第五章 图灵机

A.Turing在1936年介绍了这样一个通用的计算模型，该模型具有以下两个性质

- 该模型的每个过程都是有穷可描述的；
- 过程必须是由离散的、可以机械执行的步骤组成。

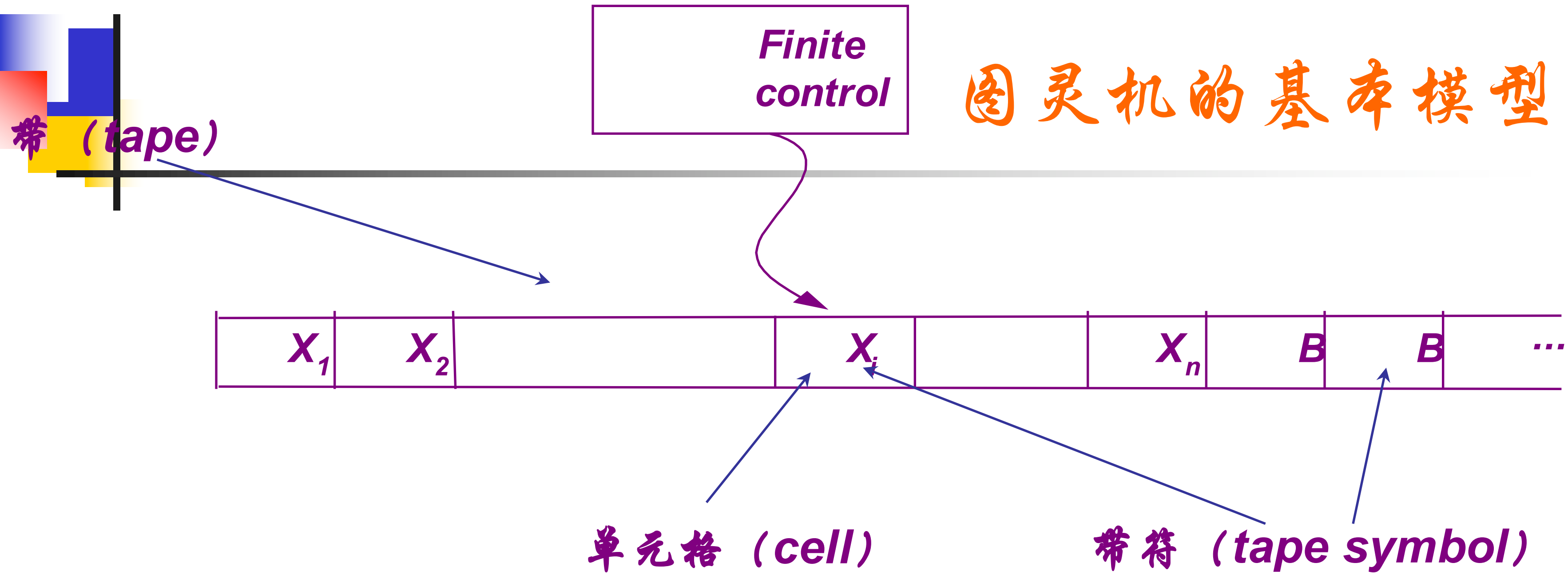
图灵机是计算机的一种简单数字模型，尽管简单，但它具有模拟通用计算机的计算能力。

- 通过研究TM来研究递归可枚举集和部分递归函数
- 为算法和可计算性研究提供了形式化描述工具。
- https://v.youku.com/v_show/id_XNjcwMjczNjky.html

主要内容

- TM的基本定义
- TM的格局
- TM接受的语言
- TM的构造技术
- TM的变形；
  - 重点: TM的定义、TM的构造。
  - 难点: TM的构造。

图灵机的基本模型



- 读写头在每一时刻扫描带上的一个单元
- 带有一个最左单元，向右则是无限的。
- 带的每个单元可容纳一个带符号

开始时，最左边n个单元装着输入（n>=0，n为有限数），它是一个字符串，符号都选自“带符号”的一个子集，即所谓的“输入符号集合”。余下的有穷个单元都存放空白符，它是一个特殊的带符号，但不是输入符号。

图灵机的工作机制

在一个图灵机的动作中，图灵机根据带头（读写头）所扫描的符号和有限控制器的状态可能作

- 改变状态
- 在被扫描的带单元上重新写一个符号，以代替原来写在该单元上的符号.
- 将带头向左或者右移一个单元。

＊ 图灵机和双向有限自动机的区别：图灵机能改变它带上的符号。

图灵机的形式化描述

✧ **形式定义** 一个图灵机 *TM (Turing machine)* 是一个七元组

$$M = (Q, T, \Sigma, \delta, q_0, B, F).$$

有限状态集

有限输入符号集

有限带符号集

转移函数

开始状态

特殊带符:空白符

终态集合

$q_0 \in Q$

$T \subseteq \Sigma$

$B \in \Sigma - T$

$F \subseteq Q$

转移函数 $\delta: Q \times \Sigma \rightarrow Q \times \Sigma \times \{L, R\}$

图灵机的函数与格局

- δ函数示例： $Q\times\Sigma \to Q\times\Sigma\times\{L,R\}$

  $\delta(q,a_i)=(p,B,L)$　$q, p \in Q$

  $\delta(q,a_i)=(p,b,R)$　$a_i\in\Sigma$　$b\in\Sigma$

- 格局

  用$w_1q\ w_2$描述图灵机的瞬间工作状态

  q为M的当前状态，　$w_1w_2\in \Sigma^*$

  w_1w_2是当前时刻从开始端（因为可写）到右边空白符号为止的内容

  当读写头已达到带的右端，则w_1w_2为读写头以左的内容。

约定：$w_1q\ w_2$表示读写头正扫描w_2的最左字符

$w_2=\varepsilon$ 则表示读写头正扫描一个空白字符。

图灵机的格局

✧ 给定图灵机 $M = (Q, T, \Sigma, \delta, q_0, B, F)$，定义格局之间的推导关系 $\vdash_M$ 如下：

1. 设 $\delta(q, X_i) = (p, Y, L)$，则有

   $X_1X_2\ldots X_{i-1}qX_iX_{i+1}\ldots X_n \vdash_M X_1X_2\ldots X_{i-2}pX_{i-1}Y\ldots X_n$，

   但有如下两个例外：

   （1）$i=1$ 时，$qX_1X_2\ldots X_n \vdash_M pYX_2\ldots X_n$，和

   （2）$i=n$ 及 $Y=B$ 时，$X_1X_2\ldots X_{n-1}qX_n \vdash_M X_1X_2\ldots X_{n-2}pX_{n-1}$ B.

2. 设 $\delta(q, X_i) = (p, Y, R)$，则有

   $X_1X_2\ldots X_{i-1}\,q\,X_iX_{i+1}\ldots X_n \vdash_M X_1X_2\ldots X_{i-1}Y\,p\,X_{i+1}\ldots X_n$，

   但有如下两个例外：

   （1）$i=n$ 时，$X_1X_2\ldots X_{n-1}q\,X_n \vdash_M X_1X_2\ldots X_{n-1}Y\,p\,B$，和

   （2）$i=1$ 及 $Y=B$ 时，$q\,X_1X_2\ldots X_n \vdash_M B\,p\,X_2\ldots X_{n-1}X_n$.

图灵机接受的语言

$L(M)=\{\omega \mid \omega\in T^*且q_0\omega \vdash^* \alpha_1 p \alpha_2, p\in F, \alpha_1\alpha_2\in \Sigma^*\}$

图灵机接受的语言是输入字母表中这样一些字符串的集合，初始时，这些字符串放在M的带上，M处于状态q_0，且M的带头处在最左单元上，这些字符串将使M进入某个终止状态。

假定：

当输入被接受时，图灵机将停止，没有下一个动作。

图灵机的停机问题

任给图灵机 $M = (Q, T, \Sigma, \delta, q_0, B, F)$，以及输入字符串 $w \in T^*$. 试问：对于w，M 是否停机？停机是指图灵机不存在下一个移动步（*move*）.

✧ 结论 图灵机的停机问题是不可解的（即不可判定的）.

✧ 结论 任给图灵机 M，很容易构造一个图灵机 M'，使得 $L(M)= L(M')$，并满足：如果$w \in L(M)$，则对于 w，M' 接受w 并一定停机.

如果没有特别指出，总是假定图灵机到达终态（接受态）后一定停机.

但是，对不能接受的字符串，图灵机可能永不停止.（只要M还在某个输入上运行，我们无法知道是因为运行的时间不够长而没有被接受，还是根本就不会停机）

图灵机举例

例1：设语言 $L=\{a^n b^n | n>=1\}$，设计图灵机接受L。

思路：最初带上为　a a … a b b… b B B B ……

n个a　n个b

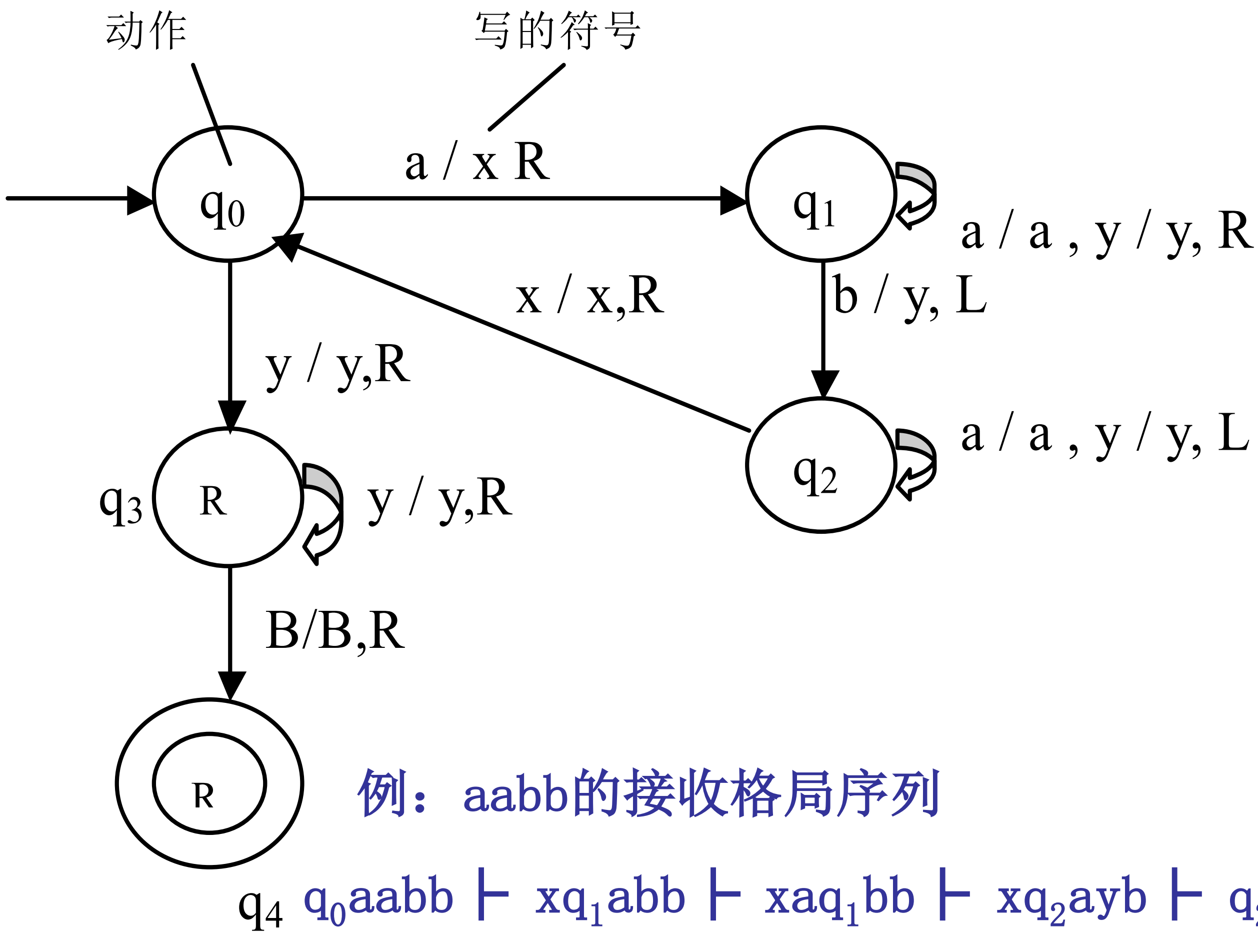
首先用x替换M最左边的a，再右移至最左边的b用y替换之，左移寻找最右的x，然后右移一单元到最左的a，重复循环。

如果

（1）当在搜寻b时，M找到了空白符B，则M停止，不接受该串。（此时，a的个数大于b的个数）

（2）当将b改为y后，左边再也找不到a，此时,若右边再无b，接受；若仍有b，则b的个数大于a的个数，不接受。

例1　L={$a^n b^n$ | n>=1}



$\delta(q_0, a) = (q_1, x, R)$

$\delta(q_0, y) = (q_3, y, R)$

$\delta(q_1, a) = (q_1, a, R)$

$\delta(q_1, y) = (q_1, y, R)$

$\delta(q_1, b) = (q_2, y, L)$

$\delta(q_2, a) = (q_2, a, L)$

$\delta(q_2, y) = (q_2, y, L)$

$\delta(q_2, x) = (q_0, x, R)$

$\delta(q_3, y) = (q_3, y, R)$

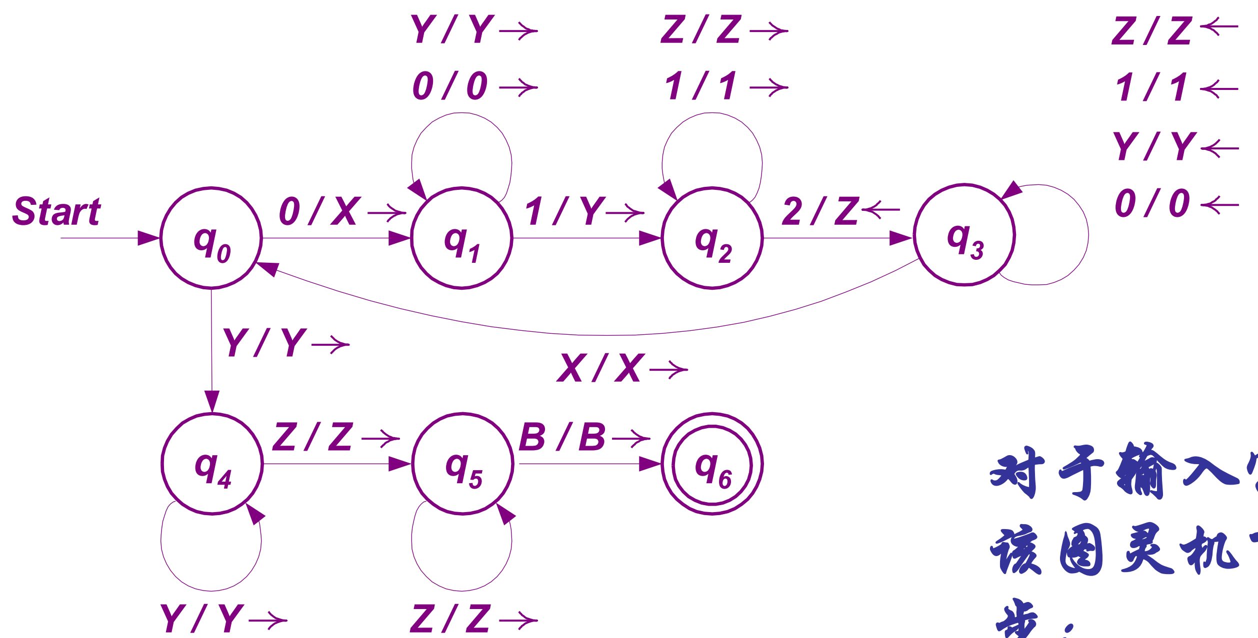
$\delta(q_3, B) = (q_4, B, R)$

例：aabb的接收格局序列

$q_0aabb \vdash xq_1abb \vdash xaq_1bb \vdash xq_2ayb \vdash q_2xayb \vdash xq_0ayb \vdash xxq_1yb$

$\vdash xxyq_1b \vdash xxq_2yy \vdash xq_2xyy \vdash xxq_0yy \vdash xxyq_3y \vdash xxyyq_3B \vdash xxyyBq_4$

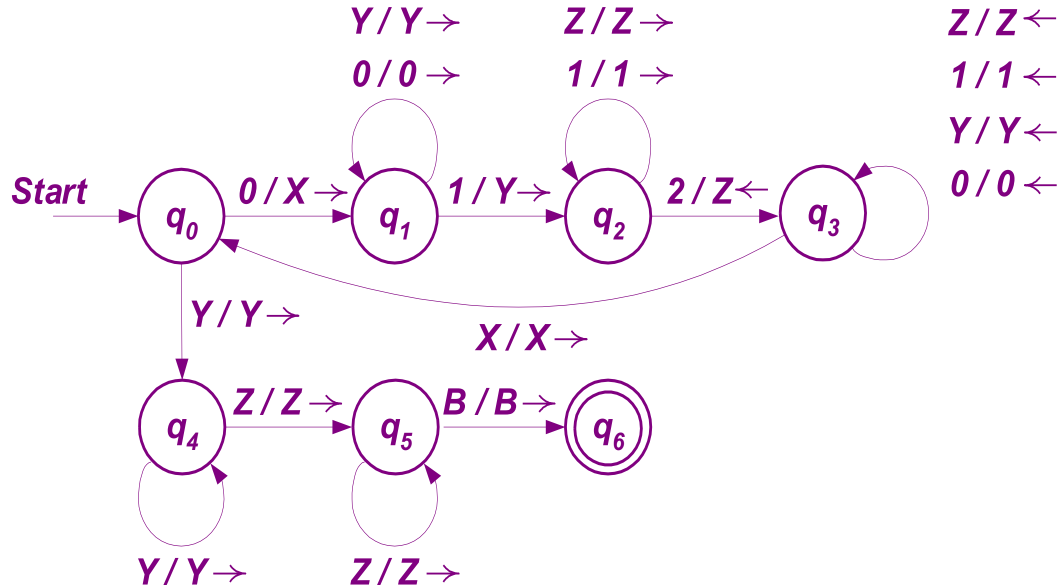
例2 $L = \{ 0^n1^n2^n \mid n \geq 1\}$.



对于输入字符串001122，该图灵机可以有如下推导步：

$$q_0001122 \vdash_M Xq_101122 \vdash_M X0q_11122 \vdash_M X0Yq_2122 \vdash_M X0Y1q_222$$
$$\vdash_M X0Yq_31Z2 \vdash^*_M q_3X0Y1Z2 \vdash_M Xq_00Y1Z2 \vdash^*_M XXYYZq_22$$
$$\vdash_M XXYYq_3ZZ \vdash^*_M Xq_3XYYZZ \vdash_M XXq_0YYZZ \vdash^*_M XXYYq_4ZZ$$
$$\vdash_M XXYYZq_5Z \vdash_M XXYYZZq_5B \vdash_M XXYYZZBq_6B$$

转移图与转移表



State	0	1	2	Symbol X	Y	Z	B
q_0	(q_1, X, R)	—	—	—	(q_4, Y, R)	—	—
q_1	$(q_1, 0, R)$	(q_2, Y, R)	—	—	(q_1, Y, R)	—	—
q_2	—	$(q_2, 1, R)$	(q_3, Z, L)	—	—	(q_2, Z, R)	—
q_3	$(q_3, 0, L)$	$(q_3, 1, L)$	—	(q_0, X, R)	(q_3, Y, L)	(q_3, Z, L)	—
q_4	—	—	—	—	(q_4, Y, R)	(q_5, Z, R)	—
q_5	—	—	—	—	—	(q_5, Z, R)	(q_6, B, R)
q_6	—	—	—	—	—	—	—

作为整数函数计算机的图灵机

- **预备知识：**图灵机除了作为语言接受器外，还可看作整数到整数的函数计算机。
- **传统方法把整数表示成一进制**

  整数 $i \geq 0$ 用字符串 0^i 表示
- 如果一个函数有k个自变量，$i_1, i_2, \ldots, i_k$，那么这些整数开始时被放在带上，并用1把他们分隔开。

  形如 $0^{i_1}\ 1\ 0^{i_2}\ 1\ 0^{i_3} \ldots\ 1\ 0^{i_k}$
- 如果图灵机停止(不论是否在一个接受状态上)且带上为 0^m，则 $f(i_1,i_2,\ldots,i_k)=m$　f是被图灵机计算的k元函数
- 如果f（$i_1,i_2\ldots,i_k$）对所有$i_1,i_2\ldots,i_k$有定义，那么称f是一个全递归函数。全递归函数对应于递归语言，因为它总是被能停下来的图灵机所计算。
- 所有常用的整数算术函数都是全递归函数。

例3：设计图灵机求真减法

$$m \quad n = \begin{cases} m \quad n & m > n \\ 0 & m \quad n \end{cases}$$

- 初始带 $0^m 1 0^n$

- 思路：
  1. 用空白符B代替带上的最左端的0
  2. 右移至紧跟1后的0,将其改为1
  3. 左移找到B，将B之后的0改为B
  4. 重复上述过程

如果（1）右移找0时，遇到B，意味着m>n

BB…B $0^{m-(n+1)}$ 1 11…1

n+1 n个

将后面n+1个1变为B，将左侧最后一个B变0，形如

BB…B 0 $0^{m-(n+1)}$ BB…B

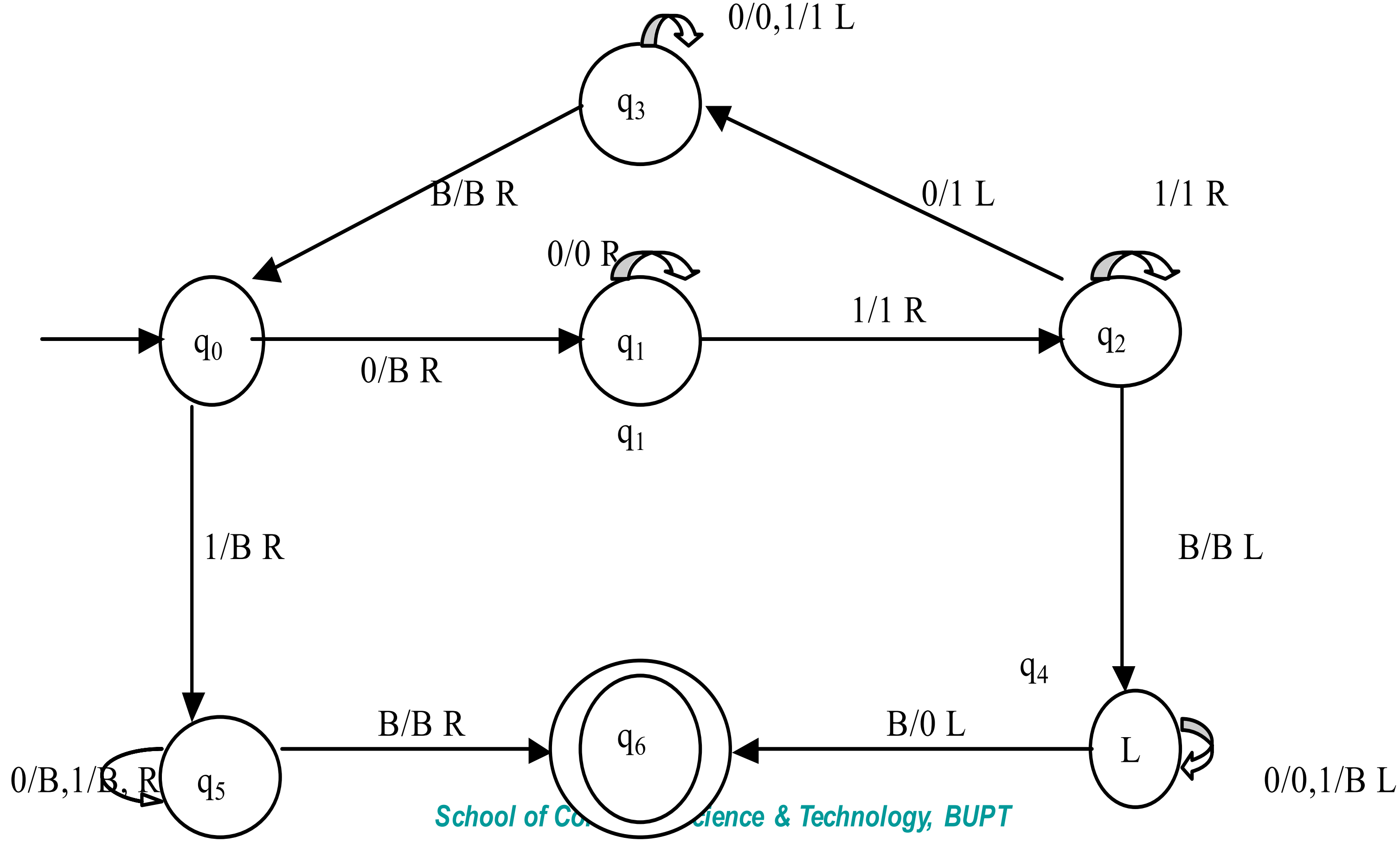
n个 n+1个 这时，带上留下m-n个0，即结果为m-n

求真减法（续）

(2) M左移找不到0，意味着 n ≥ m，形如

BB…B 1 11…1 0…0

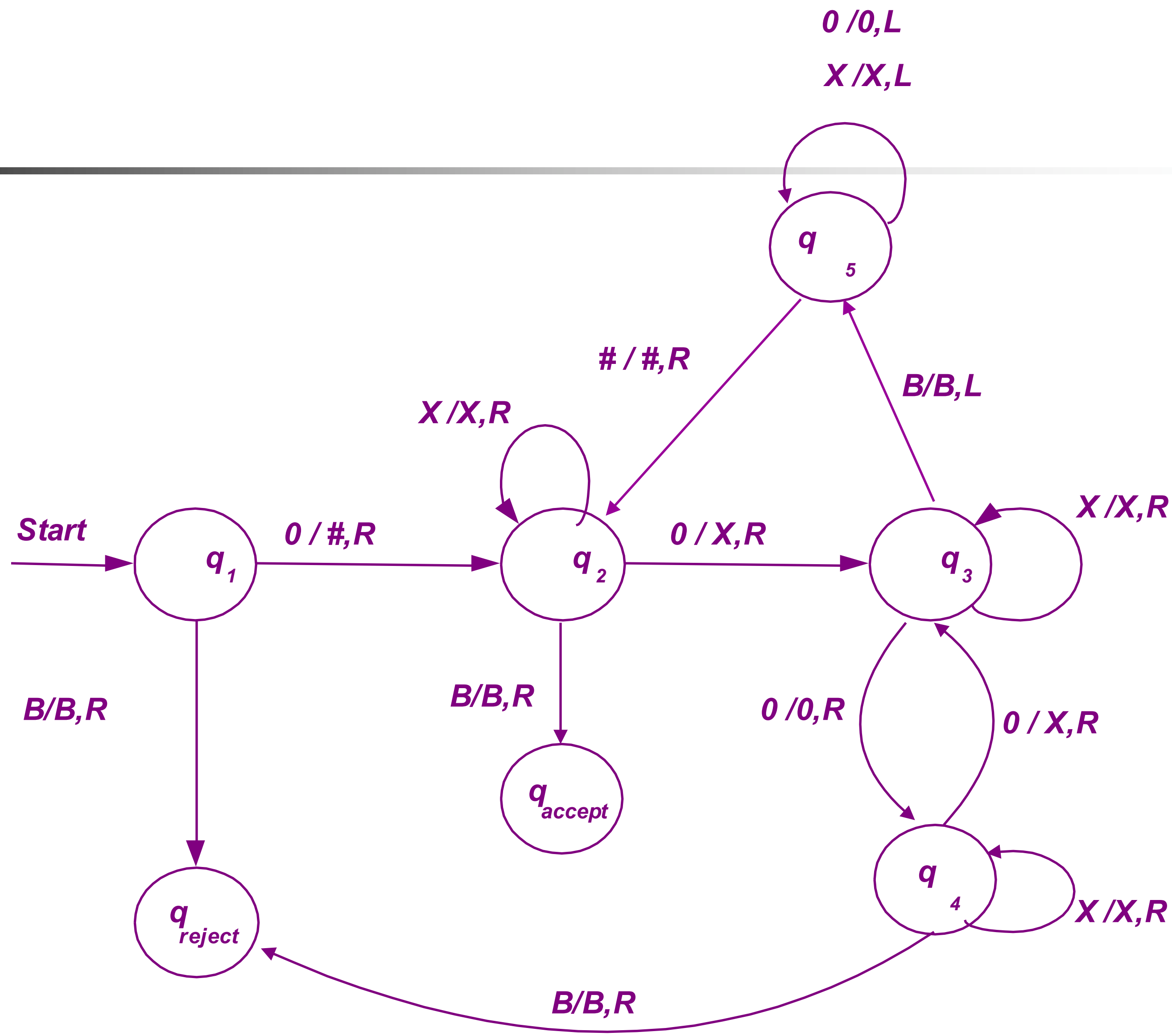
m个 m个 n-m个 此时，用B替换所有剩余的1 和0



例4：L={ 0^m | $m=2^n$, $n \geq 0$}

- **设计思路：对输入串w**

  1. 从左到右扫描带，隔一个消一个0；
  2. 若带上只剩唯一一个0,接受；
  3. 若带上不止一个0,且个数为奇数，拒绝;
  4. 让读写头返回带的最左端;
  5. 转第一步。



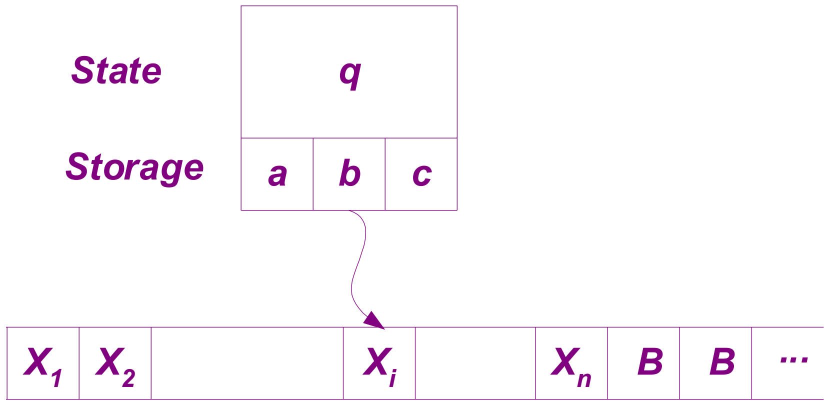
识别 $L=\{0^m \mid m=2^n, n \geq 0\}$的图灵机

5.2 图灵机的构造技术

在设计图灵机的过程中，写出δ函数很麻烦，为了构造复杂的图灵机，需探讨图灵机的若干构造技术，并引入一些新的概念和工具。

目的：设计时方便，但这些构造技术并未增加图灵机的功能。

5.2.1. 利用带存储区的状态（storage in the state）



此类图灵机 $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ 中，状态中可以包含一个具有有限个取值的存储单元，即状态集合为

$Q = S \times T = \{ [q,a] \mid q \in S \wedge a \in T \}$,

其中 $q \in S$ 通常表示控制状态，而 $a \in T$ 通常表示数据元素.

一般情况下，有限控制器内允许存储n个字符，即状态的第2个元素可存储n个字符。

例：设计一个图灵机，读写头将扫视第一个字符存入有限控制器内，然后扫视整个带，若找不到与第一个相同的字符，则M接受该字符串，否则不接受。

构造$M=(Q,\{a,b\},\{a,b,B\},\delta,q0,B,F)$

其中$Q=\{q0,q1\}\times\{a,b,B\}=\{[q0,a],[q0,b],[q0,B][q1,a][q1,b][q1,B]\}$

初态$[q0,B]$

终态$F=\{[q1,B]\}$

δ函数：$\delta([q0,B],a)=([q1,a],a,R)$

$\delta([q0,B],b)=([q1,b],b,R)$ 存第一个字符

$\delta([q1,a],b)=([q1,a],b,R)$

$\delta([q1,b],a)=([q1,b],a,R)$ 后面符号与第一个不等，继续右移

$\delta([q1,a],B)=([q1,B],B,L)$

$\delta([q1,b],B)=([q1,B],B,L)$ 进入终态$[q1,B]$

$\delta([q1,a],a)=\Phi$

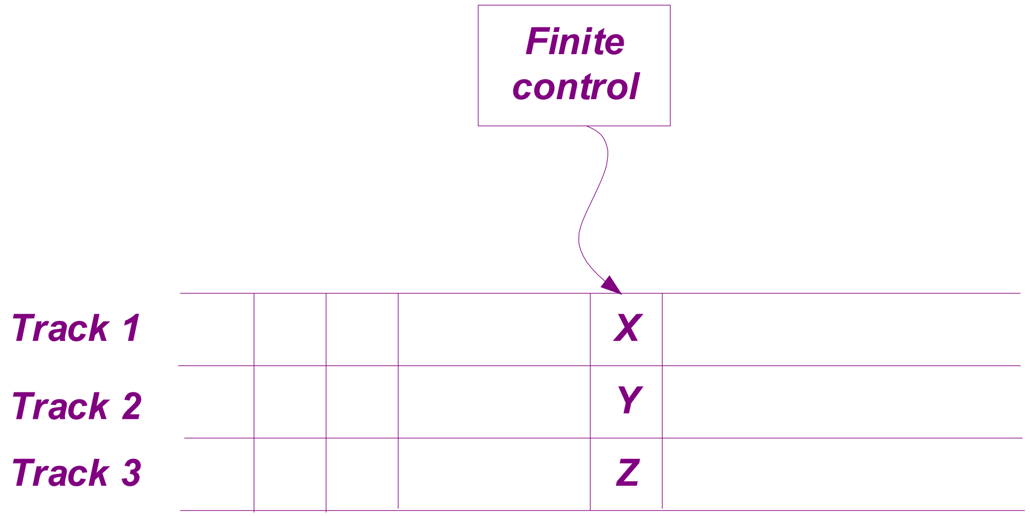
$\delta([q1,b],b)=\Phi$ 遇到相同符号，δ无定义

M停机，不接受

5.2.2 多道（multiple tracks）图灵机

把图灵机的输入带分成两层或多层，这样，原来的每一单元变成了上下两个或多个单元。

对含有n层的输入带来说，读写头一次可同时读出并改写n个单元的字符，这样的图灵机称为n道机。



例：多道图灵机

例：用三道机，检查某个数n（n>2）是否为素数。（书p196）

思路：将被检查的数n以二进制形式写在输入带的第一道上，数的两端分别用￥和Φ定界

检查方法：

在第二道上写下一个二进制数2

把第一道上的数复制到第三道上

将第三道上的数反复减去第二道上的数，余数留在第三道上

若余数为0，被检查的数不是素数

若余数不为0，将第二道数加1，将第一道数复制到第三道，重复上述1，2，3，4过程

当一，二道数相等时，该数是素数。

5.2.3 核对符

当用图灵机识别语言时，如果语言中存在有重复性或可逆性等类型的句子时，为了判定某个字符串是否属于语句中的句子， 可以使用一个核对符，以此增加图灵机的灵活性。

考虑用一个双道机，在第二道上使用核对符“√”，在第一道上放要被检查的字符串,当字符串中某个字符一旦被核对以后,可以在第二道上对应位置写上核对符“√”。

5.2.4 移位

可以让图灵机具备移位的功能，即对输入带上的字符进行移位操作。当需要在输入带上留出一部分空间时，可将输入带上的非空白符右移若干单元。

假设需要输入带上的非空白字符右移n个单元，则可让控制器状态的第二个元素具有存储n单元的功能（n是有限数）

例： 构造图灵机M，要求它将带上非空白符向移动两个单元

原带为 a b c d B， 移后为 z z a b c d B

设$M=(Q, T, \sum, \delta, q_0, B, F)$； $Q=\{[q, D_1, D_2] \mid q=q_0,q_1,$且$D_1,D_2,\ldots D_n \in \sum\}$

初始：$[q_0, B, B]$，终态$[q_1, B, B]$

δ定义：

$\delta([q_0, B, B], D_1)=([q_0, B, D_1], Z, R)$

$\delta([q_0, B, D_1], D_2)=([q_0, D_1, D_2], Z, R)$

$\delta([q_0, D_1, D_2], D3)=([q_0, D_2, D_3], D_1, R)$

$\delta([q_0, D_{n-1}, D_n], B)=([q_0, D_n, B], D_{n-1}, R)$

$\delta([q_0, D_n, B], B)=([q_1, B, B], Dn, L)$

对$D \in \sum-\{ B, Z\}$:

$\delta([q_1, B, B],D)=([q_1, B, B], D, L)$ 回到输入点

作业：

第五章第二题 第三题（1）

复习：

第二章

语言，运算，文法

第三章

有限自动机、三种类型，极小化，相互关系

右线性文法，正则文法，特性

文法与自动机的关系

有输出的自动机

第四章

推导树，二义性

上下文无关文法的变换

C范式/G范式

下推自动机

文法与自动机的关系

文法特性

第五章

图灵机