有等价的DFA

证明: 设$N = (Q_1, \Sigma, \delta_1, s_1, F_1)$是NFA,　//设计Q,F, s,δ

令 $Q = P(Q_1)$,　//全体子集

$F = \{ A \in Q : F_1 \cap A \neq \varnothing \}$,

$s = E(\{s_1\})$, $E(A) = \{ q : \exists r \in A$, r经0到多个ε箭头可达q $\}$

$\delta: Q \times \Sigma \to Q$, $\forall a \in \Sigma$, $\forall A \in Q$,

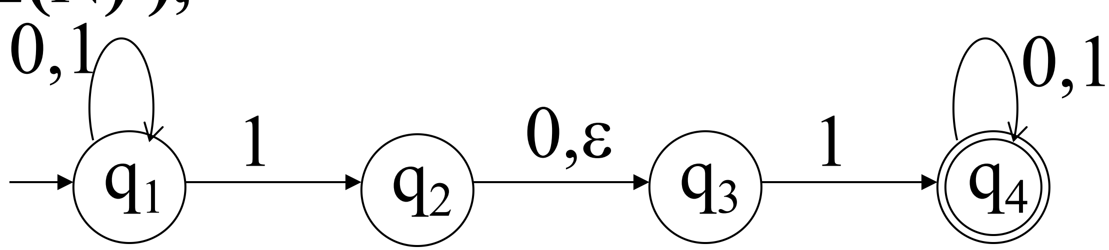
$$\delta(A, a) = E\left(\cup_{r \in A} \delta_1(r,a) \right)$$

$M = (Q, \Sigma, \delta, s, F)$,

则 $\forall x\ (x \in L(M) \leftrightarrow x \in L(N))$,

即 $L(M) = L(N)$.

证毕



正则运算的封闭性

定理: 正则语言对并运算封闭.

定理: 正则语言对连接运算封闭.

定理: 正则语言对星号运算封闭.

证明: 画状态图.

证明若A, B正则, 则A∪B正则

DFA: $M_1=(Q_1,\Sigma,\delta_1,s_1,F_1)$, $M_2=(Q_2,\Sigma,\delta_2,s_2,F_2)$, $L(M_1)=A$, $L(M_2)=B$,

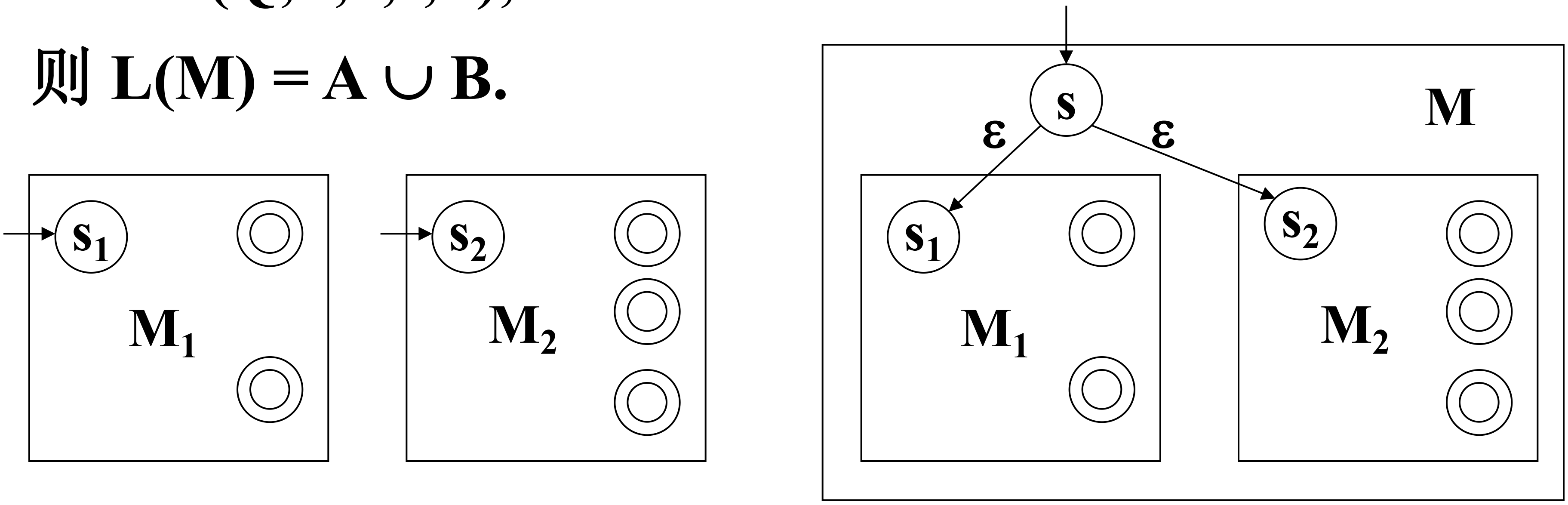
令 $Q = Q_1 \cup Q_2 \cup \{s\}$ 不交并, $F = F_1 \cup F_2$ 不交并

$\forall i=1,2$, $\forall r\in Q_i$, $\forall a\in\Sigma$, $\delta(r,a) = \{\delta_i(r,a)\}$

$\delta(s,\varepsilon) = \{s_1,s_2\}$

$M=(Q,\Sigma,\delta,s,F)$,

则 $L(M) = A \cup B$.



证明若A, B正则, 则A°B正则

DFA: $M_1=(Q_1,\Sigma,\delta_1,s_1,F_1)$, $M_2=(Q_2,\Sigma,\delta_2,s_2,F_2)$, $L(M_1)=A$, $L(M_2)=B$,

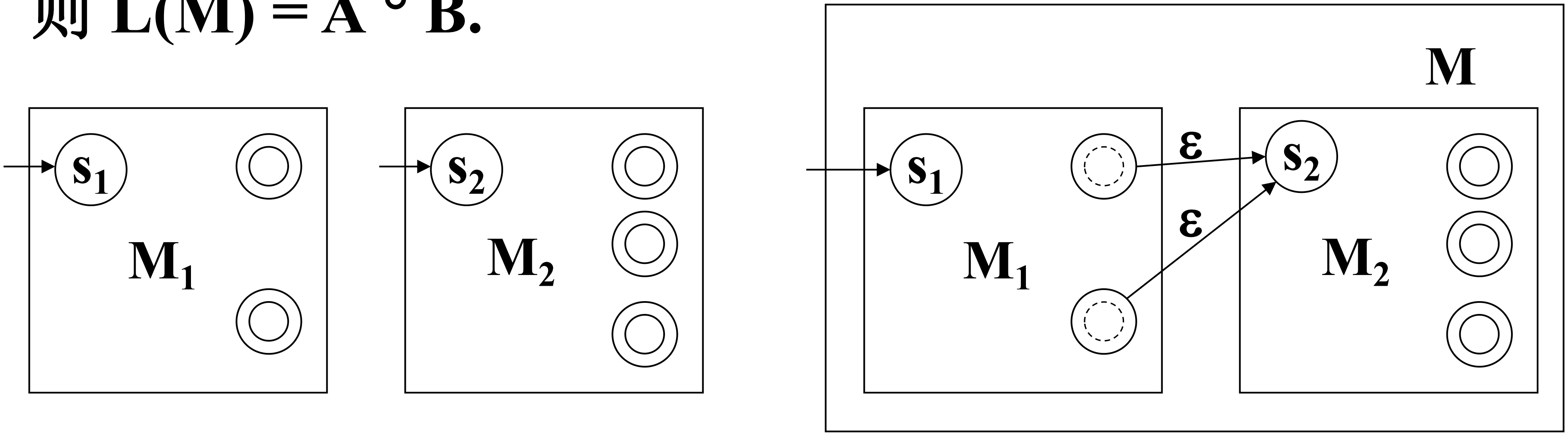
令 $Q = Q_1 \cup Q_2$ 不交并, $F = F_2$,

$\forall r \in F_1,\ \delta(r,\varepsilon) = \{s_2\}$

$\forall i=1,2,\ \forall r \in Q_i,\ \forall a \in \Sigma,\ \delta(r,a) = \{\delta_i(r,a)\}$

$M=(Q,\Sigma,\delta,s_1,F)$,

则 $L(M) = A \circ B$.



证明若A正则, 则A^*正则

DFA: $M_1=(Q_1,\Sigma,\delta_1,s_1,F_1)$, $L(M_1)=A$,

令 $Q = Q_1 \cup \{s\}$ 不交并, $F = F_1 \cup \{s\}$ 不交并

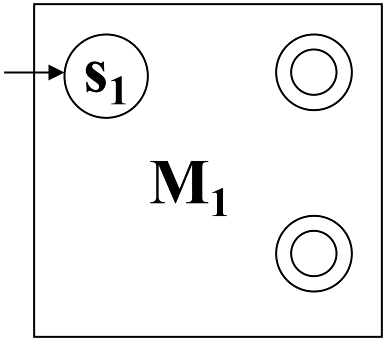
$\forall r \in Q_1, \forall a \in \Sigma,\ \delta(r,a) = \{\delta_1(r,a)\}$

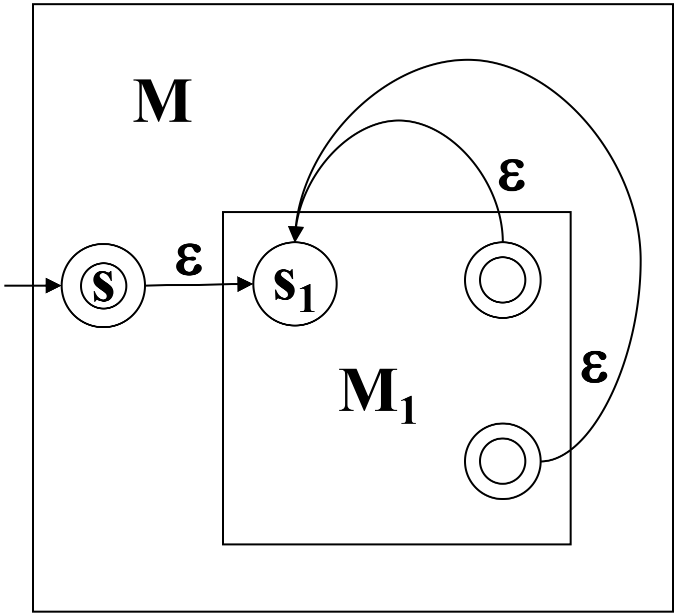
$\forall r \in F_1, \delta(r,\varepsilon) = \{s_1\}$,

$\delta(s,\varepsilon) = \{s_1\}$,

$M=(Q,\Sigma,\delta,s,F)$,

则 $L(M) = A^*$.





正则表达式

定义: 称R是一个正则表达式, 若R是

1) a, $a \in \Sigma$;

2) ε ;

3) $\varnothing$;

4) $(R_1 \cup R_2)$, R_1和R_2是正则表达式;

5) $(R_1 \circ R_2)$, R_1和R_2是正则表达式;

6) (R_1^*), R_1是正则表达式;

每个正则表达式R表示一个语言(?),记为L(R).

例: 0^*10^*, $01 \cup 10$, $(\Sigma\Sigma)^*$, $1^*\varnothing$, $\varnothing^*$.

正则表达式与DFA等价

定理2.3.1: 语言A正则⇔A可用正则表达式描述.

(⇐) 若 语言A可用正则表达式描述,

则 A正则. (容易)

(⇒) 若语言A正则,

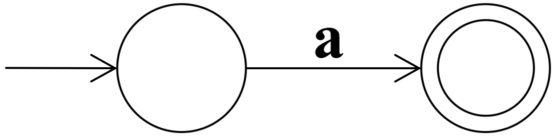
则A可用正则表达式描述. (困难)

A有正则表达式⇒A正则

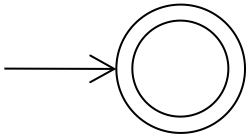
数学归纳法

R是一个正则表达式, 若R是

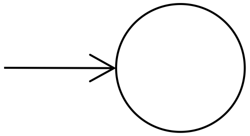
1) a, $a \in \Sigma$



2) ε



3) $\varnothing$



4) $(R_1 \cup R_2)$

5) $(R_1 \circ R_2)$

6) (R_1^*)

A正则⇒A有正则表达式

构造广义非确定有限自动机(GNFA)

- 非确定有限自动机
- 转移箭头可以用任何正则表达式作标号

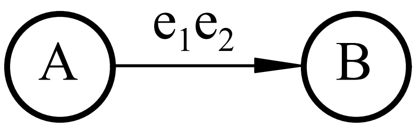
证明中的特殊要求:

- 起始状态无射入箭头.
- 唯一接受状态(无射出箭头).

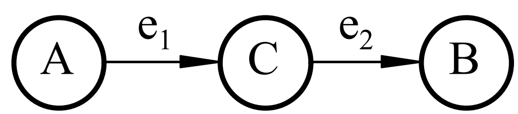
手段: 一个一个地去掉中间状态.

正则表达式到NFA的转换

（1） 替换成

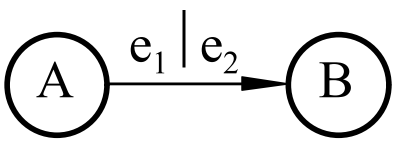


(a)

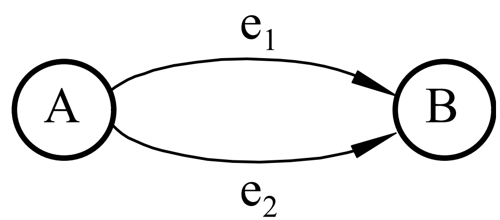


(b)

（2） 替换成

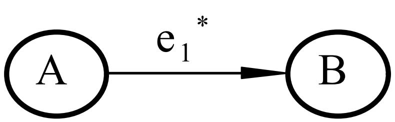


(c)

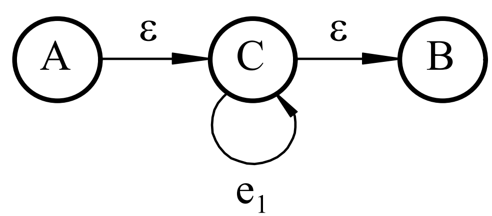


(d)

（3） 替换成

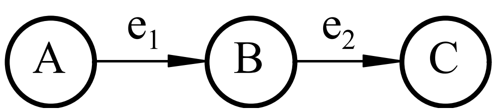


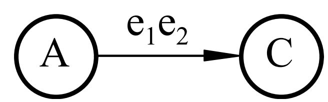
(e)

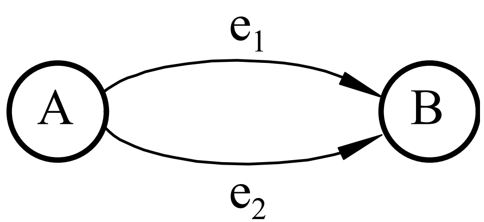


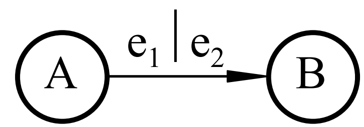
(f)

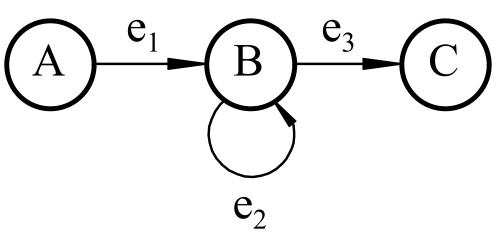
NFA到正则表达式的转换

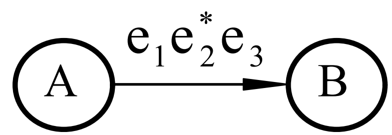
（1）

替换成


（2）

替换成


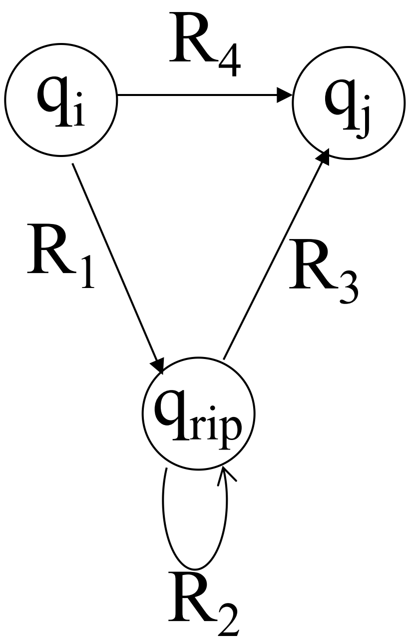
（3）

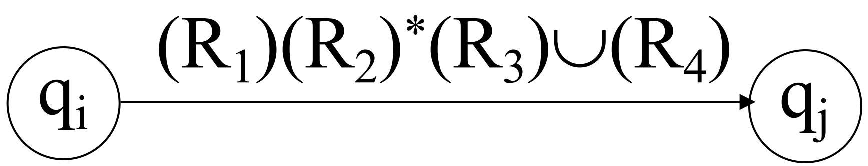
替换成


删除一个中间状态

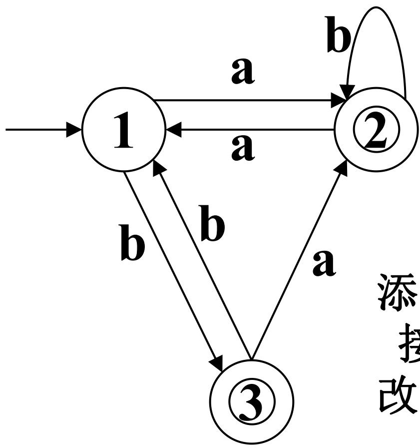
设q_{rip}为待删中间状态,

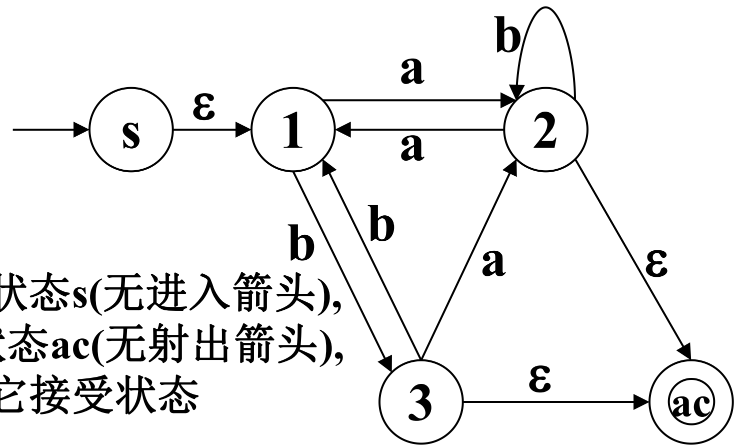
对任意两个状态q_i, q_j都需要修改箭头标号



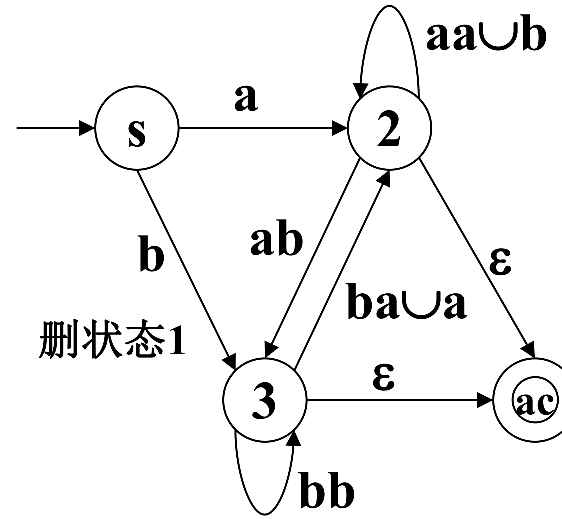


举例: A正则⇒A有正则表达式

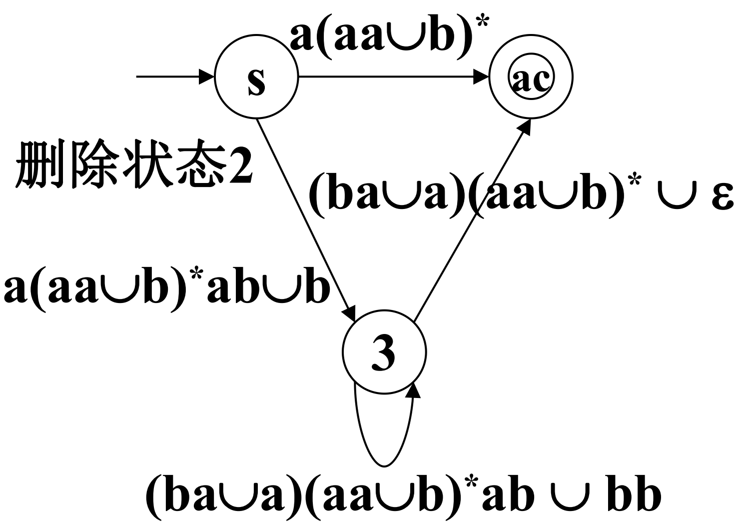




添起始状态s(无进入箭头),
接受状态ac(无射出箭头),
改掉其它接受状态

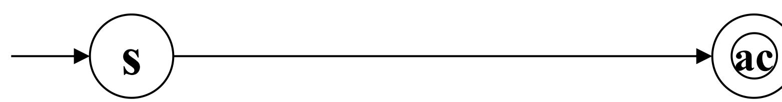


删状态1



删除状态2

$\color{red}{(}\ (a(aa\cup b)^{*}ab\cup b)((ba\cup a)(aa\cup b)^{*}ab \cup bb)^{*}$
$((ba\cup a)(aa\cup b)^{*}\cup\varepsilon)\ \color{red}{)} \cup \color{red}{(}\ a(aa\cup b)^{*}\color{red}{)}$



删除状态3

非正则语言

B={ 0^n1^n | n≥0 }

C={ w | w中0和1的个数相等 }

D={ w | w中01和10的个数相等 }

哪些是正则语言?

泵引理

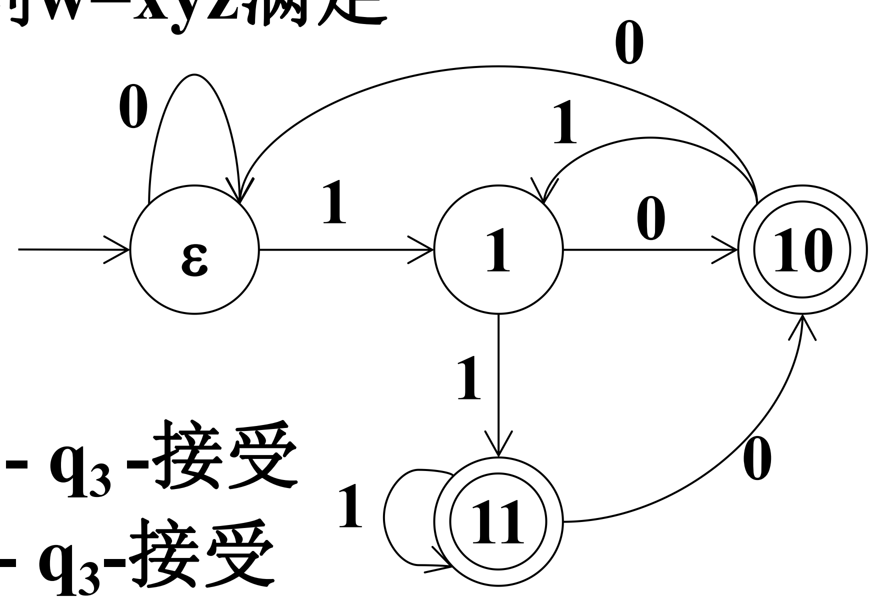
定理(泵引理): 设A是正则语言,则存在p>0使得

对任意$w \in A$, $|w| \geq p$, 存在分割$w=xyz$满足

1) 对任意 $i \geq 0$, $xy^iz \in A$;

2) $|y|>0$;

3) $|xy| \leq p$.



11011 : q_0-1- q_1- 101 - q_1 - 1 - q_3 -接受

$1(101)^i1$: q_0-1- q_1- $(101)^i$ - q_1 - 1 - q_3-接受

11011=xyz

x=1, y=101, z=1. xy^iz 被接受的原因?

取p为DFA状态个数.

由鸽巢原理, 读前p个符号必有状态重复

泵引理的等价描述

定理(泵引理): 设A是正则语言,则存在$p>0$使得

对任意$w\in A$, $|w|\geq p$, 存在分割$w=xyz$满足

1) 对任意 $i\geq 0$, $xy^iz\in A$;

2) $|y|>0$;

3) $|xy|\leq p$.

若A是正则语言,

则$\exists p>0$

$\forall w\in A(|w|\geq p)$

$\exists x,y,z(|y|>0, |xy|\leq p, w=xyz)$

$\forall i\geq 0,$

$xy^iz\in A.$

若$\forall p>0$

$\exists w\in A(|w|\geq p)$

$\forall x,y,z(|y|>0, |xy|\leq p, w=xyz)$

$\exists i\geq 0,$

$xy^iz\notin A.$

则A非正则语言

$B = \{ 0^n1^n \mid n \ge 0 \}$ 非正则

∵ $\forall p>0$,

令 $w=0^p1^p$,

$\forall x,y,z(|y|>0, |xy|\le p, w=xyz)$

令 $i=0$,

$xz = 0^{p-|y|}1^p \notin B$

∴ B非正则语言

若 $\forall p>0$

$\exists w \in A(|w| \ge p)$

$\forall x,y,z(|y|>0, |xy|\le p, w=xyz)$

$\exists i \ge 0$,

$xy^iz \notin A$.

则A非正则语言

$C = \{ ww \mid w \in \{0,1\}^* \}$ 非正则

$\because \forall p>0,$

令$w=0^p10^p1,$

$\forall x,y,z(|y|>0, |xy|\le p, w=xyz)$

令$i=0,$

$xz = 0^{p-|y|}10^p1 \notin C$

$\therefore$ C非正则语言

若$\forall p>0$

$\exists w\in A(|w|\ge p)$

$\forall x,y,z(|y|>0, |xy|\le p, w=xyz)$

$\exists i\ge 0,$

$xy^iz \notin A.$

则A非正则语言

泵引理的证明

定理(泵引理): 设A是正则语言,则存在$p>0$使得

对任意$w\in A$, $|w|\geq p$, 存在分割$w=xyz$满足

1) 对任意 $k\geq 0$, $xy^kz\in A$;

2) $|y|>0$;

3) $|xy|\leq p$.

证明: 令$M=(Q,\Sigma,\delta,s,F)$ 且 $L(M)=A$, 令$p=|Q|$,

设 $w = w_1w_2\dots w_n\in A$, $w_i\in\Sigma$, 且$n\geq p$, 则有

$$s=r_0\xrightarrow{w_1} r_1 \xrightarrow{w_2} r_2 \overset{\cdots}{\dashrightarrow} r_{n-1} \xrightarrow{w_n} r_n\in F$$

由鸽巢原理, 存在$i<j\leq p$使得$r_i=r_j$, 令$x=w_1\dots w_i$, $y=w_{i+1}\dots w_j$, $z=w_{j+1}\dots w_n$. 那么对$\forall k\geq 0$, $xy^kz\in A$.

第3章 图灵机

1. 图灵机基础

1.1 图灵机的定义

1.2 图灵机举例

1.3 图灵机的描述

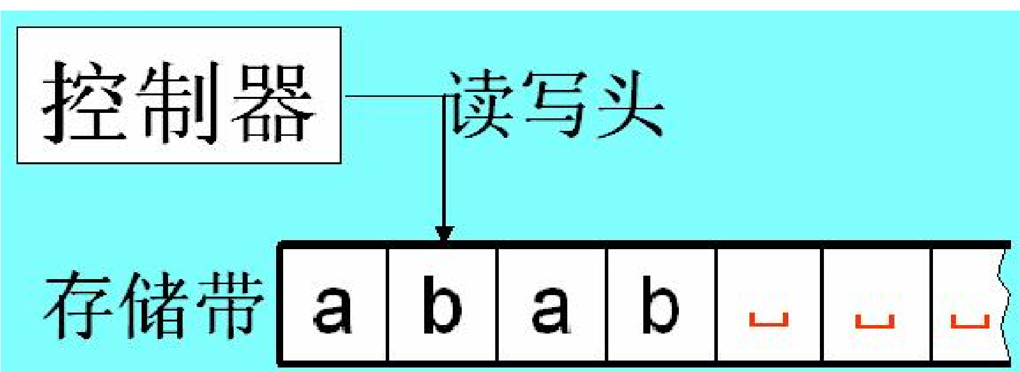
图灵对计算的观察

图灵: 计算通常是一个人拿着笔在纸上进行的.

他根据
- 眼睛看到的纸上符号,
- 脑中的若干法则,

指示笔
- 在纸上擦掉或写上一些符号,
- 再改变他所看到的范围.

继续, 直到他认为计算结束.



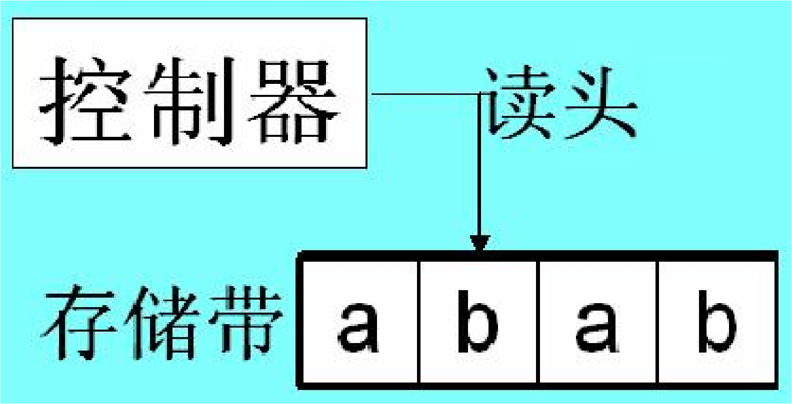
脑:控制器　纸:存储带

眼睛和笔:读写头

法则:转移函数

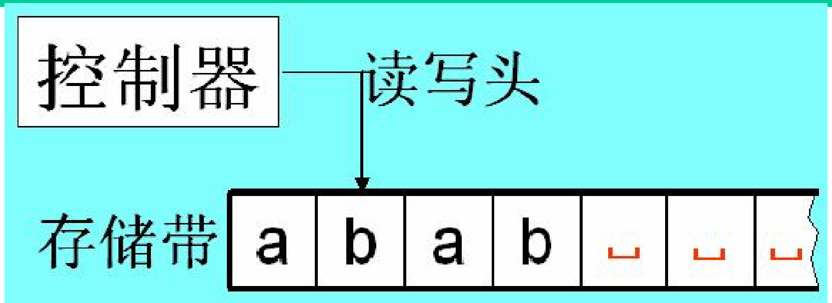
与有限自动机的区别

有限自动机:



- 输入带长度有限
- 只能读和右移, 不能写和左移
- 读完输入停机

图灵机(TM)的形式化定义



TM是一个7元组$(Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$

1) Q是状态集.

2) Σ是输入字母表,不包括空白符␣.

3) Γ是带字母表,其中␣$\in\Gamma$, $\Sigma\subset\Gamma$.

4) $\delta: Q\times\Gamma\rightarrow Q\times\Gamma\times\{L,R\}$是转移函数.

5) $q_0\in Q$是起始状态. 6) $q_a\in Q$是接受状态.

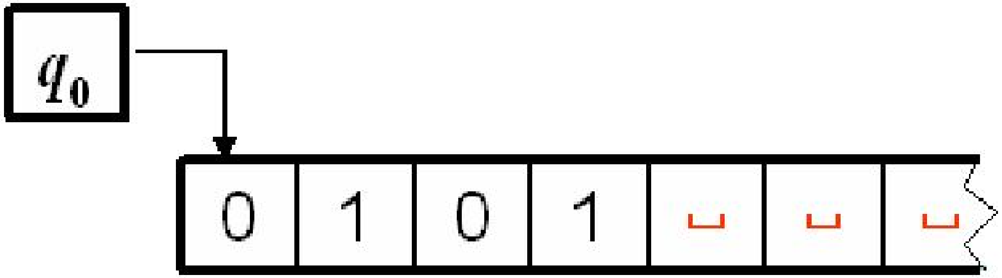
7) $q_r\in Q$是拒绝状态, $q_a \neq q_r$.

图灵机的初始化

设M=(Q, Σ, Γ, δ, q_0, q_a, q_r), w=w$_1$…w$_n$∈Σ^n,

- 输入带: 将输入串w放在最左端n格中, 带子其余部分补充空格␣.
- 读写头: 指向工作带最左端.

例:设输入串为0101, 则其初始形态为



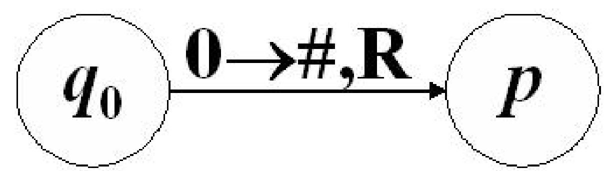
图灵机的运行

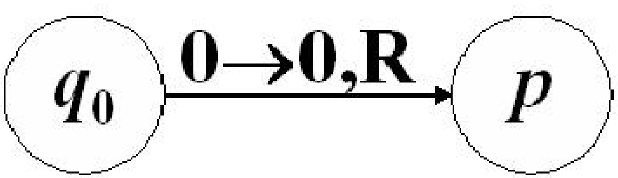
- 图灵机根据转移函数运行.

例:设输入串为0101, 且$\delta(q_0,0)=(p,\#,R)$, 则有

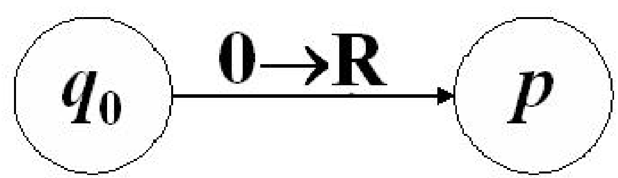
- 注: 若要在最左端左移, 读写头保持不动.

$\delta(q_0,0)=(p,\#,R)$的状态图表示:





简记为



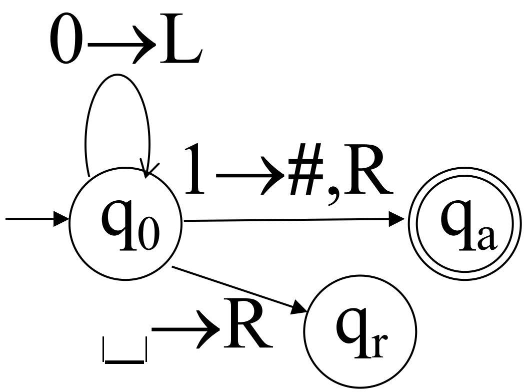
判定器与语言分类

- 图灵机运行的三种结果

  1. 若TM进入接受状态,则停机且接受输入,

  2. 若TM进入拒绝状态,则停机且拒绝输入,

  3. 否则TM一直运行,不停机.



- 定义: 称图灵机M为判定器,

  若M对所有输入都停机.

- 定义不同语言类:

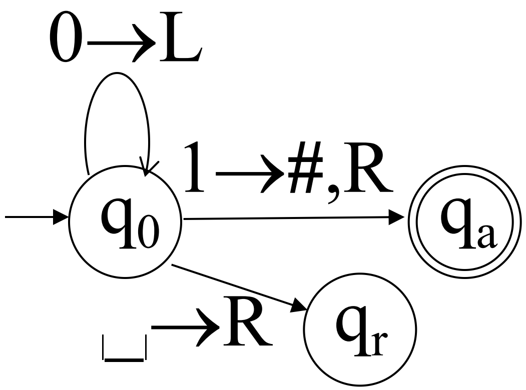
  图灵可判定语言: 某个判定器的语言(也称递归语言)

  图灵可识别语言: 某个图灵机的语言,

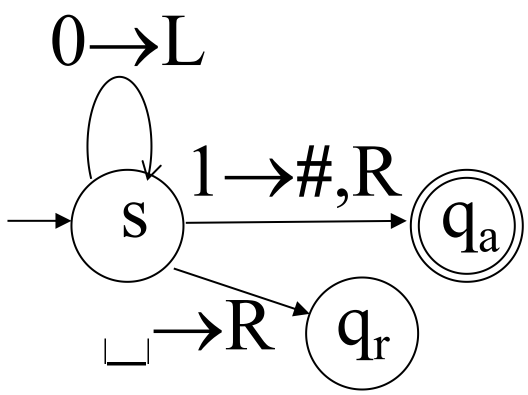
  也称为递归可枚举语言

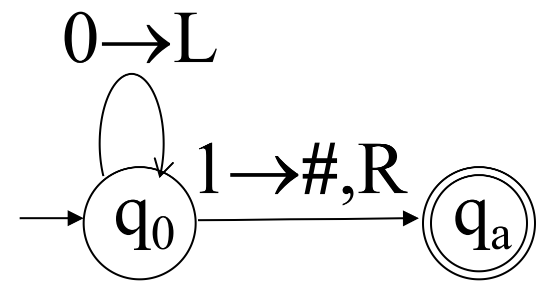
图灵机的格局

- 描述图灵机运行的每一步需要如下信息:
  控制器的状态; 存储带上字符串; 读写头的位置.
- 定义: 对于图灵机$M=(Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$,
  设$q\in Q$, $u,v\in\Gamma^*$, 则格局 uqv 表示
  1) 当前控制器状态为q ;
  2) 存储带上字符串为uv(其余为空格);
  3) 读写头指向v的第一个符号.
- 起始格局, 接受格局, 拒绝格局.



格局演化举例





省略拒绝状态

s 0 1

s 0 1

…

循环

s 1 0

\# q_a 0

接受

s _ _

_ q_r _

拒绝

图灵机计算的形式定义

称图灵机**M接受**字符串**w**,

若存在格局序列$C_1,C_2,\ldots,C_k$使得

1) C_1是M的起始格局q_0w;

2) C_i产生C_{i+1}, $i=1,\ldots,k-1$;

3) C_k是M的接受格局.

M的**语言**: **M**接受的所有字符串的集合,

记为**L(M)**.

1. 图灵机基础

1.1 图灵机的定义

1.2 图灵机举例

1.3 图灵机的描述

图灵机举例

$\Sigma=\{0,1\}$, $A=\{0w1 : w\in\Sigma^*\}$ 正则语言

$B=\{0^n1^n : n\geq 0\}$ 上下文无关语言

$\Sigma=\{0\}$, $C=\{0^k: k=2^n, n\geq 0\}$ 图灵可判定语言

M="对于输入串w,

1) 若w=ε, 则拒绝.

2) 若只有1个0,则接受.

3) 若有奇数个0,则拒绝.

4) 隔一个0,删一个0. 转(2)."

L(M)=C, 即M识别C.

状态图

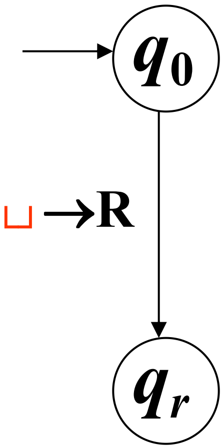
M="对于输入w,

1) 若w=ε, 则拒绝.

2) 若只有1个0,则接受.

3) 若有奇数个0,则拒绝.

4) 隔一个0,删一个0. 转(2)."



状态图

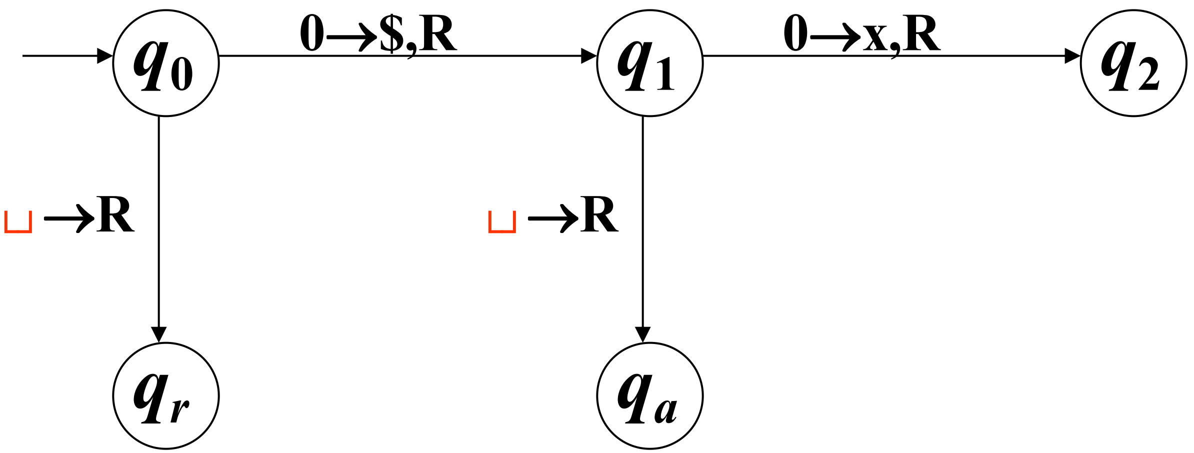
M="对于输入w,

1) 若w=ε, 则拒绝.

2) 若只有1个0,则接受.

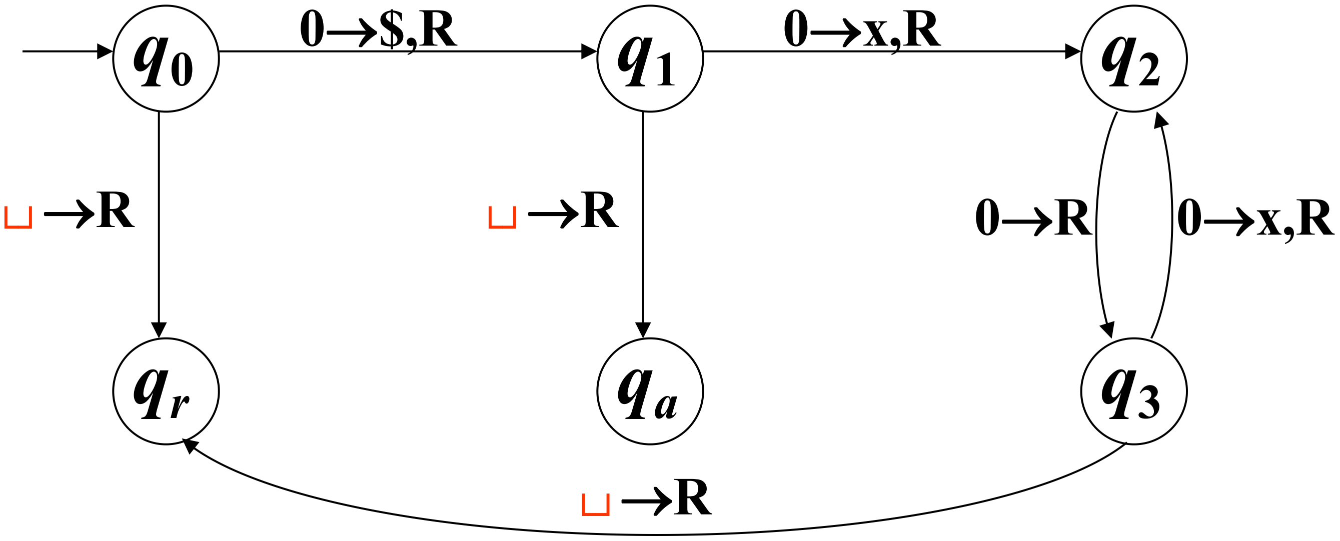
3) 若有奇数个0,则拒绝.

4) 隔一个0,删一个0. 转(2)."



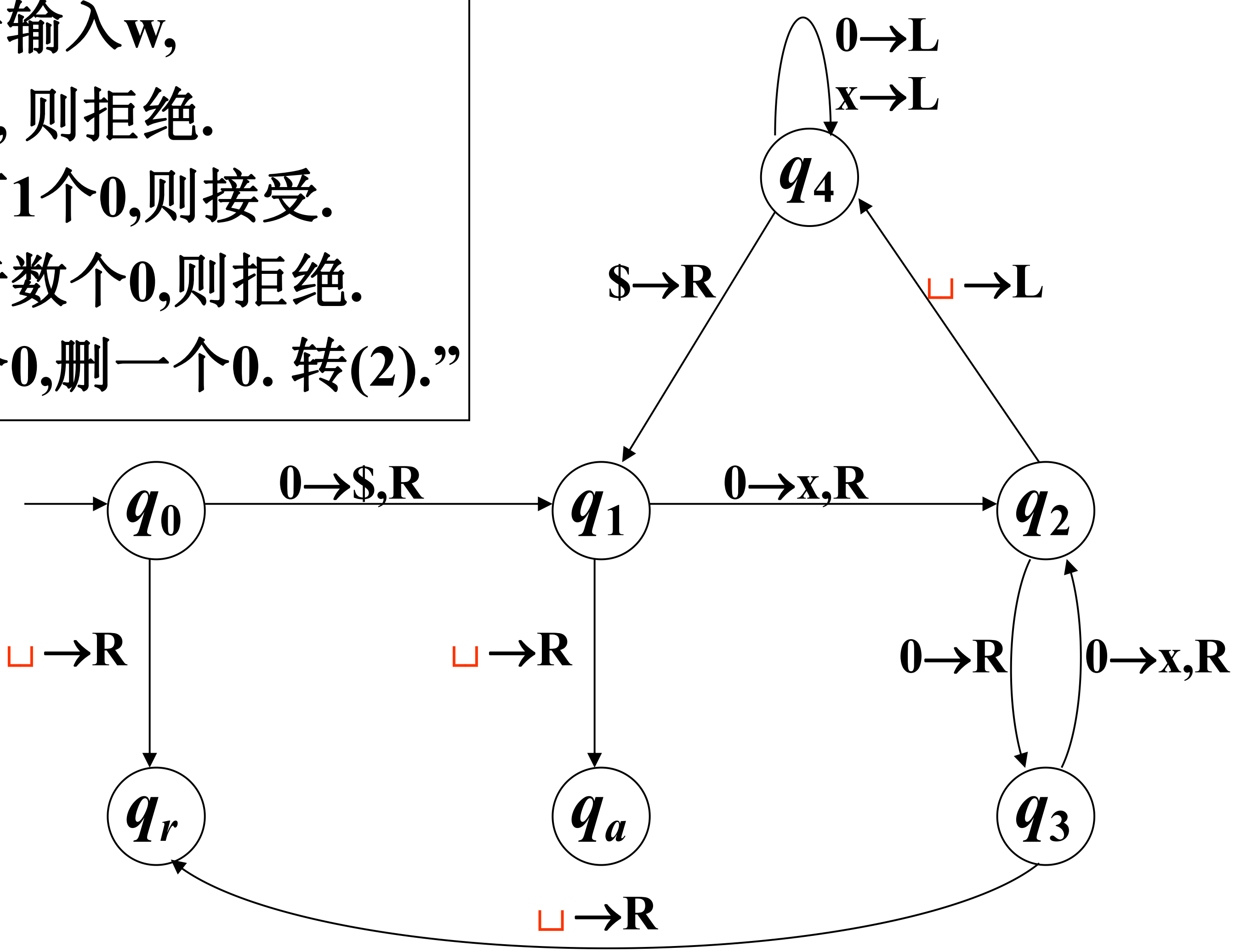
状态图

M="对于输入w,
1) 若w=ε, 则拒绝.
2) 若只有1个0,则接受.
3) 若有奇数个0,则拒绝.
4) 隔一个0,删一个0. 转(2)."



状态图

M="对于输入w,
1) 若w=ε, 则拒绝.
2) 若只有1个0,则接受.
3) 若有奇数个0,则拒绝.
4) 隔一个0,删一个0. 转(2)."



状态图

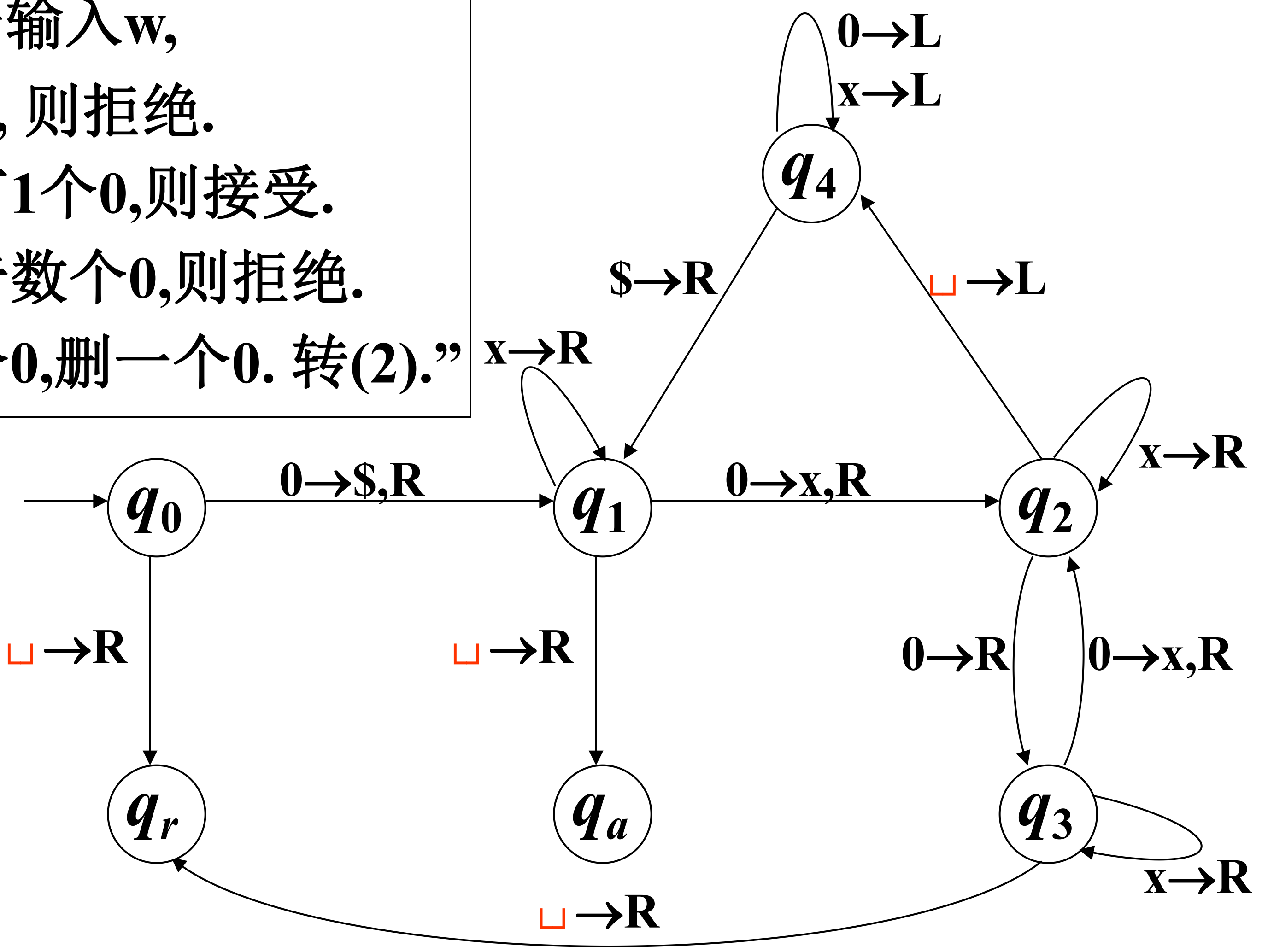
M="对于输入w,

1) 若w=ε, 则拒绝.

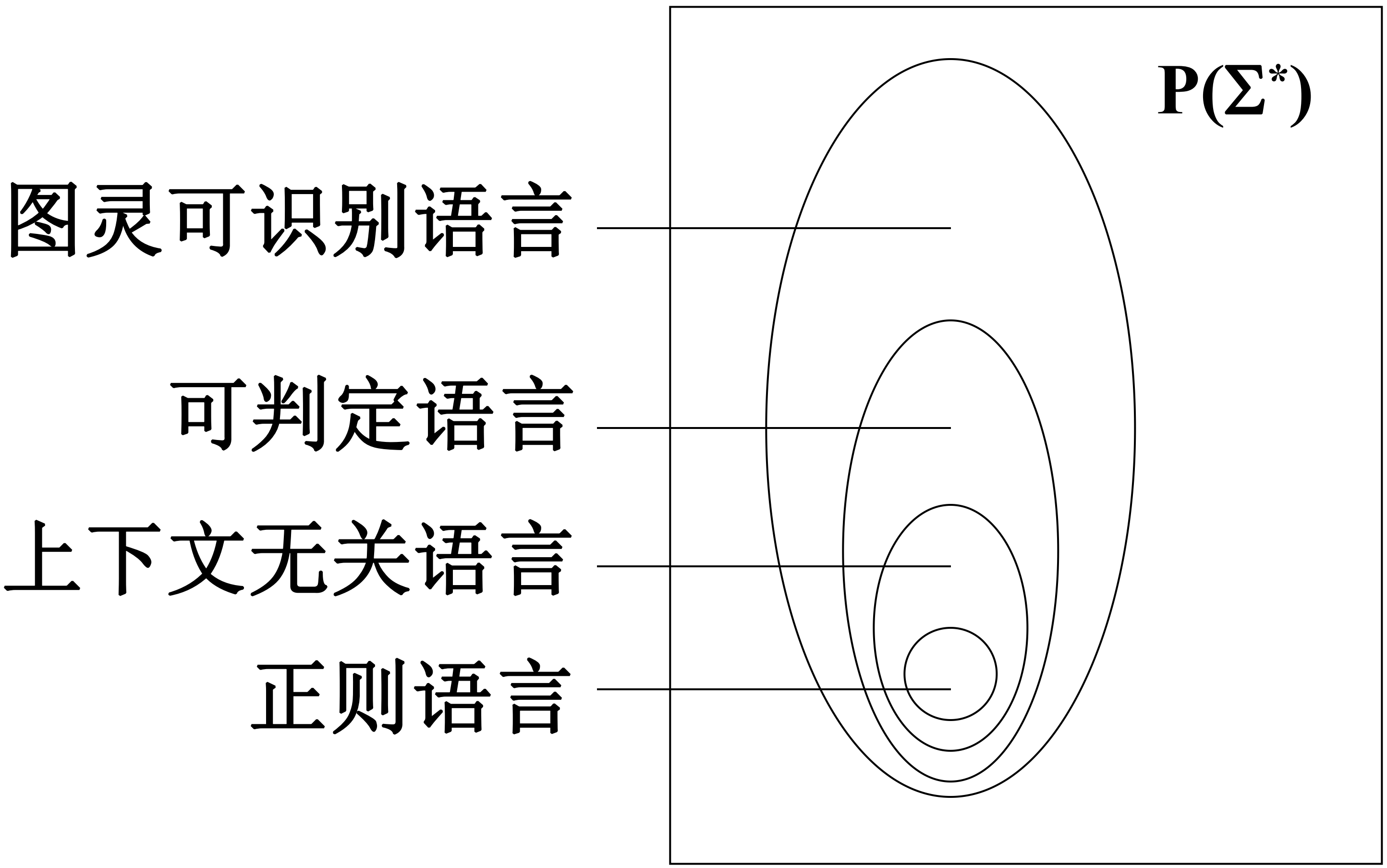
2) 若只有1个0,则接受.

3) 若有奇数个0,则拒绝.

4) 隔一个0,删一个0. 转(2)."



各种语言类的包含关系



图灵机的描述

(1) 形式水平的描述(状态图或转移函数)

(2) 实现水平的描述(读写头的移动,改写)

(3) 高水平描述(使用日常语言)

用带引号的文字段来表示图灵机. 例如:

M="对于输入串w,

1) 若w=ε, 则拒绝.

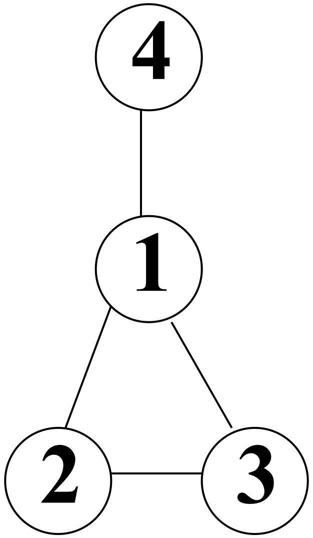
2) 若只有1个0, 则接受.

3) 若0的个数为奇数, 则拒绝.

4) 从带左端隔一个0, 删一个0. 转(2)."

图灵机的输入

- 由定义, TM的输入总是字符串.
- 有时候要输入数, 图, 或图灵机等对象.
  那么要将对象编码成字符串.
- 记对象O的编码为<O>.
- 本课程中一般不关心实际编码方式.
  数: 可取二进制,十进制,或其它编码.
  图: 例如左边的图可以编码为:
  G=(1,2,3,4)((1,2),(2,3),(3,1),(1,4))
- 特别的, 图灵机是有向带权图
  也可以编码为字符串.



输入为对象的图灵机举例

M_1="对于输入<G>, G是一个无向图,

1) 选择G的一个顶点, 并做标记.

2) 重复如下步骤, 直到没有新标记出现.

3) 对于G的每个未标记顶点, 若有边将它连接到已标记顶点, 则标记它.

4) 若G的所有顶点已标记, 则接受; 否则, 拒绝."

分析M_1的语言可知:

$L(M_1)=\{<G> \mid G\text{是连通的无向图}\}$

1. 图灵机基础

2. 图灵机的变形

图灵机的变形

图灵机有多种变形:

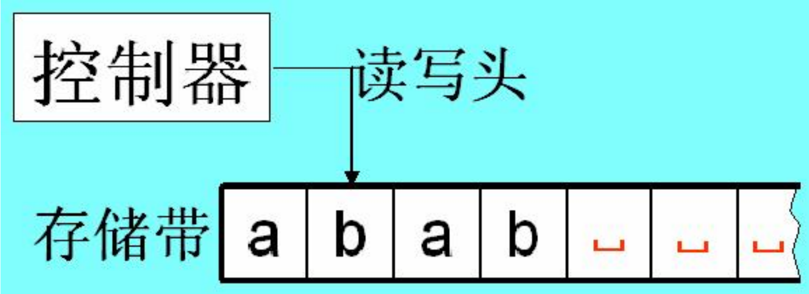
例如多带图灵机, 非确定图灵机

还有如枚举器, 带停留的图灵机等等

只要满足必要特征,

它们都与这里定义的图灵机等价.

非确定型图灵机(NTM)



- NTM的转移函数

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

- NTM转移函数举例

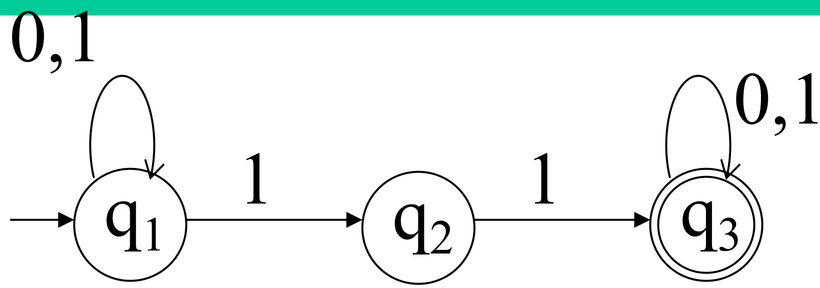
  $\delta(q_3,0)=\{(q_2,x,R), (q_1,1,L), (q_3,\$,R)\}$

- 称NTM M接受x, 若在x上运行M时有接受分支.
- 称一NTM为判定的,

  若它对所有输入,所有分支都停机.

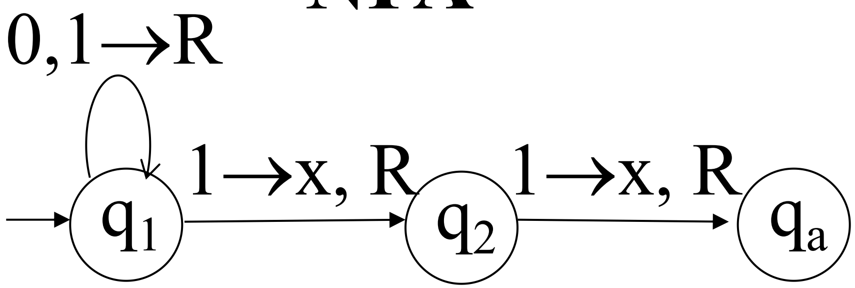
- 定理: 每个NTM都有等价的确定TM.
- 定理: 每个判定NTM都有等价的判定TM.

举例



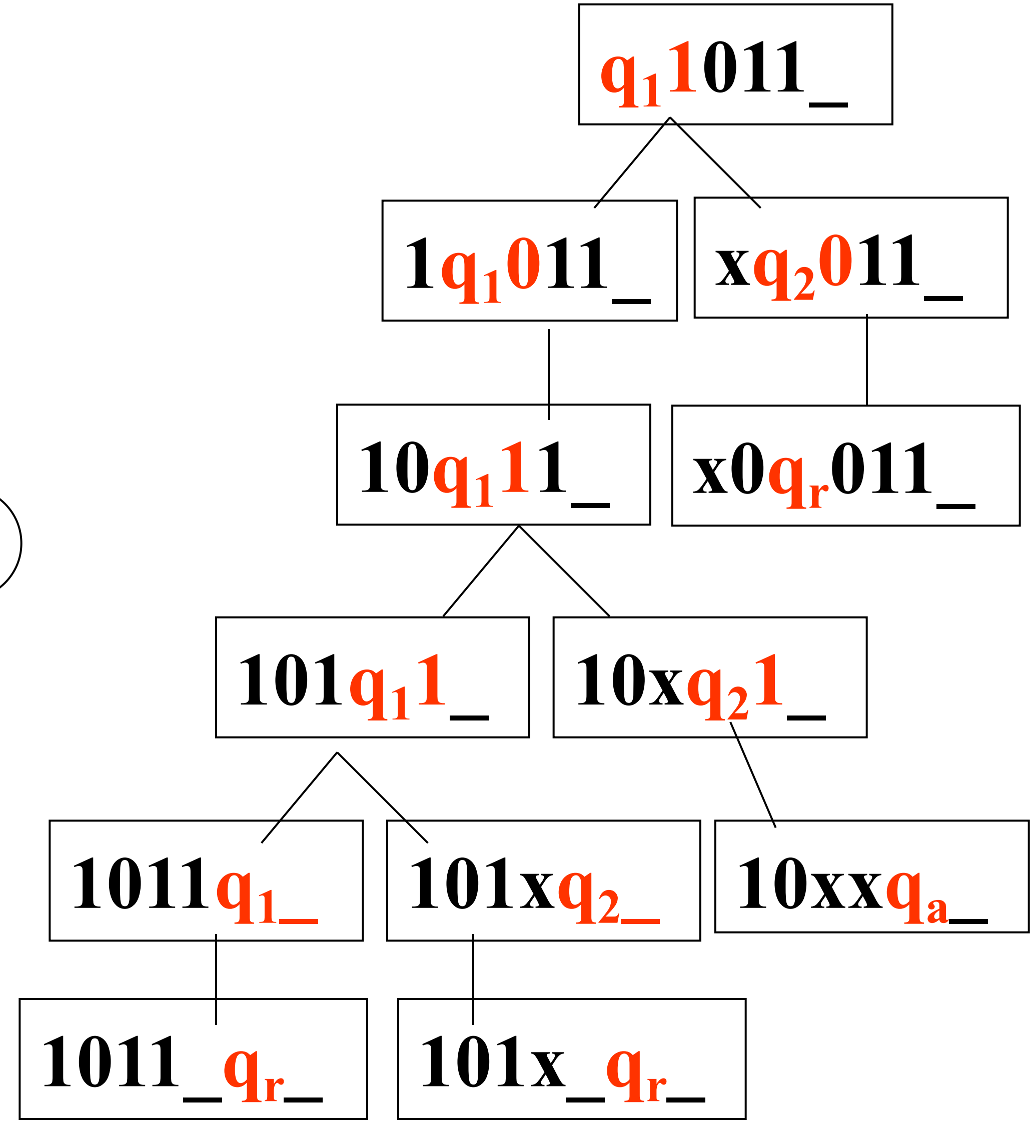
0,1
1
1
0,1
q_1
q_2
q_3

NFA



0,1→R
1→x, R
1→x, R
q_1
q_2
q_a

NTM



$q_1$1011_
1$q_1$011_
x$q_2$011_
10$q_1$11_
x0q_r011_
101$q_1$1_
10x$q_2$1_
1011q_1_
101xq_2_
10xxq_a_
1011_q_r_
101x_q_r_

计算理论第1章作业

1.1 下图给出了两台DFA M_1和M_2的状态图。

回答下述关于这两台机器的问题。

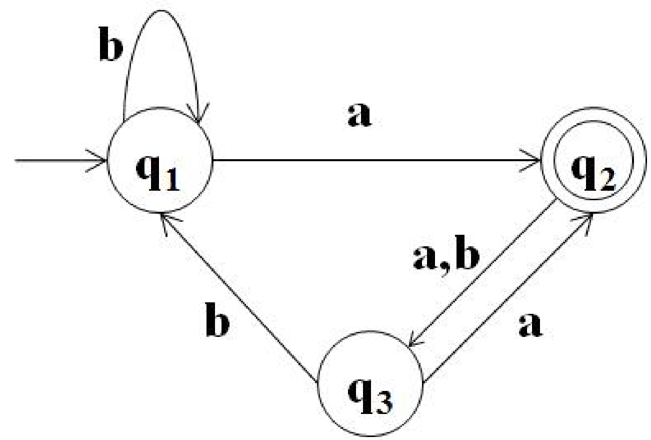
a. 它们的起始状态是什么?

b. 它们的接受状态集是什么?

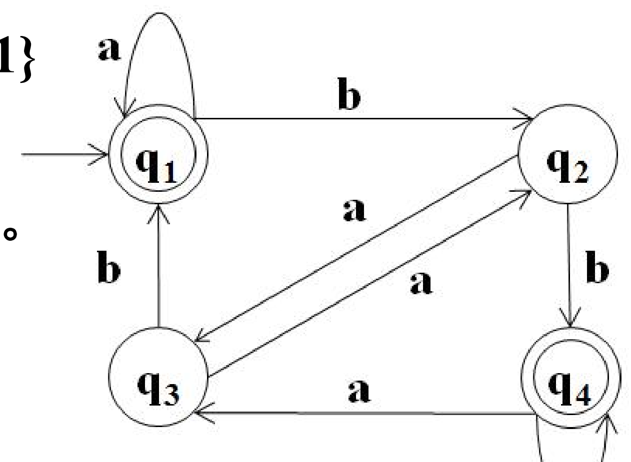
c. 对输入aabb，它们经过的状态序列是什么?

d. 它们接受字符串aabb吗?

e.它们接受字符串ε吗?



M_1



M_2

1.6 画出识别下述语言的DFA状态图。字母表为{0,1}

d. { w | w的长度不小于3, 并且第3个符号为0};

1.7. 给出下述语言的NFA，并且符合规定的状态数。

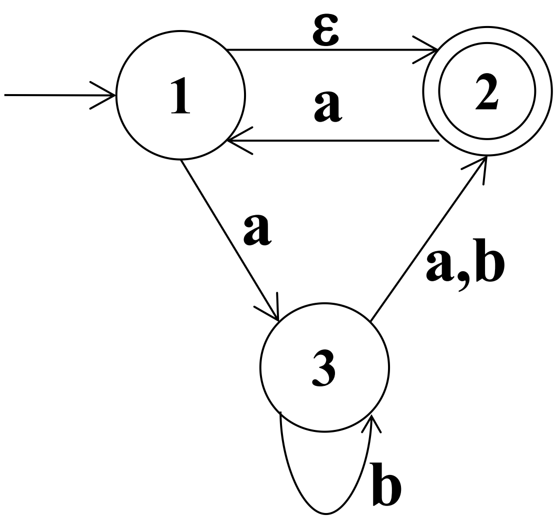
字母表为{0,1}

e. 语言$0^*1^*0^*0$, 3个状态。

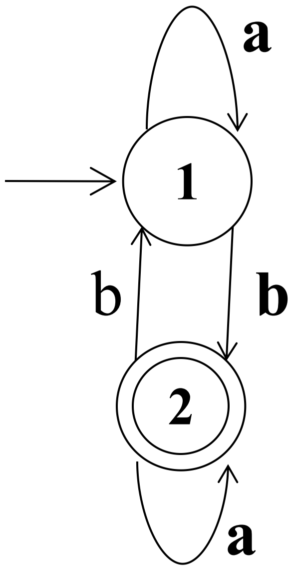
计算理论第1章作业

1.16(b) 将如右图的非确定有限自动机转换成等价的确定有限自动机.

1.21(a) 将如右图的有限自动机转换成等价的正则表达式.



1.16(b)题图



1.21(a)题图

计算理论第1章作业

1.22 在某些程序设计语言中, 注释出现在两个分隔符之间, 如/#和#/. 设C是所有有效注释串形成的语言. C中的成员必须以/#开始, /#结束, 并且在开始和结束之间没有#/. 为简便起见, 所有注释都由符号a和b写成; 因此C的字母表 $\Sigma=\{a, b, /, \#\}$.

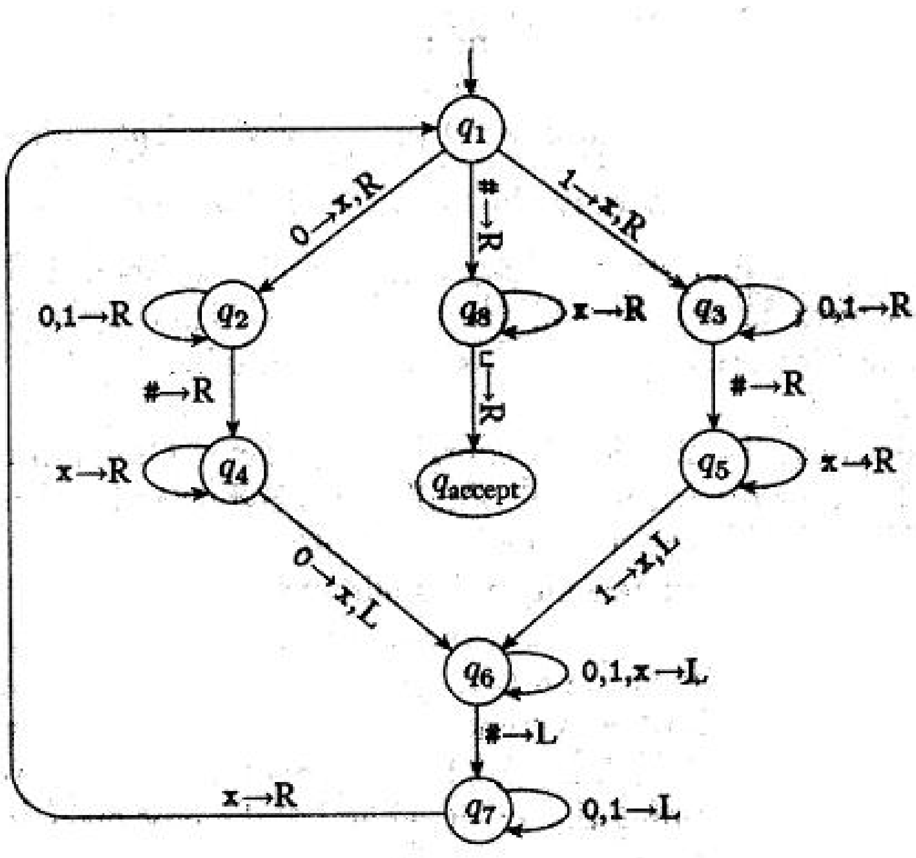
a. 给出识别C的DFA

b. 给出生成C的正则表达式.

1.29 使用泵引理证明下述语言不是正则的。

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

计算理论第3章作业



补充说明: 没有画出的箭头指向拒绝状态

3.2 对于识别{w|w=u#u, u∈{0,1}*}的图灵机M_1 (见左图), 在下列输入串上, 给出M所进入的格局序列.

c. 1##1, d. 10#11, e. 10#10

3.8 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:

b. {w|w所包含的0的个数是1的个数的两倍}

c. {w|w所包含的0的个数不是1的个数的两倍}

3.15b 证明图灵可判定语言类在连接运算下封闭.

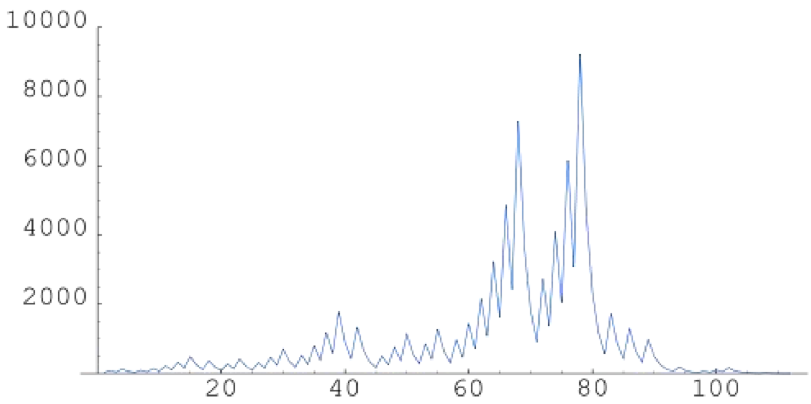
3.16d证明图灵可识别语言类在交运算下封闭.

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

$$|x_0|\le (n+1)\,c_{\max}/|c_1|$$

3n+1问题目前不知道有没有算法

- 输入: 一个正整数n,
- 映射: $f(n) = n/2$, 若n是偶数;
  $f(n) = 3n+1$, 若n是奇数.
- 迭代: 5→16→8→…, 到1则停止
- 输出: n可在f迭代下是否能到1停止
- 直接模拟是正确的算法吗?
- 27需迭代111步(见右图)
- $1 \sim 5 \times 10^{18}$都能到1.([wiki])



不可判定问题(没有算法)举例

Hilbert第十问题: “多项式是否有整数根”有没有算法?

1970’s 被证明不可判定.

M = “对于输入 “p”, p是k元多项式,

1. 取k个整数的向量x (绝对值和从小到大)

2. 若p(x) = 0, 则停机接受.

3. 否则转1. ”

这个图灵机对输入 **p(x,y) = x^2+y^2-3不停机**