



北京理工大学  
BEIJING INSTITUTE OF TECHNOLOGY

# 算法与计算理论

# 课程内容



## 数据结构

概述

线性表

栈与队列

数组与广义表

串

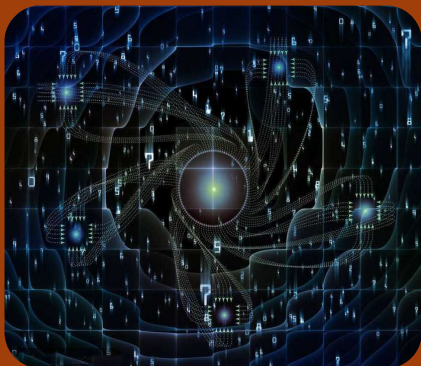
树

图

查找

内部排序

外部排序



## 算法与计算理论

概述

分治

动态规划

贪心

回溯

.....

.....

.....

.....

计算模型

可计算理论

计算复杂性



图灵机基础

图灵机的定义

图灵机举例

图灵机的描述

# Contents

## 本章内容

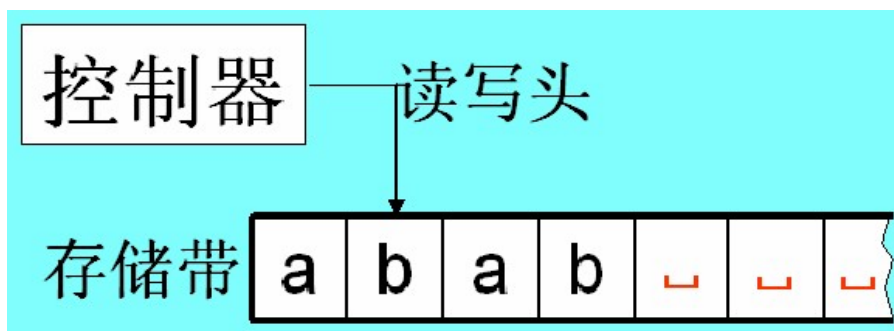
[S] 唐常杰等译, Sipser著, 计算理论导引(第3版), 机械工业.

参考资料:

[L] Lewis等著, 计算理论基础, 清华大学.

## 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

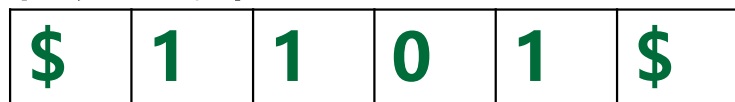
眼睛和笔:读写头

法则:转移函数

# 与有限自动机的区别

有限自动机:

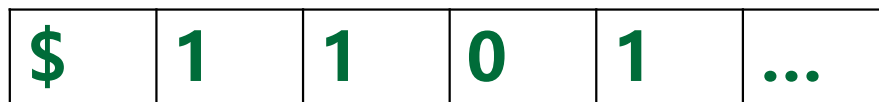
有穷纸带 单向读头 →



状态控制器

图灵机:

无穷纸带 双向读写头 ↔



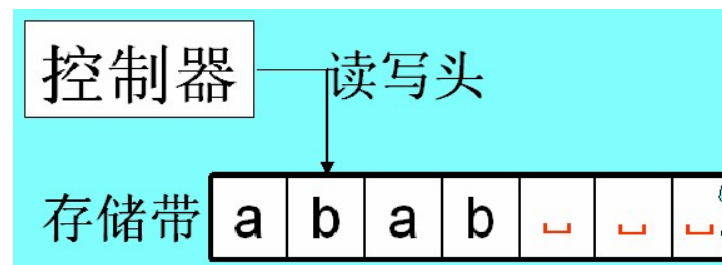
状态控制器

	有限自动机	图灵机
输入带长度	有限	无限
读头移动方向	右移	右移左移
是否可写	不可写	可写
如何停机	读完输入后停机	进入接受或拒绝状态后停机
是否停机	停机	不一定停机

## 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)  $\Sigma$ 是输入字母表,不包括空白符  $\sqcup$ .
- 3)  $\Gamma$ 是纸带字母表,其中  $\sqcup \in \Gamma, \Sigma \subset \Gamma$ .
- 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,  $q_a \neq q_r$ .



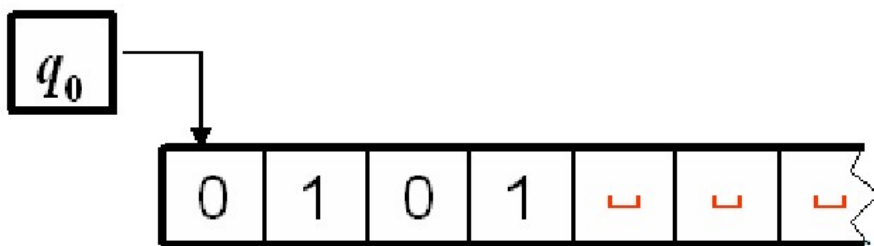
## 图灵机的初始化

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

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  $\sqcup$ .

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

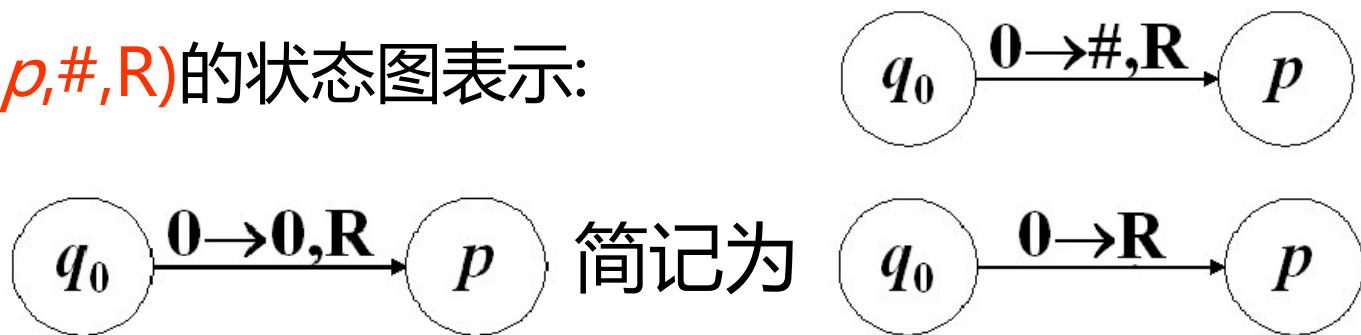
图灵机根据转移函数运行.

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:



## 判定器与语言分类

- 图灵机运行的三种结果

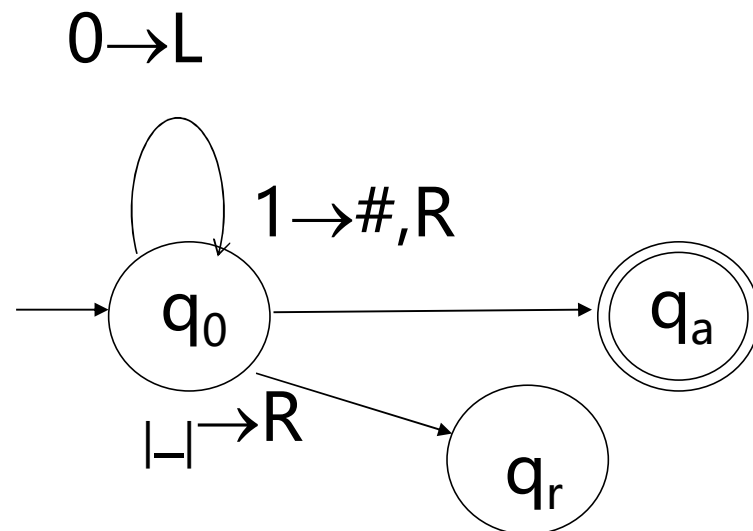
1. 若TM进入接受状态,则停机且接受输入,
2. 若TM进入拒绝状态,则停机且拒绝输入,
3. 否则TM一直运行,不停机.

- 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



## 图灵可判定语言与图灵可识别语言

如果存在一个图灵机，对于任意给定的输入字符串，该图灵机都能在有限步骤内停止，并且能正确地判定该字符串是否属于这个语言，那么这个语言是**图灵可判定语言**（也称为递归语言）。

- 1.如果输入字符串属于该语言，图灵机最终会进入一个接受状态并停止。
- 2.如果输入字符串不属于该语言，图灵机最终会进入一个拒绝状态并停止。

简单来说，对于图灵可判定语言，有一个图灵机能够对任何输入给出“是”或“否”的答案，并且保证在有限时间内给出答案。

对于**图灵可识别语言**，存在一个图灵机可以识别属于该语言的所有字符串（即对这些字符串，图灵机会在有限时间内停止并接受），但对于不属于该语言的字符串，图灵机可能永远运行下去而不会停止。

**图灵可判定语言是图灵可识别语言的一个子集。每个图灵可判定语言都是图灵可识别的，但不是每个图灵可识别语言都是图灵可判定的。**

## 图灵机的格局

图灵机的“格局”（也称为“配置”或“瞬时描述”）是指图灵机在计算过程的某一特定时刻的完整状态描述。一个格局包括以下三个要素：

- 1.当前的带状符号：图灵机的带子上每个格子的符号序列。带子是图灵机的存储介质，通常被想象为左右无限延伸的，并且分成连续的格子，每个格子上可以写有一个符号。
- 2.头的位置：图灵机的读写头所在的位置。头是图灵机用来读写符号的部件，它可以在带子上移动（左移、右移或保持不动）。
- 3.当前状态：图灵机当前的内部状态。一个图灵机拥有一套有限的状态集合，这些状态决定了机器如何根据带子上的符号进行操作。

每个格局可以看作是图灵机在计算过程中的一个快照，记录了图灵机的所有必要信息，以便从这一点继续计算。计算的每一步都可能改变图灵机的格局，这通常发生在读写头读取带子上的符号，并根据转移函数（图灵机的规则集合）来更新带子上的符号、移动读写头的位置以及改变当前状态。

## 图灵机的格局

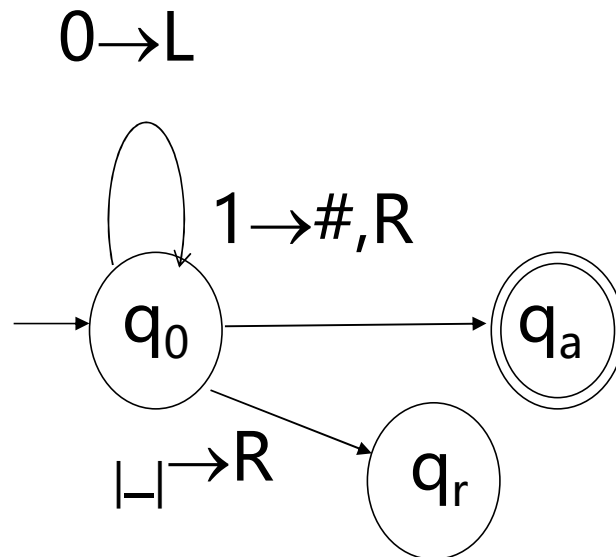
描述图灵机运行的每一步需要如下信息:  
存储带上**字符串**; 读写头的**位置**. 控制器的**状态**;

定义: 对于图灵机  $M = (Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$ ,

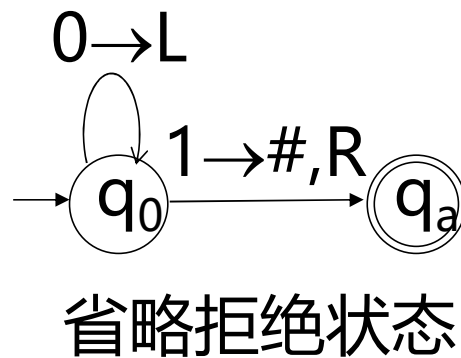
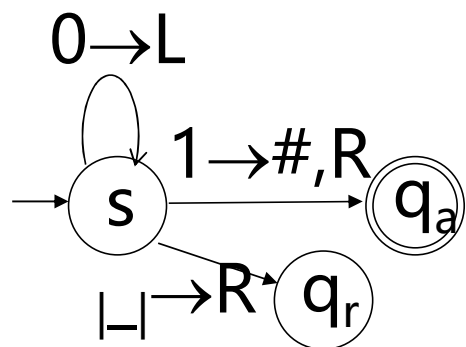
设  $q \in Q$ ,  $u, v \in \Gamma^*$ , 则**格局**  $uqv$  表示

- 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



s	0	1
s	0	1
...		
循环		

s	1	0
#	q <sub>a</sub>	0
接受		

s	_	_
_	q <sub>r</sub>	_
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 $C_1, C_2, \dots, C_k$ 使得

- 1)  $C_1$ 是 $M$ 的起始格局 $q_0w$ ;
- 2)  $C_i$ 产生 $C_{i+1}$ ,  $i=1, \dots, k-1$ ;
- 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 $L(M)$ .



图灵机基础

图灵机的定义

图灵机举例

图灵机的描述

# Contents

## 本章内容



## 图灵机举例

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

$B = \{0^n 1^n : n \geq 0\}$  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  $w = \varepsilon$ , 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

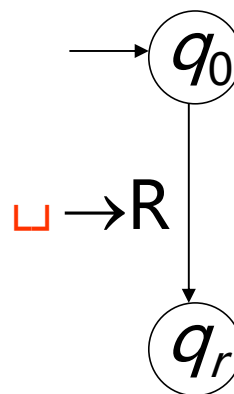
$M =$  “对于输入  $w$ ,

1) 若  $w = \varepsilon$ , 则拒绝.

2) 若只有1个0, 则接受.

3) 若有奇数个0, 则拒绝.

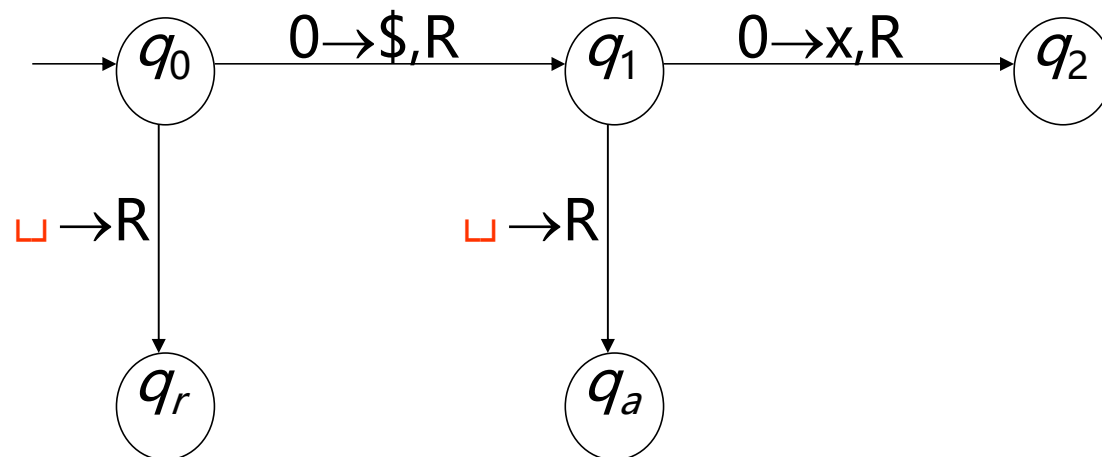
4) 隔一个0, 删一个0. 转(2).”



## 状态图

$M =$  “对于输入  $w$ ,

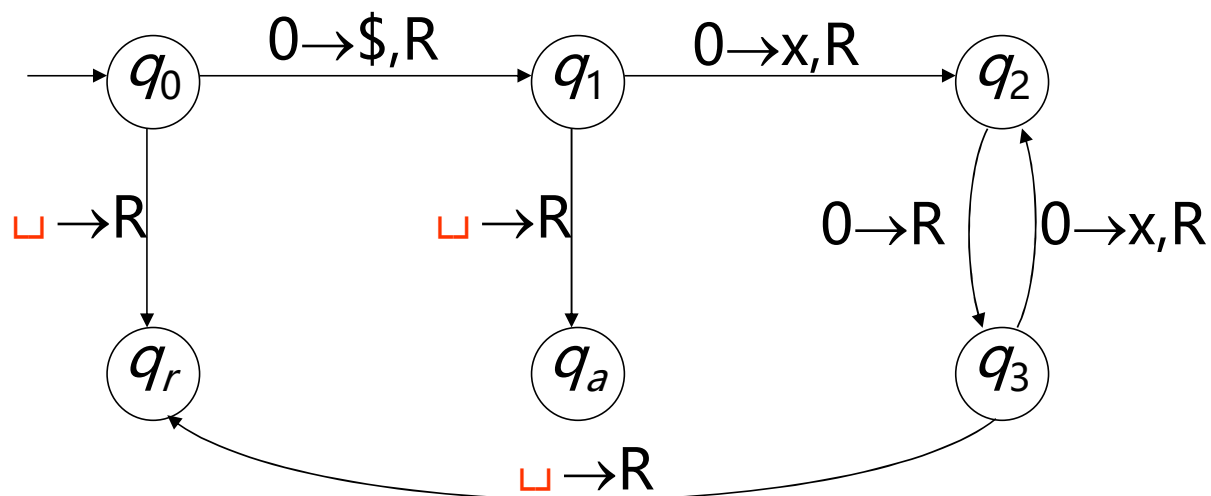
- 1) 若  $w = \varepsilon$ , 则拒绝.
- 2) 若只有1个0, 则接受.
- 3) 若有奇数个0, 则拒绝.
- 4) 隔一个0, 删一个0. 转(2).”



## 状态图

$M =$  “对于输入  $w$ ,

- 1) 若  $w = \varepsilon$ , 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

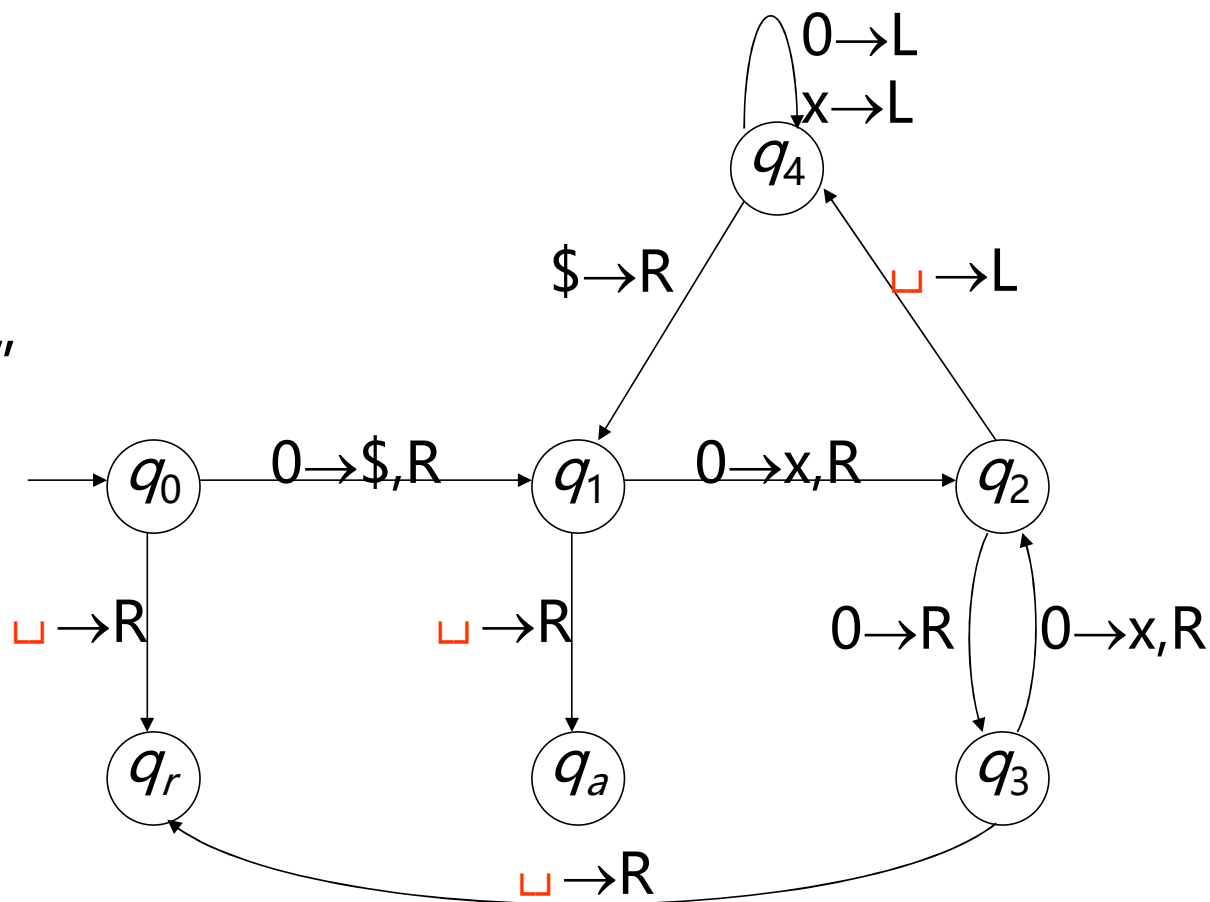
$M =$  “对于输入  $w$ ,

1) 若  $w = \varepsilon$ , 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

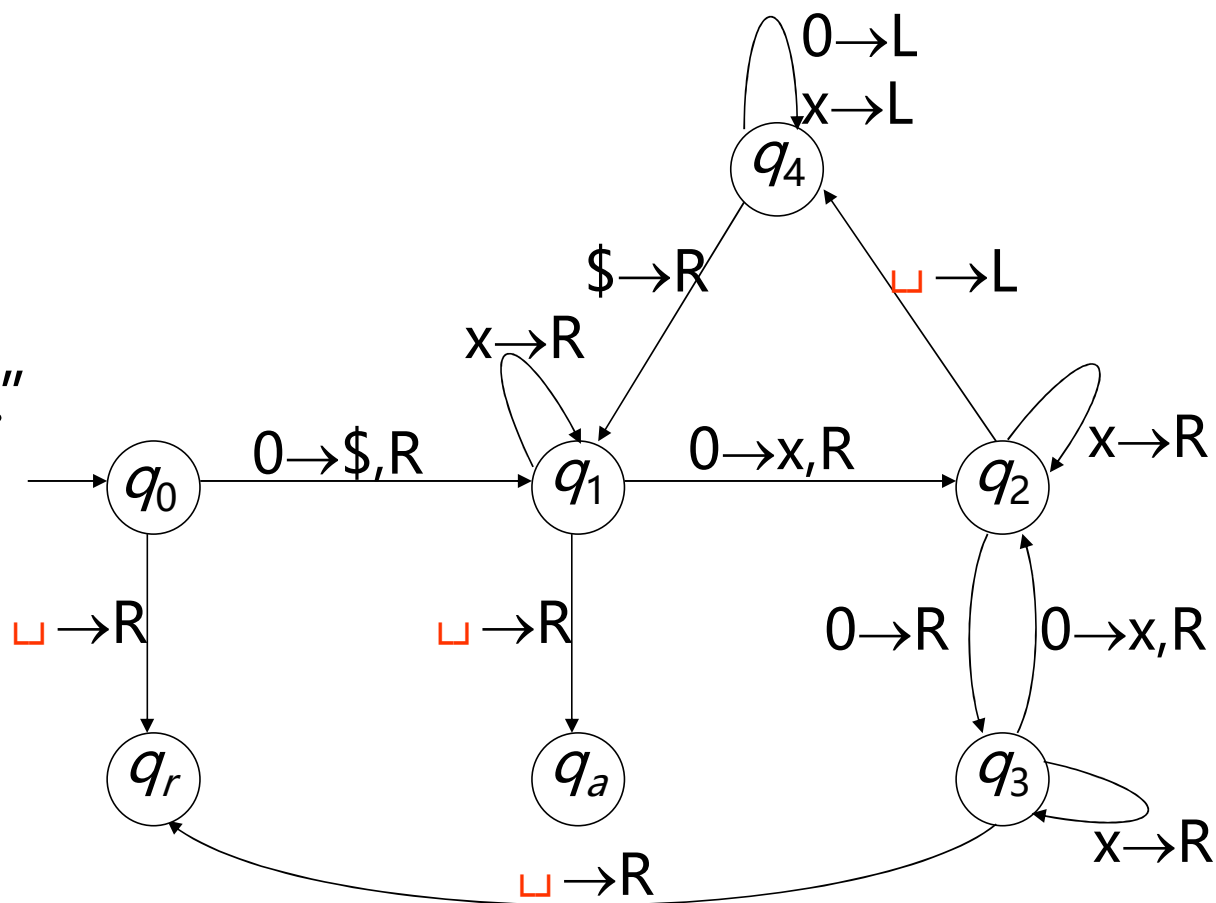
$M =$  “对于输入  $w$ ,

1) 若  $w = \varepsilon$ , 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).”



◆ M1= “对于输入字符串x:

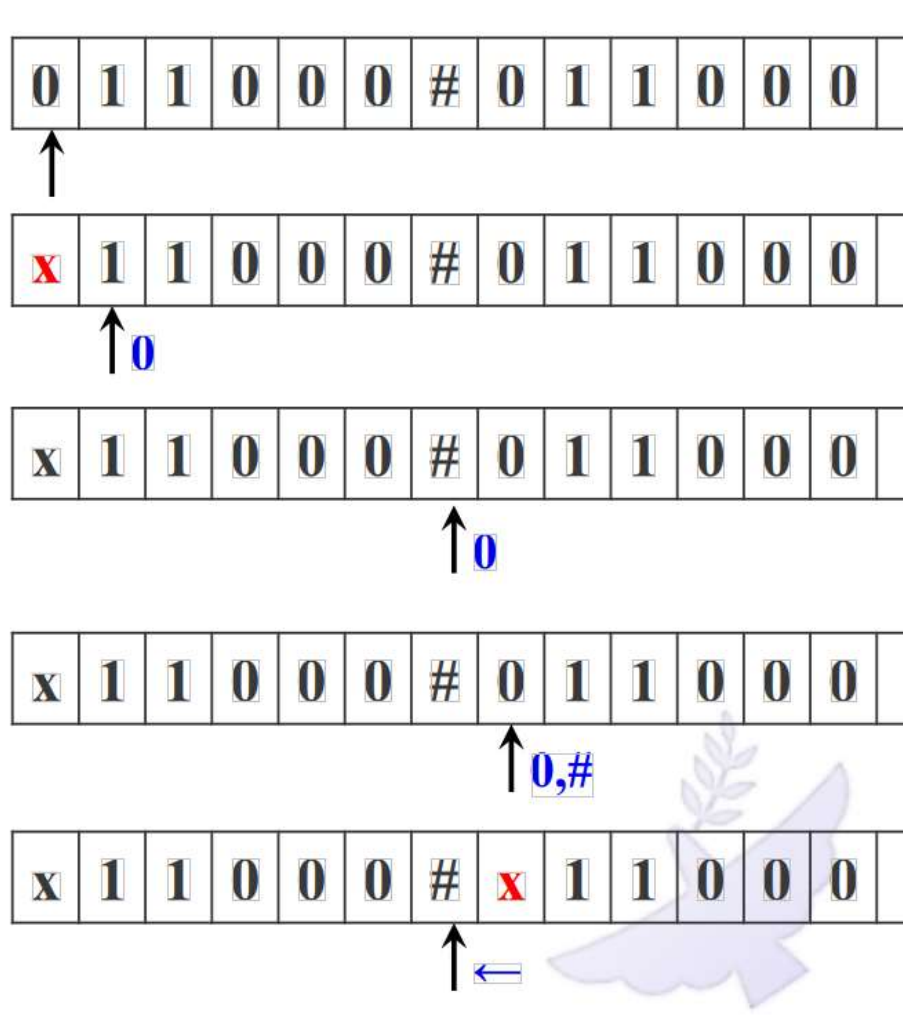
◆ 2)在#两边对应位置来回移动,检查是否含相同符号.

是则消去已检查过符号.

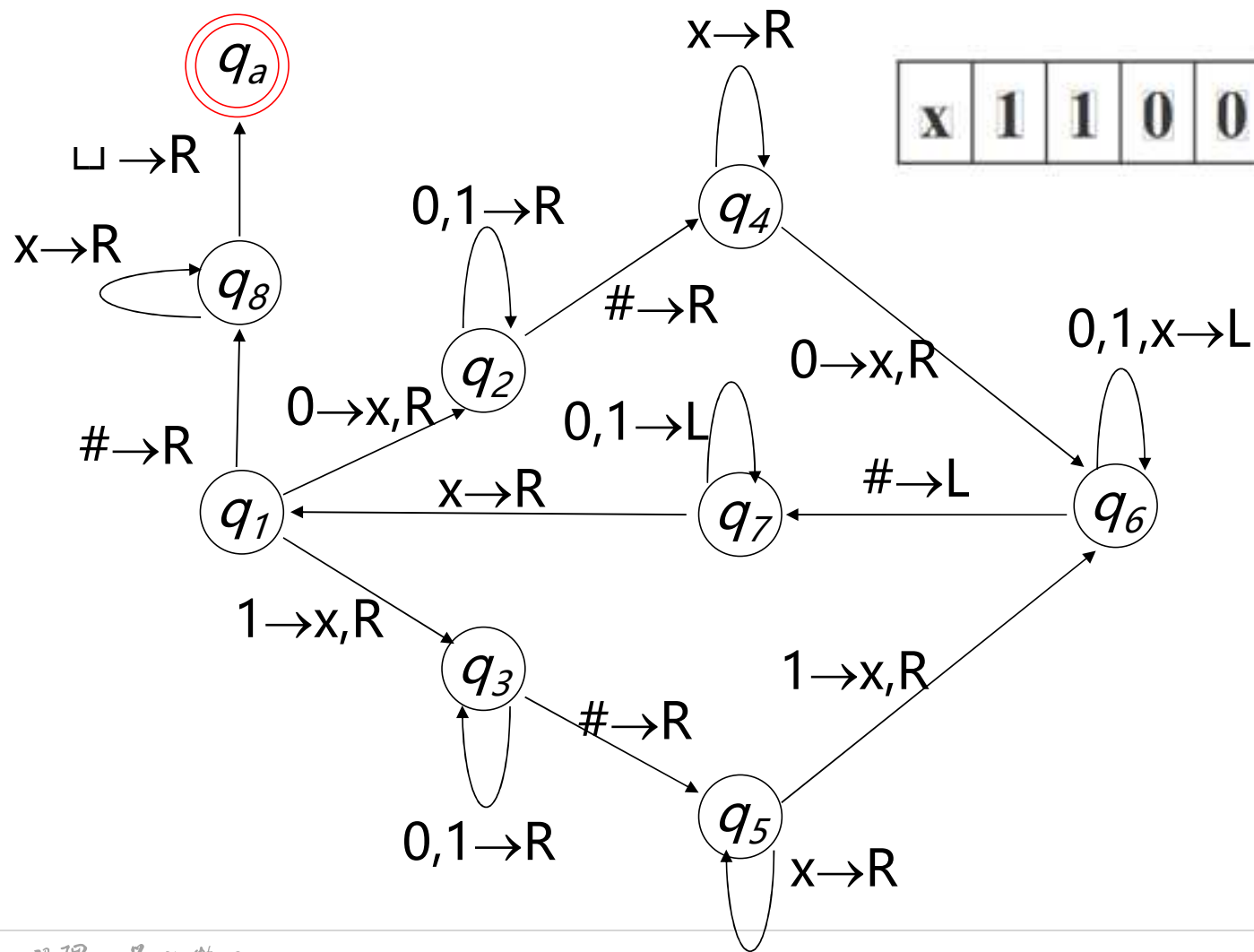
若不是,则拒绝.

◆ 3)当消去#左边所有符号时,检查#右边是否还有符号,若是,则拒绝.

否则接受。”



# 状态转换图 $\{w#w \mid w \in \{0,1\}^*\}$



x	1	1	0	0	0	#	0	1	1	0	0	0	
---	---	---	---	---	---	---	---	---	---	---	---	---	--

在#两边对应位置来回移动,  
检查是否含相同符号.  
是则消去已检查过符号.  
若不是,则拒绝



$$C = \{a^i b^j c^k \mid i \times j = k\}$$

M3 = “对输入串w:

- 1) 从左向右扫描输入, 确认输入具有形式  $a^*b^*c^*$ , 否则拒绝.
- 2) 让读写头回到带子左端.
- 3) 消去1个a, 向右扫描直到b出现.  
在b和c之间来回移动, 成对消去b和c, 直到消去所有b.
- 4) 若还有a未消去, 则  
恢复所有已消去的b, 重复第三步.  
若所有a已消去, 则检查是否消去所有c,  
若是则接受, 否则拒绝.”

$$E = \{ \#x_1\#x_2\#\dots\#x_k \mid x_i \in \{0,1\}^*, \forall i \neq j, x_i \neq x_j \}$$

M4 = “对输入w:

1)在最左端带符号上做记号.

若此符号是空白符,则接受;

若此符号是#,则进行下一步.否则拒绝.

2)向右扫描下一个#,在其顶上做第二个记号.

若在遇到空格之前没有遇到#,则只有x1,因此接受.

3)来回移动比较做记号#右边两个串,若相等,则拒绝.

4)将右边#上记号向右移到下一个#上.

若在遇到空白符之前没有遇到#,

则将左边#上记号向右移到下一个#上,并将右边记号移到后面#上.

若此时右边仍然找不到#,则接受.

5)转到第3)步.”

上述语言都是可判定的

# 各种语言类的包含关系

$A = \{0^k 1^n : k \geq n, n \geq 0\}$

正则语言

$B = \{0^n 1^n : n \geq 0\}$

上下文无关语言

$C = \{0^k : k = 2^n, n \geq 0\}$

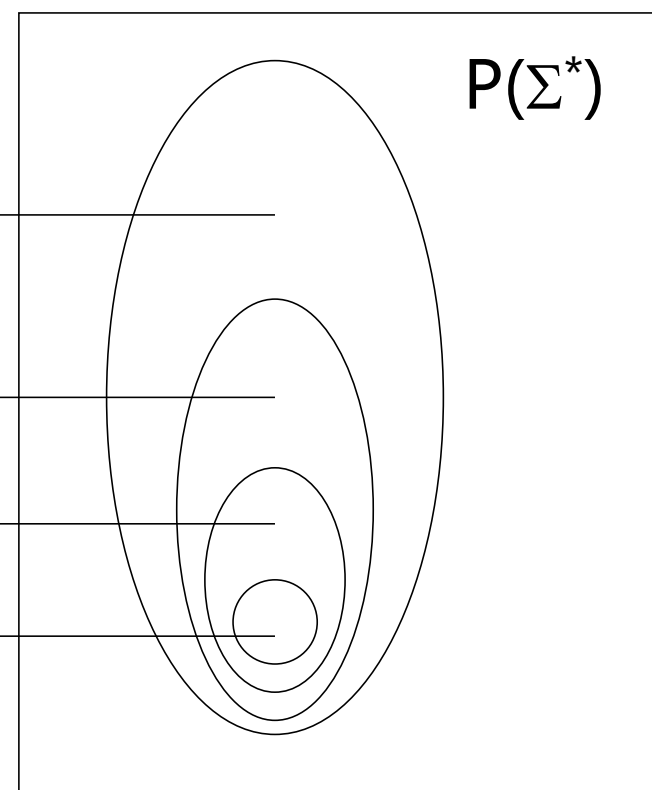
图灵可判定语言

图灵可识别语言

可判定语言

上下文无关语言

正则语言



## 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 $w=\varepsilon$ , 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>.

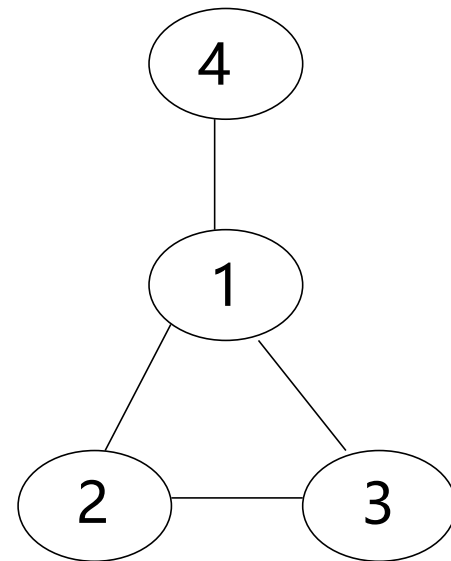
本课程中一般不关心实际编码方式.

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  $\langle G \rangle$ ,  $G$  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:

$L(M_1) = \{ \langle G \rangle \mid G \text{ 是连通的无向图} \}$

## 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

多头图灵机

枚举器

带停留的图灵机等等

.....

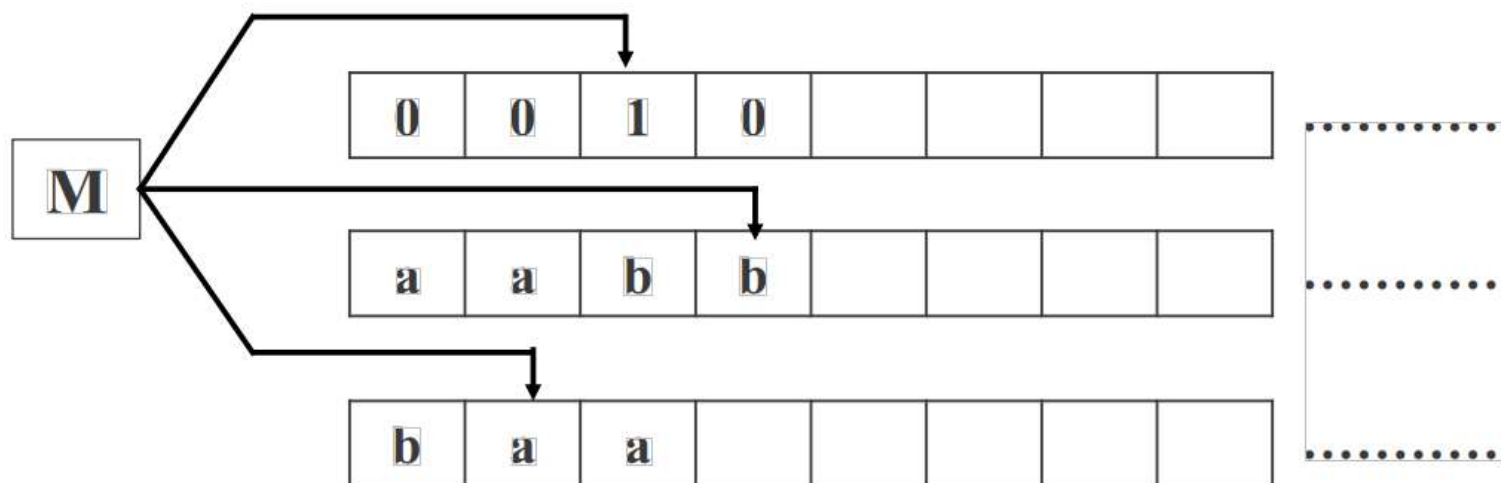
只要满足必要特征,它们都与这里定义的图灵机等价.

# 多带图灵机

多带图灵机的转移函数:

$\delta: Q \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{L, R\}^k.$

$(q_i, a_1, \dots, a_k) = (q_j, b_1, \dots, b_k, L, R, \dots, L)$





# 多带图灵机

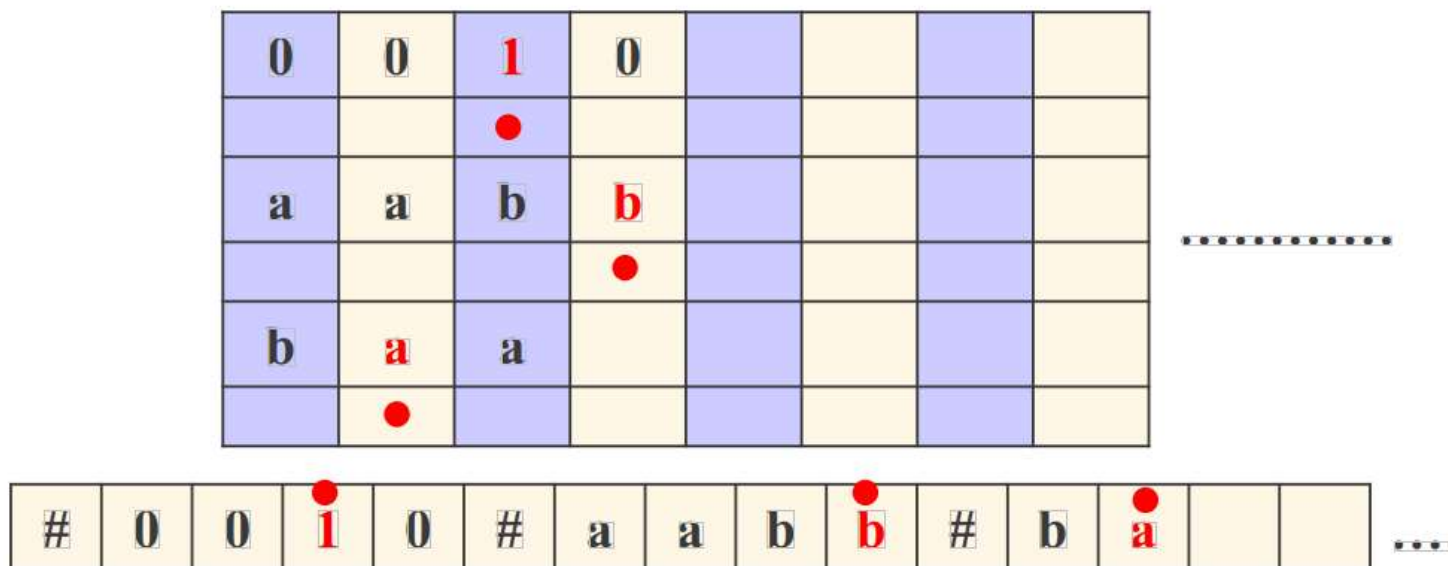
定理:每个多带TM都有等价单带TM.

◆ 分析:用单带图灵机模拟多带TM运行过程

方法一: 一次模拟一条带(改变字母表, 生成新转移函数)

$\Gamma = \Gamma_1 \times \{-, \cdot\} \times \Gamma_2 \times \{-, \cdot\} \times \dots \times \Gamma_k \times \{-, \cdot\}$

方法二: “拼接”: 每段模拟一条带



## 证明:每个多带TM都有等价单带TM.

证明:设计单带TM  $S$ 来模拟多带TM  $M$ .

◆  $S =$  “对于输入  $w = w_1 \dots w_n$ :

◆ 1)  $S$ 在自己带上放上  $\#w_1w_2\dots w_n\# \sim \# \sim \# \dots \#$ .

◆ 2)为了模拟一步移动,

$S$ 从标记左端点的第一个 $\#$ 开始扫描,

直到标记右端点的第 $k+1$ 个 $\#$ , 确定虚拟读写头下的符号.

然后 $S$ 进行第二次扫描,根据 $M$ 的转移函数来更新带子.

◆ 3)任何时候,只要 $S$ 将某个虚拟读写头向右移动到某个 $\#$ 上,  $S$ 就这个位置写下空白符,

并把这个位置以右的所有内容向右平移一格.

然后继续模拟.” 证毕.

定理:每个多带TM都有等价单带TM.

推论: 图灵可识别当且仅当可用多带图灵机识别.

# 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

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

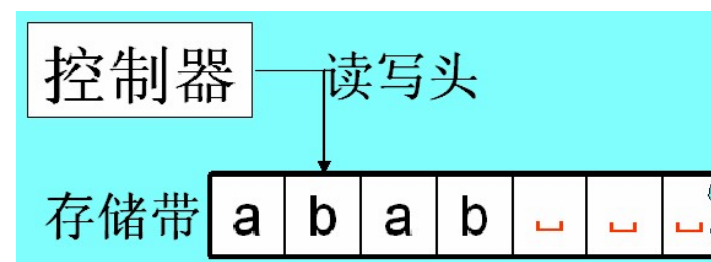
- 称NTM M接受x, 若在x上运行M时有接受分支.

- 称一NTM为判定的,

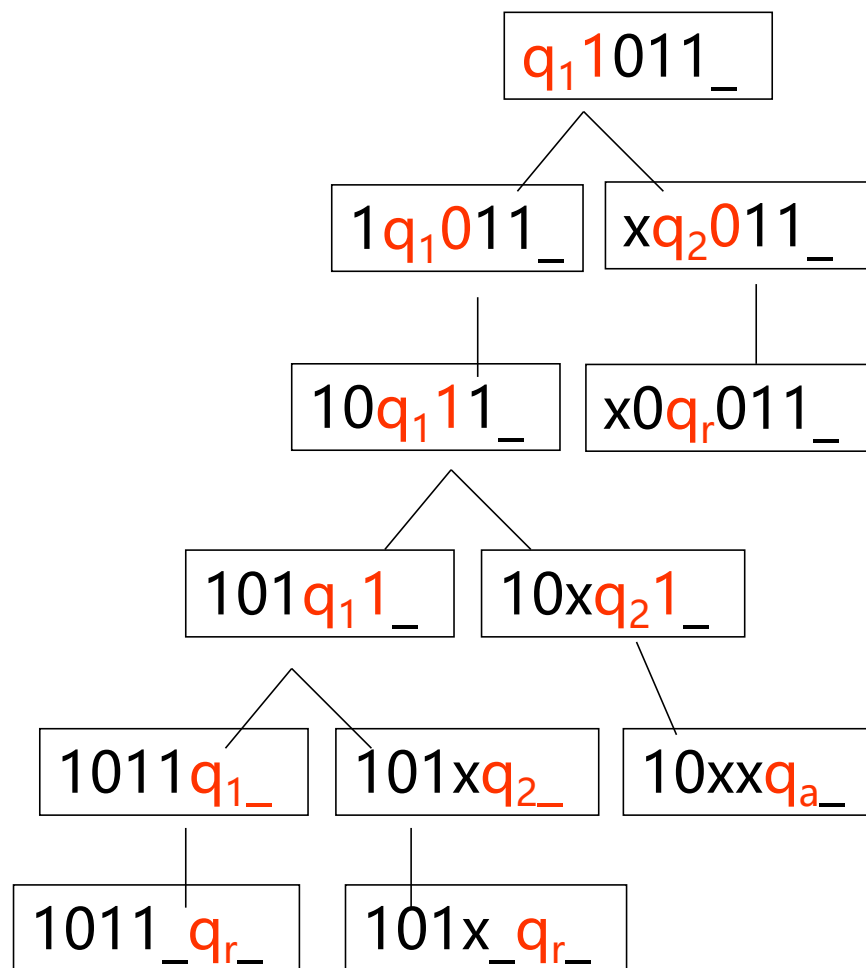
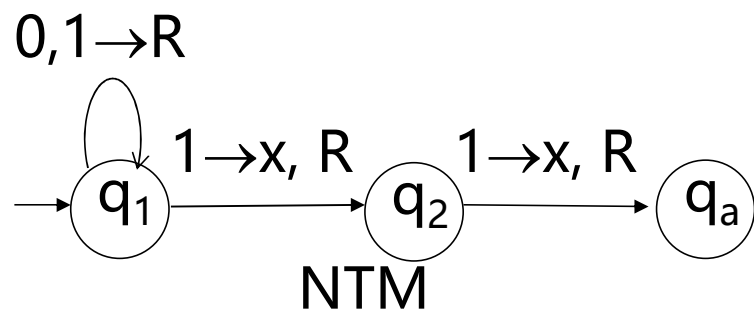
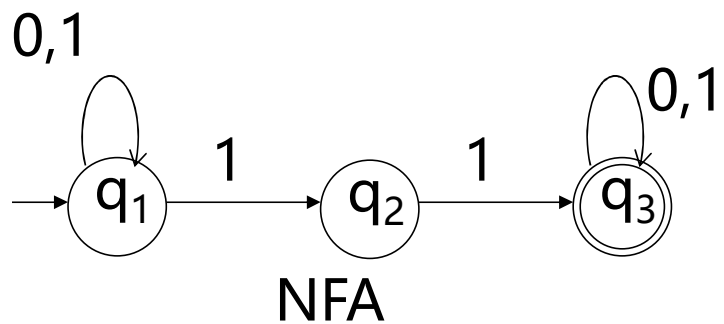
若它对所有输入, 所有分支都停机.

- 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



# 非确定型图灵机(NTM)

每个NTM都有等价的确定TM.

- ◆ 推论1:图灵可识别当且仅当可用非确定型图灵机识别。
- ◆ 推论2:图灵可判定当且仅当可用非确定型图灵机判定。

计算装置的工作方式:

识别器:输入 $x$ ,  $M$ 输出 $0/1/?$

判定器:输入 $x$ ,  $M$ 输出 $0/1$

转换器:输入 $x$ ,  $M$ 输出 $y$

产生器:输入 $0^n$ ,  $M$ 输出 $x^n$

枚举器:输入 $\varepsilon$ ,  $M$ 输出 $L(M)$ 中的所有串:  $x_1, x_2, x_3, \dots$

▶ 无遗漏, 无多余(可重复), 无顺序

## 定理：图灵可识别等价于可枚举

定理：图灵可识别等价于可枚举

◆ 分析：

1. 可识别  $\Leftarrow$  可枚举：

要识别某元素

只要等待枚举器输出该元素

2. 可识别  $\Rightarrow$  可枚举：

要枚举语言A的元素

只要列出  $\Sigma^*$  中的各个元素, 逐个识别是否属于A。

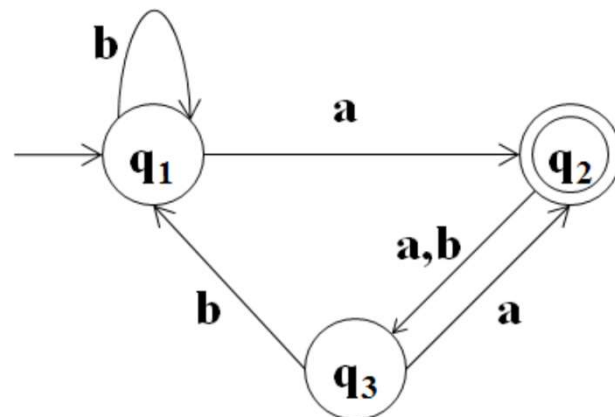


# 计算理论第1章作业

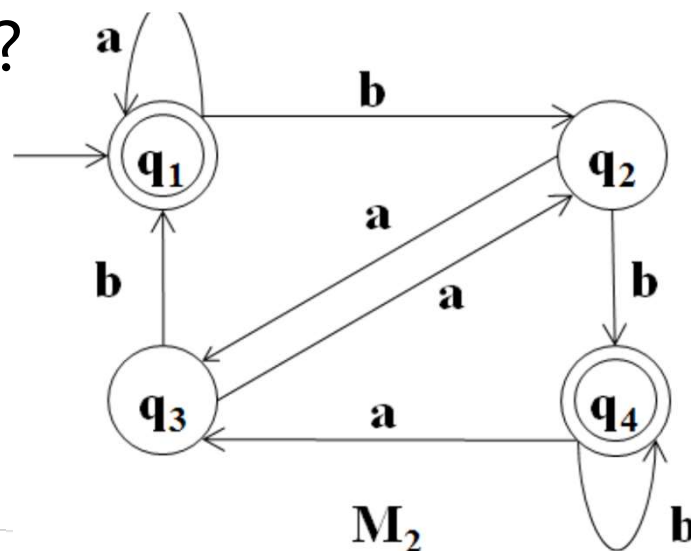
1.1 下图给出了两台DFA  $M_1$ 和 $M_2$ 的状态图。

回答下述关于这两台机器的问题。

- 它们的起始状态是什么？
- 它们的接受状态集是什么？
- 对输入aabb，它们经过的状态序列是什么？
- 它们接受字符串aabb吗？
- 它们接受字符串 $\varepsilon$ 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.  $\{ w \mid w \text{ 的长度不小于3, 并且第3个符号为0} \}$ ;

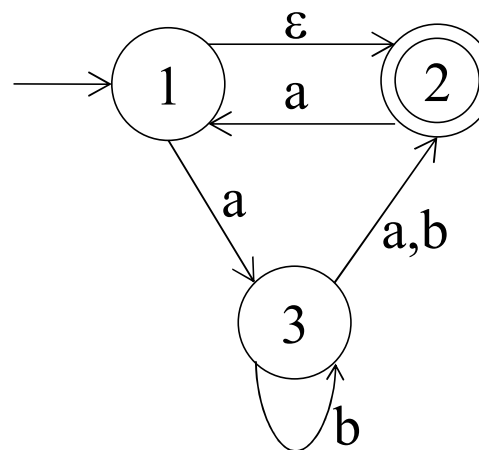
1.7. 给出下述语言的NFA, 并且符合规定的状态数。

字母表为 $\{0,1\}$

e. 语言 $0^*1^*0^*0$ , 3个状态。

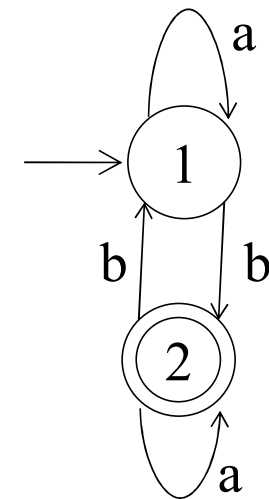
## 计算理论第1章作业

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.



1.16(b)题图

1.21(a) 将如右图的有限自动机转换成等价的正则表达式.



1.21(a)题图

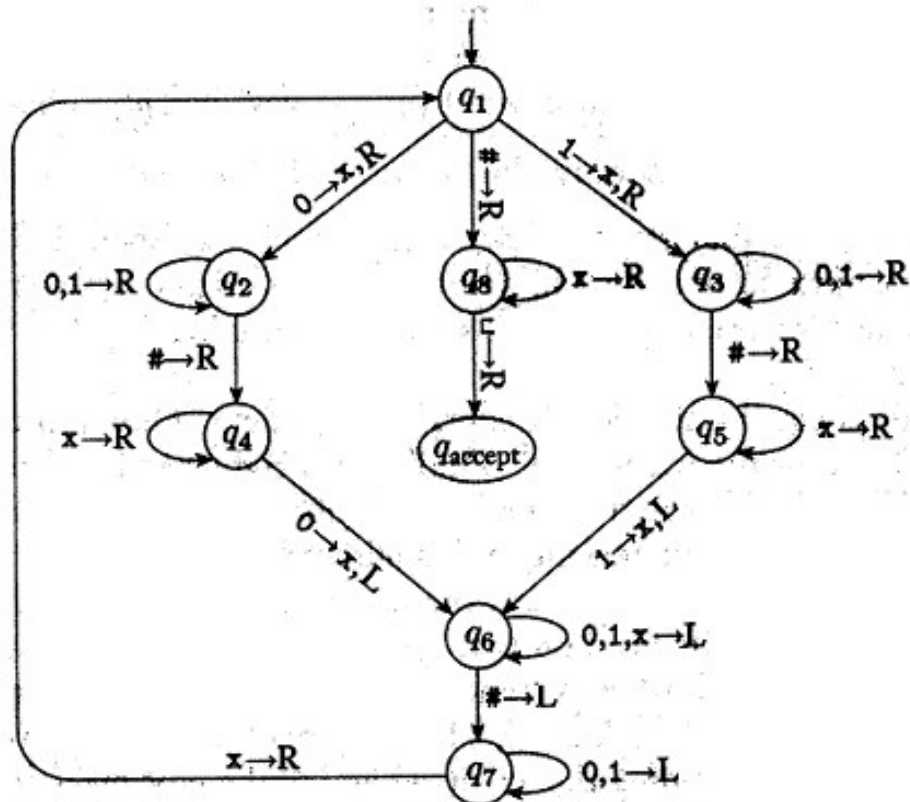
## 计算理论第1章作业

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



补充说明: 没有画出的箭头指向拒绝状态

3.2 对于识别 $\{w|w=u\#u, u \in \{0,1\}^*\}$ 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.  $\{w|w \text{ 所包含的 } 0 \text{ 的个数是 } 1 \text{ 的个数的两倍}\}$

c.  $\{w|w \text{ 所包含的 } 0 \text{ 的个数不是 } 1 \text{ 的个数的两倍}\}$

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  $c_1x^n + c_2x^{n-1} + \dots + c_nx + c_{n+1}$  有根  $x = x_0$ ,  $c_{\max}$  是  $c$  的最大绝对值. 证明  $|x_0| \leq (n+1) c_{\max} / |c_1|$

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  $c_1x^n + c_2x^{n-1} + \dots + c_nx + c_{n+1}$  有根  $x = x_0$ ,  $c_{\max}$  是  $c_i$  的最大绝对值. 证明  $|x_0| \leq (n+1) c_{\max} / |c_1|$

