计算理论
第一部分 计算模型

[S] 第1章 有限自动机

第2章 上下文无关语言

第3章 图灵机

算法核心: 问题和程序

计算理论研究关于计算的一般理论

如何描述问题？ 语言

机器是指什么？ 算法或程序

第1章 有限自动机

0. 引论--什么是问题

1. 确定有限自动机

2. 非确定有限自动机

3. 正则表达式

4. 正则语言的泵引理

第1章 有限自动机

0. 引论--什么是问题
1. 确定有限自动机
2. 非确定有限自动机
3. 正则表达式
4. 正则语言的泵引理

决定性问题与字符串集合

决定性问题(Dicision Prob): 只需回答是与否的问题

“一数是否是偶数” -----------{ 以0结尾的01串 }

“串长度是否是2的幂次” ---{ 0^{2^n} : $n \geq 0$ }

“图是否连通” ----------------{ <G> | G是连通图 }

其中<G>是图G编码成的字符串.

“图是否有k团”---{<G> | 图G有k团} //等价于最大团问题

给定有限字母表Σ, 例如{0,1}

- 每个输入是一个01串, 任意01串都可以是输入
- 决定性问题一一对应满足某性质的串的集合

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

Σ^*, 语言, 字典序

令字母表 $\Sigma = \{0,1\}$, Σ上的语言举例:

A={0,00,0000}, B={0,00,01,000,001,…}

- Σ上所有有限长串记为Σ^*.
- Σ上的任一语言都是Σ^*的子集.
- Σ^*(字典序): ε, 0, 1, 00, 01, 10, 11, 000, …
- Σ上所有无限长串记为Σ^N.
- Σ上的语言与Σ^N一一对应. //例"是否偶数"

Σ^*	ε	0	1	00	01	10	11	000	001	…
B	×	0	×	00	01	×	×	000	001	…
f(B)	0	1	0	1	1	0	0	1	1	…

等势, 可数, 不可数

- 等势: 若两集合间存在一一对应, 则称它们等势
- 可数: 若集合与有限集或与自然数集等势

  或者说集合元素可以按次序列出
- 不可数: 若集合不是可数的

  或者说集合元素不能按次序列出
- 自然数集可数, 正偶数集可数, $\{0,1\}^*$可数

可数集合举例

- 正有理数集可数

n	1	2	3	4		5	6	7	8	…
f(n)=p/q	1/1	2/1	1/2	3/1	2/2	1/3	4/1	3/2	2/3	…
p+q	2	3	3	4	4	4	5	5	5	…

正有理数集={p/q}, 其中p,q是互素的自然数.

- 给定字母表{0,1}, $\{0,1\}^*$可数.

{0,1}上所有有限长字符串的字典序排列:

ε,0,1,00,01,10,11,000,…

定理 $\{0,1\}^N$不可数

$\{0,1\}^N$是全体无限长的01串

证明: 假设$\{0,1\}^N$可数, 即可以排成一列$(f(i))$

按下面方法在$\{0,1\}^N$中取一点 x ,

x的第i位与$f(i)$的第i位相反

n	f(n)
1	1 1 1 0 1 …
2	0 0 0 0 0 …
3	0 1 1 1 1 …
4	1 1 1 0 0 …
…	…
x	0 1 0 1 …

x在不在列表中

所以(0,1]不可数.

计算理论研究对象:字符串集合

- 等势: 若两集合间存在一一对应, 则称它们等势
- 可数: 若集合与有限集或与自然数集等势
- 不可数: 若集合不是可数的
- 全体程序是$\{0,1\}^*$ 的子集, 至多可数
- 全体决定性问题与$\{0,1\}^N$ 等势, 不可数
- 程序可数, 问题不可数
- 数学的研究对象有数, 函数, 函数空间等
- 计算理论的研究对象: 问题 即 语言 即 字符串集合
- 问题难易程度: 计算模型, 时间复杂度

第1章 有限自动机

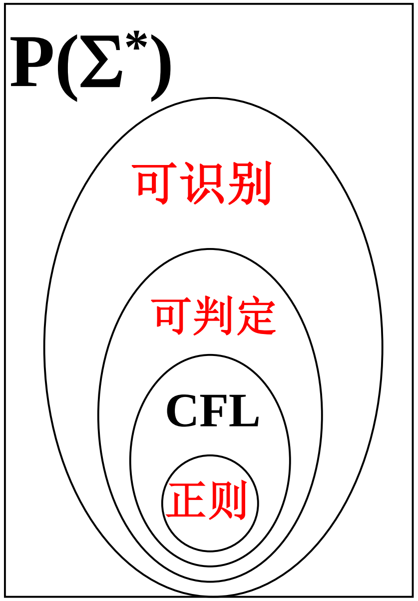
0. 引论--什么是问题

1. 确定有限自动机

2. 非确定有限自动机

3. 正则表达式

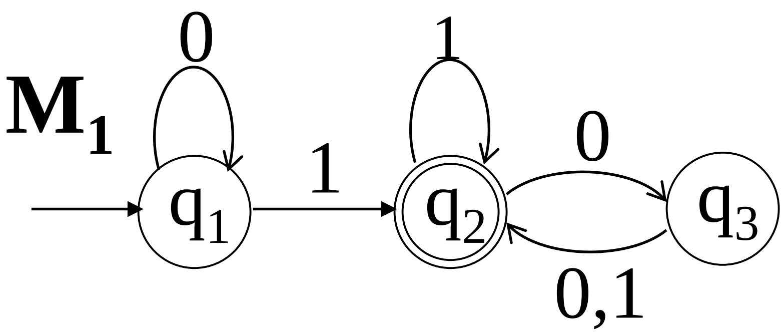
4. 正则语言的泵引理



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) $\delta: Q\times\Sigma\to Q$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

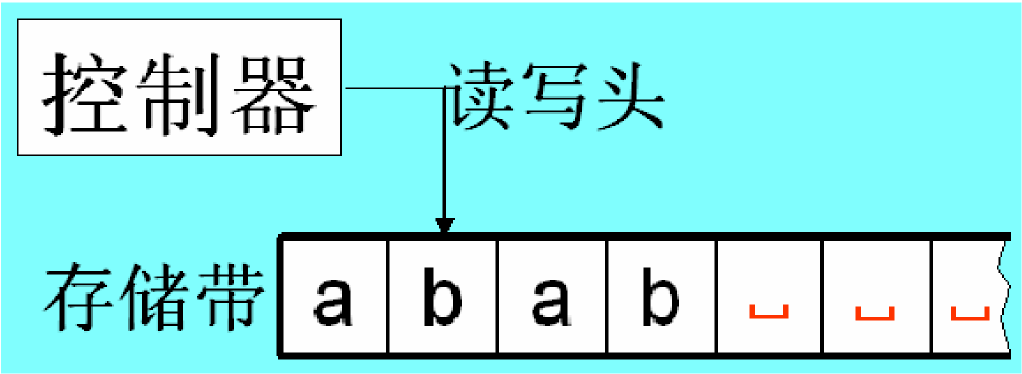
图灵机

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.



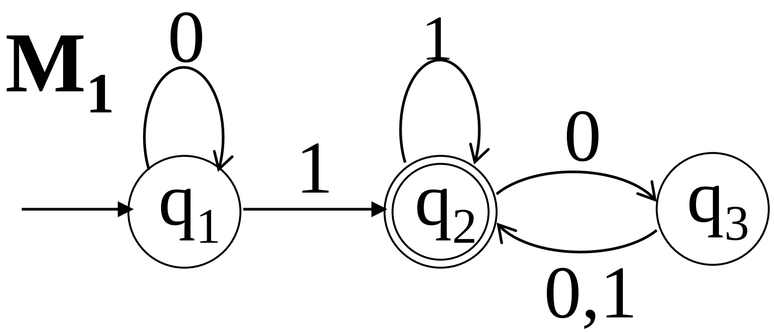
脑:控制器　纸:存储带

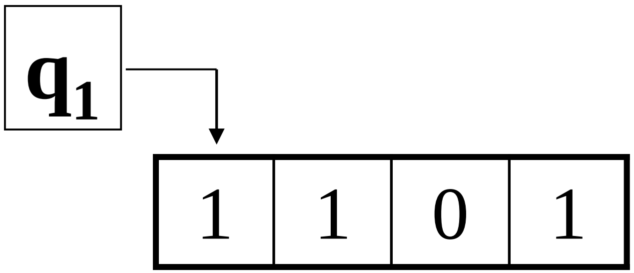
眼睛和笔:读写头

法则:转移函数

有限自动机是简化的图灵机

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计算的定义和描述

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.

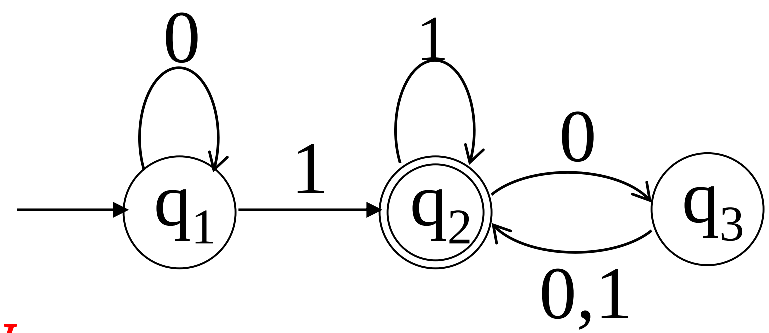
若存在Q中的状态序列$r_0,r_1,\ldots,r_n$, 满足

1) $r_0 = s$;　　　　　　　//从起始状态开始

2) $\delta(r_i, w_{i+1}) = r_{i+1}$; //按转移函数运行

3) $r_n \in F$　　　　　　　//停止于接受状态

则M接受w.



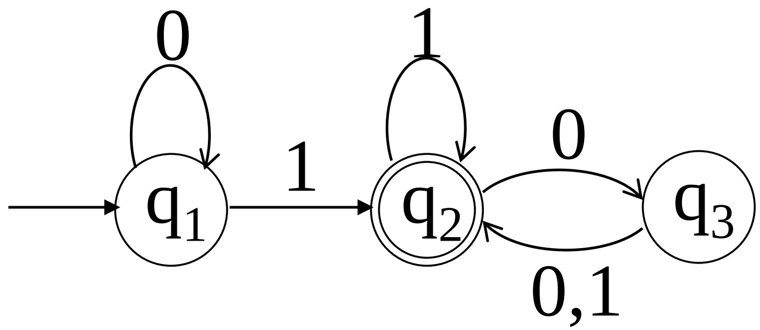
$$s \xrightarrow{w_1} r_1 \xrightarrow{w_2} r_2 \overset{\cdots}{\dashrightarrow} r_{n-1} \xrightarrow{w_n} r_n$$

有限自动机的语言

对有限自动机**M**, 若 **A** = { **w**∈**Σ*** | **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.**

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,**

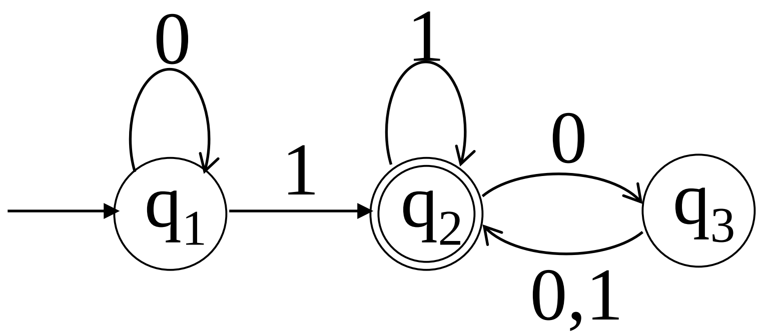
且最后一个**1**后面含有偶数个**0** }

注: **M**的语言唯一? **M**不识别任何其它语言. 比如真子集

注: 给定语言**A**, 设计**DFA**: **A**中串要接受, 非**A**串要拒绝

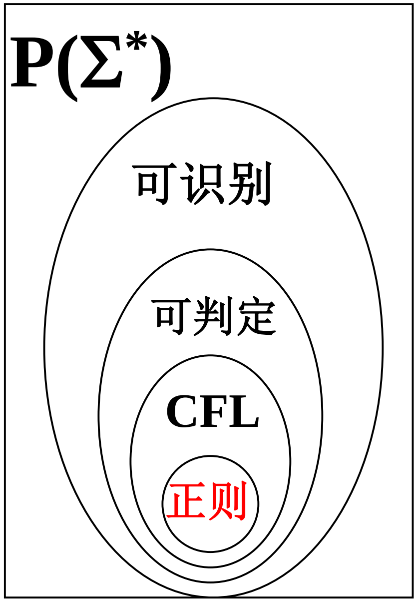
正则语言

$\mathbf{M_1}$



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 }



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.

有限自动机的设计

- 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 $\}$

$\{ w \in \{0,1\}^* \mid$ 二进制数w模3余1 $\}$

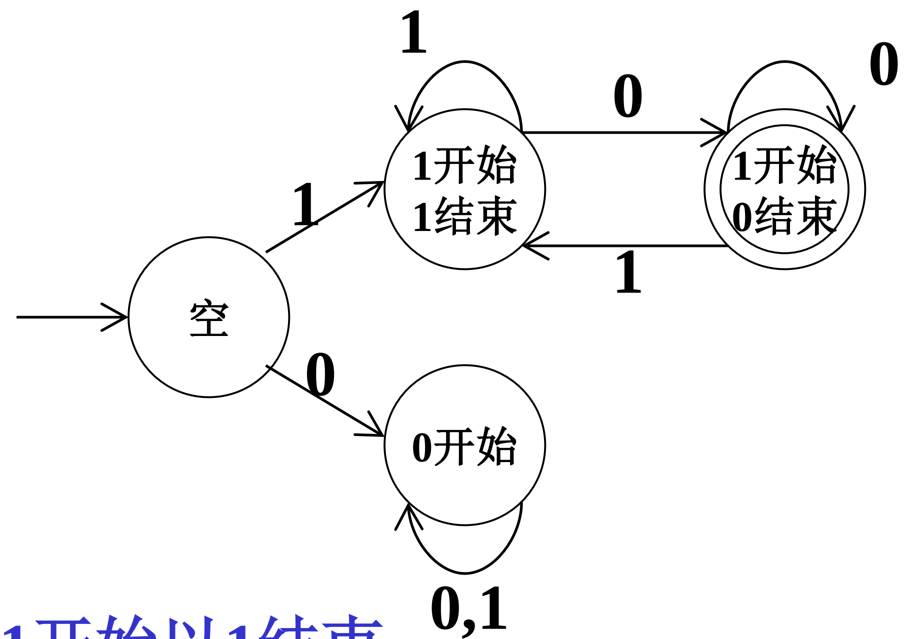
有限自动机的设计和理解

{ 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,

算法:

1. 读输入数组A[1:n]
2. 若A[1]=1,A[n]=0,则返回真
3. 否则返回假



空, 以0开始, 以1开始以0结束, 以1开始以1结束

对应自动机算法:

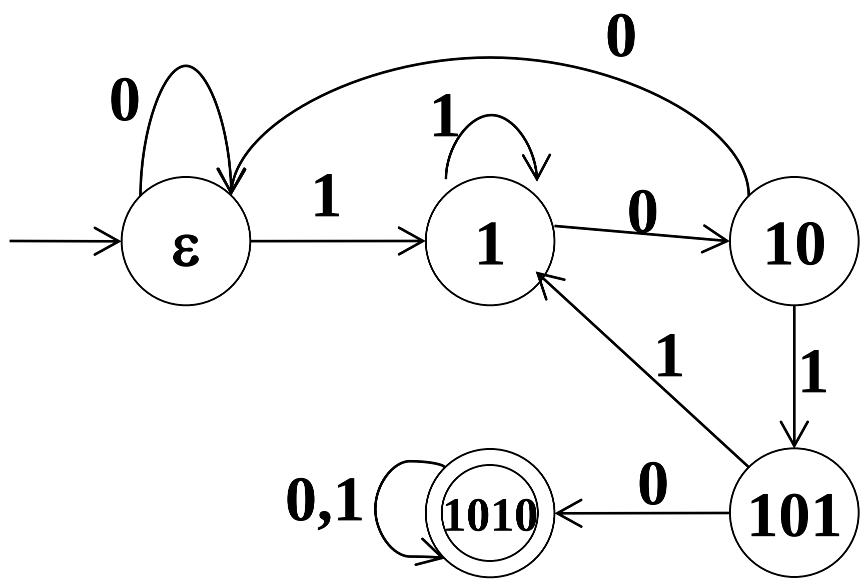
1. 初始状态
2. 当有输入, 根据转移函数转移状态
3. 若当前处于接受状态, 返回真, 否则返回假

运行举例: 1100, 101

有限自动机的设计

{ w∈{0,1}* | w含有子串1010 }

Σ={0,1}，关键信息: ε, 1, 10, 101, 1010



对应自动机算法:

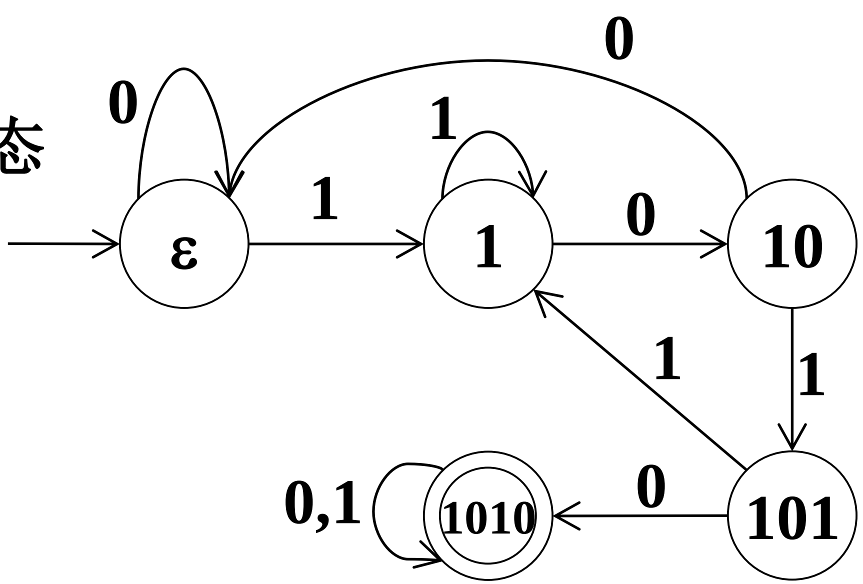
1. 初始状态
2. 当有输入, 根据转移函数转移状态
3. 若当前处于接受状态, 返回真, 否则返回假

运行举例: 1011, 010101

有限自动机对应的算法

对应自动机算法:

1. 初始状态
2. 当有输入, 根据转移函数转移状态
3. 若当前处于接受状态, 返回真,
   否则返回假



记状态分别为0,1,2,3,4.

1. 设初始状态q=0
2. 读字母s, 若 (q,s) =
3. (0,0), 则q=0; (0,1), 则q=1;
4. (1,0), 则q=2; (1,1), 则q=1;
5. (2,0), 则q=0; (2,1), 则q=3;
6. (3,0), 则q=4; (3,1), 则q=1;
7. (4,0), 则q=4; (4,1), 则q=4;
8. 若q==4, 则回答是; 否则回答否.

字符串匹配算法

算法	预处理时间	匹配时间
朴素	**0**	$O(nm)$
自动机	$O(m\|\Sigma\|)$	$\Theta(n)$
Knuth-Morris-Pratt	$\Theta(m)$	$\Theta(n)$

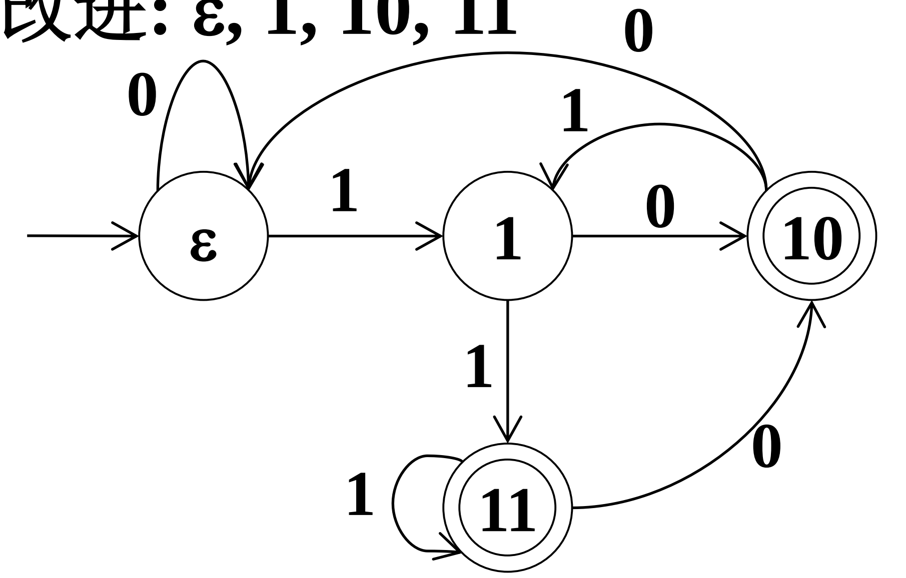
有限自动机的设计

{ w∈{0,1}* | w倒数第2个符号是1 }

只需关注最后两个符号

Σ={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

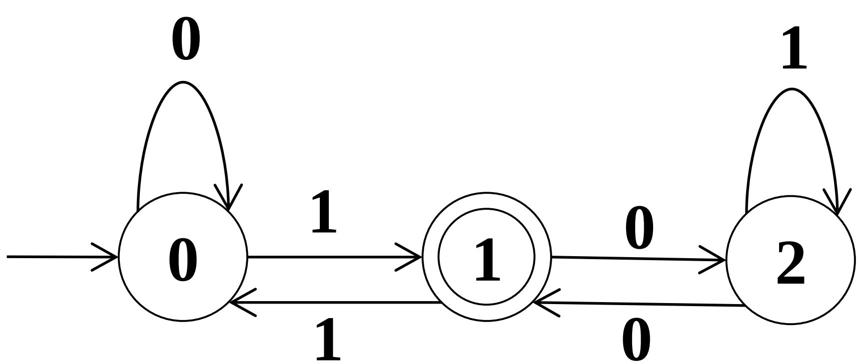
{ w∈{0,1}* | 二进制数w模3余1 }

Σ={0,1}, 关键信息?

模3余0,1,2

起始状态

接受状态



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: $A\cup B = \{x|x\in A或x\in B\}$
- 连接: $A\circ B = \{xy|x\in A且y\in B\}$
- 星号: $A^* = \{x_1x_2\ldots x_k|k\geq 0且每个x_i\in A\}$

定理: 如果A,B正则, 那么A∪B正则

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∪B={ good, bad, boy, girl }

A°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?

第1章 有限自动机

0. 引论--什么是问题

1. 确定有限自动机

2. 非确定有限自动机

3. 正则表达式

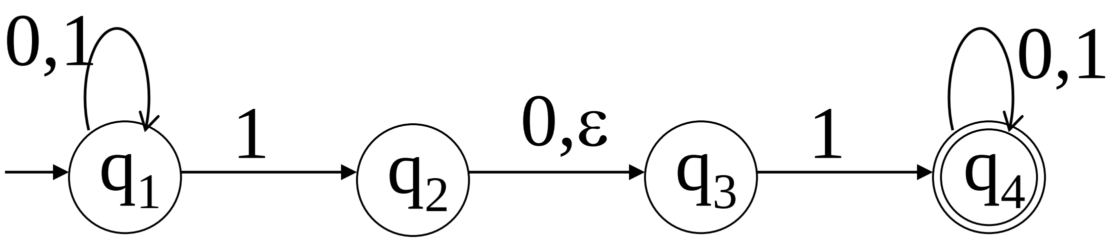
4. 正则语言的泵引理

非确定型机器(难点)

DFA中因为δ: $Q\times\Sigma\rightarrow Q$是一个函数, 所以

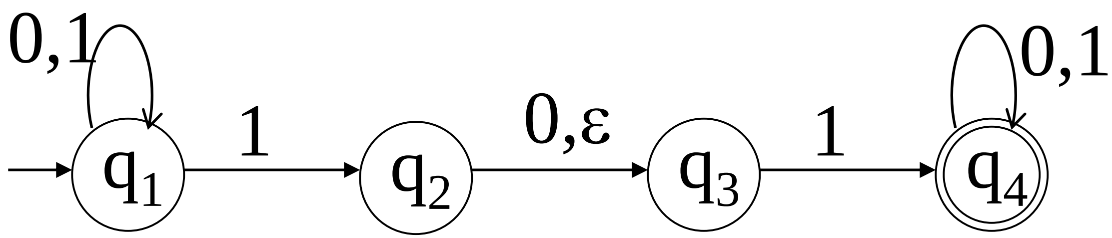
- 每步存在唯一的方式进入下一状态
- 称为确定型有限自动机(**DFA**)

现在引入非确定型有限自动机(**NFA**)

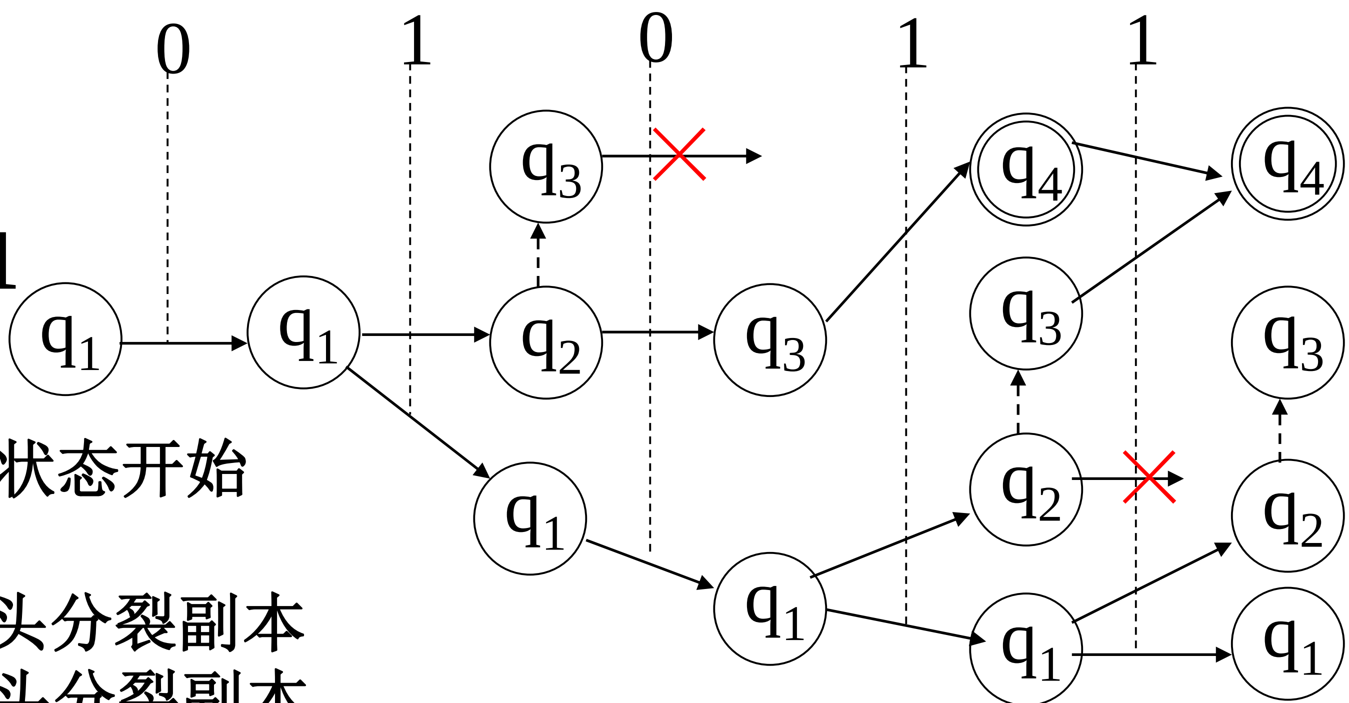


- 每步可以**0**至多种方式进入下一步
- 转移箭头上的符号可以是空串**ε,**
  表示不读任何输入就可以转移过去

非确定型计算



输入
01011



1. 从起始状态开始
2. 读s,
3. 沿s箭头分裂副本
4. 沿ε箭头分裂副本
5. 若副本中有接受状态则接受.

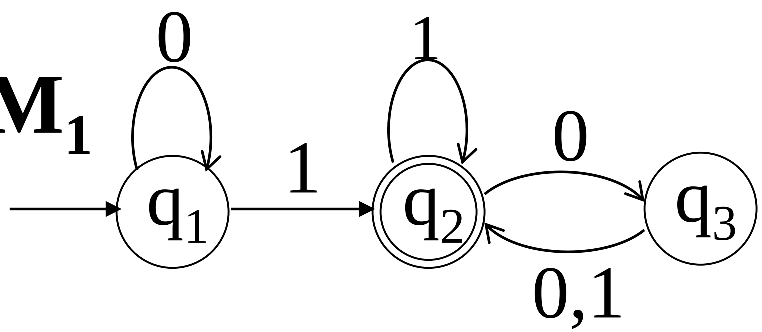
注: 若起始状态有射出的ε箭头? 定义NFA? 定义计算? 转化DFA?

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;

M_1



- 状态图等价于形式定义

$Q=\{q_1,q_2,q_3\}$, 状态集

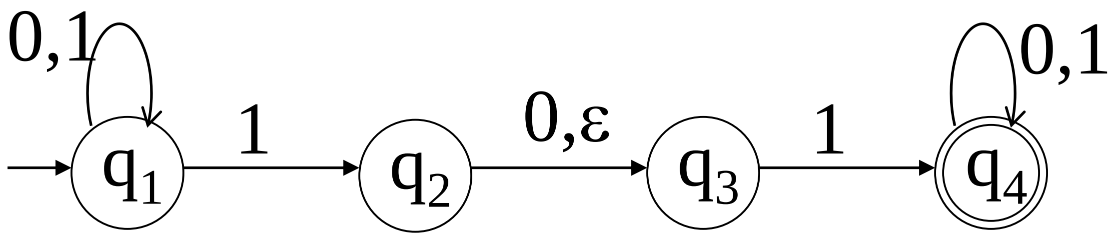
$\Sigma=\{0,1\}$, 字母表

$s=q_1$, 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

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

4) $s\in Q$是起始状态;

5) $F\subseteq Q$是接受状态集;

注:$\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$, $P(Q)$: Q的全体子集

试写出该状态图对应的形式定义

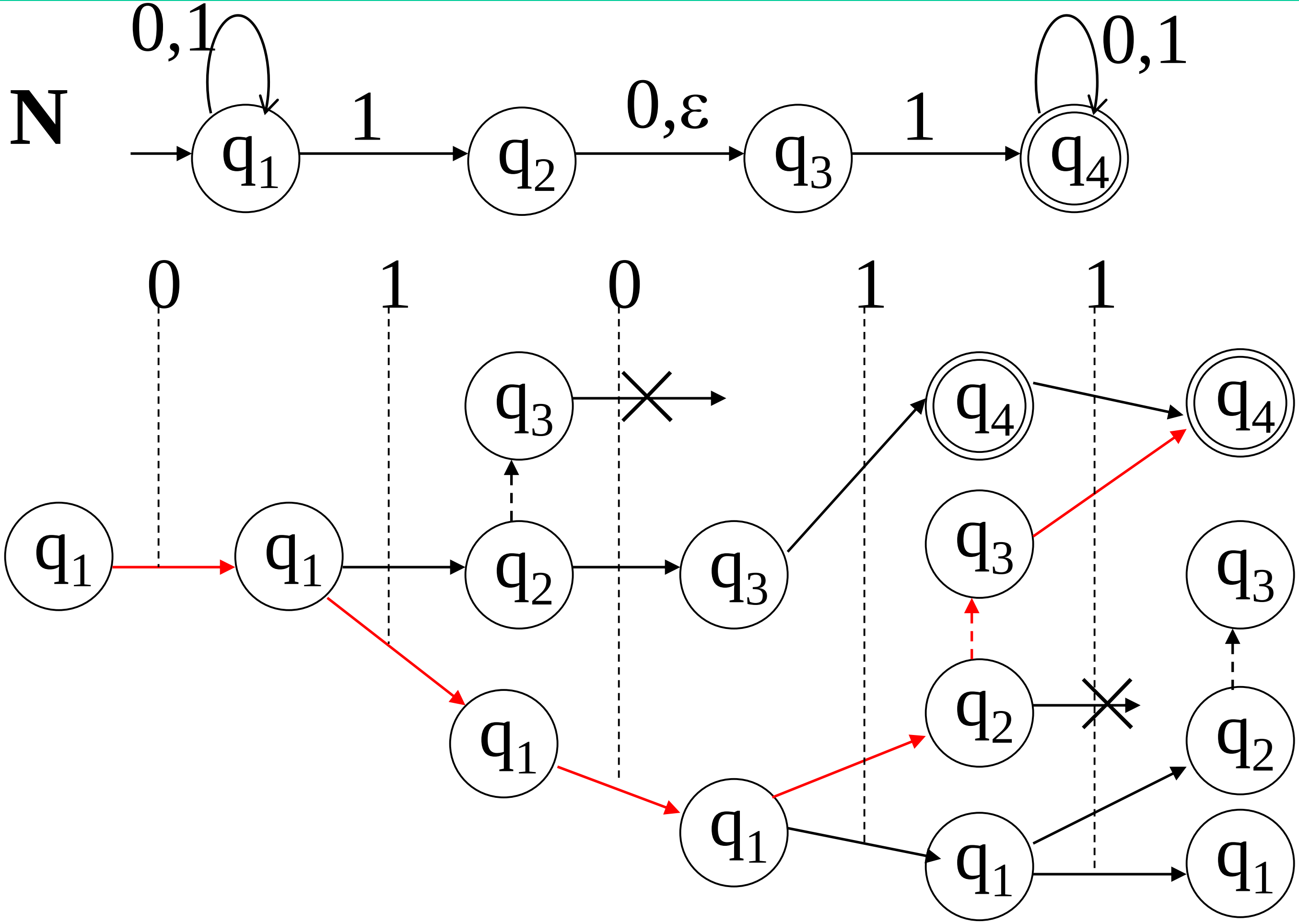
$\delta(q_1,1) = \{q_1,q_2\}$

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

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

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

如何定义NFA的计算



N
0,1
1
0,ε
1
0,1
q_1
q_2
q_3
q_4
0
1
0
1
1
q_1
q_1
q_2
q_3
q_1
q_3
q_1
q_4
q_3
q_2
q_1
q_4
q_3
q_2
q_1

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) $r_0 = q_0$;

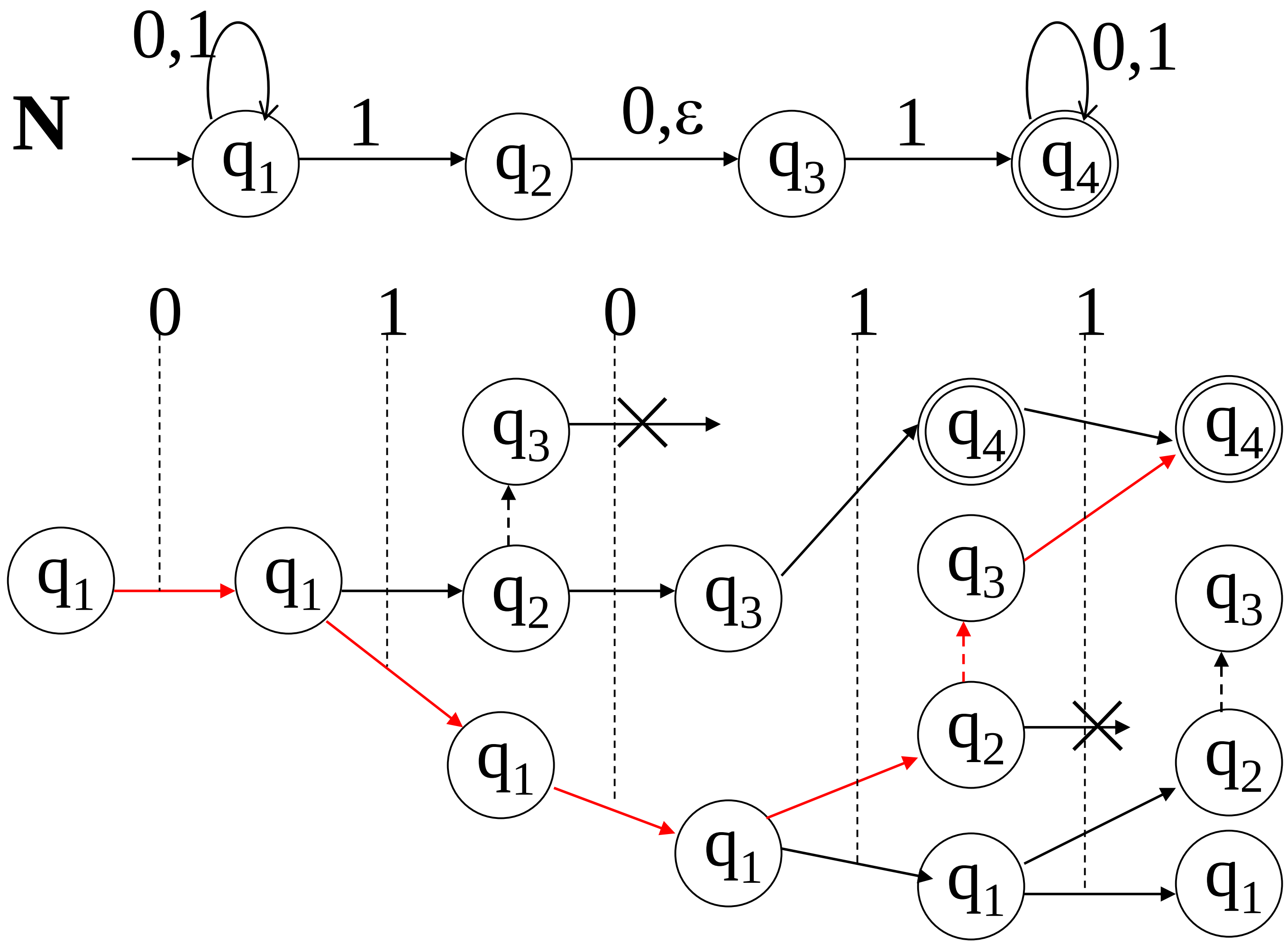
2) $r_{i+1}\in\delta(r_i, w_{i+1})$;

3) $r_n \in F$

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

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计算的形式定义

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

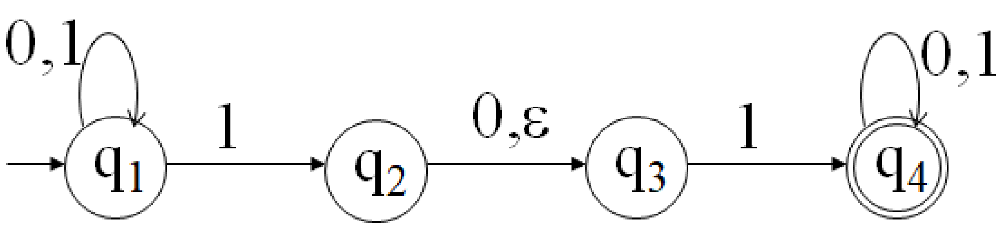
1) $r_0 = q_0$;

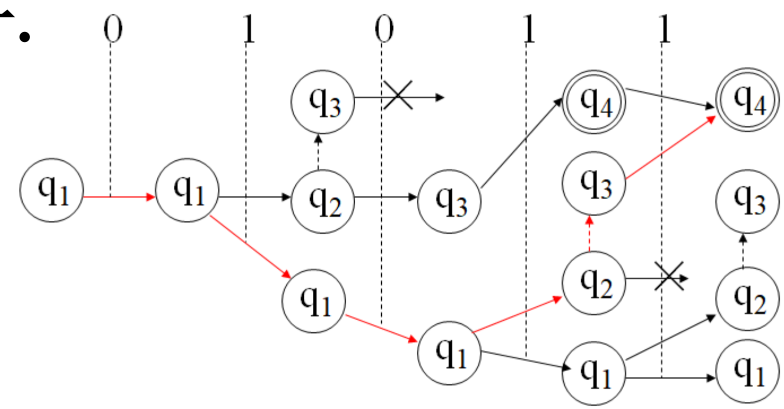
2) $r_{i+1}\in\delta(r_i, w_{i+1})$;

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.





$$w = w_1w_2\ldots w_6 = 0101\varepsilon1$$

$$(r_0,r_1,\ldots,r_6) = (q_1,q_1,q_1,q_1,q_2,q_3,q_4)$$

NFA的设计

- 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 $\}$

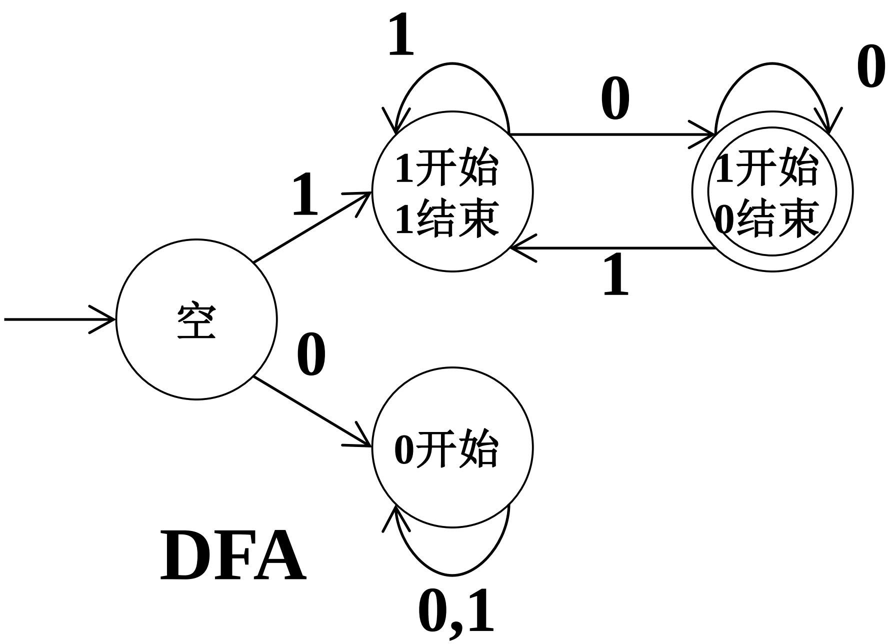
$\{ w \in \{0,1\}^* \mid$ 二进制数w模3余1 $\}$

NFA的设计

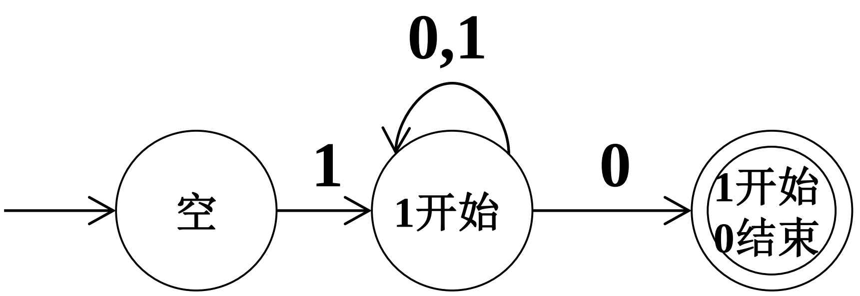
{ w∈{0,1}* | w从1开始, 以0结束 }

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

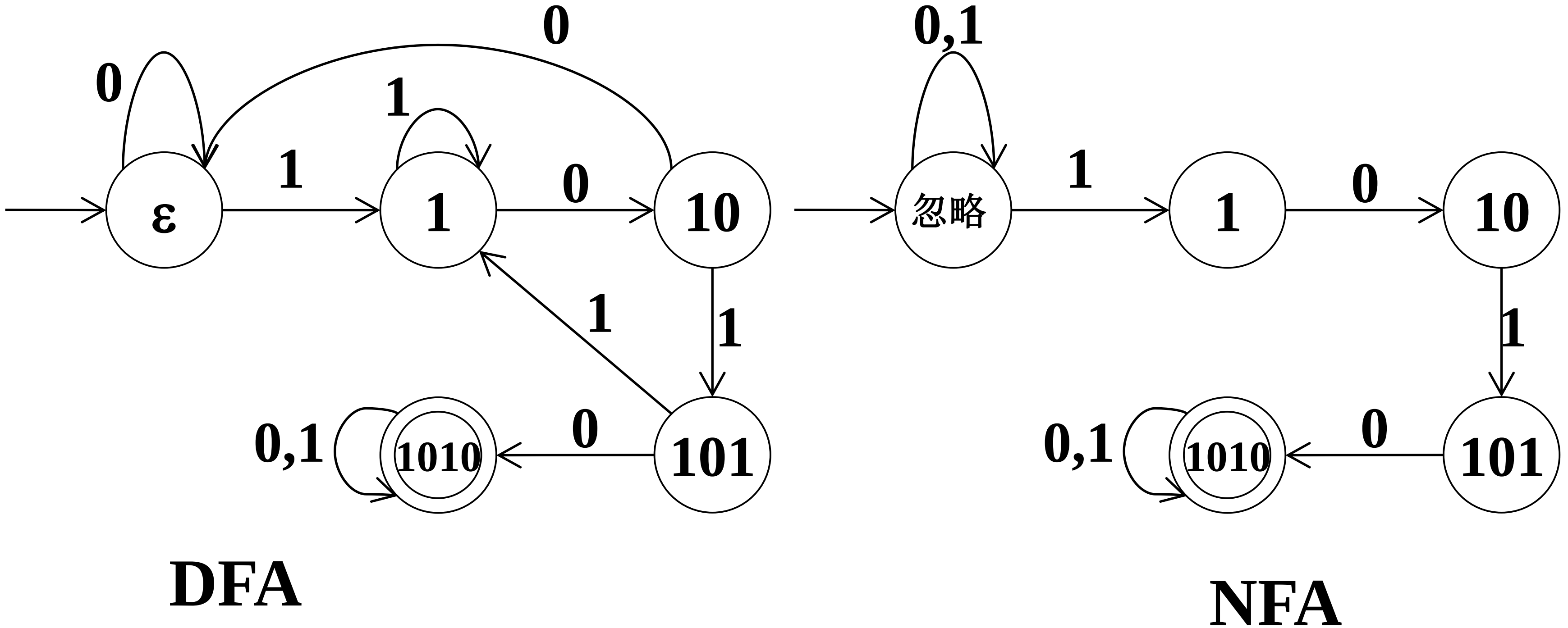


NFA

NFA的设计

{ w∈{0,1}* | w含有子串1010 }

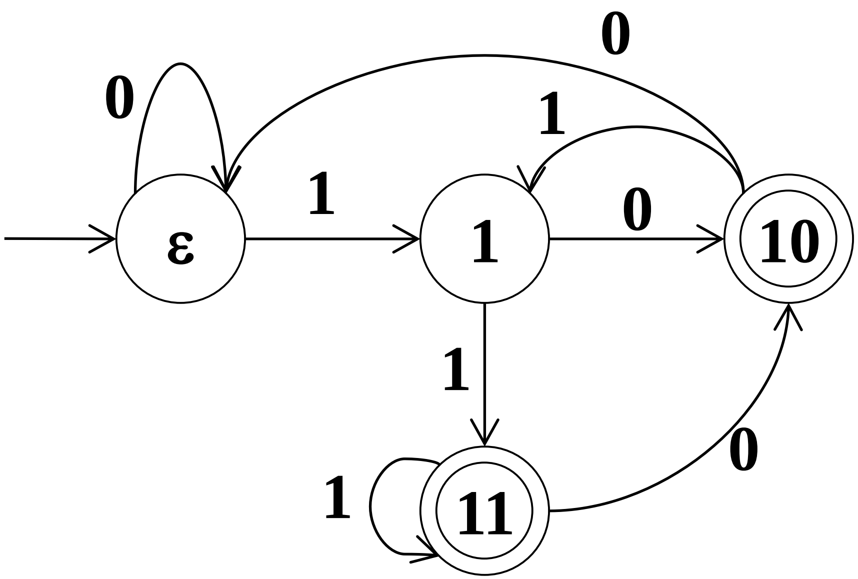
Σ={0,1}，关键信息: 忽略(ε), 1, 10, 101, 1010



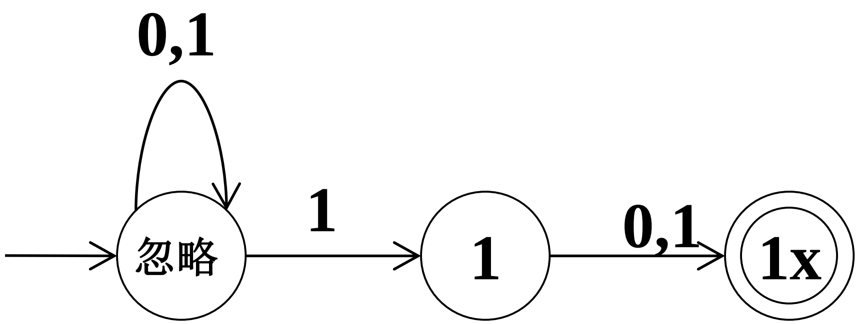
NFA的设计

{ 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,



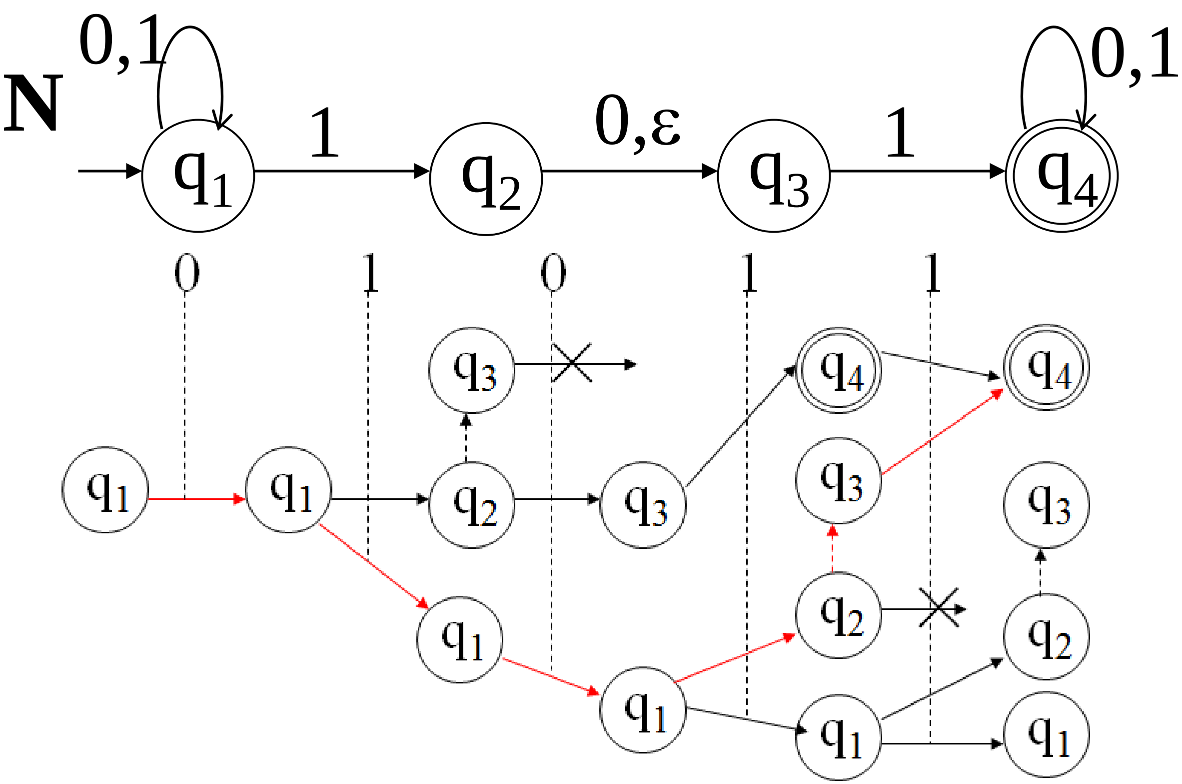
DFA



NFA

NFA与DFA等价

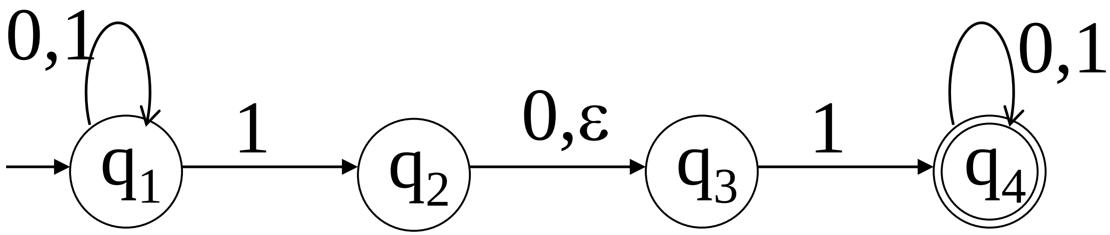
定理: 每个NFA都有一台等价的DFA.



困难在于不确定:
在状态q读到a,
进入哪个状态?
寻找确定:
在状态q读到a,
进入哪些状态?
进一步寻找确定:
给定副本状态集,
读到符号a,
得到的副本状态集

构造DFA关键信息: 副本状态的集合.
转移? 起始? 接受状态集?

每个NFA都有等价的DFA



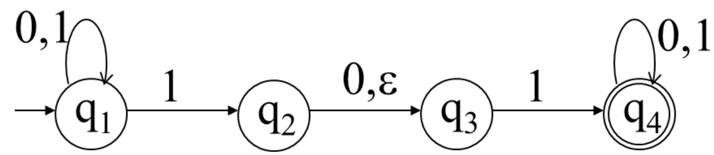
以原状态的子集为新机器的状态

编号	δ	0	1
1	$\{q_1\}$	$\{q_1\}$	$\{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, //依次设计 Q, F, s, δ

令 $Q = P(Q_1)$, //Q_1的全体子集



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

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

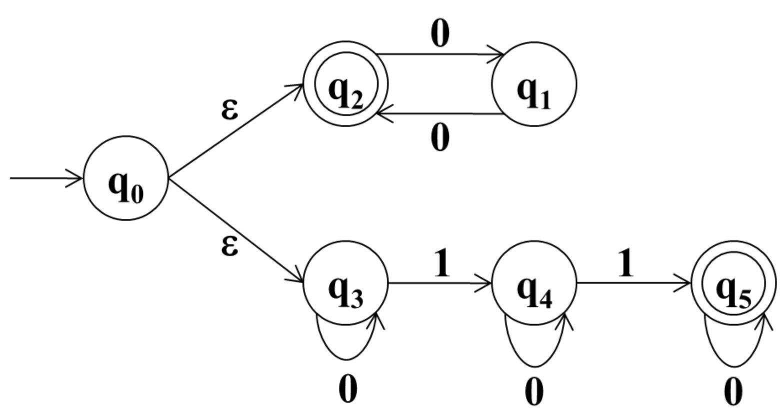
$\delta: Q \times \Sigma \to Q$, $\forall a \in \Sigma, \forall A \in Q$,

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

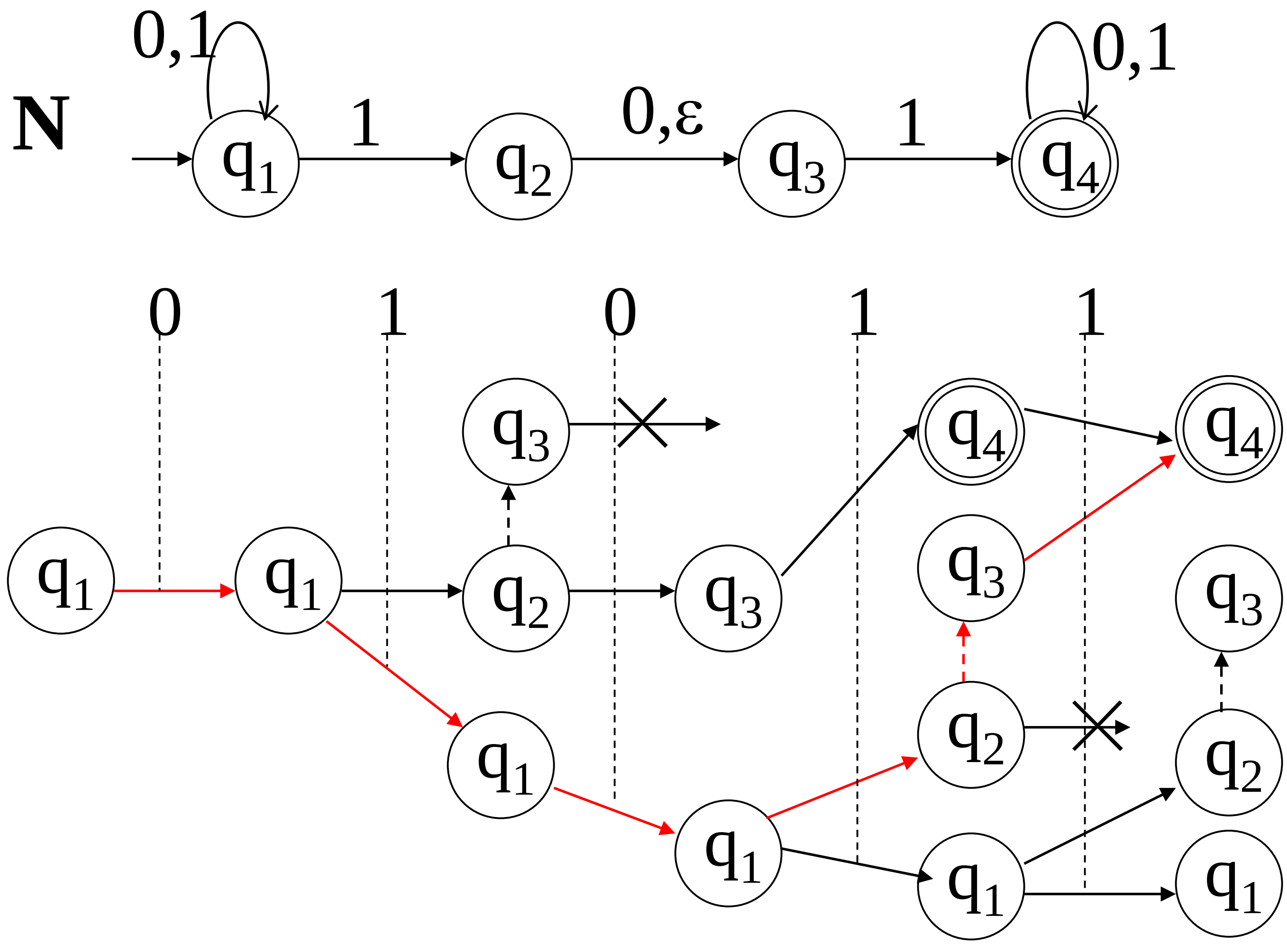
$M = (Q, \Sigma, \delta, s, F)$,

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

即 $L(M) = L(N)$. 证毕



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



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

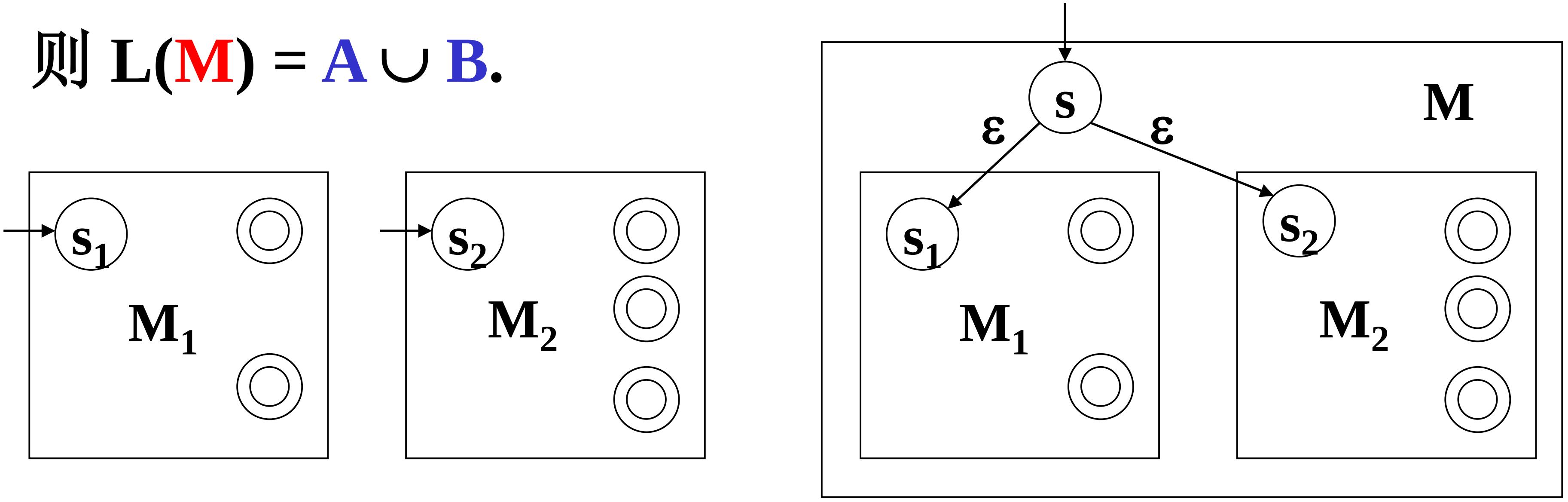
令 $Q = Q_1 \cup Q_2 \cup \{s\}$ 不交并, $F = F_1 \cup F_2$ 不交并

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

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

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

则 $L(M) = A \cup B$.



证明若A, B正则, 则A°B正则

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

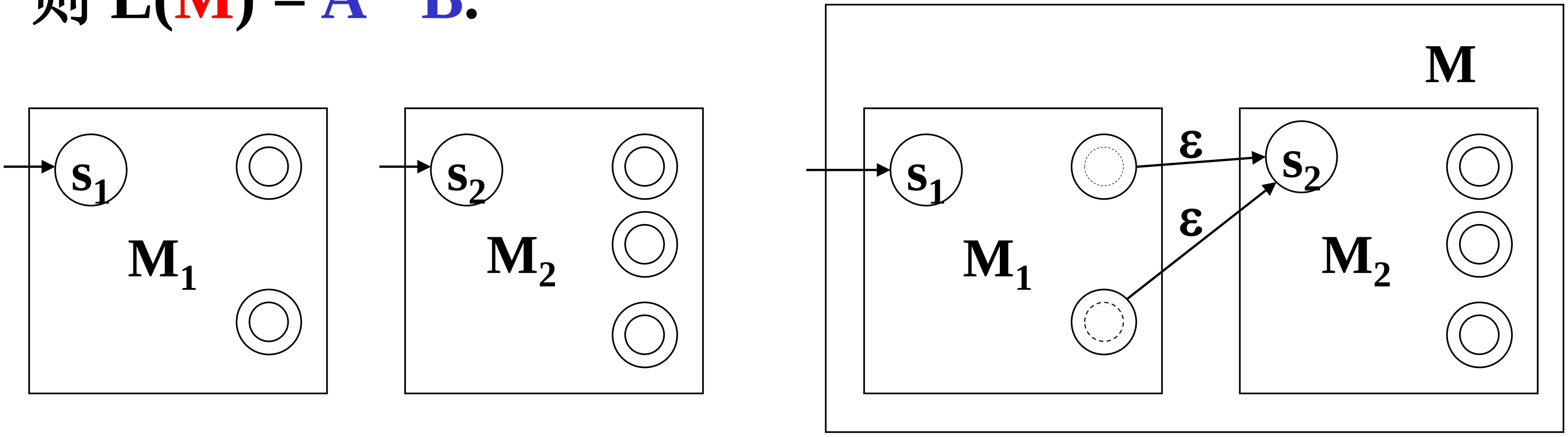
令 $Q = Q_1 \cup Q_2$ 不交并, $F = F_2$,

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

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

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

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

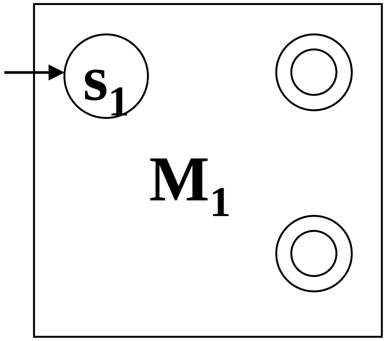
$\forall r\in Q_1, \forall a\in\Sigma,\ \delta(r,a) = \{\delta_1(r,a)\}$

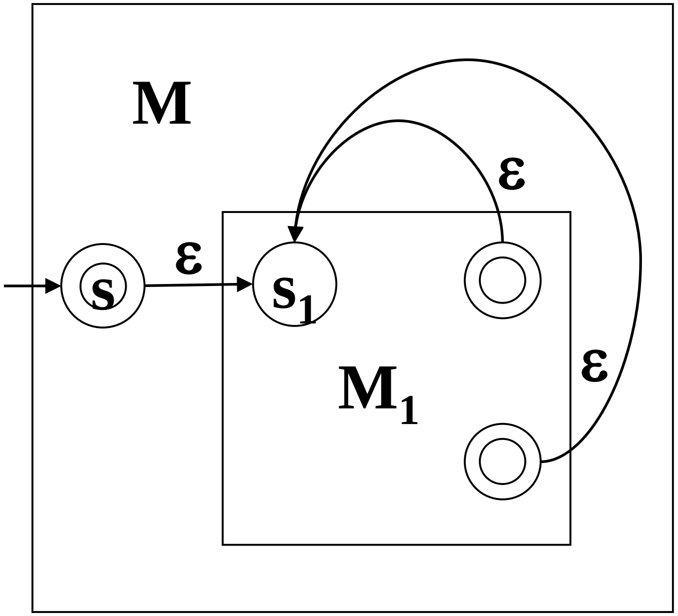
$\forall r\in F_1, \delta(r,\varepsilon) = \{s_1\}$,

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

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

则 $L(M) = A^*$.





第1章 有限自动机

0. 引论--什么是问题
1. 确定有限自动机
2. 非确定有限自动机
3. 正则表达式
4. 正则语言的泵引理

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,

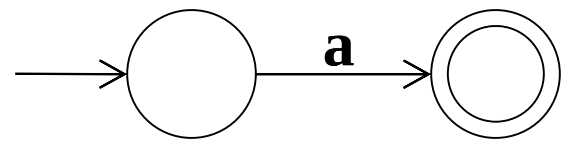
则A可用正则表达式描述. (困难)

A有正则表达式⇒A正则

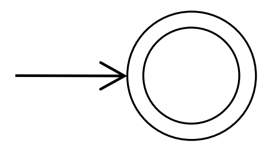
数学归纳法

R是一个正则表达式, 若R是

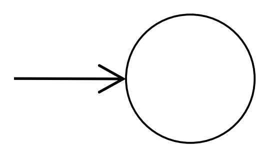
1) $a, a \in \Sigma$



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:

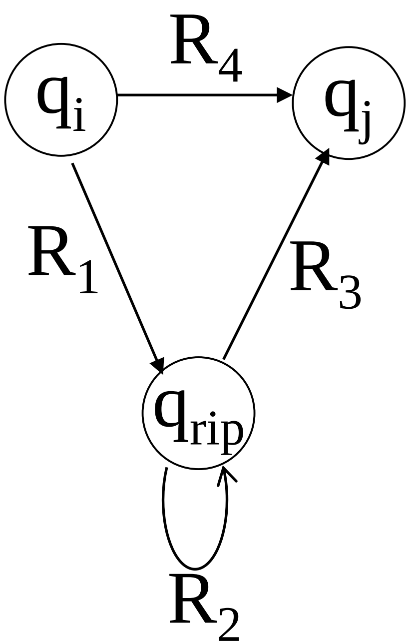
- 起始状态无射入箭头.
- 唯一接受状态且接受状态无射出箭头.

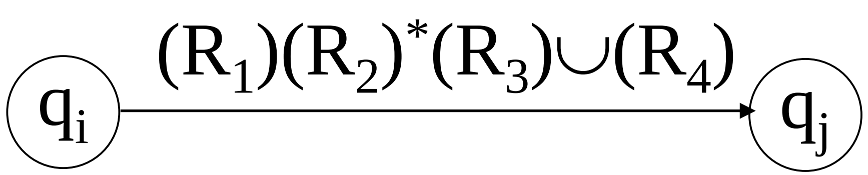
方法: 一个一个地去掉中间状态.

删除一个中间状态

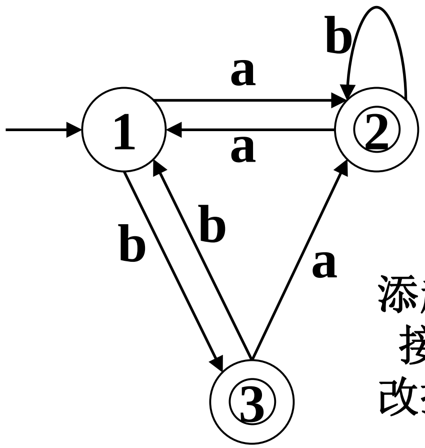
设q_{rip}为待删中间状态,

对任意两个状态q_i, q_j都需要修改箭头标号

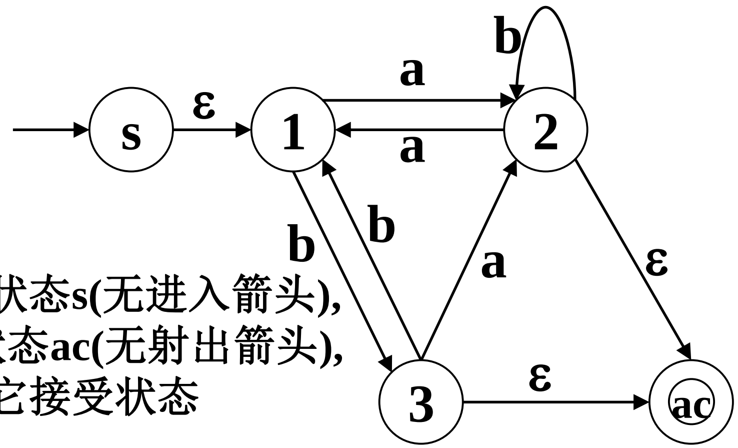




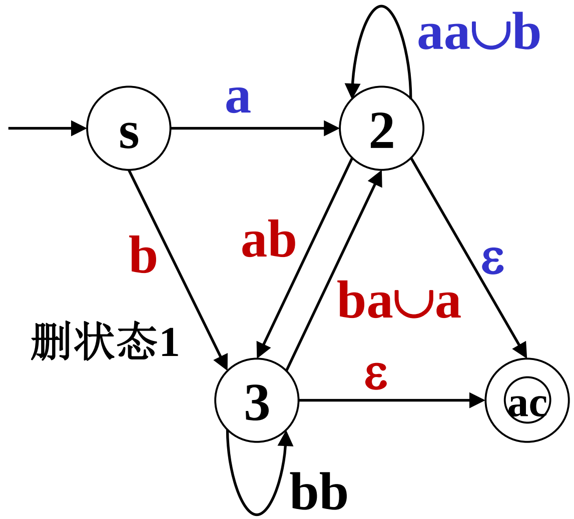
举例: A正则⇒A有正则表达式



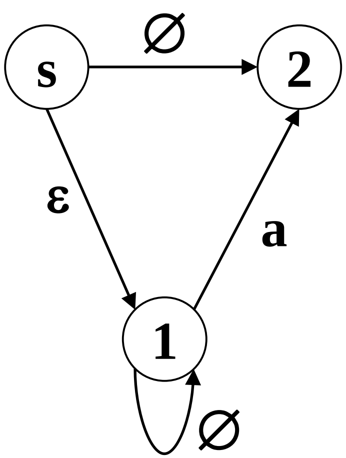
添起始状态s(无进入箭头),
接受状态ac(无射出箭头),
改掉其它接受状态



删状态1

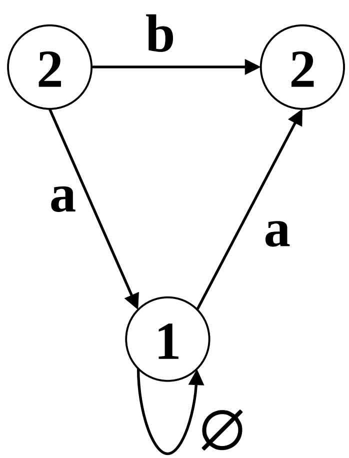


在s,2间去掉1的影响



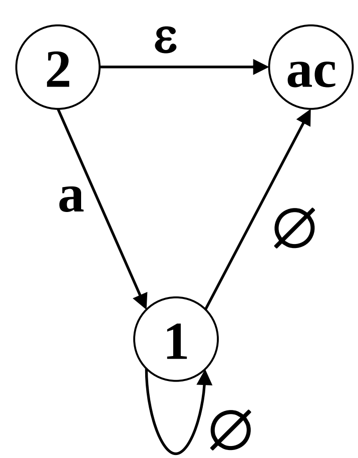
$\varnothing \cup (\varepsilon\varnothing^{*}a) = a$

在2,2间去掉1的影响



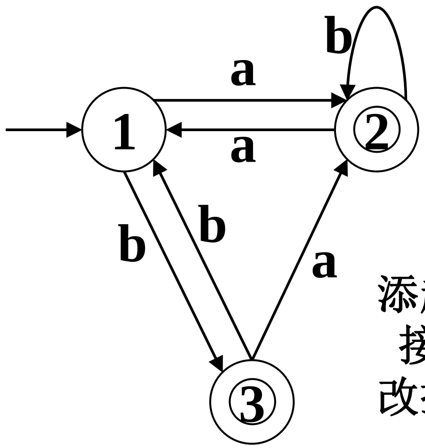
$b\cup(a\varnothing^{*}a)=b\cup aa$

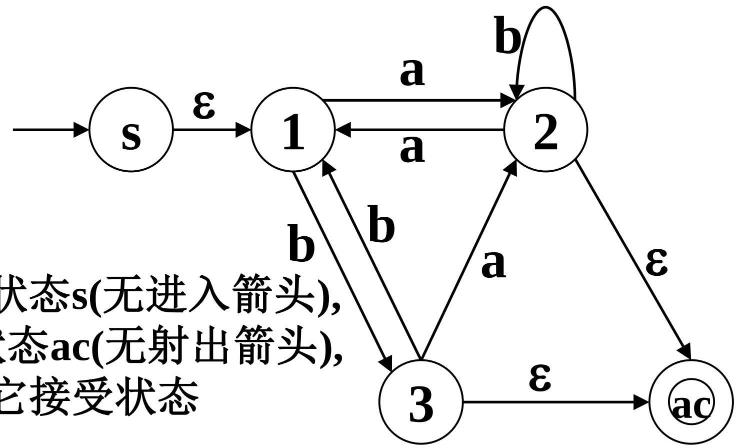
在2,ac间去掉1的影响



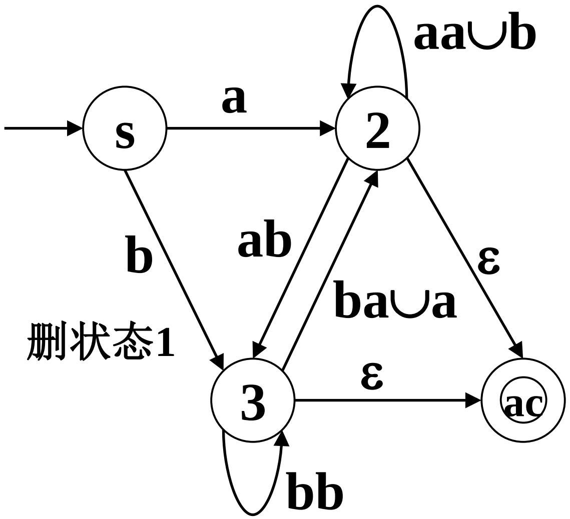
$\varepsilon\cup (a\varnothing^{*}\varnothing) = \varepsilon$

举例: A正则⇒A有正则表达式

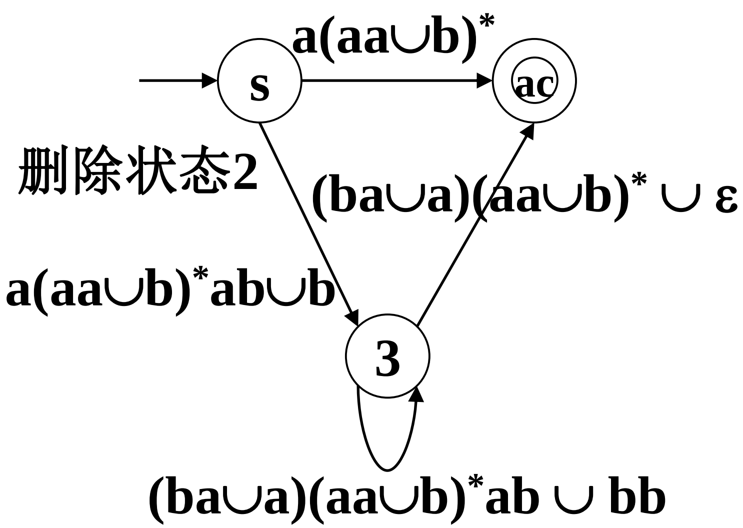




添起始状态s(无进入箭头),
接受状态ac(无射出箭头),
改掉其它接受状态

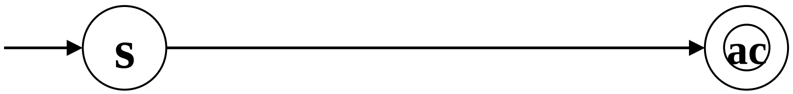


删状态1



删除状态2

$$((a(aa\cup b)^*ab\cup b)((ba\cup a)(aa\cup b)^*ab \cup bb)^* ((ba\cup a)(aa\cup b)^*\cup\varepsilon)) \cup (a(aa\cup b)^*)$$



删除状态3

第1章 有限自动机

0. 引论--什么是问题

1. 确定有限自动机

2. 非确定有限自动机

3. 正则表达式

4. 正则语言的泵引理

非正则语言

下面哪些是正则语言?

$B = \{ 0^n1^n \mid n \geq 0 \}$

$C = \{ w \mid w$中0和1的个数相等 $\}$

$D = \{ 1^k \mid k=2^n, n \geq 0 \}$

$E = \{ w \mid w$中01和10的个数相等 $\}$

$= \{ w \mid w$以0开始0结束或1开始1结束 $\}$

泵引理

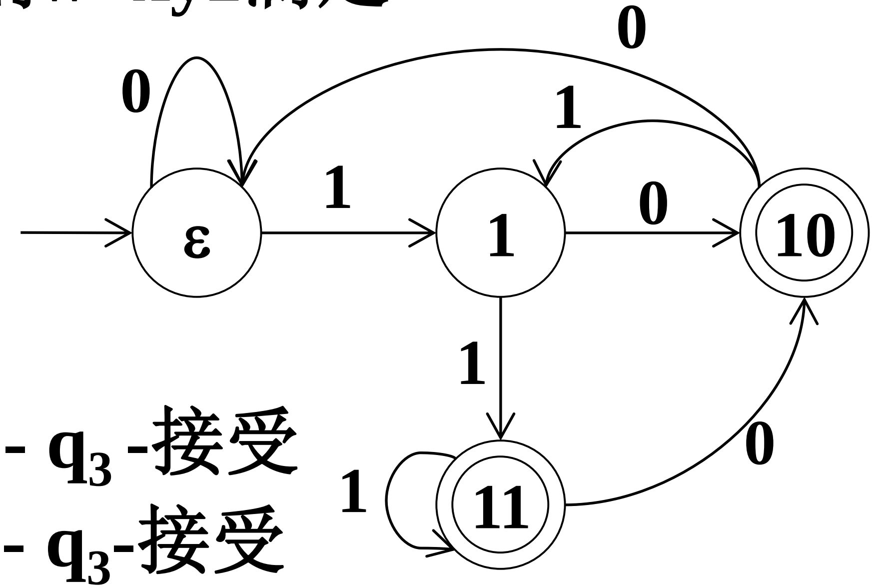
定理(泵引理): 设A是正则语言,则存在p>0使得

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

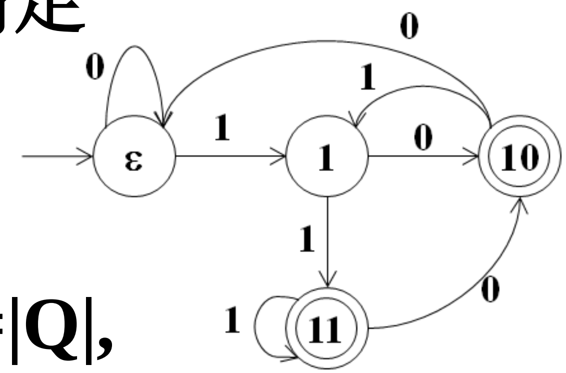
泵引理的证明

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 $w = w_1w_2\ldots w_n\in A$, $w_i\in\Sigma$, 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

$\exists i<j\leq p$, $r_i=r_j$.

令$x=w_1\ldots w_i$, $y=w_{i+1}\ldots w_j$, $z=w_{j+1}\ldots w_n$.

$$s=r_0\xrightarrow{x}r_i\xrightarrow{y}r_i\xrightarrow{z}r_n\in F$$

对$\forall k\geq 0$, $xy^kz\in A$?

$$r_0\xrightarrow{x}r_i\xrightarrow{z}r_n \qquad r_0\xrightarrow{x}r_i\xrightarrow{y}r_i\xrightarrow{\cdots}r_i\xrightarrow{y}r_i\xrightarrow{z}r_n$$

泵引理的等价描述

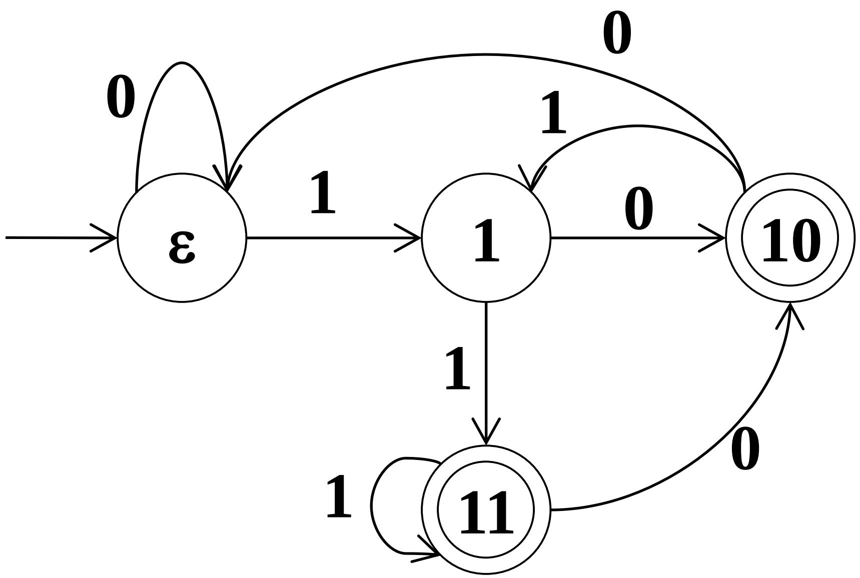
定理(泵引理): 设A是正则语言,则存在p>0使得

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$\exists p>0$

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

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

$\forall i \geq 0$,

$xy^iz \in A$.

泵引理的逆否命题

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

$B = \{ 0^n1^n \mid n \geq 0 \}$ 非正则

∵ $\forall p>0$,

令 $w=0^p1^p$,

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

令 $i=0$,

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

∴ B非正则语言

若 $\forall p>0$

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

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

$\exists i \geq 0$,

$xy^iz \notin A$.

则A非正则语言

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

∵ $\forall p>0$,

令$w=0^p10^p1$,

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

令$i=0$,

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

∴ C非正则语言

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

$D = \{ 1^k \mid k=2^n, n \geq 0 \}$ 非正则

∵ $\forall p>0$,

令$w=1^k$, $k=2^{p+1}$,

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

令$i=2$,

$2^{p+1}<|xy^2z|=k+|y|<2^{p+2}$,

即 $xy^2z \notin D$

∴ D非正则语言

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

本章作业

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。

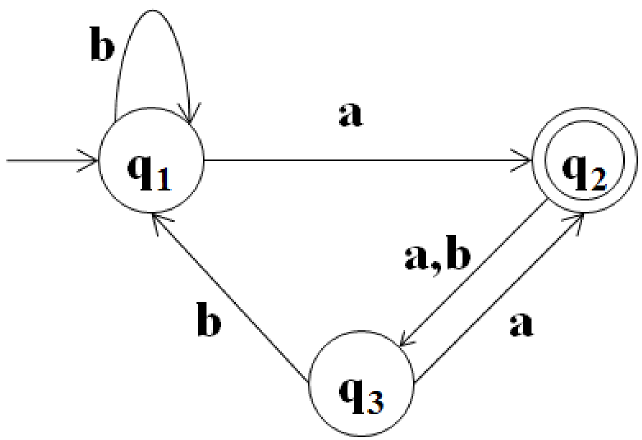
a. 它们的起始状态是什么?

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

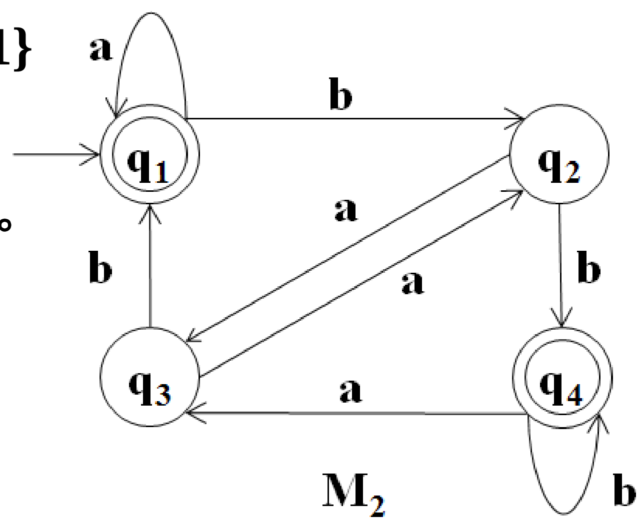
1.6 画出识别下述语言的DFA状态图。字母表为{0,1}

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。



M_2

本章作业

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.

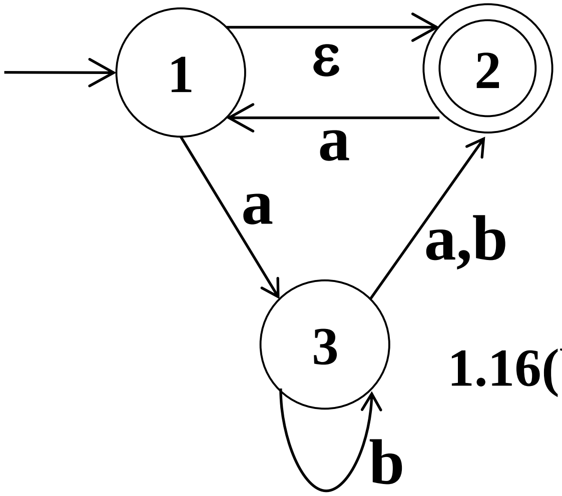
1.22 在某些程序设计语言中, 注释出现在两个分隔符之间, 如/#和#/. 设C是所有有效注释串形成的语言. C中的成员必须以/#开始, #/结束, 并且在开始和结束之间没有#/. 为简便起见, 所有注释都由符号a和b写成; 因此C的字母表Σ={a, b, /, #}.

a. 给出识别C的DFA.

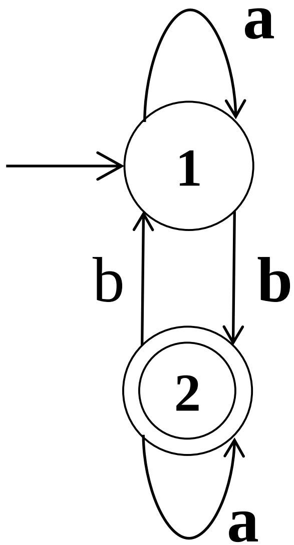
b. 给出产生C的正则表达式.

1.29 使用泵引理证明下述语言不是正则的。

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



1.16(b)题图



1.21(a)题图

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.

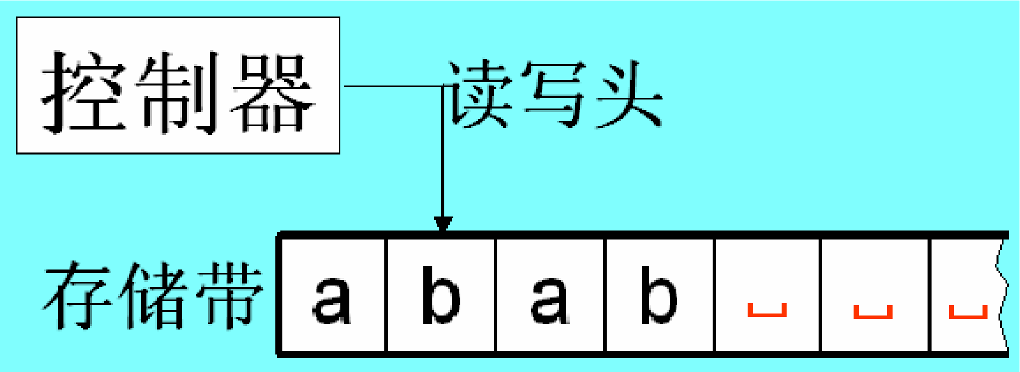
他根据 • 眼睛看到的纸上符号,

• 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

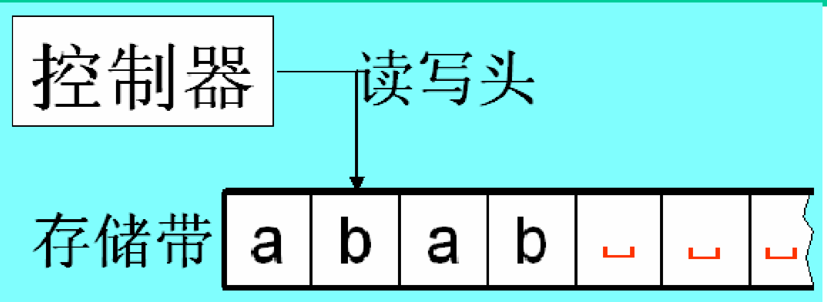
DFA确定型有限自动机:

- 输入带长度有限 = 输入串长度
- 只能读和右移, 不能写和左移
- 读完输入停机

图灵机:

- 输入带长度无限 (可以打草稿)
- 可以改写, 可以左右移
- (1) 什么时候结束? (2) 转移函数? (3) 图灵机定义?
- 直到他认为计算结束!
- 设计接受状态, 拒绝状态
- δ: $Q\times\Gamma \rightarrow Q\times\Gamma\times\{L,R\}$, 其中$\Gamma$是工作字母表

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, $q_a \neq q_r$.

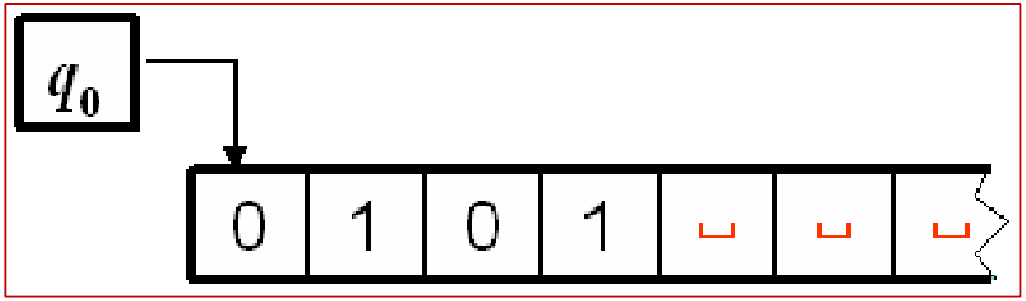
图灵机的初始化

设$M=(Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$, $w=w_1\ldots w_n\in\Sigma^n$,

输入带? 读写头?

- 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.
- 比汇编还原始, 一切都自己设计

例:设输入串为0101, 则其初始形态为



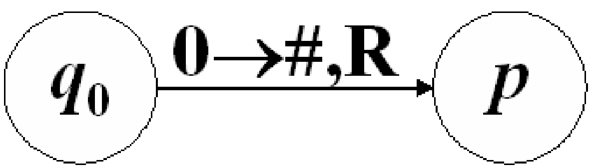
图灵机的运行

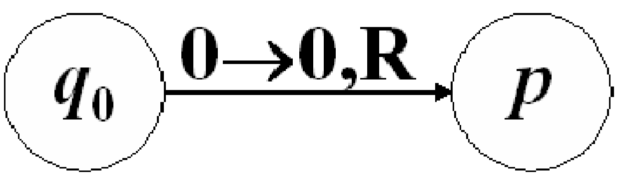
- 图灵机根据转移函数运行.

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

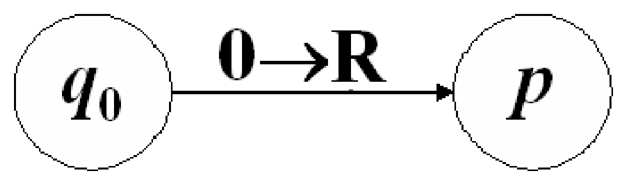
- 注: 若要在最左端左移, 读写头保持不动.

$\delta(q_0,0) = (p,\#,R)$的状态图表示:



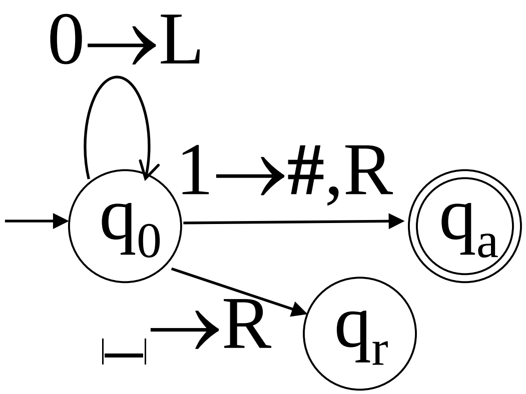


简记为

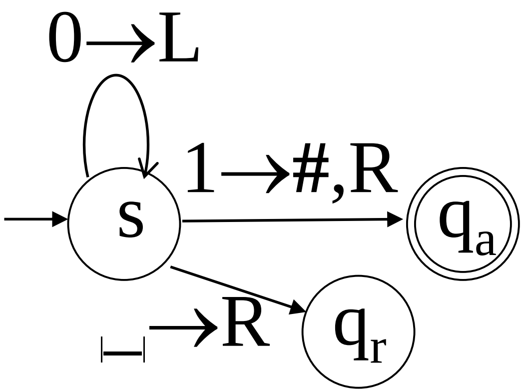


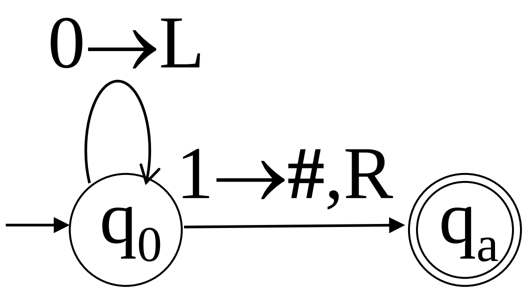
图灵机的格局(configuration)

- 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.

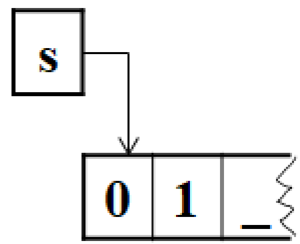


格局演化举例





省略拒绝状态



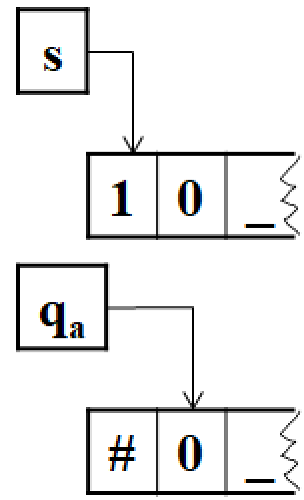
01:

s 0 1

s 0 1

…

循环

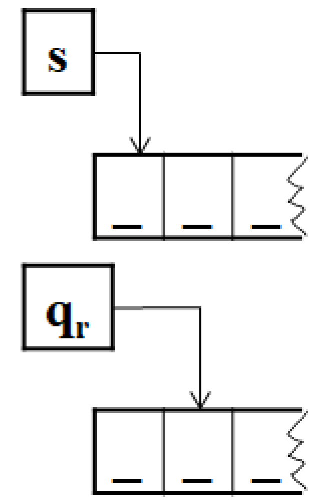


10:

s 1 0

q_a 0

接受



ε:

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).

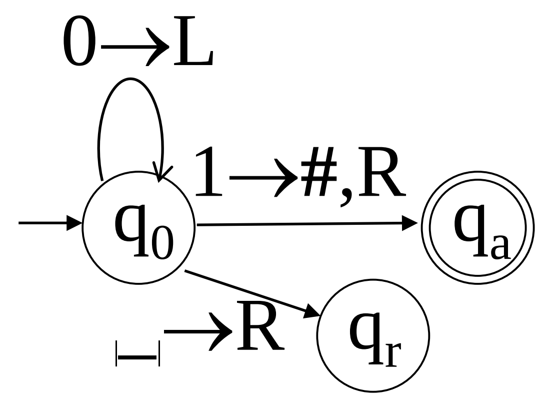
判定器与语言分类

- 对一个输入, 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. 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

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. (3)(4)**结合设计

状态图

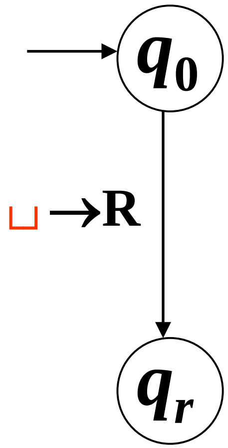
M="对于输入w,

1) 若w=ε, 则拒绝.

2) 若只有1个0,则接受.

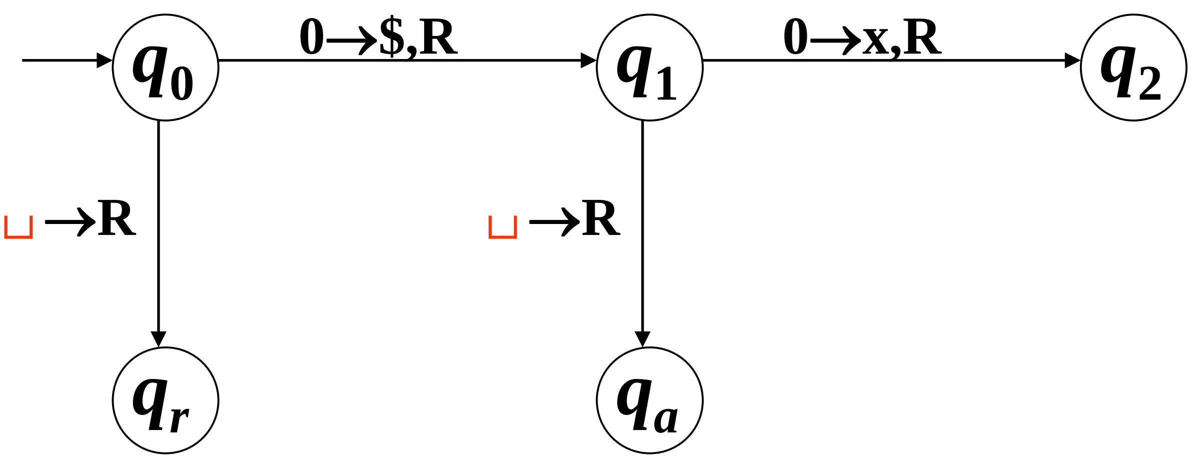
3) 若有奇数个0,则拒绝.

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

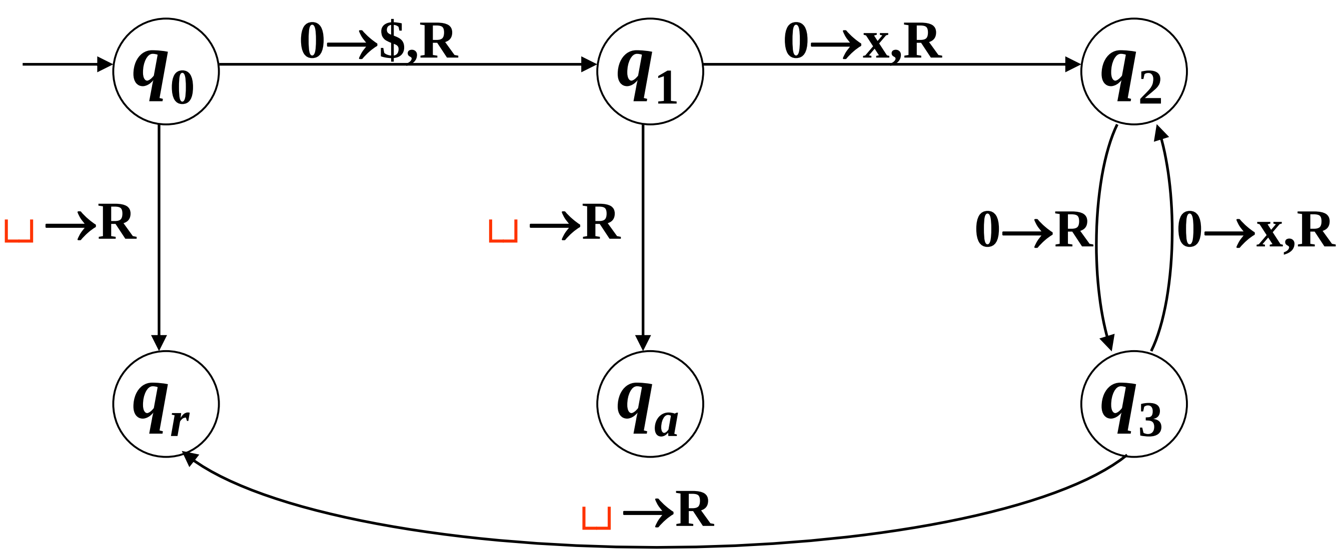
M="对于输入w,

1) 若w=ε, 则拒绝.

2) 若只有1个0,则接受.

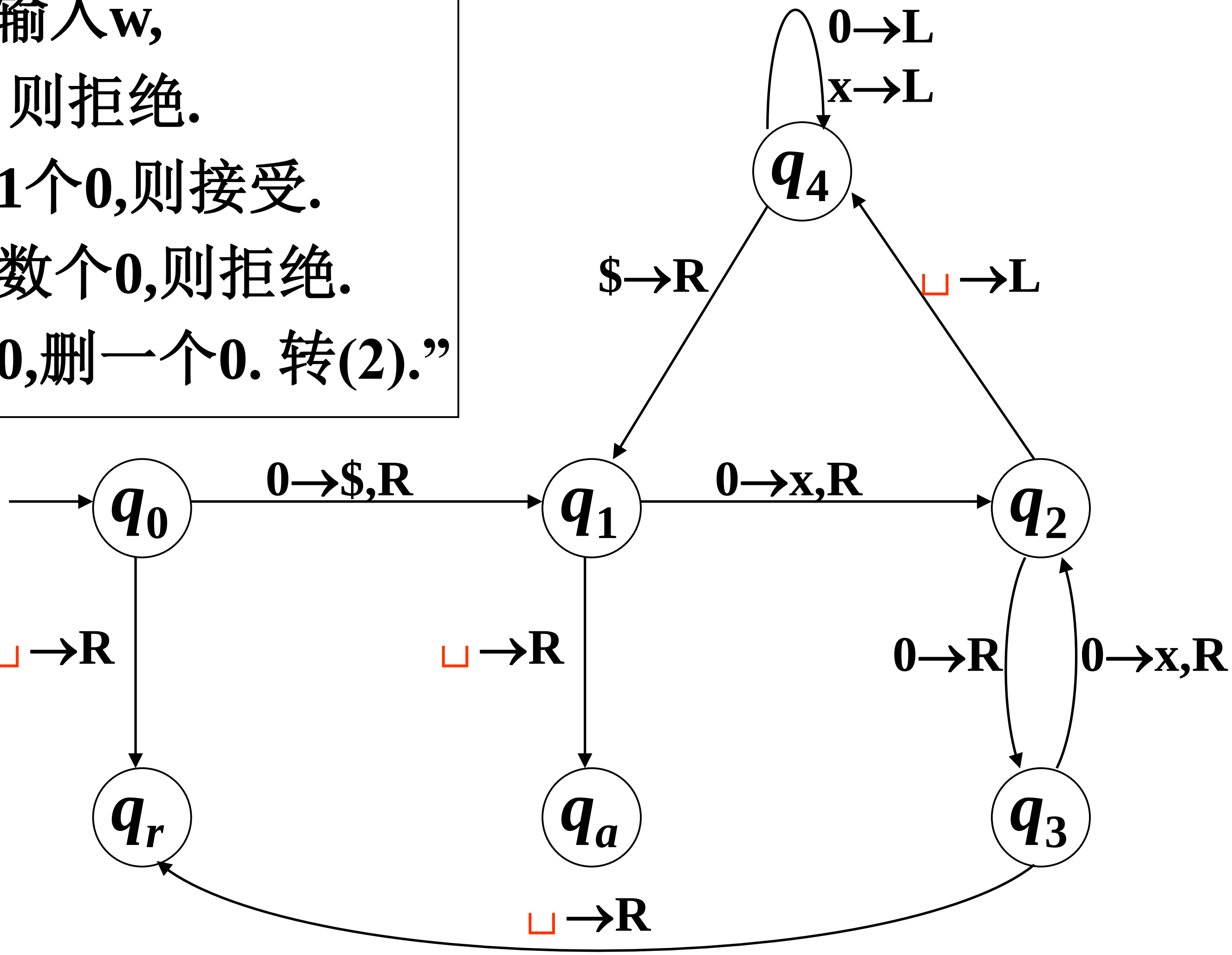
3) 若有奇数个0,则拒绝.

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

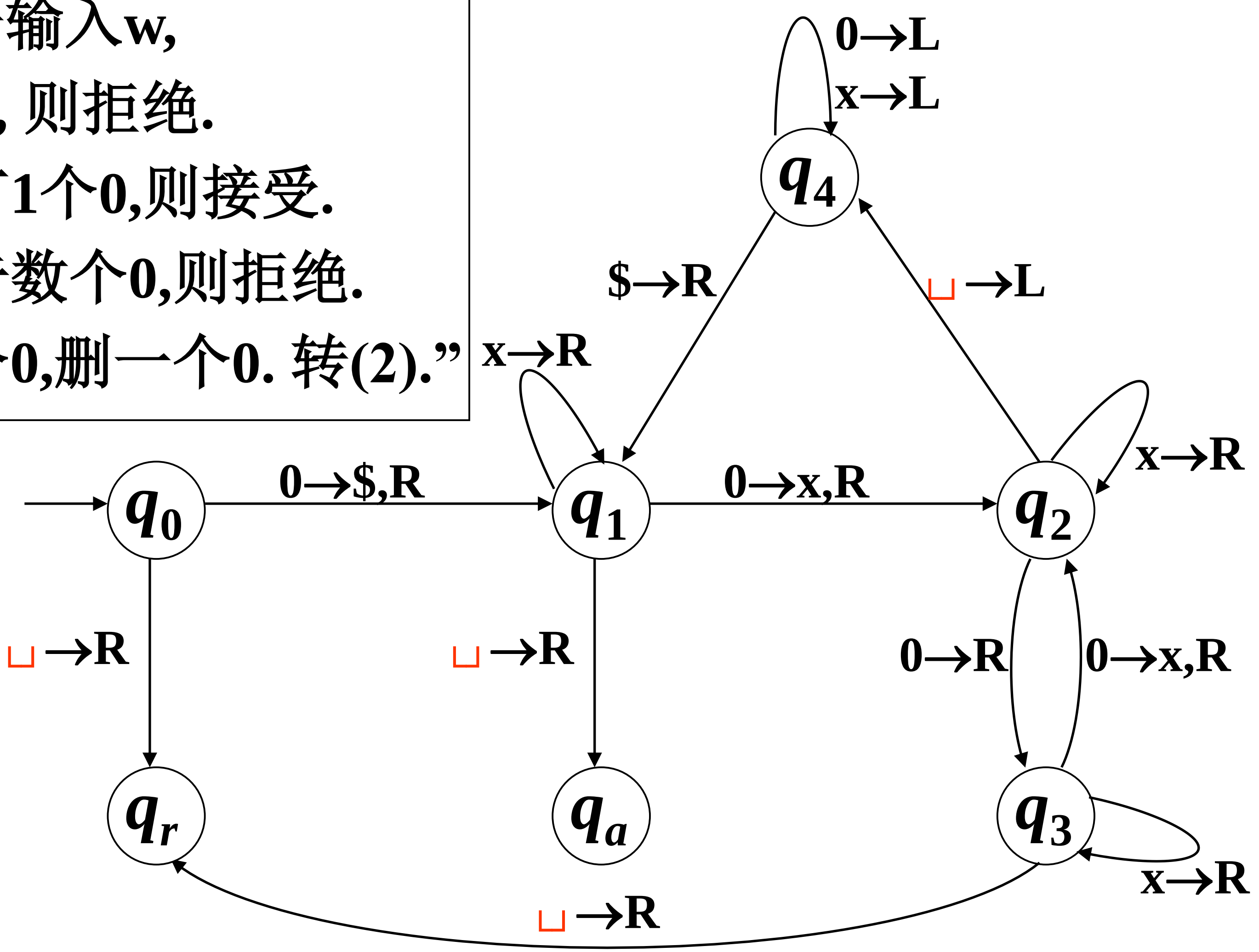
M="对于输入w,

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)."



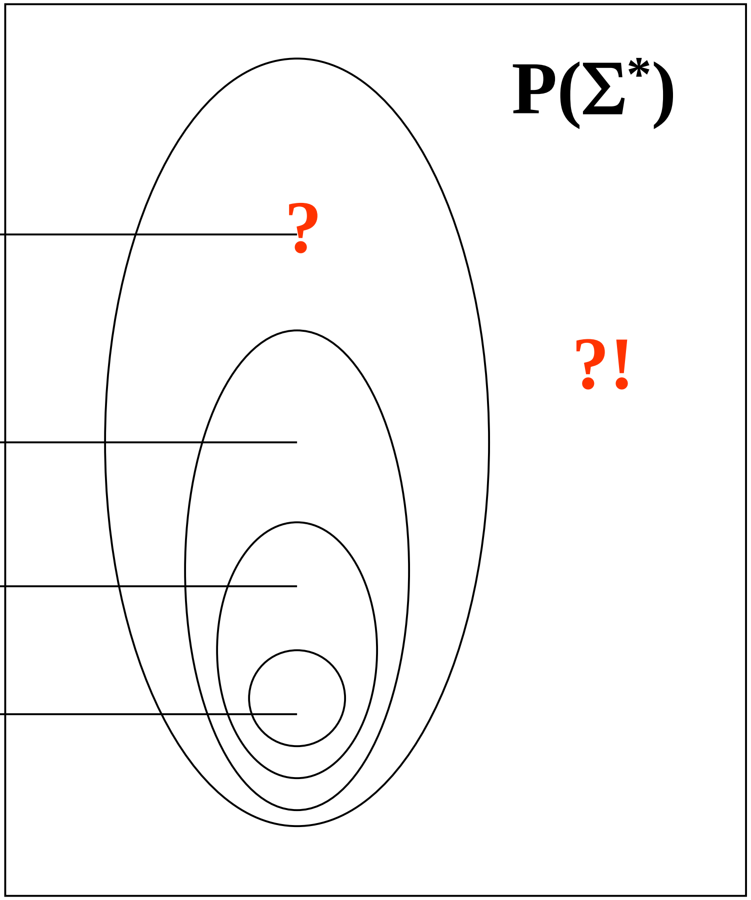
各种语言类的包含关系

图灵可识别语言

可判定语言

上下文无关语言

正则语言



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

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.

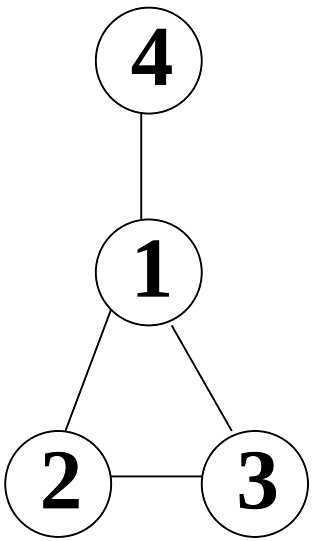
3) 若0的个数为奇数, 则拒绝.

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> | G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:

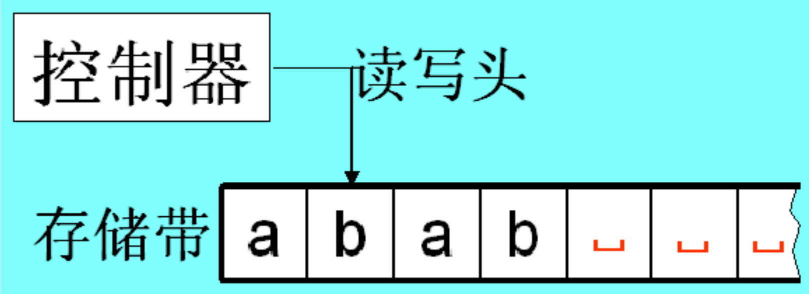
例如多带图灵机, 非确定图灵机

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)



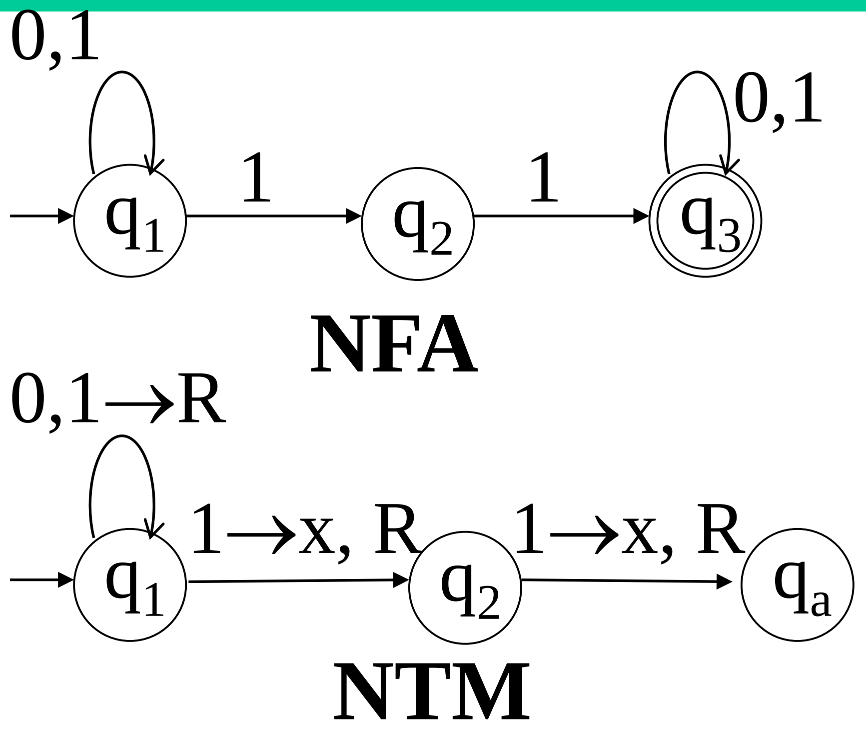
- NFA: $\delta: Q\times\Sigma_\varepsilon \rightarrow P(Q)$
- NTM的转移函数

  $\delta: Q\times\Gamma\rightarrow P(Q\times\Gamma\times\{L,R\})$
- NTM转移函数举例

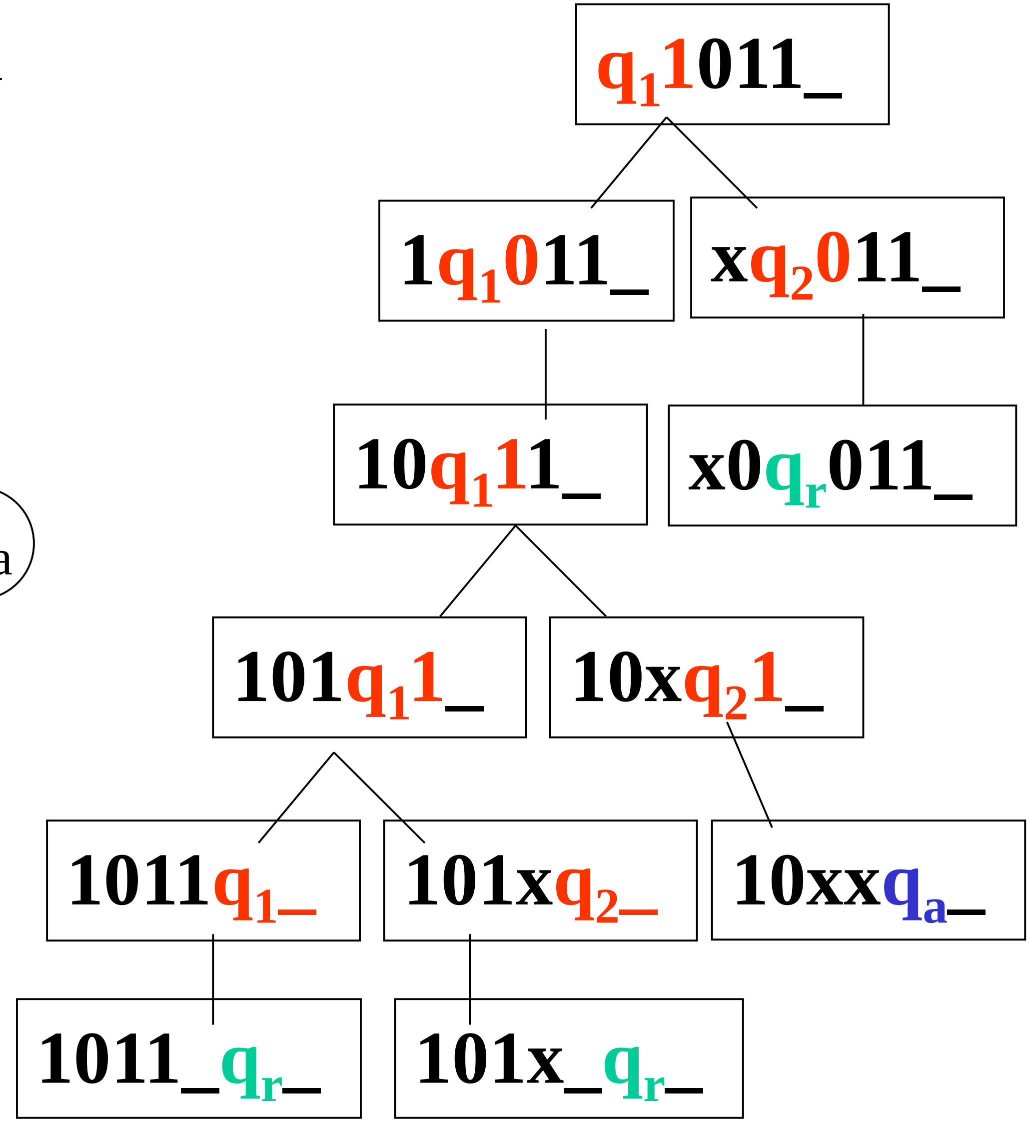
  $\delta(q_3,0)=\{(q_2,x,R), (q_1,1,L), (q_3,\$,R)\}$
- 称NTM M接受x, 若在x上运行M时有接受分支.
- 称一NTM为判定的,

  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

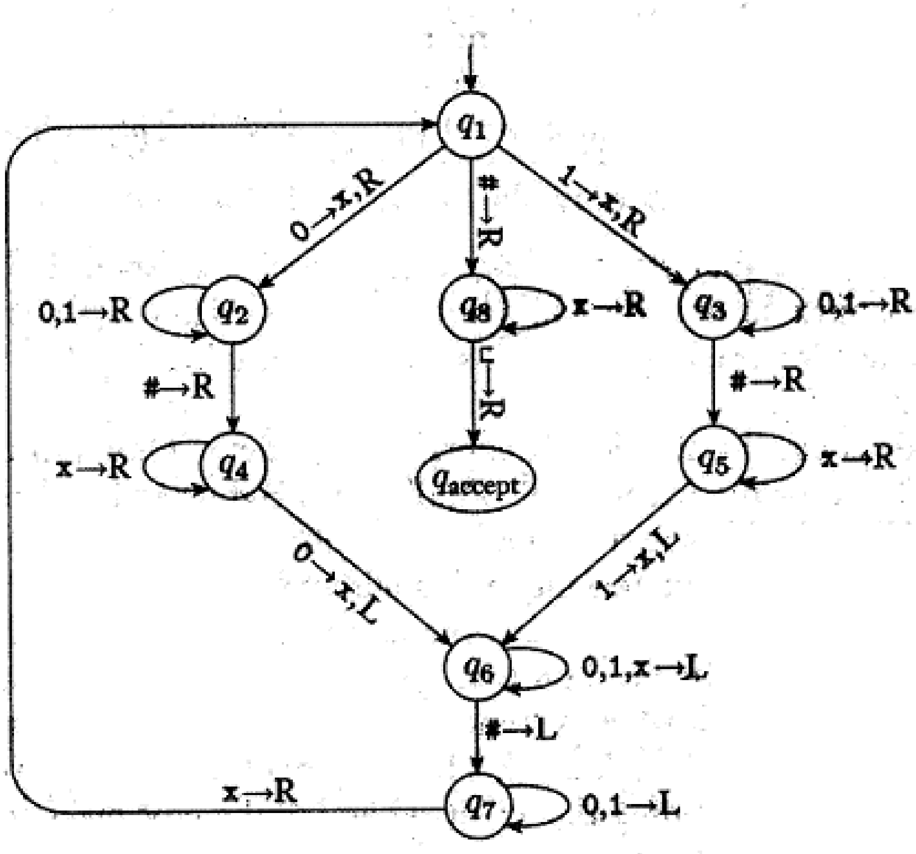


0,1
0,1
1
1
q_1
q_2
q_3
NFA
0,1→R
1→x, R
1→x, R
q_1
q_2
q_a
NTM



$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

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

附录

从字符串匹配问题说起

- 输入: 两个字符串x, y, (|x|=n, |y|=m)
- 输出: 所有y在x中出现的起点位置
- 例: x=abaabababbabababbababaa, y=ababbababaa
- 输出13
- 直接法: 以每个位置为起点对比一遍y. 时间?

  O((n-m+1)m). 能否利用已经看到的信息?

x: a b a a b a b a b b a b a b a b b a b a b a a

位置: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23

y: |a b a b b a b a b a a

y: |a b a b b a b a b a a

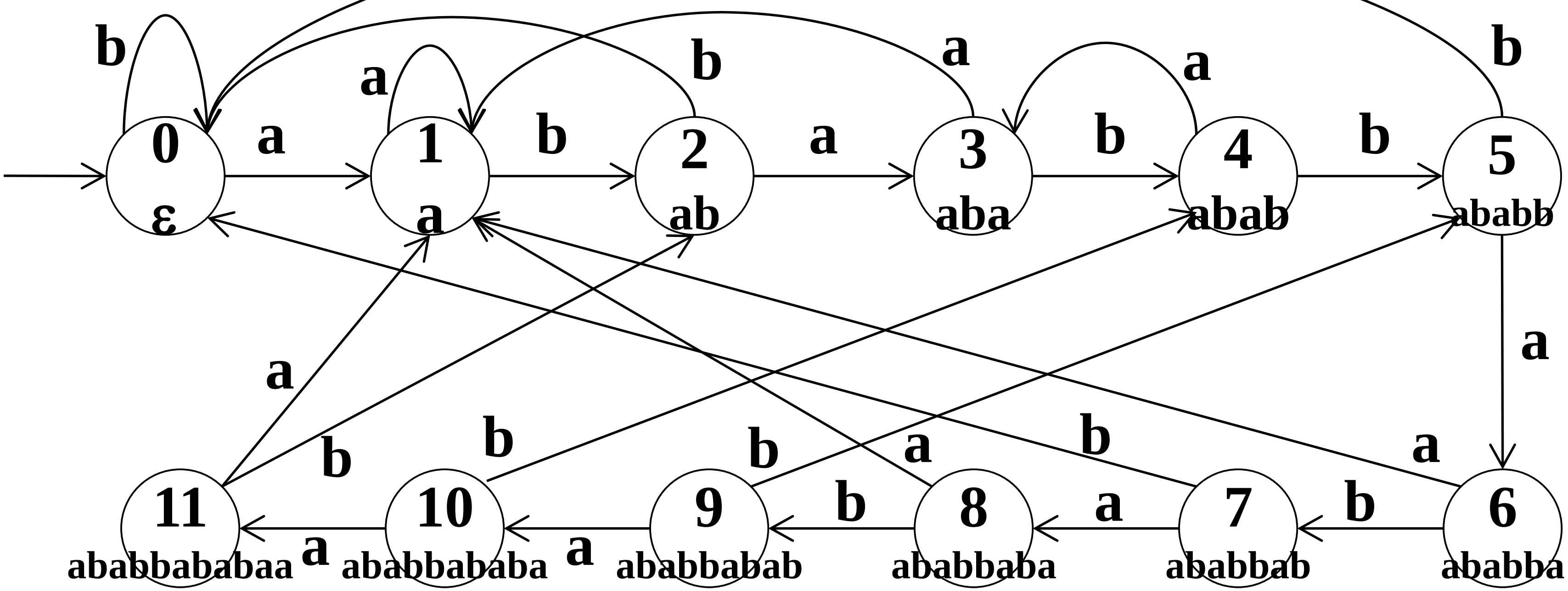
y: |a b a b b a b a b a a

输出 13

y: |a b a b b a b a b a a

构造状态和转移函数

y=ababbababaa　　节点0:11　　图 f: {0:11}×{a,b}→{0:11}



x:	a	b	a	a	b	a	b	a	b	b	a	b	a	b	a	b	b	a	b	a	b	a	a
位置:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
0	1	2	3	1	2	3	4	3	4	5	6	7	8	9	10	4	5	6	7	8	9	10	11

输出 23-10 = 13

输入x,y, 串匹配的有限自动机算法

符号集Σ, m=Length(y), 计算转移函数

1. 对 q = 0:m　　　　　// $Y_q=y_1y_2\ldots y_q$
2. 　对每个符号a∈Σ, 　//计算Y_qa
3. 　　　k=min{m,q+1}　//从Y_k开始对比
4. 　　　当 (Y_k不是Y_qa的后缀)　k--
5. 　　　f(q,a)=k

y=ababbababaa

Σ = {a,b}

|a,　　f(0,a)=1

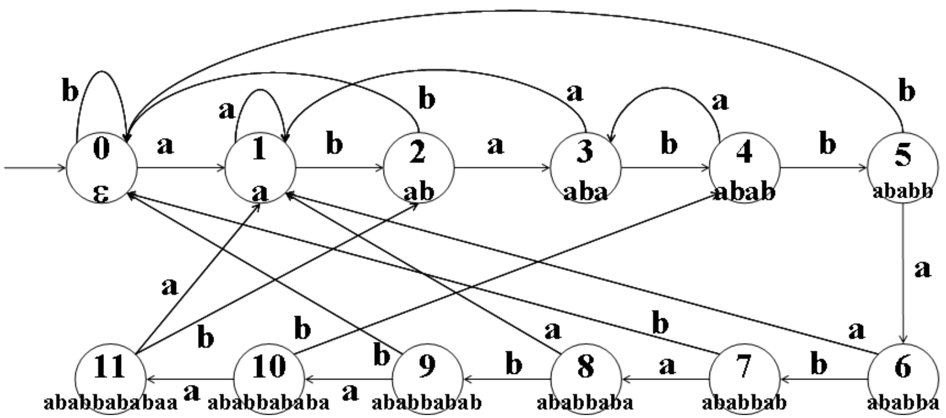
|b,　　f(0,b)=0

abab|b, f(4,b)=5

abab|a, f(4,a)=3

字符串匹配自动机算法

1. n=Length(x), q=0
2. 对 i = 1:n
3. 　q=f(q,x[i]),
4. 　若q=m, 打印 i-m+1



---

x:	a	b	a	a	b	a	b	a	b	b	a	b	a	b	a	b	b	a	b	a	b	a	a
0	1	2	3	1	2	3	4	3	4	5	6	7	8	9	10	4	5	6	7	8	9	10	11

字符串匹配算法

算法	预处理时间	匹配时间
朴素	0	$O((n-m+1)m)$
自动机	$O(m\|\Sigma\|)$	$\Theta(n)$
Knuth-Morris-Pratt	$\Theta(m)$	$\Theta(n)$

其中Σ是字母表

怎么会想到自动机? 为什么构造KMP?

有没有自动机做不了的事情?

第2章 上下文无关语言

本章不是考试内容

上下文无关文法(CFG)举例

G:

$$\boxed{\begin{array}{l} A \rightarrow 0A1 \\ A \rightarrow B \\ B \rightarrow \# \end{array}}$$

变元A,B, 终结符0,1,#,
替换规则,
起始变元A

$A \Rightarrow 0A1$ (1步派生)

$\Rightarrow 00A11 \Rightarrow 000A111 \Rightarrow 000B111 \Rightarrow 000\#111$

$A \Rightarrow^* 000\#111$ (0至多步派生)

$A \Rightarrow^* A$, $A \Rightarrow^* 0\#1$, $0B1 \Rightarrow^* 0\#1$

文法G的语言$L(G)=\{0^n\#1^n \mid n \geq 0\}$

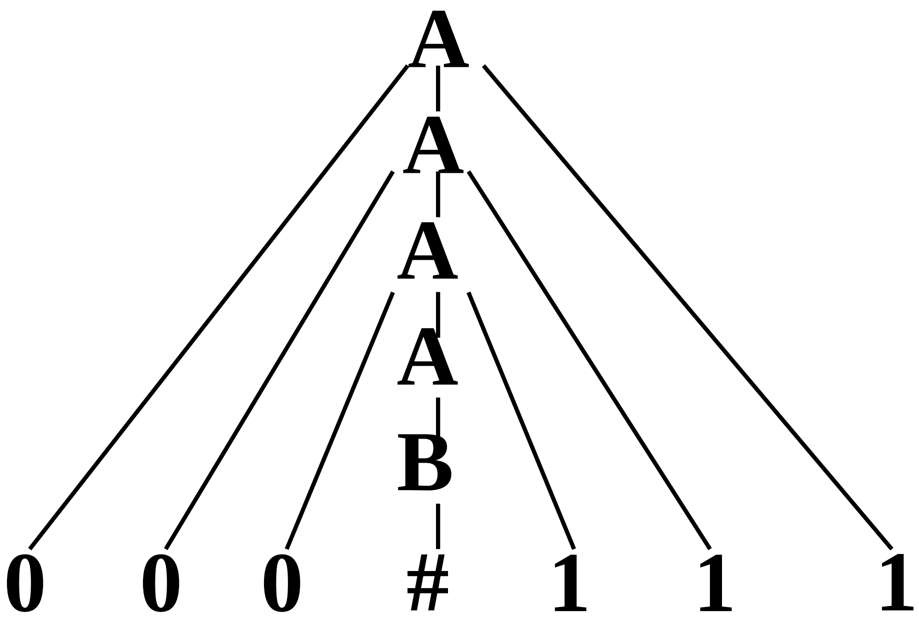
语法分析树

G:
$$
\begin{array}{l}
A \rightarrow 0A1 \\
A \rightarrow B \\
B \rightarrow \#
\end{array}
$$

通常简记为

$$
\begin{array}{l}
A \rightarrow 0A1 \mid B \\
B \rightarrow \#
\end{array}
$$

文法G派生串
000#111
的语法分析树:



上下文无关文法的形式定义

定义: CFG是一个4元组(V,Σ,R,S), 其中

1) V, 变元集(variables)

2) Σ, 终结符集(terminals)

3) R, 规则集 $A \to u$, $A \in V$, $u \in (V \cup \Sigma)^*$,

4) $S \in V$, 起始变元(start variable).

$$
\boxed{\begin{array}{l} A \to 0A1 \mid B \\ B \to \# \end{array}}
$$

变元集: $V = \{A,B\}$

终结符集: $\Sigma = \{0,1,\#\}$

起始变元: A

规则集:

$R = \{A \to 0A1, A \to B, B \to \#\}$

派生和语言的形式定义

定义: 一次派生 $\Rightarrow$

若 $u,v,w \in (\Sigma \cup V)^*$, $A \in V$, $(A \to w) \in R$,

则称 uAv 生成 uwv, 记为 $uAv \Rightarrow uwv$

定义: 0至多次派生 $\Rightarrow^*$

若 $u = v$, 或有序列 $u_1, u_2, \ldots, u_k$ $(k \geq 0)$使得

$$u \Rightarrow u_1 \Rightarrow \ldots \Rightarrow u_k \Rightarrow v$$

则称 u 派生 v, 记为 $u \Rightarrow^* v$.

定义: CFG的语言, 设 $G = (V, \Sigma, R, S)$

$$L(G) = \{ w \in \Sigma^* \mid S \Rightarrow^* w \}$$

上下文无关语言与CFG设计

定义: Σ为字母表, $A \subseteq \Sigma^*$,

若有上下文无关文法G使得A=L(G)

则称A为上下文无关语言(CFL).

CFG设计: 化繁为简, 利用正则,

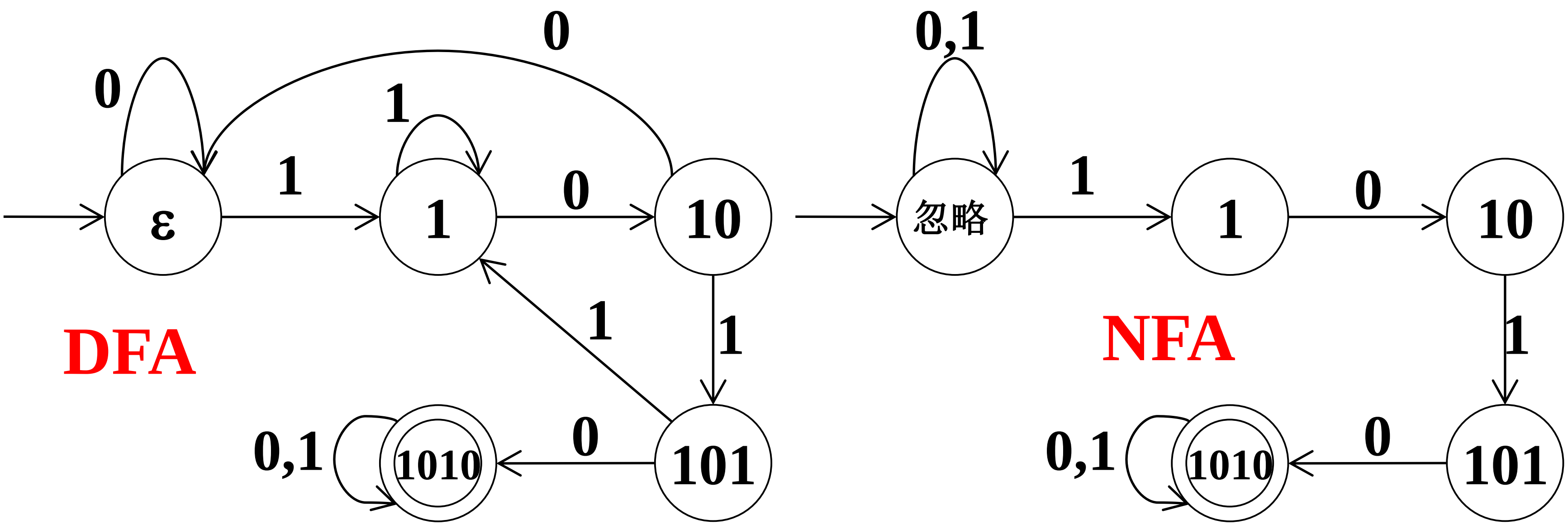
考察子串, 利用递归, 等.

{ 含有1010的01串 } 利用正则 或 直接构造

$\{ ww^R \mid w \in \{0,1\}^* \}$ 考察对应符号, 递归

$\{ 0^i 1^j \mid i < j < 2i \}$ 逐步构造

{ 含有1010的01串 }



由DFA构造

$$
\begin{aligned}
S &\to 0S|1A \\
A &\to 1A|0B \\
B &\to 0S|1C \\
C &\to 1A|0D \\
D &\to 0D|1D|\varepsilon
\end{aligned}
$$

由NFA构造

$$
\begin{aligned}
S &\to 0S|1S|1A \\
A &\to 0B \\
B &\to 1C \\
C &\to 0D \\
D &\to 0D|1D|\varepsilon
\end{aligned}
$$

直接构造

$$
\begin{aligned}
S &\to 0S|1S|1010D \\
D &\to 0D|1D|\varepsilon
\end{aligned}
$$

$\{ ww^R \mid w \in \{0,1\}^* \}$

考察对应符号, 递归

1 (0 (1 (0 0) 1) 0) 1, 剥大白菜

$S \rightarrow 1S1 \mid 0S0$

$S \rightarrow \varepsilon$

$\{ 0^i1^j \mid i < j < 2i \}$ 逐步构造

CFG设计: 化繁为简, 利用正则, 考察子串,递归, 等.

抽丝剥茧

$\{ 0^i1^j \mid i = j \}$: $S \rightarrow 0S1 \mid \varepsilon$

$\{ 0^i1^j \mid 2i = j \}$: $S \rightarrow 0S11 \mid \varepsilon$

$\{ 0^i1^j \mid i \leq j \leq 2i \}$: $S \rightarrow 0S1 \mid 0S11 \mid \varepsilon$

$\{ 0^i1^j \mid i < j < 2i \}$: $S \rightarrow 0S1 \mid 0S11 \mid 00111$

泵引理

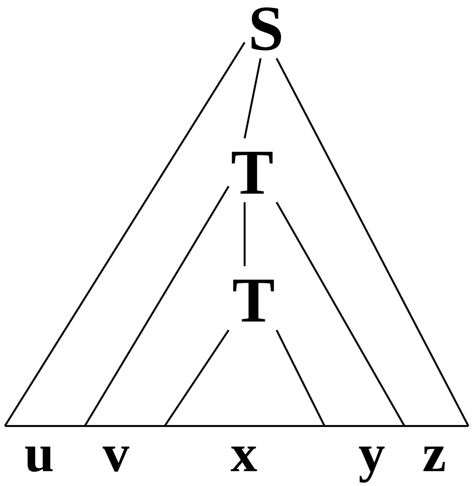
定理(泵引理): 设A是CFL,则存在p>0使得

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

1) $\forall\ i \geq 0,\ uv^ixy^iz \in A$;

2) $|vy|>0$;

3) $|vxy|\leq p$.

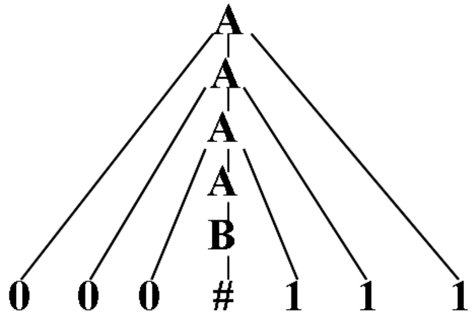


泵引理的证明.

应用: $B=\{\ 0^n1^n2^n \mid n\geq 0\ \}$ 非CFL

$C=\{\ ww \mid w\in\{0,1\}^*\ \}$ 非CFL

$D=\{\ 1^k \mid k=2^n,\ n\geq 0\ \}$ 非CFL



$C = \{ ww \mid w \in \{0,1\}^* \}$ 非CFL

∵ $\forall p>0$,

令$w=0^p1^p0^p1^p$,

(分4个区)

$\forall u,v,x,y,z$

($|vy|>0$, $|vxy|\le p$, $w=uvxyz$)

(vxy可能在12, 23, 34区)

令$i = 0$,

$uxz \notin A$.

∴ C非上下文无关语言

若$\forall p>0$

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

$\forall u,v,x,y,z$

($|vy|>0$, $|vxy|\le p$, $w=uvxyz$)

$\exists i\ge 0$,

$uv^ixy^iz \notin A$.

则A非上下文无关语言