第3章 图灵机

第3章 图灵机

- 3.1. 图灵机基础
  - 图灵机的定义
  - 图灵机举例
  - 图灵机的描述
- 3.2. 图灵机的变形
- 3.3. 算法的定义

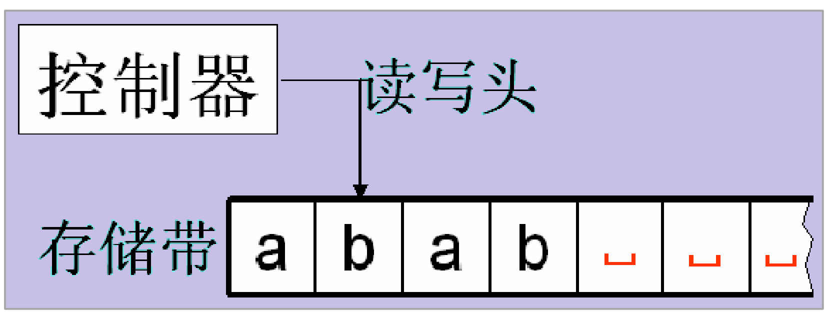
图灵对计算的观察

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,

用笔
- 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.



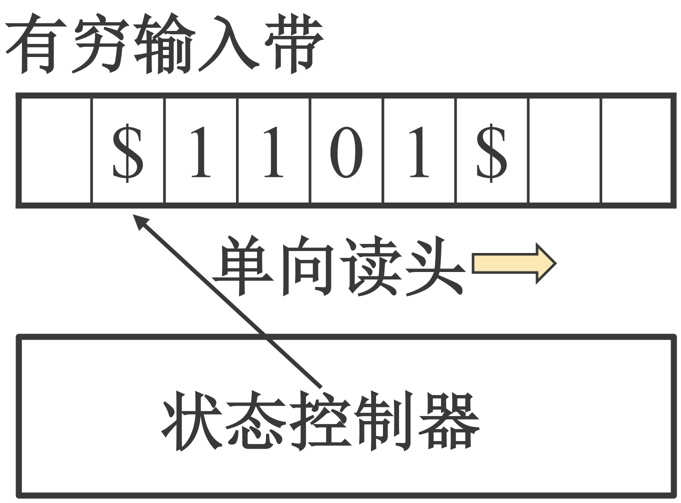
脑:控制器　纸:存储带

眼睛和笔:读写头

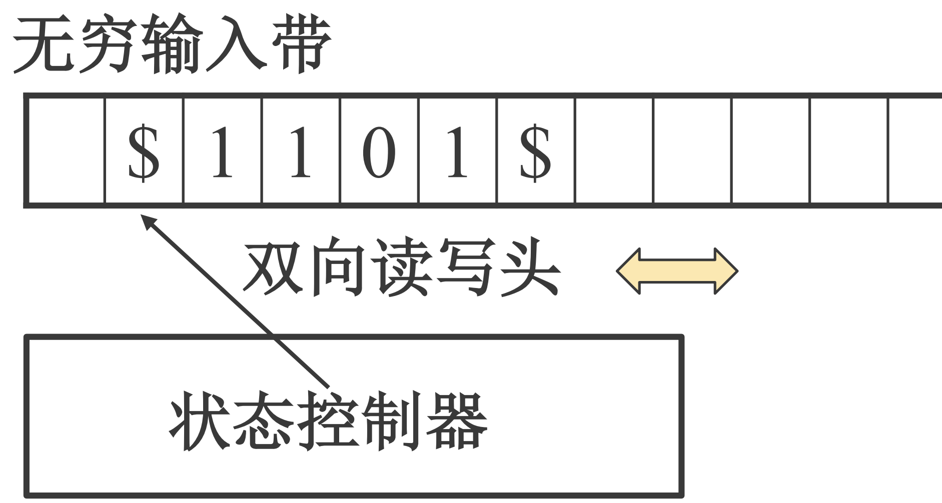
法则:转移函数

图灵机与有限自动机的区别

有限自动机



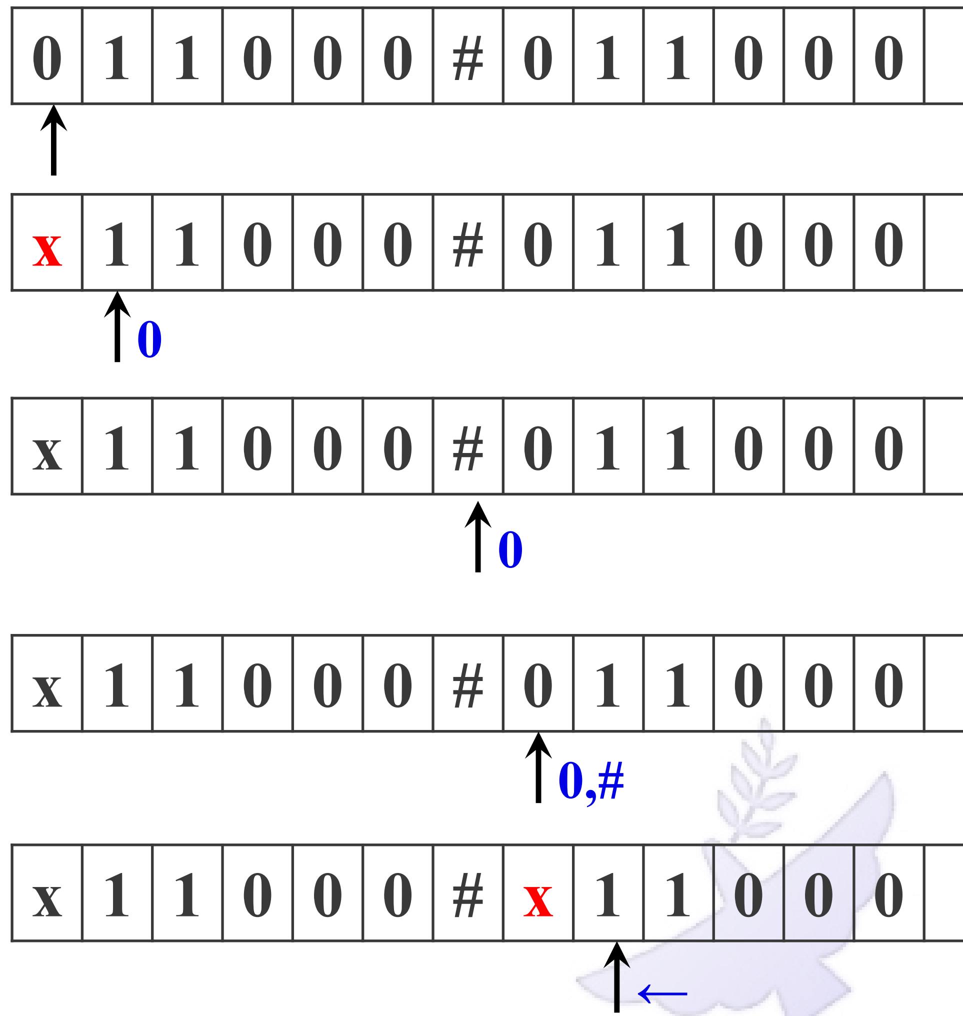
图灵机



	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

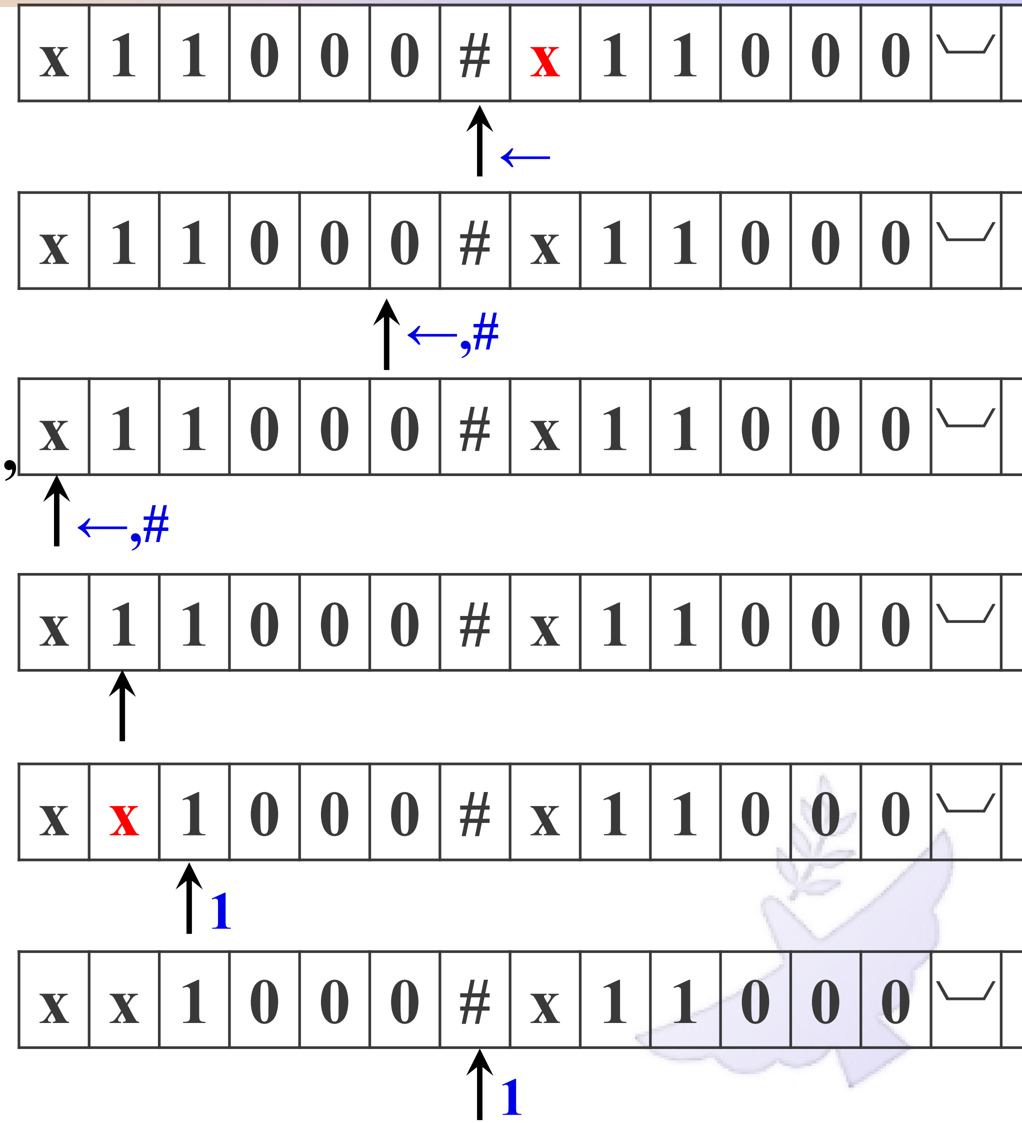
识别w#w的单带图灵机M_1

- B = { w#w | w∈{0,1}* }
- M_1=“对于输入字符串x:
- 1) 扫描输入,确认只含一个#.
  否则拒绝.
- 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.”



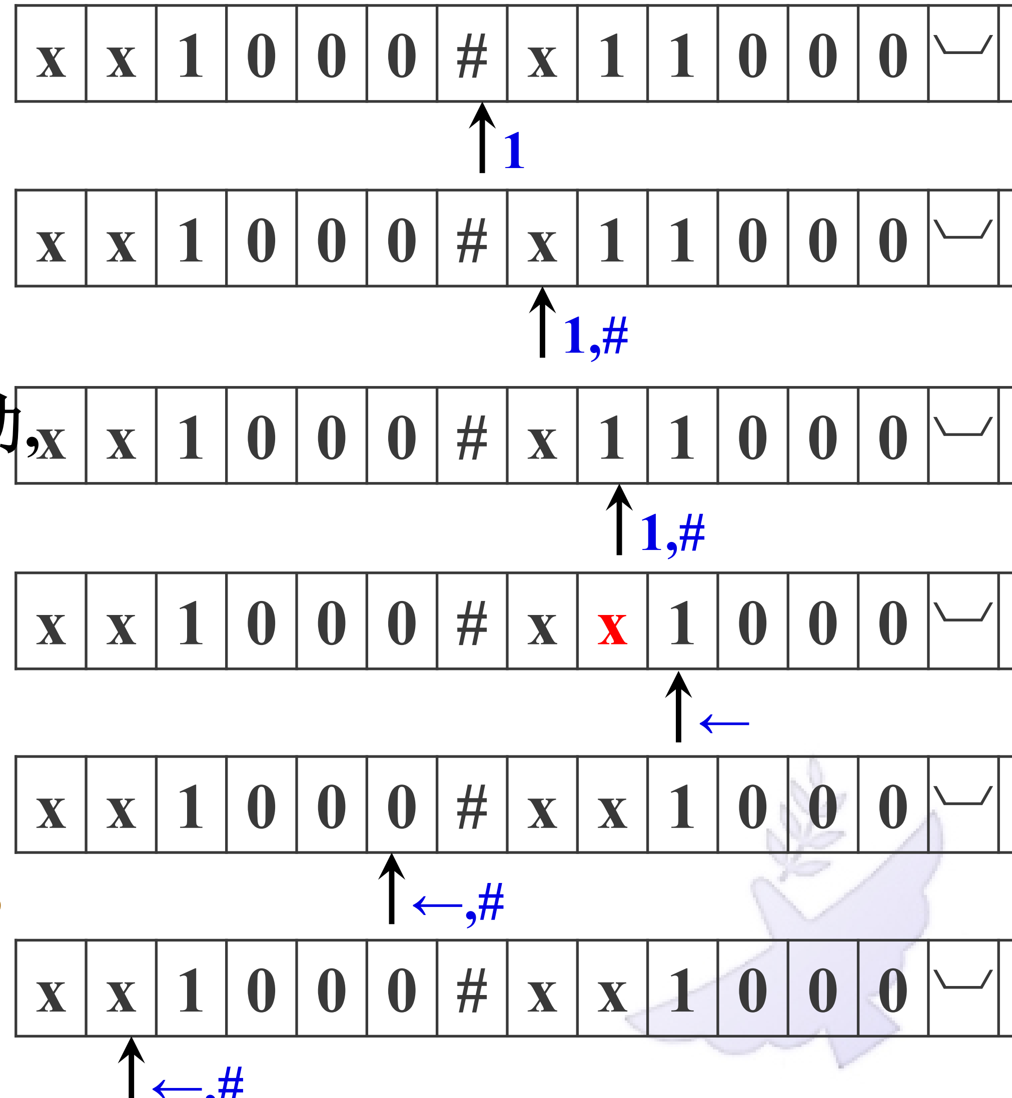
识别w#w的单带图灵机M_1

- B = { w#w | w∈{0,1}* }
- M1="对于输入字符串x:
- 1) 扫描输入,确认只含一个#.
  否则拒绝.
- 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."



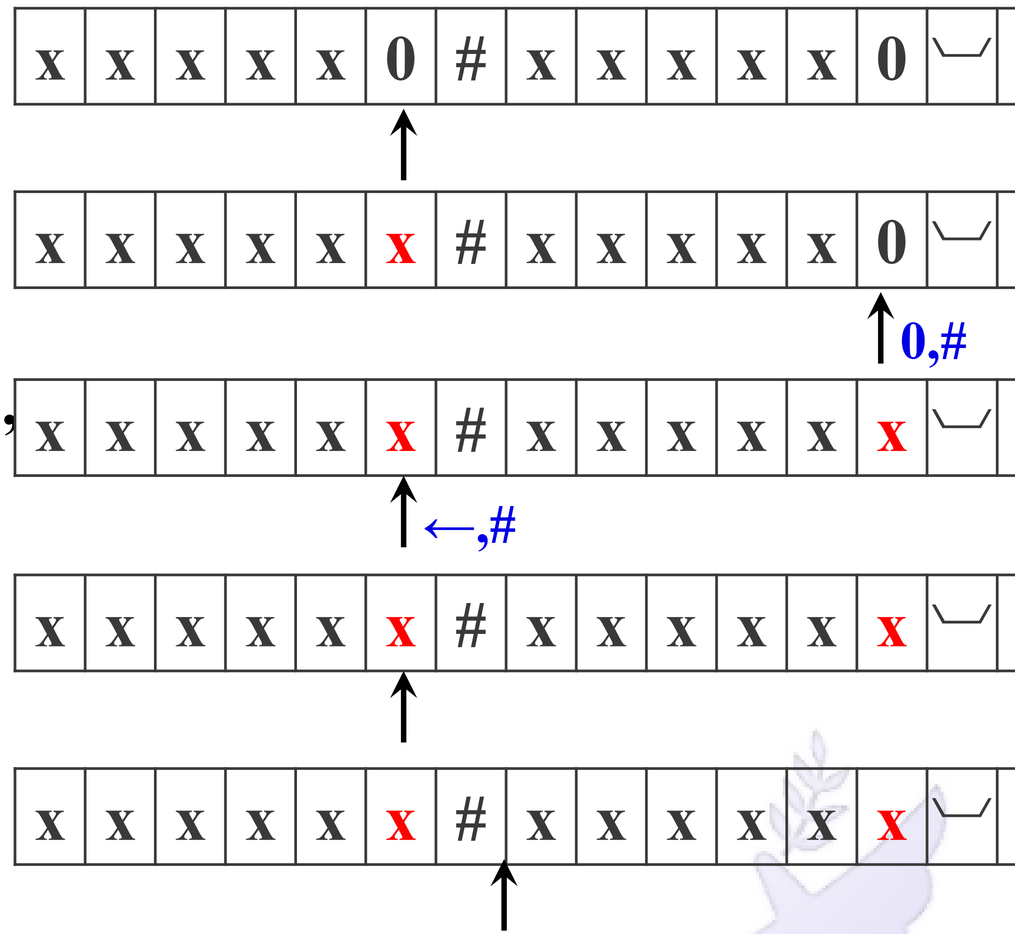
识别w#w的单带图灵机M_1

- B = { w#w | w∈{0,1}* }
- M1="对于输入字符串x:
- 1) 扫描输入,确认只含一个#.
  否则拒绝.
- 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."



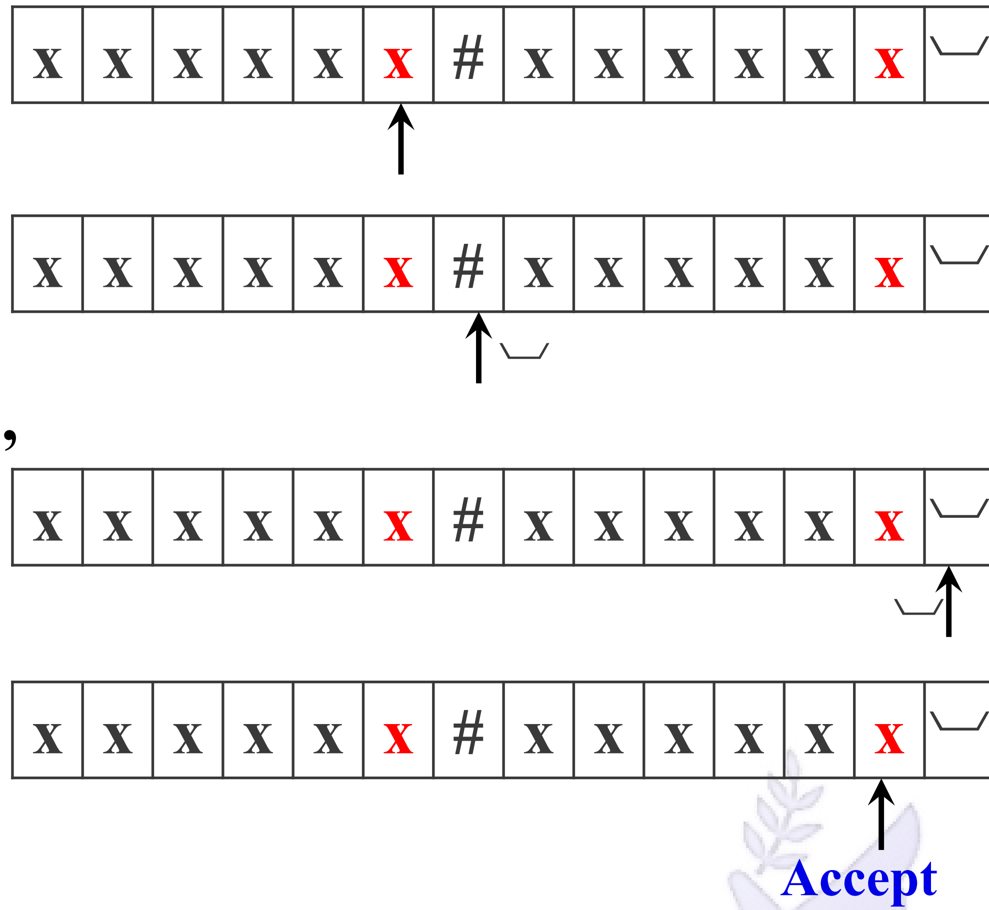
识别w#w的单带图灵机M_1

- B = { w#w | w∈{0,1}* }
- M1="对于输入字符串x:
- 1) 扫描输入,确认只含一个#.
  否则拒绝.
- 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."



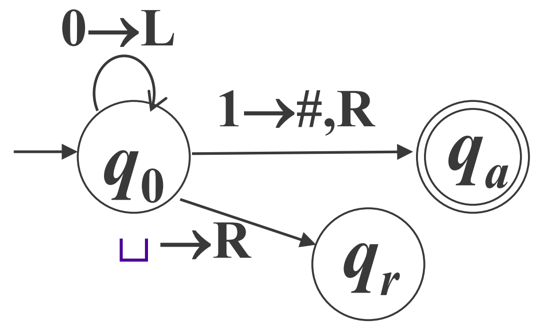
识别w#w的单带图灵机M_1

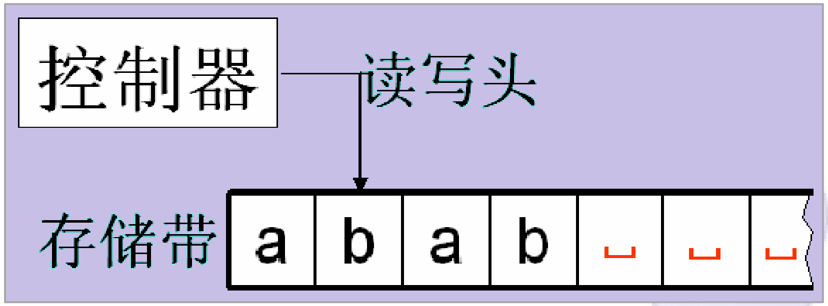
- B = { w#w | w∈{0,1}* }
- M1="对于输入字符串x:
- 1) 扫描输入,确认只含一个#.
  否则拒绝.
- 2) 在#两边对应位置来回移动,
  检查是否含相同符号.
  是则消去已检查过符号.
  若不是,则拒绝.
- 3) 当消去#左边所有符号时,
  检查#右边是否还有符号,
  若是,则拒绝.
  否则接受."



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$\sqcup$.
  3) Γ是带字母表,其中$\sqcup \in \Gamma$, $\Sigma \subset \Gamma$.
  4) $\delta: Q\times\Gamma \to Q\times\Gamma\times\{L, R\}$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, $q_a \neq q_r$.

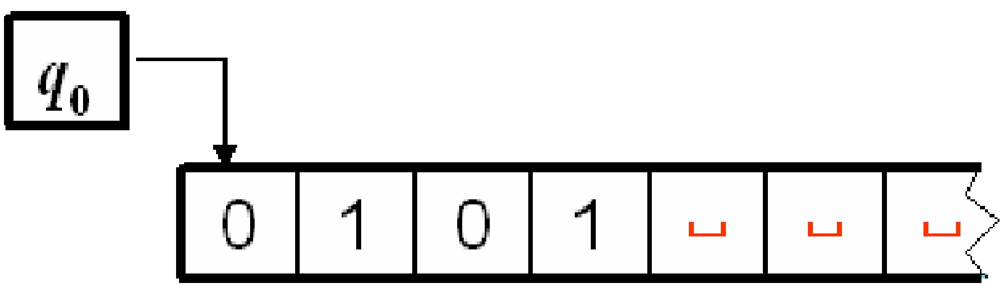




图灵机的初始化

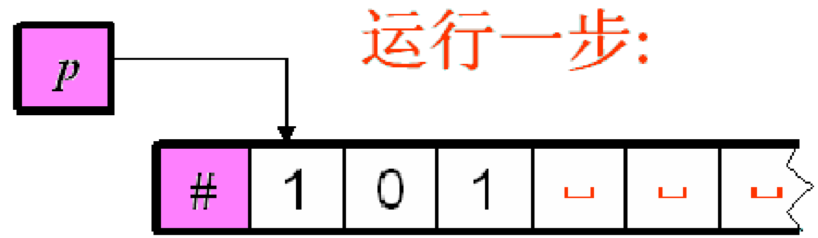
- 设M=(Q, Σ, Γ, δ, q_0, q_a, q_r), w=$w_1 \ldots w_n \in \Sigma^n$,
  - 输入带: 将输入串w放在最左端n格中,
  - 带子其余部分补充空格␣.
  - 读写头: 指向工作带最左端.

例:设输入串为0101, 则其初始形态为



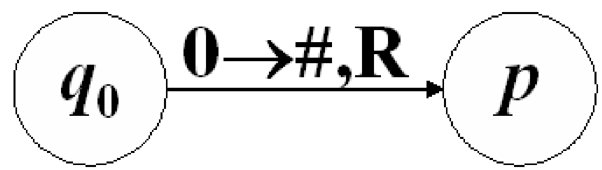
图灵机的运行

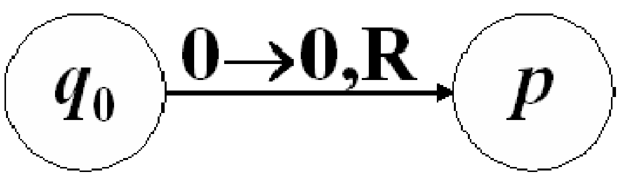
- 图灵机根据转移函数运行.
- 例:设输入串为0101, 且$\delta(q_0,0)=(p,\#,R)$, 则有



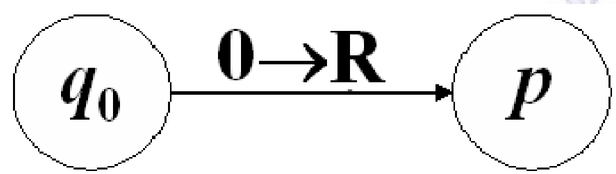
• 注: 若要在最左端左移, 读写头保持不动.

$\delta(q_0,0)=(p,\#,R)$的状态图表示:



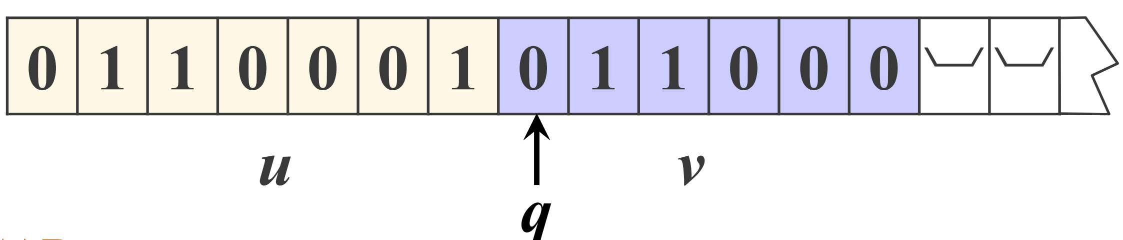


简记为



图灵机的格局

- 图灵机的格局: 描述图灵机运行的每一步需要的信息:
  - 1) 控制器的状态;
  - 2) 存储带上字符串;
  - 3) 读写头的位置.
- 定义: 对于图灵机M=(Q, Σ, Γ, δ, 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.



图灵机的格局

- 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 。
  - 起始格局： q_0w， w是输入串
  - 接受格局： $uq_{acc}v$
  - 拒绝格局： $uq_{rej}v$
  - 停机格局: $uq_{acc}v$, $uq_{rej}v$
- 格局示例

0	1	1	0	0	0	1	0							…

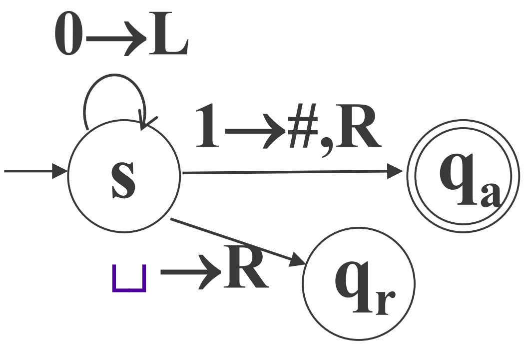
q_7

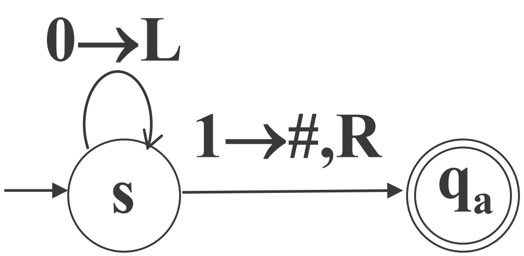
$0110q_7\textcolor{red}{0}010$

图灵机格局的产生

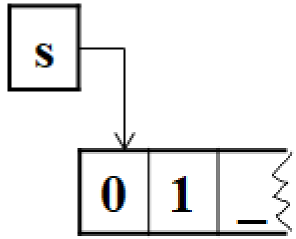
- 格局C_1产生格局C_2：
- 如果$\delta(q_i, b)=(q_j, c, L)$, 则
  - uaq_ibv 产生 uq_jacv
  - q_ibv 产生 q_jcv (若读头在带左端不能向左移)
- 如果$\delta(q_i, b)=(q_j, c, R)$, 则
  - uaq_ibv 产生 $uacq_jv$

格局演化举例





省略拒绝状态



01:

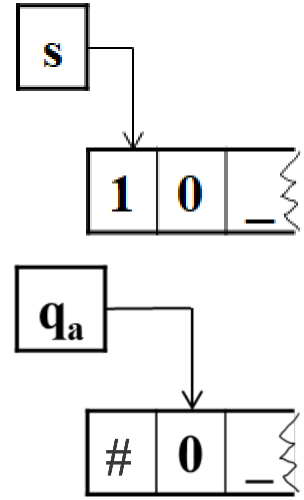
s 0 1

s 0 1

…

循环

不停机



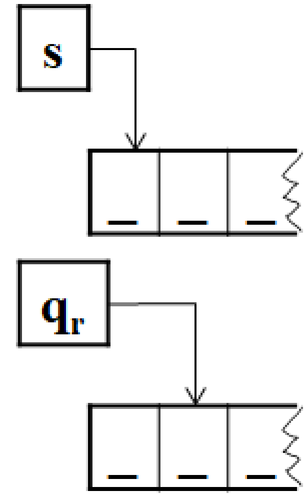
10:

s 1 0

q_a 0

接受

停机



ε:

s _ _

_ q_r _

拒绝

停机

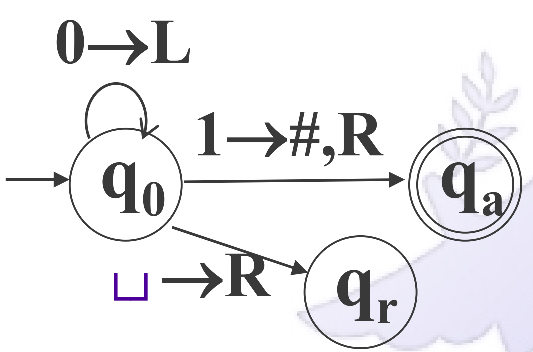
图灵机计算的形式定义

- 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.
- **图灵机M(识别/接受)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

- 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.



判定器与语言分类

- 图灵可识别: A=L(M).
  - x∈A时, M在x上停机接受
  - x∉A时, M在x上停机拒绝 或 不停机.
- 图灵可判定: A=L(M)
  - x∈A时, M在x上停机接受
  - x∉A时, M在x上停机拒绝(处处停机)
- 定义不同语言类:
  - 图灵可识别语言: 某个图灵机的语言(也称递归可枚举语言) .
  - 图灵可判定语言: 某个判定器的语言(也称递归语言)

概念

- 灵可识别
  - ＝递归可枚举
  - ＝计算可枚举
  - ＝半可判定
  - ＝半可计算

- 图灵可判定
  - ＝递归
  - ＝可解
  - ＝可行
  - ＝可判定
  - ＝可计算

第3章 图灵机

- 3.1. 图灵机基础
  - 图灵机的定义
  - 图灵机举例
  - 图灵机的描述
- 3.2. 图灵机的变形
- 3.3. 算法的定义

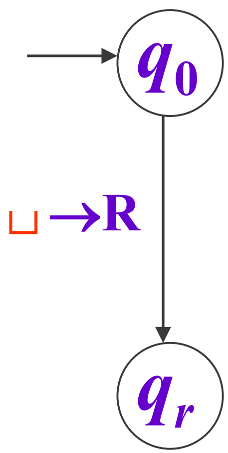
图灵机举例

- $\Sigma=\{0,1\}$, $A=\{0w1 : w\in\Sigma^*\}$ 正则语言
- $\Sigma=\{0,1\}$, $B=\{0^n1^n : n\geq 0\}$ 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

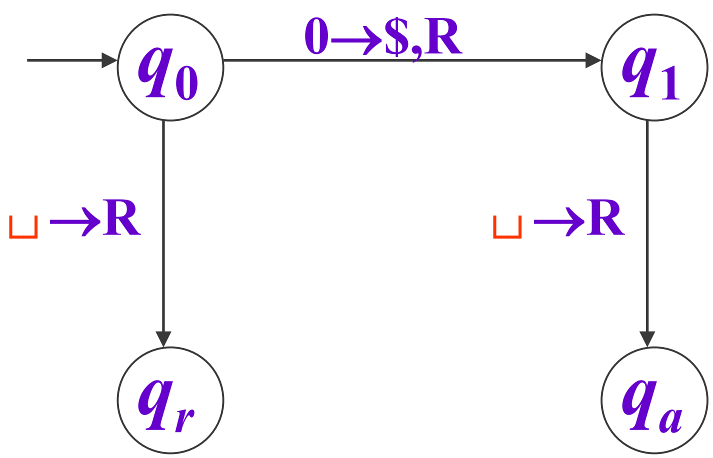
状态图:Σ={0}，C={0^k:$k=2^n,n\geq 0$}

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)."



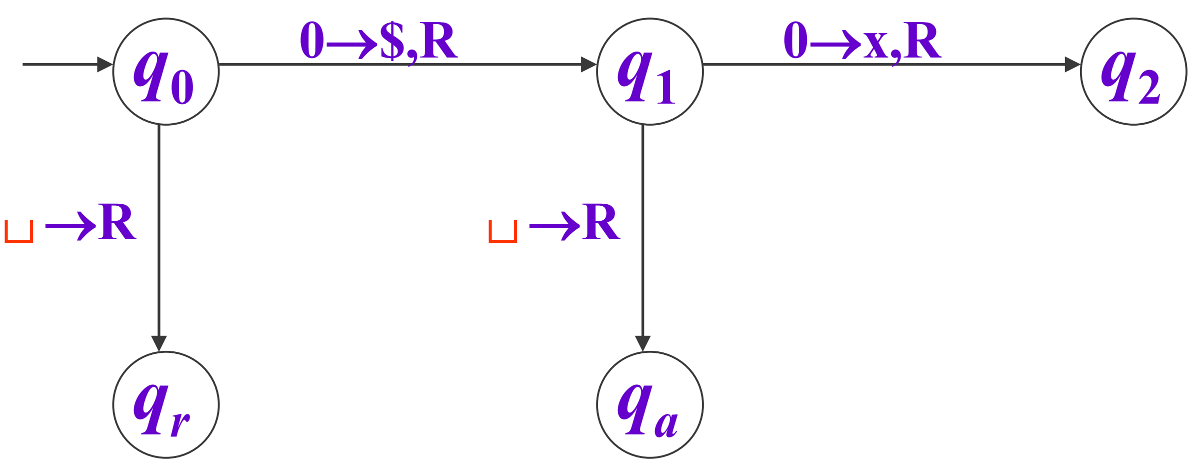
状态图:$\Sigma=\{0\}$，$C=\{0^k:k=2^n,n\geq 0\}$

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)."



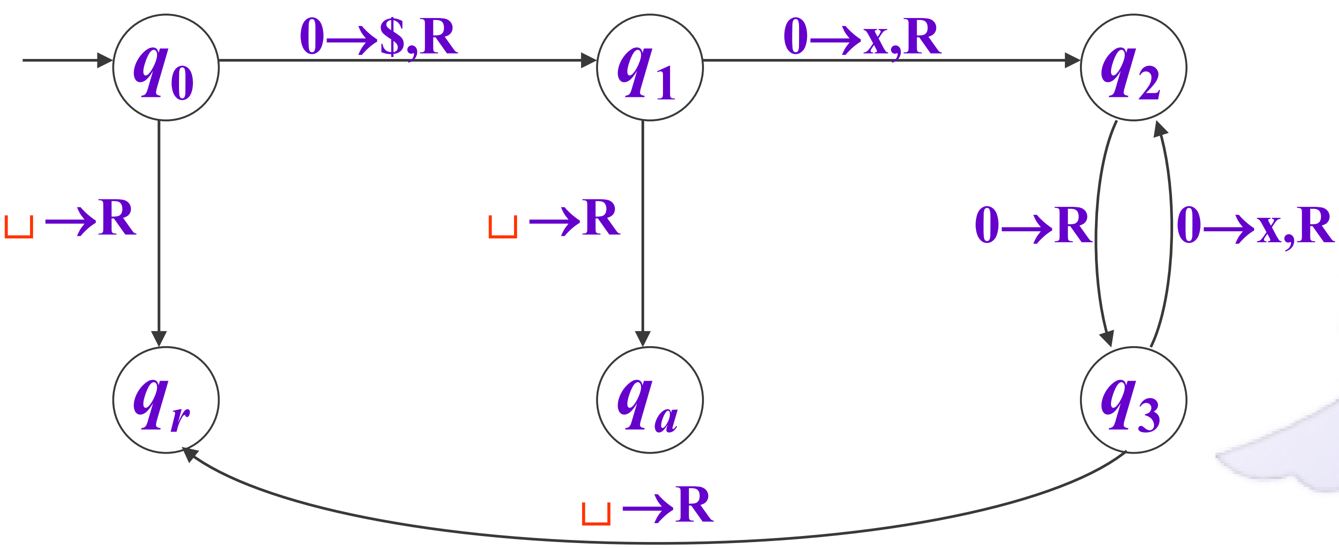
状态图:$\Sigma=\{0\}$，$C=\{0^k:k=2^n,n\geq 0\}$

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)."



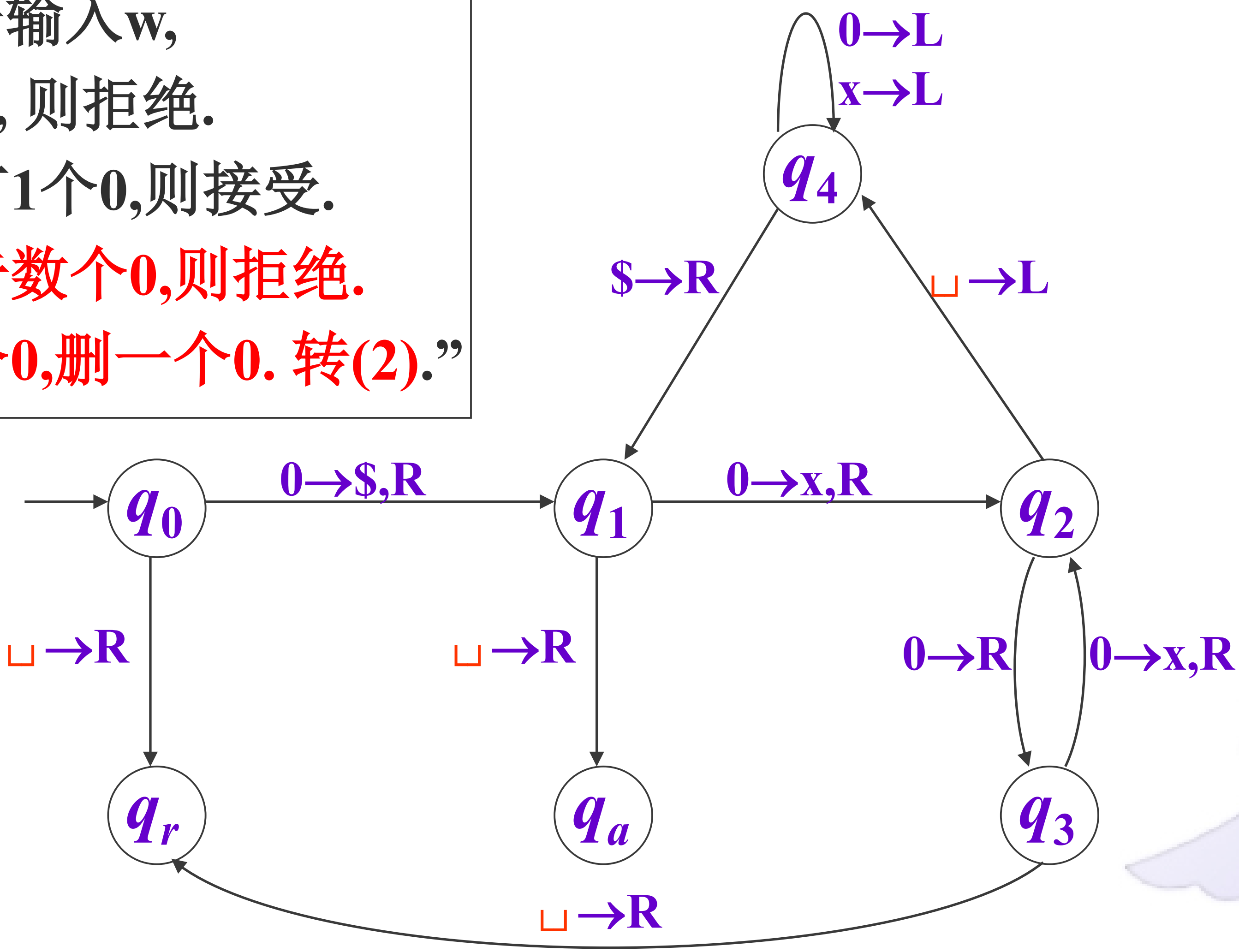
状态图:$\Sigma=\{0\}$，$C=\{0^k: k=2^n, n\geq 0\}$

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

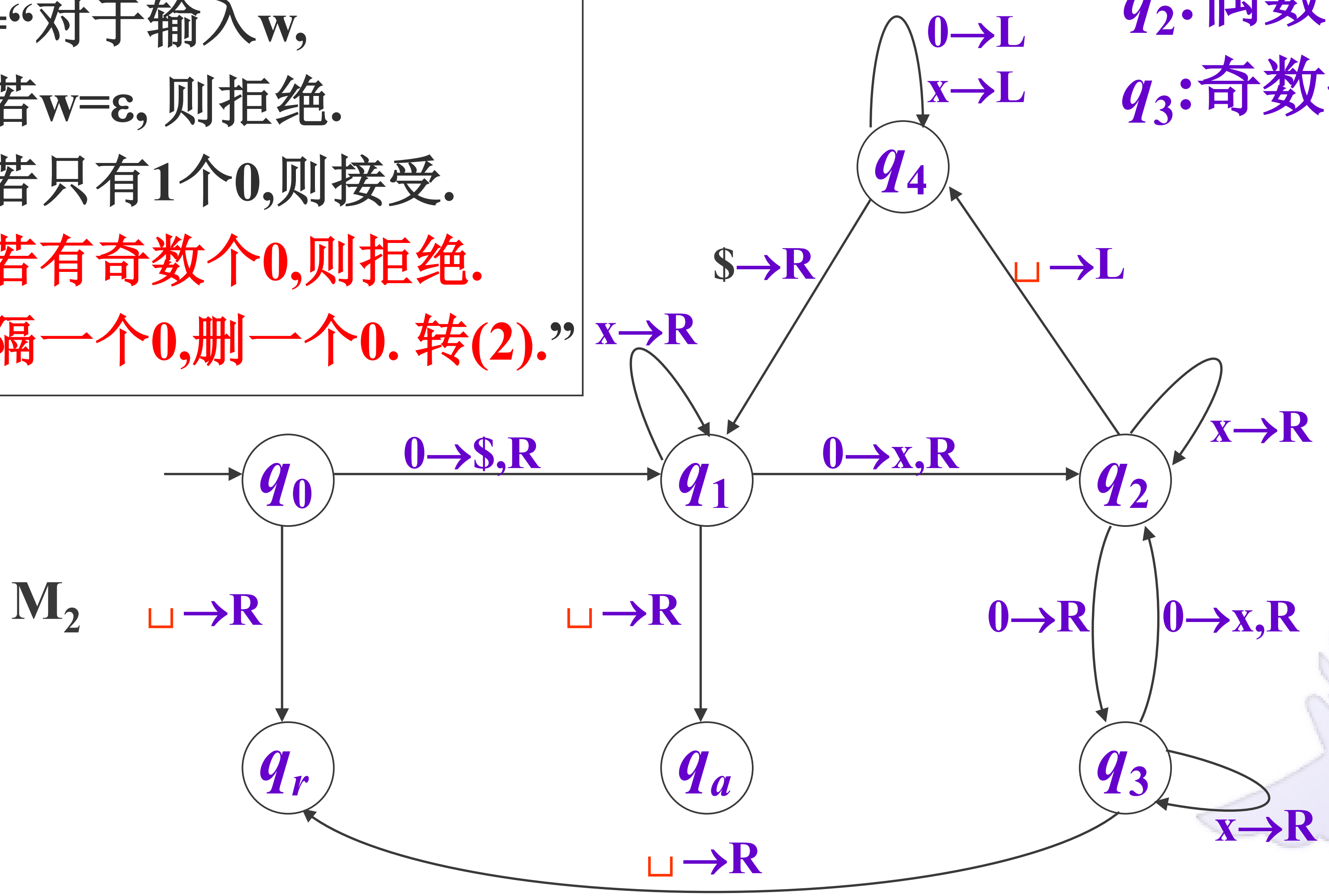
M="对于输入w,
1) 若w=ε, 则拒绝.
2) 若只有1个0,则接受.
3) 若有奇数个0,则拒绝.
4) 隔一个0,删一个0. 转(2)."



状态图:Σ={0}，C={0^k:$k=2^n$,$n\geq 0$}

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)."

q_2:偶数个零
q_3:奇数个零



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

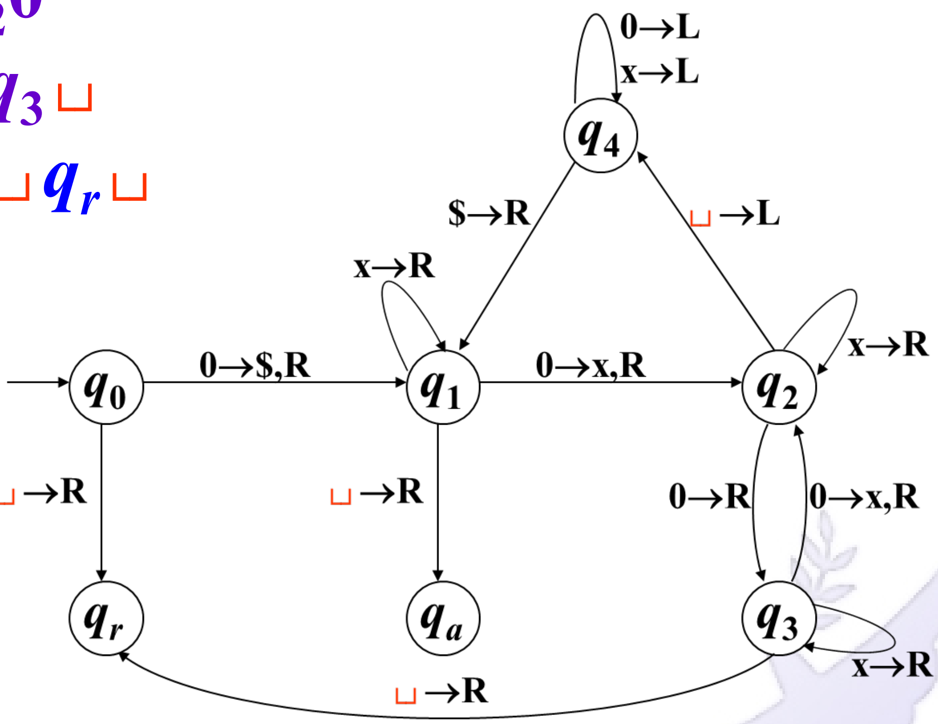
- $\Sigma=\{0\}$，$C=\{0^k:k=2^n,n\geq 0\}$ 图灵可判定语言
- $M_2=(Q,\Sigma,\Gamma,\delta,q1,q_a,q_r)$
  - $Q=\{q_0, q_1, q_2, q_3, q_4, q_a,q_r\}$
  - $\Sigma=\{0\}$
  - $\Gamma=\{0,x,\$,\sqcup\}$，用\$作为左端点定界
  - δ

分析图灵机M_2

$q_0 0$
$\$ q_1 ␣$
$\$ ␣ q_a ␣$

$q_0 00$
$\$ q_1 0$
$\$ x q_2 ␣$
$\$ q_4 x$
$q_4 \$ x$
$\$ q_1 x$
$\$ x q_1 ␣$
$\$ x ␣ q_a ␣$

$q_0 000$
$\$ q_1 00$
$\$ x q_2 0$
$\$ x 0 q_3 ␣$
$\$ x 0 ␣ q_r ␣$



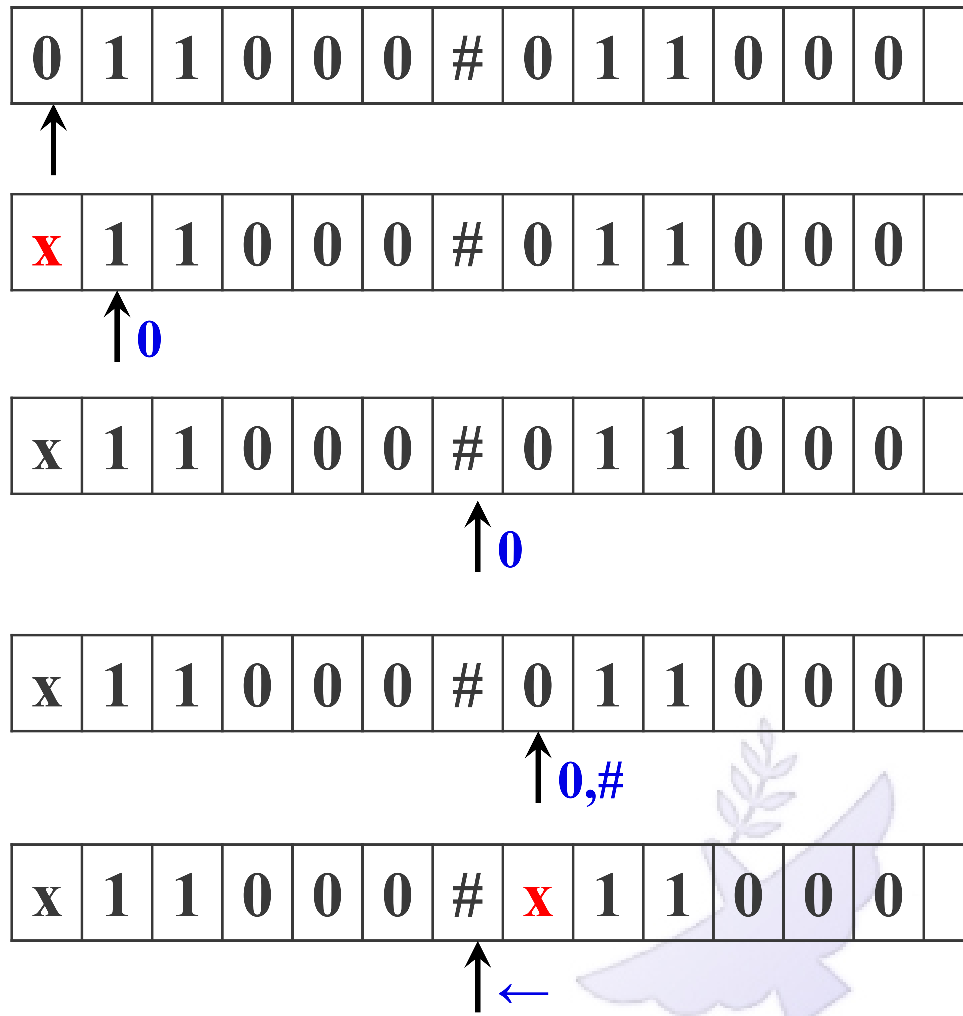
M 是判定器

$L(M)=\{0^k : k=2^n, n\geq 0\}$.

$\{0^k : k=2^n, n\geq 0\}$是图灵可判定语言

识别w#w的单带图灵机M_1

- B = { w#w | w∈{0,1}* }
- M1="对于输入字符串x:
- 1) 扫描输入,确认只含一个#.
  否则拒绝.
- 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."



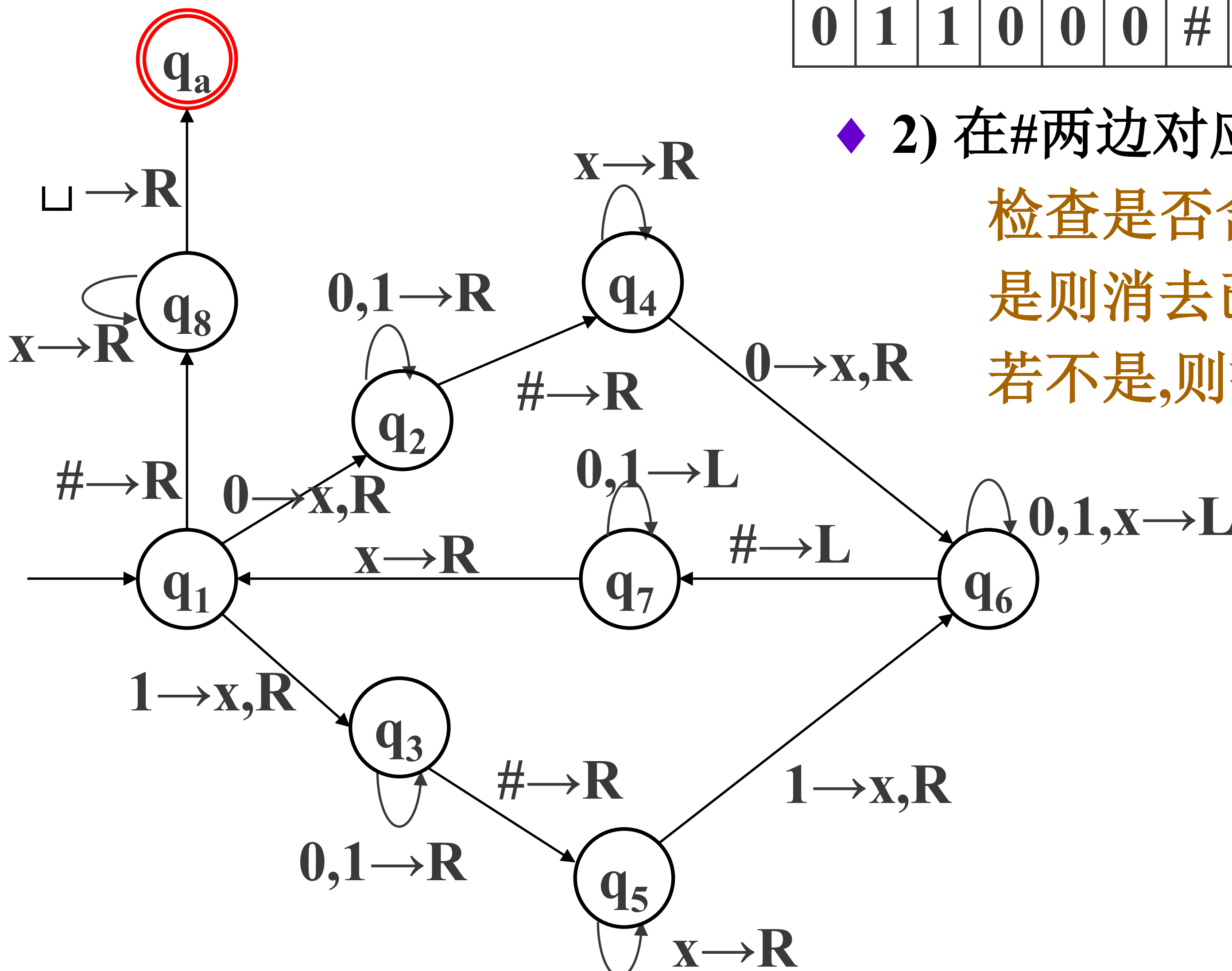
识别w#w的单带图灵机M_1

- $M_2=(Q, \Sigma, \Gamma, \delta, q_1, q_a, q_r)$
  - $Q=\{q_0, q_1, q_2, \ldots q_8, q_a, q_r\}$
  - $\Sigma=\{0, 1, \#\}$
  - $\Gamma=\{0, 1, \#, x, \sqcup\}$
  - δ
  - 省略到拒绝状态的转移

M_1的状态转移图

- { w#w | w∈{0,1}* }

0	1	1	0	0	0	#	0	1	1	0	0	0



- 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.

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

- M_3=“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. ”

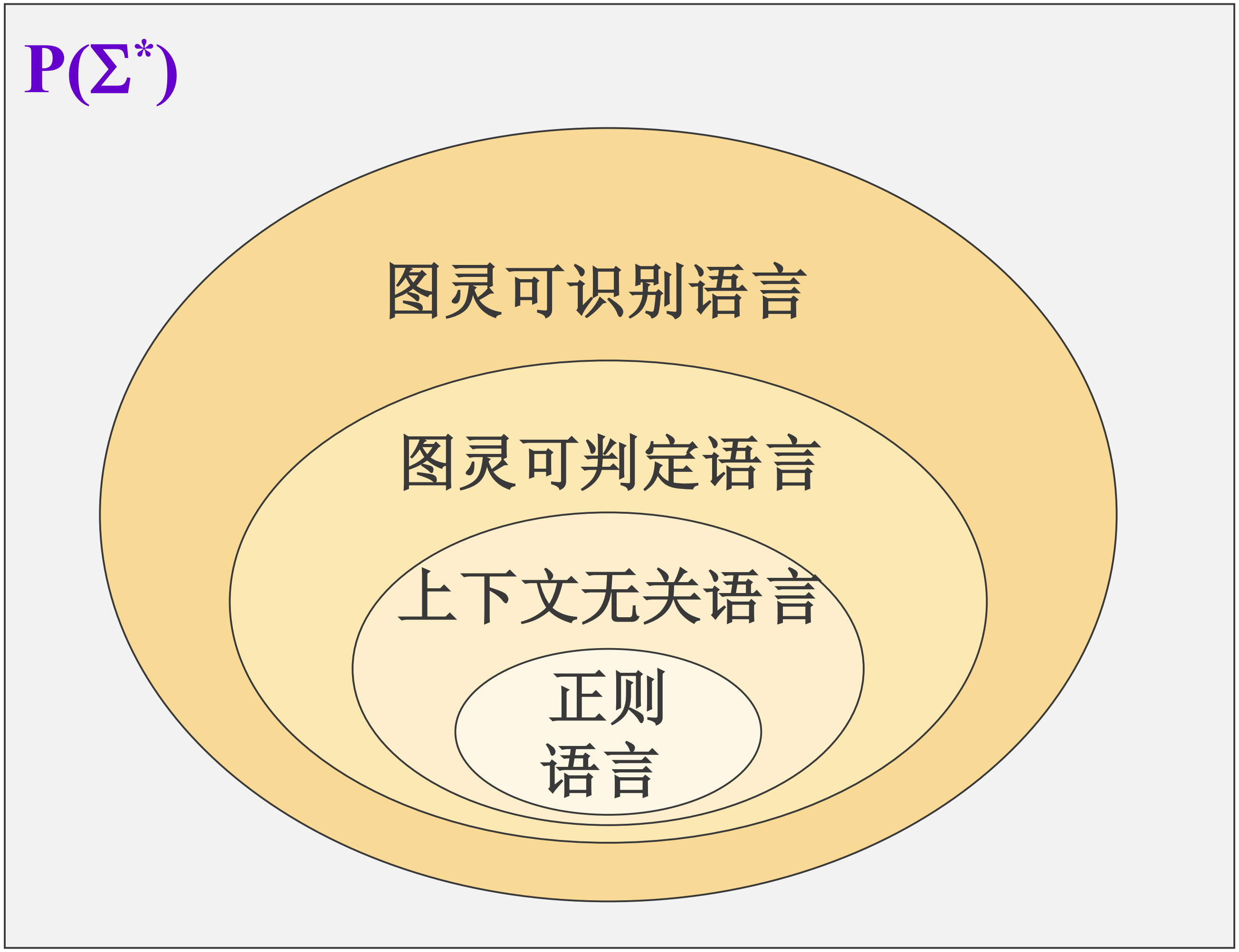
$E=\{\#x_1\#x_2\#\ldots\#x_k \mid x_i\in\{0,1\}^*, \forall i\neq j, x_i\neq x_j\}$

- M_4="对输入w:

1) 在最左端带符号顶上做记号.
   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x_1, 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



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

正则语言

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

上下文无关语言

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

图灵可判定语言

第3章 图灵机

- 3.1. 图灵机基础
  - 图灵机的定义
  - 图灵机举例
  - 图灵机的描述
- 3.2. 图灵机的变形
- 3.3. 算法的定义

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.

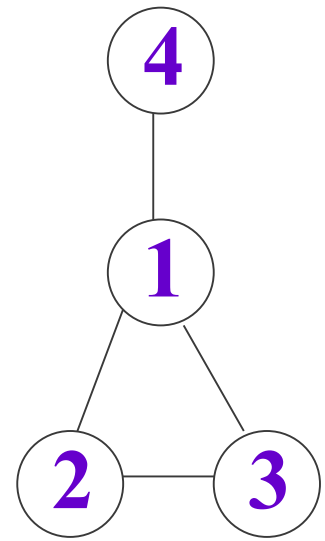
3) 若0的个数为奇数, 则拒绝.

4) 从带左端隔一个0, 删一个0. 转(2)."

图灵机的输入

- 由定义, 图灵机的输入总是字符串.
- 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

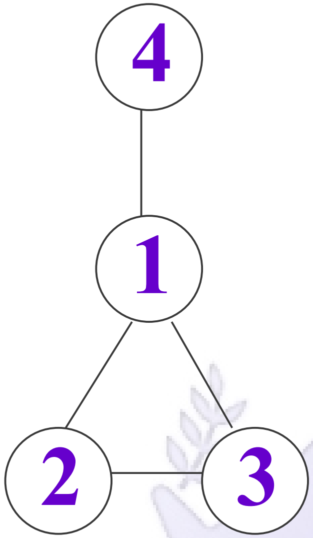
- M_5=“对于输入<G>, G是一个无向图,

  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_5的语言可知:

$L(M_5)$={<G> | G是连通的无向图 }

G=(1,2,3,4)((1,2),(2,3),(3,1),(1,4))

A={<G>|G 是连通无向图

- **图G的编码：**

  <G>=(1,2,3,4)((1,2),(2,3),(3,1),(1,4))

- **检查输入合法性：**

  顶点序列不含重复元素

  边序列中的顶点应该在顶点序列中

- **算法步骤：**

  1) 在最左端数字上加个点

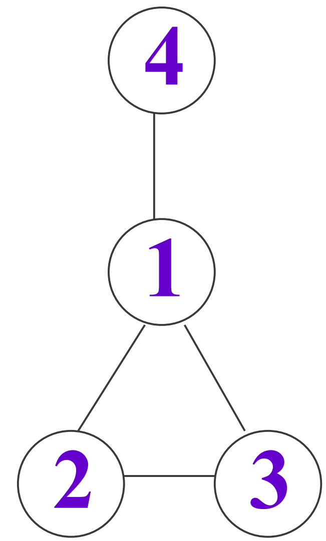
  2) 在一个未加点的顶点n_1下画线，

     在一个已加点的顶点n_2下画线，

  3) 扫描边序列寻找(n_1,n_2)，

     重新设置画线顶点

  4) 扫描顶点序列,检查是否都已加点



G=(1,2,3,4)
((1,2),(2,3),(3,1),(1,4))

第3章 图灵机

- 3.1. 图灵机基础
  - 图灵机的定义
  - 图灵机举例
  - 图灵机的描述
- 3.2. 图灵机的变形
  - 多带图灵机
  - 非确定图灵机
  - 枚举器
- 3.3算法的定义

图灵机的变形

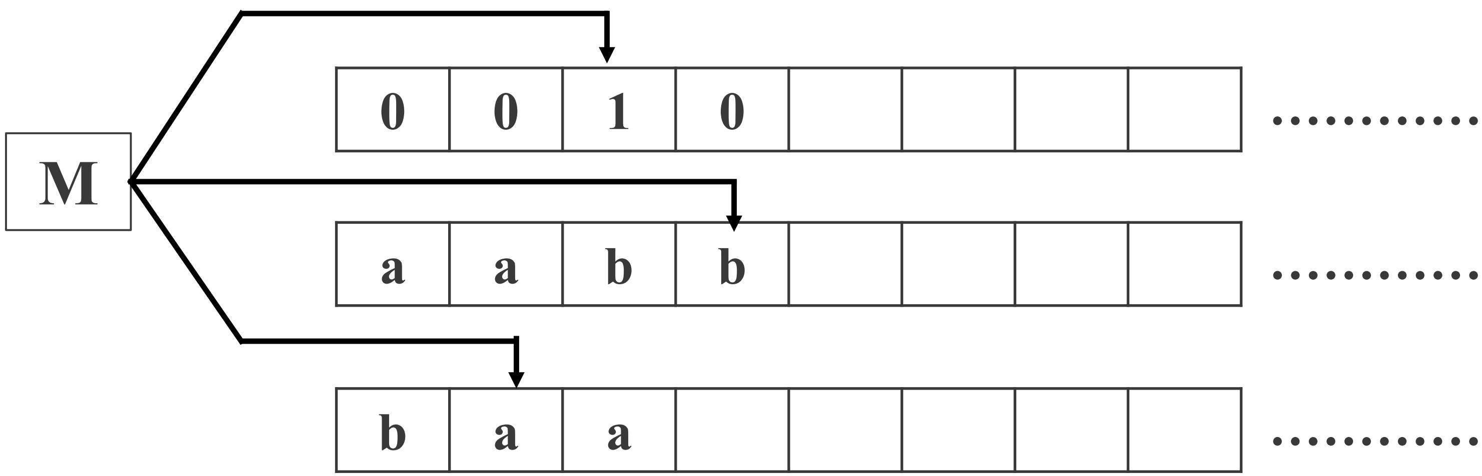
- 图灵机有多种变形: 例如
  - 多带图灵机
  - 非确定图灵机
  - 多头图灵机
  - 枚举器
  - 带停留的图灵机等等
  - ……
- 只要满足必要特征, 它们都与这里定义的图灵机等价.

多带图灵机

- 多带图灵机的转移函数:

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

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



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)
  - $\Gamma = \Gamma_1\times\{-,\cdot\}\times\Gamma_2\times\{-,\cdot\}\times\ldots\Gamma_k\times\{-,\cdot\}$
  - 方法二：“拼接”：每段模拟一条带

0	0	1	0				
		●					
a	a	b	b				
			●				
b	a	a					
	●						

…………

			●						●			●		
#	0	0	1	0	#	a	a	b	b	#	b	a		

…………

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$w=w_1\ldots w_n$:
- 1) S在自己带上放上$\#\dot{w}_1 w_2 \ldots w_n\#\dot{\sqcup}\#\dot{\sqcup}\#\ldots\#$.
- 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就在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.

#	0	0	$\dot{1}$	0	#	a	a	b	$\dot{b}$	#	b	$\dot{a}$			....

多带图灵机

- 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.

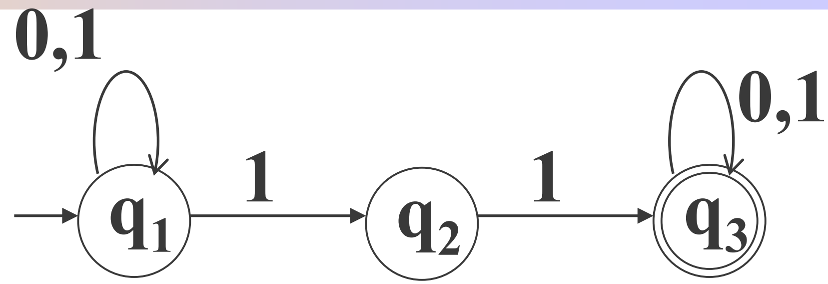
非确定型图灵机(NTM)

- **DFA的转移函数**

  $\delta: Q\times\Sigma\rightarrow Q$

- **NFA的转移函数:**

  $\delta: Q\times\Sigma_\varepsilon\rightarrow P(Q)$



NFA

- **TM的转移函数**

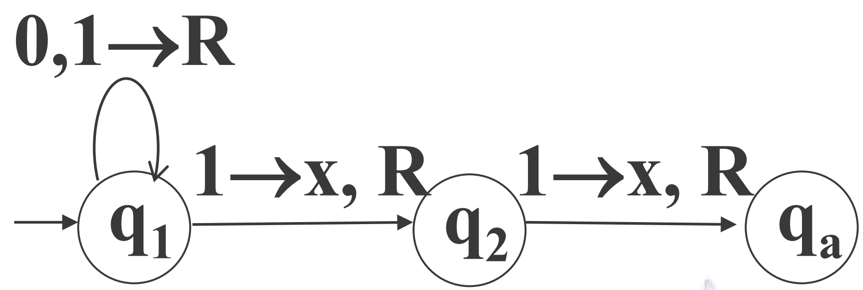
  $\delta: Q\times\Gamma\rightarrow Q\times\Gamma\times\{L,R\}$

- **NTM的转移函数 (没有ε箭头)**

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

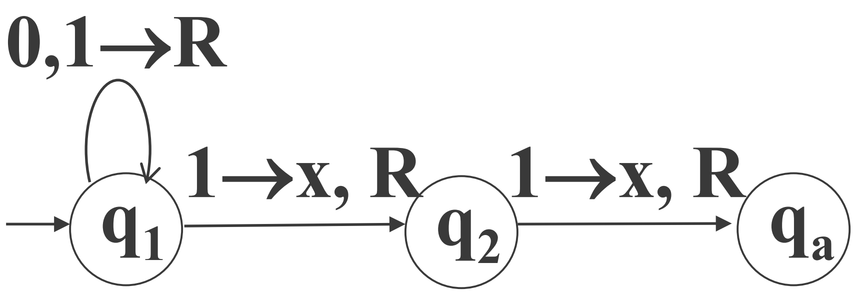
- **NTM转移函数举例**

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

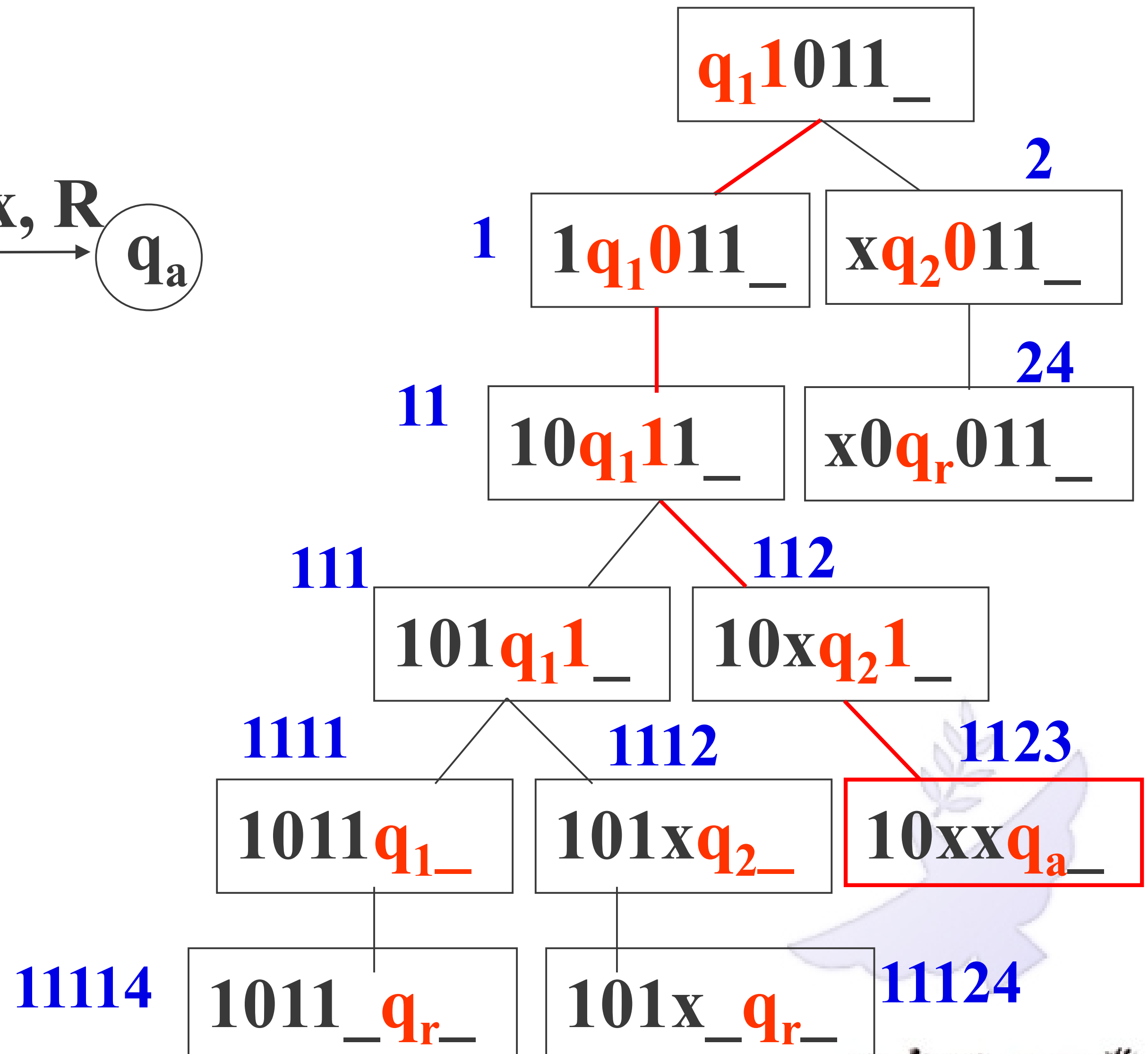


NTM

非确定型图灵机(NTM)举例

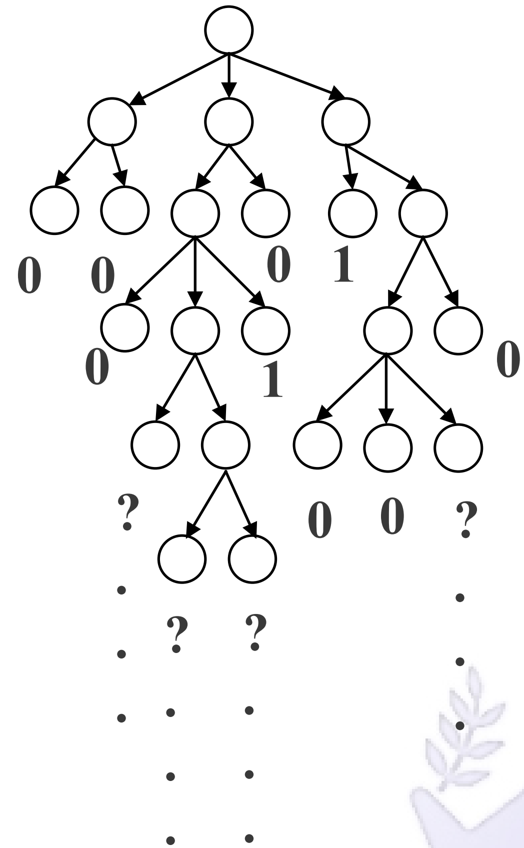


NTM



非确定型图灵机的计算

- 设读头读入的符号为s,
- 对(每个副本)机器状态q,若q有多个射出s箭头，则机器把自己复制为成多个副本.
- 称NTM M接受x, 若在x上运行M时有接受分支.

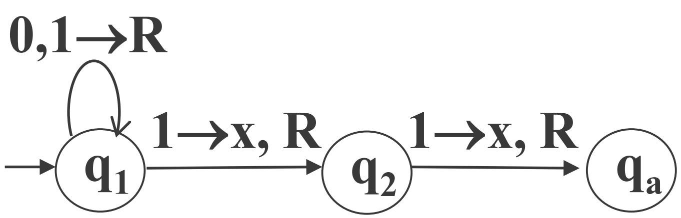


非确定型图灵机(NTM)

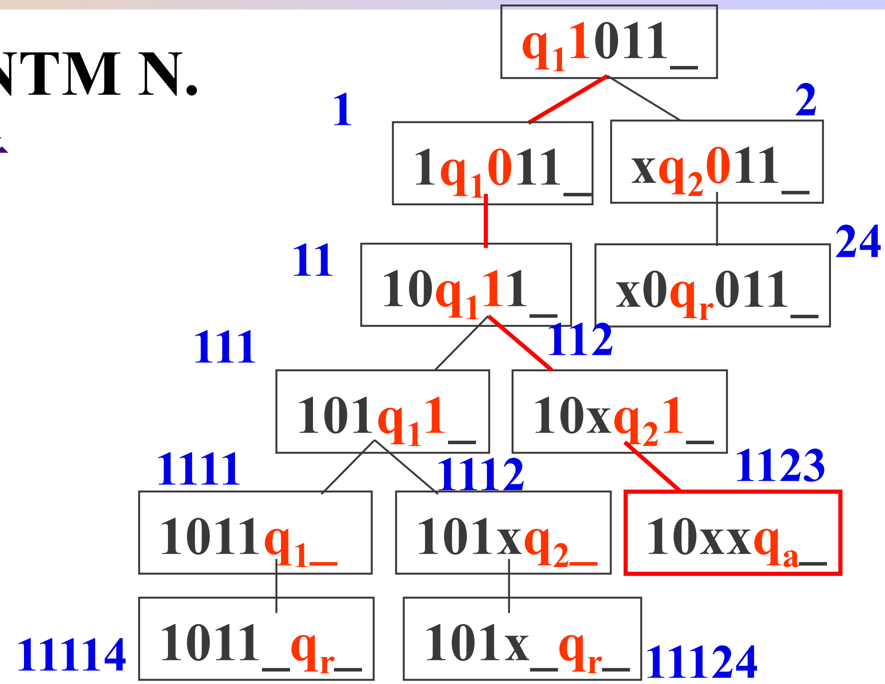
- 称**NTM M接受x**, 若在x上运行M时有接受分支.
- 称一个**NTM为判定的**, 若它对所有输入, 所有分支都停机.
- **定理: 每个NTM都有等价的DTM.**
- **定理: 每个判定NTM都有等价的判定DTM.**

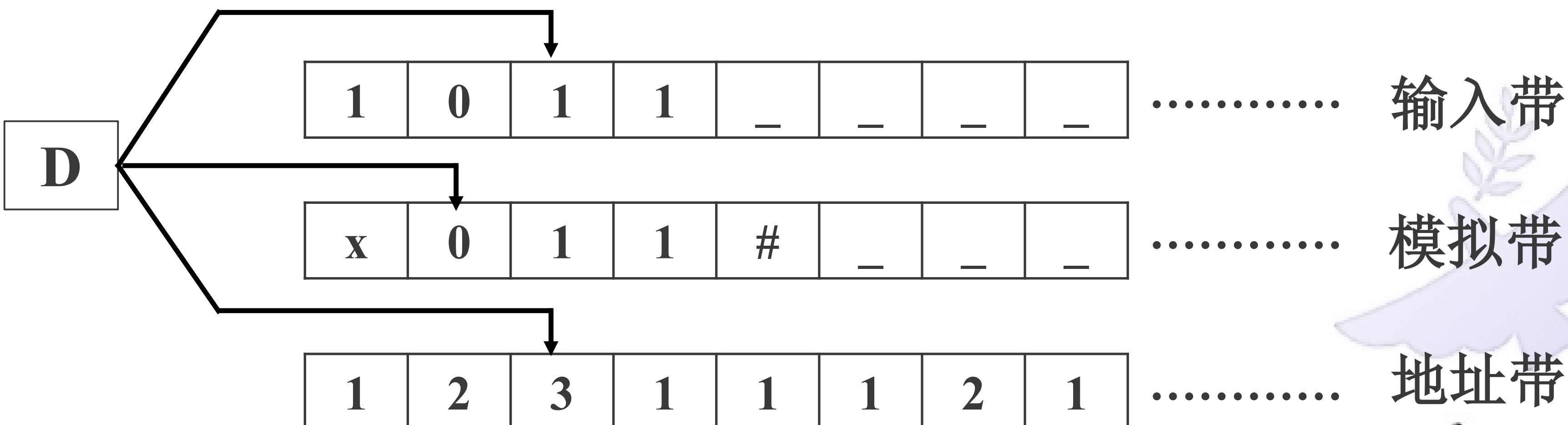
证明: 每个NTM都有等价的DTM.

- 证明思路: 用DTM D来模拟NTM N.
  - DTM“遍历” NTM计算树
  - DFS？ BFS？



NTM





证明: 每个NTM都有等价的DTM.

- 证明: 用DTM D来模拟NTM N.

D = “对输入w:

1) 开始时,第一个带包含w, 其余两个带为空.

2) 把第一个带复制到第二个带上.

3) 在第二个带上模拟N在输入w上的某个非确定性计算分支.

   查询第三个带，确定计算分支.

   若该分支编码无效/无编码/拒绝，则进入第四步.

   若该分支遇到接受格局, 则接受.

4) 在第三个带上, 用标准序下的下一个串来替代原用的串.

   转到第二步.”

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。

枚举器与识别器

- 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ε, M输出L(M)中的所有串：$x_1, x_2, x_3, \ldots$
    - 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：
  - 可识别 ⇐可枚举:

    要识别某元素

    只要等待枚举器输出该元素
  - 可识别 ⇒可枚举:

    要枚举语言A的元素

    只要列出 Σ^*中的各个元素, 逐个识别是否属于A。

证明：图灵可识别等价于可枚举

- 证明: 可识别⇐可枚举

  设枚举器E枚举语言A, 则设计TM M识别A.

  M="对输入w:

  1) 运行E, 每当E输出一个串时,

     将这个串与w进行比较.

  2) 若w曾在E的输出中出现过,

     则接受."

证明：图灵可识别等价于可枚举

- 证明:可识别 ⇒可枚举.

  设TM M识别语言A, 则设计枚举器E枚举A.

  要枚举语言A的元素

  只要列出 Σ*中的各个元素, 逐个识别是否属于A。

- 设$\Sigma^*=\{s_1, s_2, s_3,\ldots\}$.
- E="输入:无.

  1) 对i=1,2,3,…重复下列步骤:

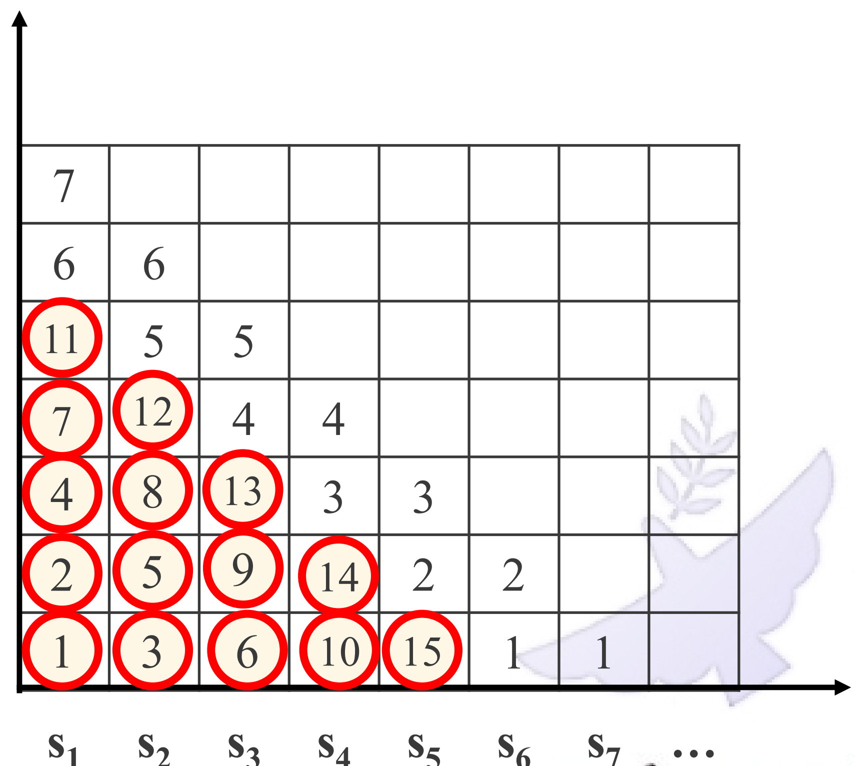
  2) 对$s_1,s_2,s_3,\ldots, s_i$中每一个,

  让M以其作为输入运行i步.

  (防止在某个串上不停机)

  3) 如果有计算接受,

  则打印相应的s_j."



证明：图灵可识别等价于可枚举

- 证明:可识别 ⇒可枚举.

- 设$\Sigma^*=\{s_1, s_2, s_3,\ldots\}$.
- E="输入:无.

  1) 对i=1,2,3,…重复下列步骤:

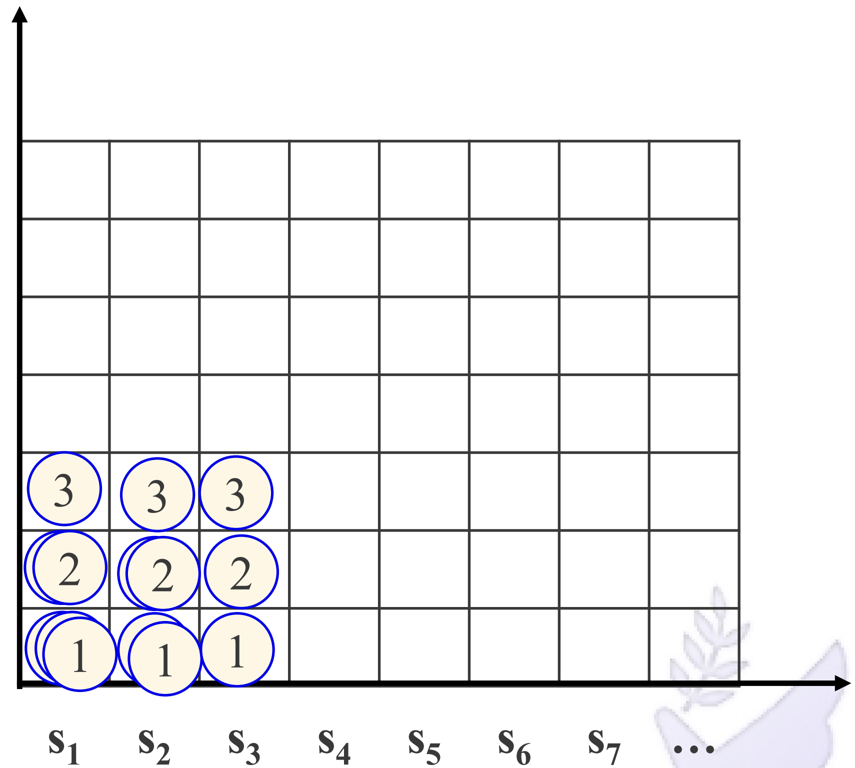
  2) 对$s_1,s_2,s_3,\ldots, s_i$中每一个,

     让M以其作为输入运行i步.

     (防止在某个串上不停机)

  3) 如果有计算接受,

     则打印相应的s_j."



第3章 图灵机

- 3.1. 图灵机基础
  - 3.1.1 图灵机的定义
  - 3.1.2 图灵机举例
  - 3.1.3 图灵机的描述
- 3.2. 图灵机的变形
- 3.3 算法的定义

算法的定义

- 算法导论[C]:

  解决可计算问题的一个工具，

  将输入转换为输出的一个可计算步骤序列。

- 韦氏大学词典:

  求解问题的一个过程, 步骤有限, 通常有重复操作;
  广义地说, 是按部就班解决一个问题的过程.

- 这些都是算法的直观解释, 包含了严格定义的所有要素

不可判定问题(没有算法)举例

- **Hilbert第十问题:**
  - “整数系数多项式方程是否有整数根”有没有算法?
  - 有没有求多项式整数根的算法
  - Hilbert: “通过有限多次运算就可以决定的过程”

$x^2+y^2-1=0$: $(\pm1,0),(0,\pm1)$

$6x^3yz^2+3xy^2-x^3-10=0$: $(5, 3, 0)$

$x^2+y^2-3=0$: 无整数根

$x^n+y^n-z^n=0$: $(n>2)$ 无正整数解，费马大定理

Hilbert第十问题

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. "

图灵机M对输入 $p(x,y) = x^2+y^2-3$不停机

图灵机M是识别器，不是判定器

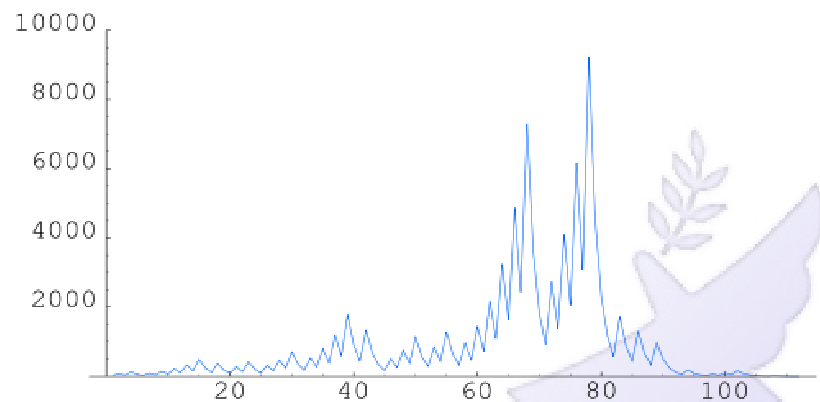
3n+1问题目前不知道有没有算法

- 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: $5 \to 16 \to 8 \to \ldots$, 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)
- $1 \sim 5\times10^{18}$都能到1.([wiki])



问题分类

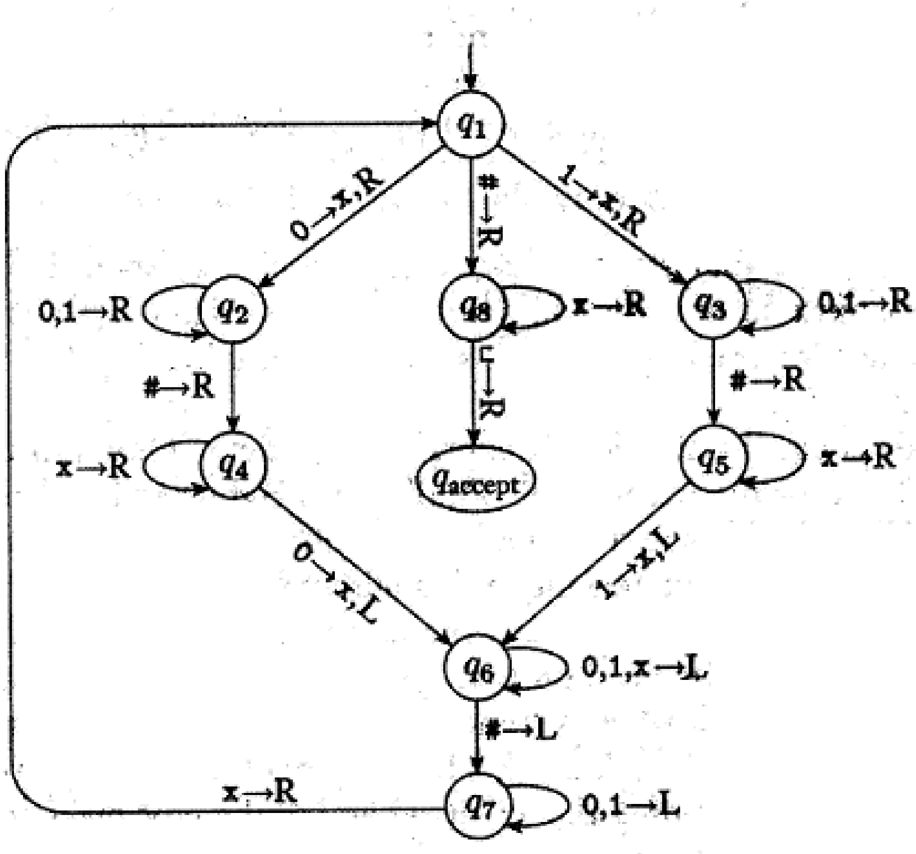
- 有算法的问题.　　　(找最大数/图连通性/最短路径)
- 没有算法的问题.　　(Hilbert第十问题)
- 不知道有没有算法的问题.　　(3n+1问题)

丘奇-图灵论题

- 算法 ≅ 处处停机的图灵机
  - = λ 演算 = 0 型文法
  - = ……
- 丘奇-图灵论题(Church-Turing Thesis)：算法的非形式化概念和精确定义之间的联系。
  - 算法的直觉概念 = 图灵机算法

END

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$\{w|w=u\#u, u\in\{0,1\}^*\}$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

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