图灵机的计算与扩展

Turing 机

- 图灵机的计算
- 图灵机的编程技术
- 图灵理论
- 图灵机带的扩展
- 图灵机移动的扩展
- 受限图灵机
- 图灵机与其它自动机
- 图灵机与计算机

Turing 机

- 图灵机的计算
- 图灵机的编程技术
- 图灵理论
- 图灵机带的扩展
- 图灵机移动的扩展
- 受限图灵机
- 图灵机与其它自动机
- 图灵机与计算机

整数表示

整数5的表示:

十进制:	5
二进制:	101
单位制:	00000

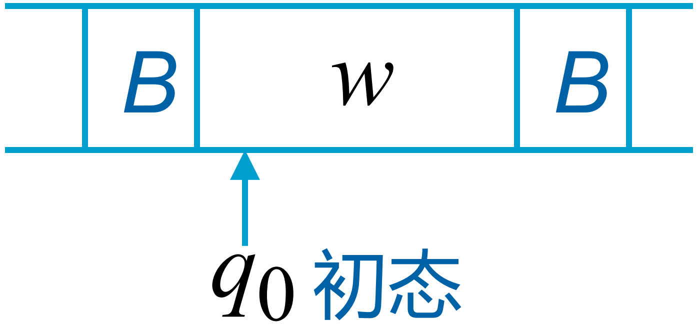
选用单位制表示: 易于图灵机的操作

函数计算

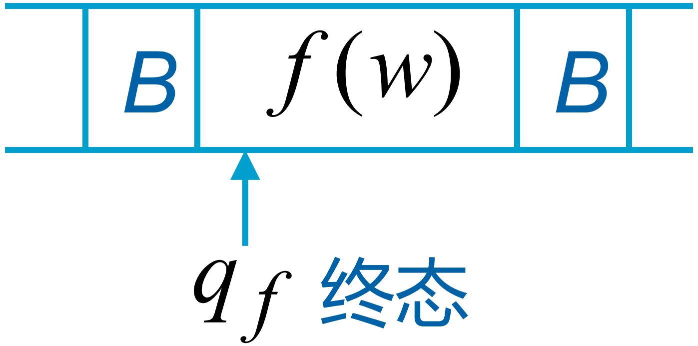
定义:

函数 f 是可计算的，如果存在图灵机M，对任意 $w \in D$, 满足

开始结构



最终结构



函数计算

换言之:

函数 f 是可计算的，如果存在图灵机M，对任意 $w \in D$，满足

$$f: \quad q_0 w \vdash^* q_f f(w)$$

初始ID　　最终ID

函数计算

例：

函数 $f(x,y)=x+y$ 是可计算的

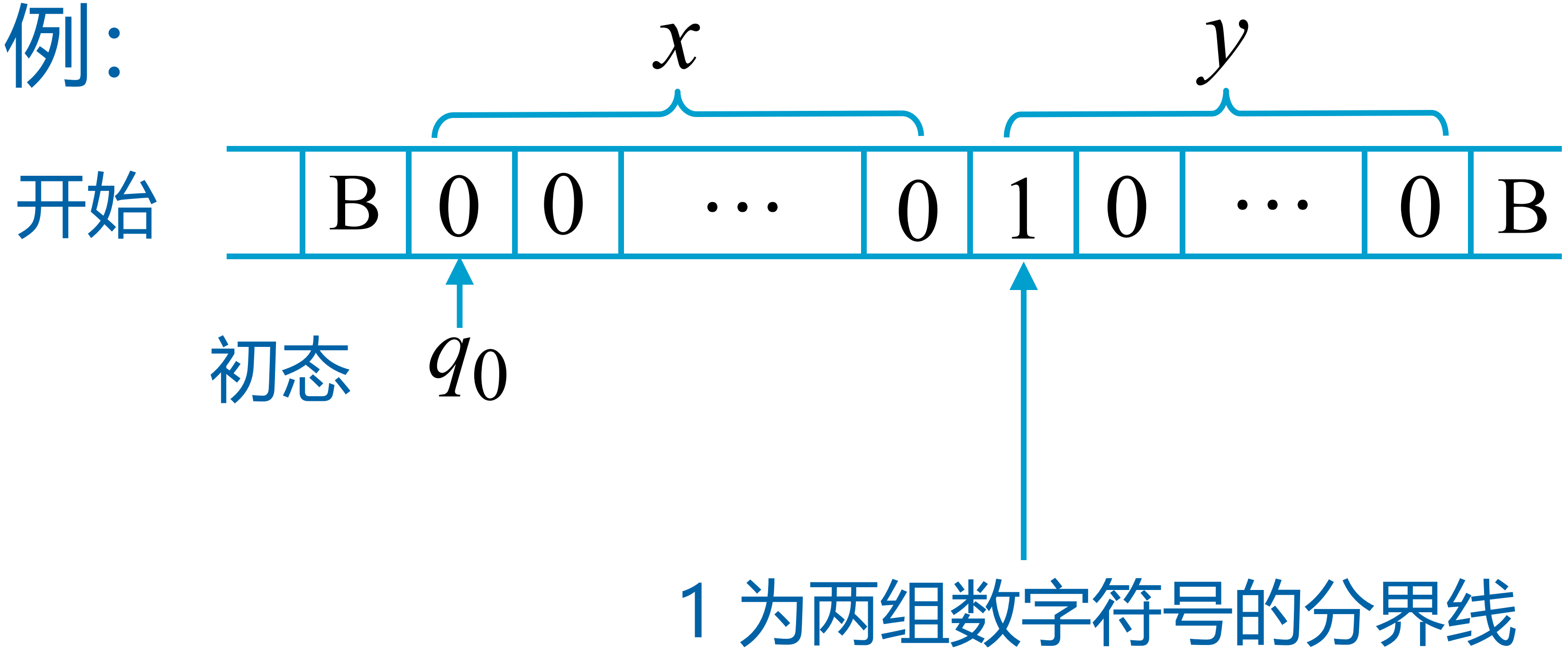
x,y 为整数

f 的图灵机M：

输入串: $x1y$ （单位制）

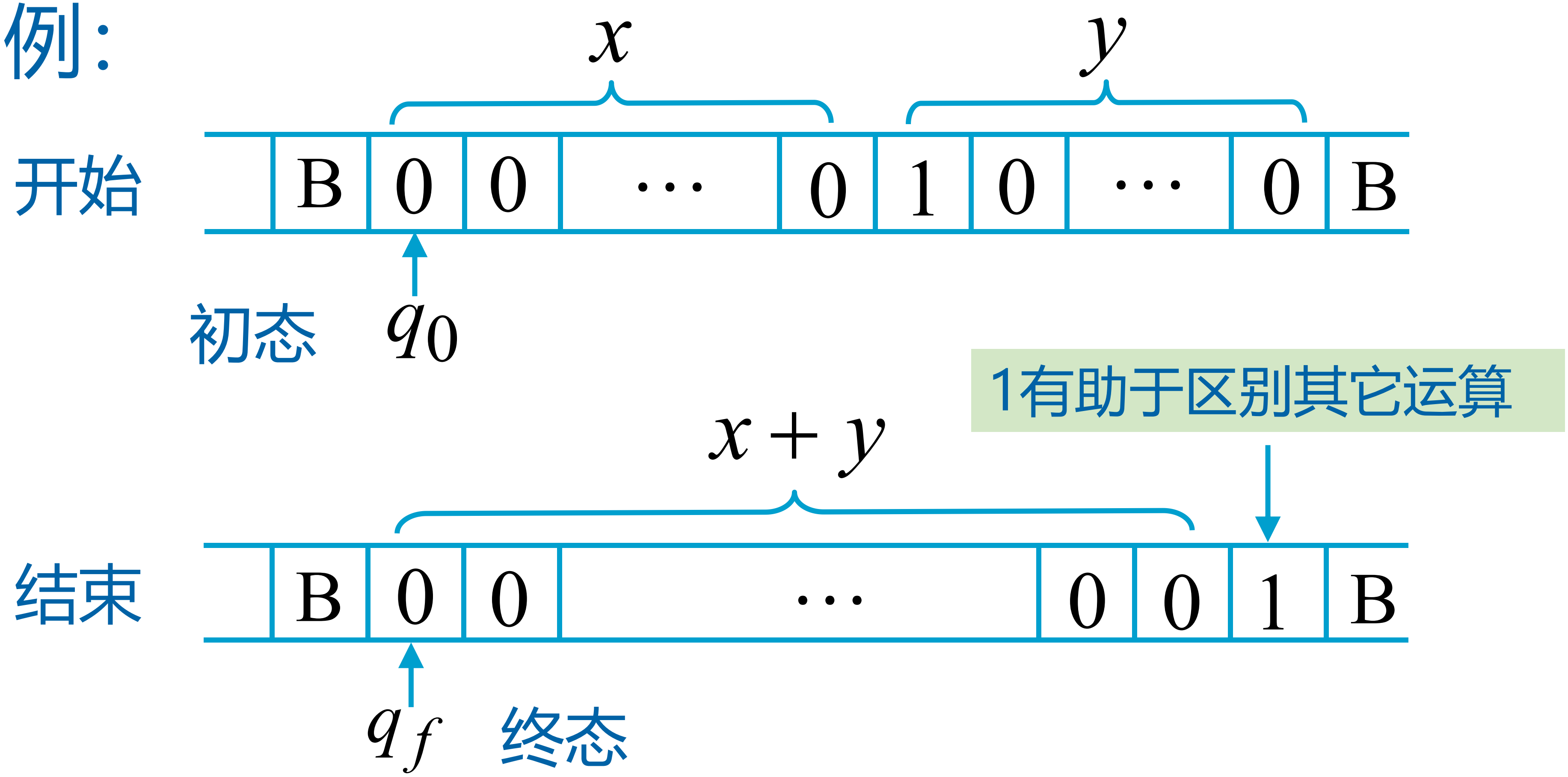
输出串: $xy1$ （单位制）

例:



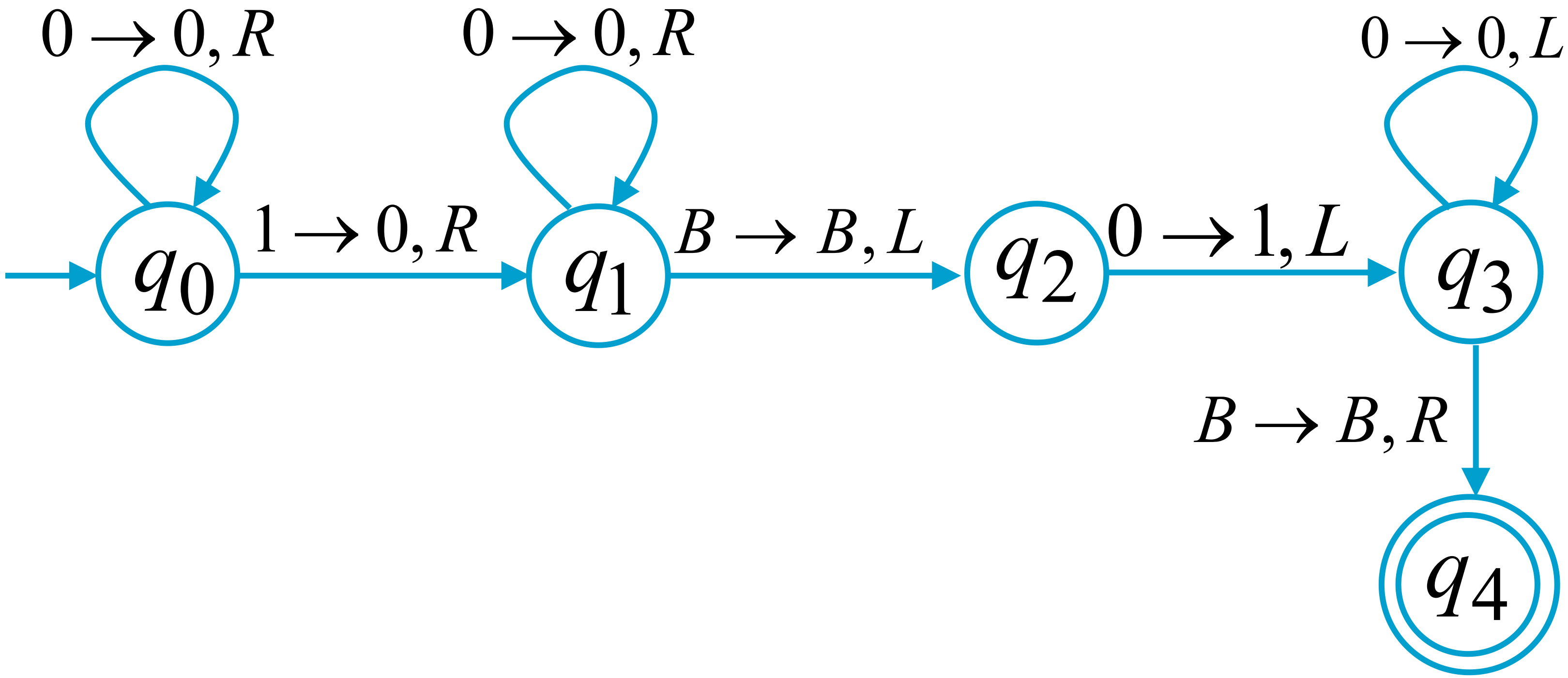
函数计算

例:



函数计算的图灵机

例： $f(x,y)=x+y$ 的图灵机



函数计算

例:

函数 $f(x)=2x$ 是可计算的

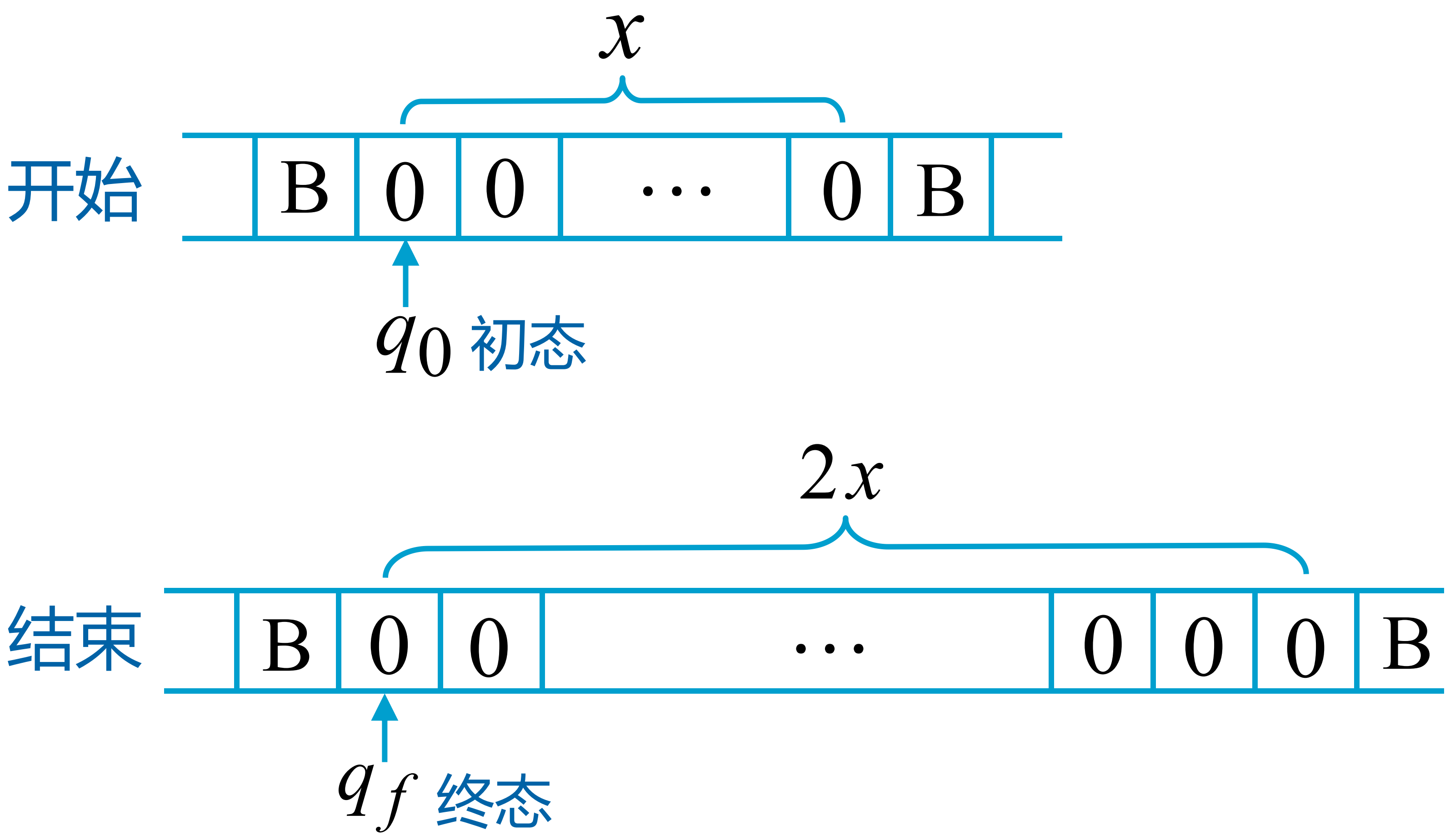
x 为整数

图灵机:

输入串: x (单位制)

输出串: xx (单位制)

函数计算



图灵机拟码

$f(x)=2x$ 图灵机拟编码

- 用$代替每一个0
- 重复:

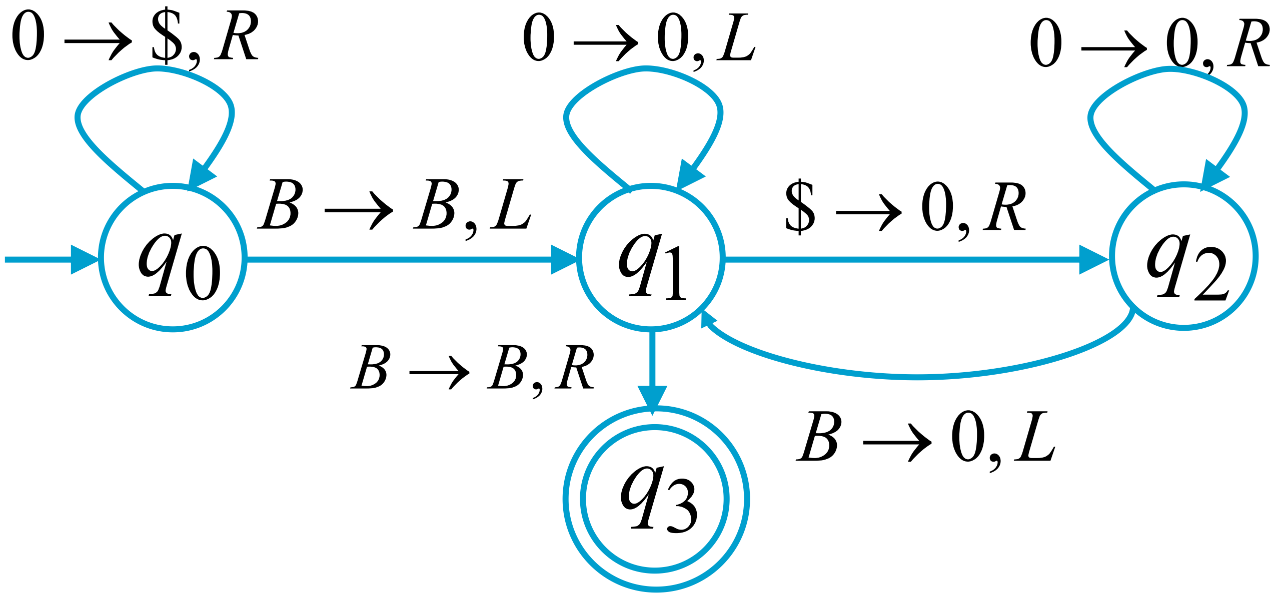
  找到字符串最右$, 替换为0

  移动到带的最右端，插入一个 0

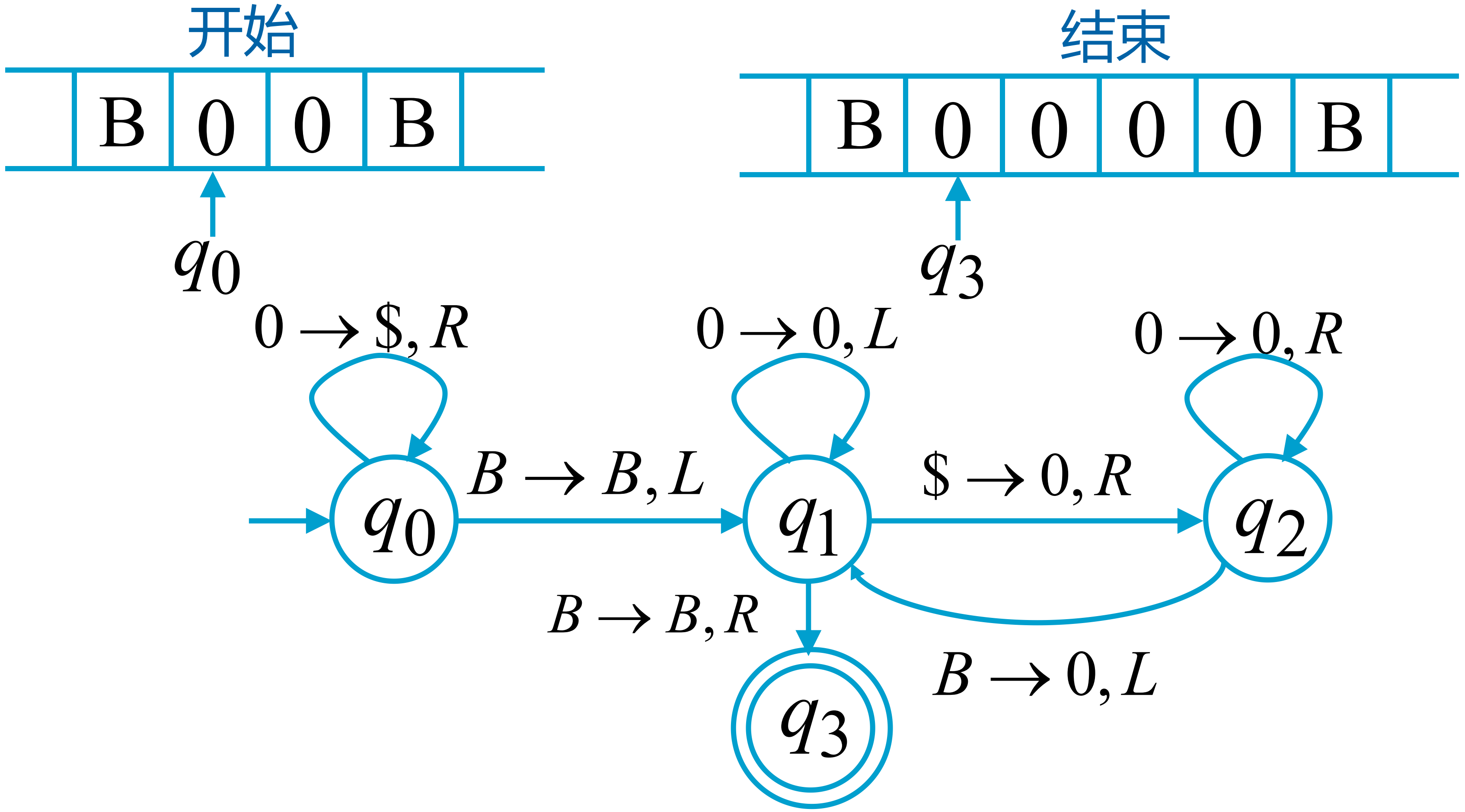
直到替换完所有的符号$

函数计算的图灵机

$f(x) = 2x$ 的图灵机为:



函数计算的图灵机



开始
B
0
0
B
q_0
结束
B
0
0
0
0
B
q_3
$0 \to \$, R$
$0 \to 0, L$
$0 \to 0, R$
$B \to B, L$
$\$ \to 0, R$
q_0
q_1
q_2
$B \to B, R$
q_3
$B \to 0, L$

函数计算

例:

函数 $$f(x,y)=\begin{cases}1 & \text{若}\quad x>y\\ 0 & \text{若}\quad x\le y\end{cases}$$

是可计算的.

函数的图灵机:

输入串: $x1y$

输出符: 1 或 0

图灵机拟码

- 重复

  匹配 x 中 0 与 y 中的 0

  直到 x 与 y 匹配完毕.

- 若 x 仍有剩余的0，

  擦去带中符号, 写 1 $(x > y)$

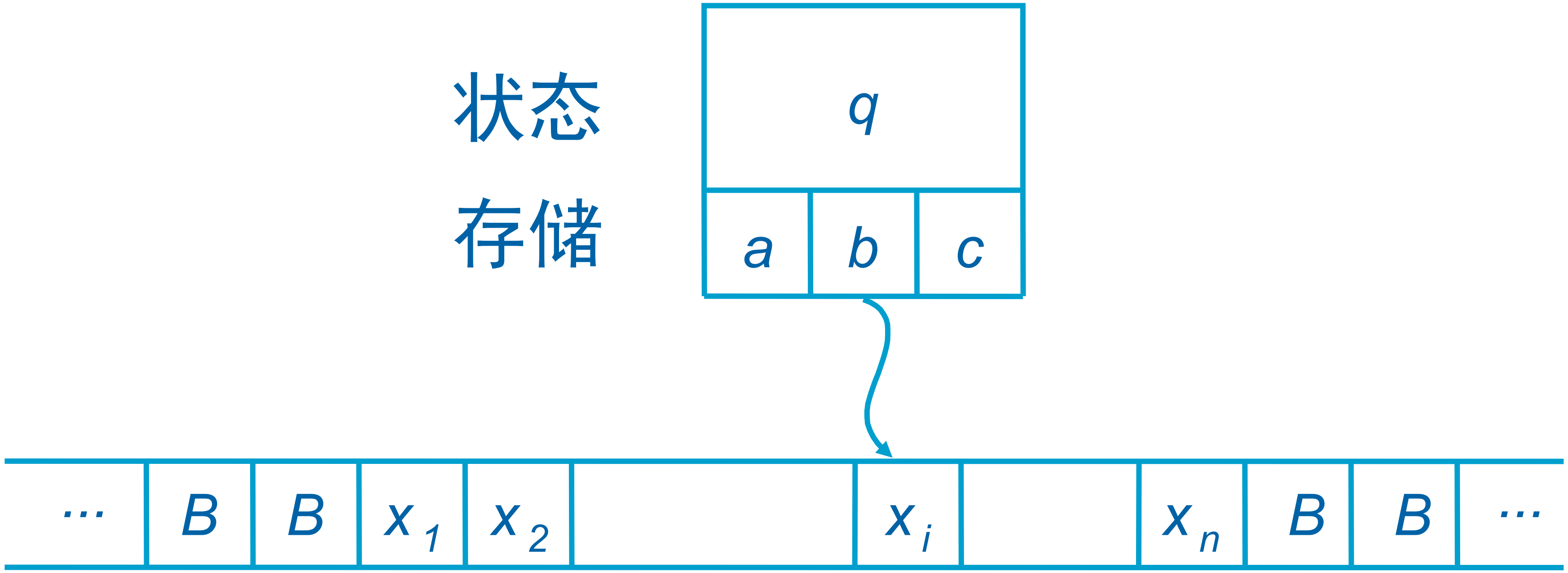
  除此之外，擦去带中符号, 写 0 $(x \le y)$

Turing 机

- 图灵机的计算
- 图灵机的编程技术
- 图灵理论
- 图灵机带的扩展
- 图灵机移动的扩展
- 受限图灵机
- 图灵机与其它自动机
- 图灵机与计算机

图灵机的编程技巧

带存储区的状态 (*storage in the state*)



图灵机的编程技巧

图灵机

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

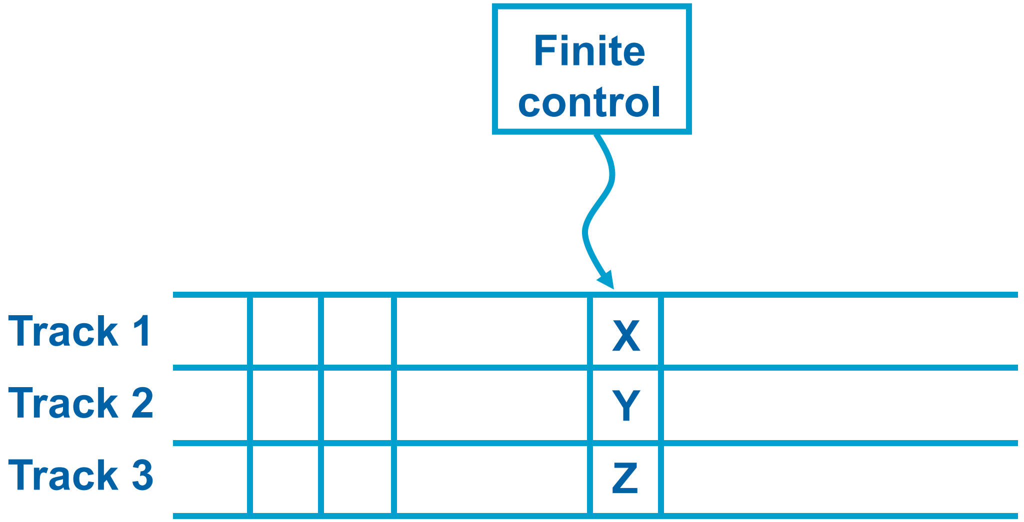
状态中可以包含一个具有有限个取值的存储单元，即状态集合为:

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

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

图灵机的编程技巧

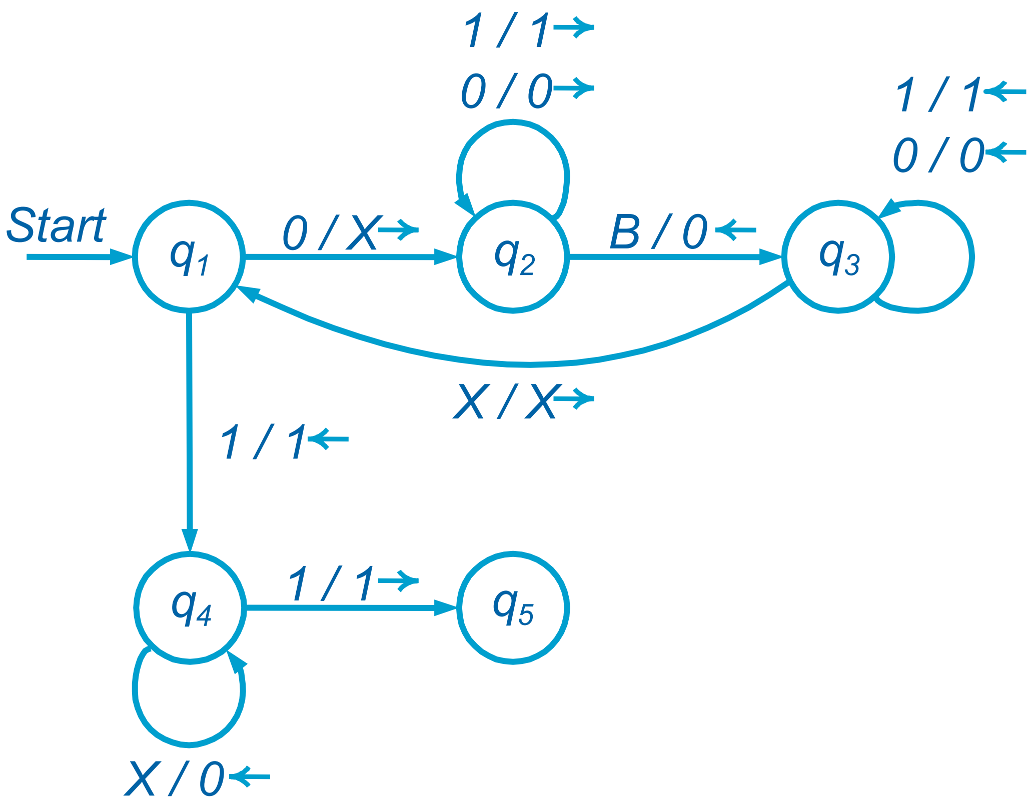
多道 (*multiple tracks*) 图灵机



此类图灵机 $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ 中，带符号是向量的形式。如上图中的图灵机，带符号的形式为三元组。

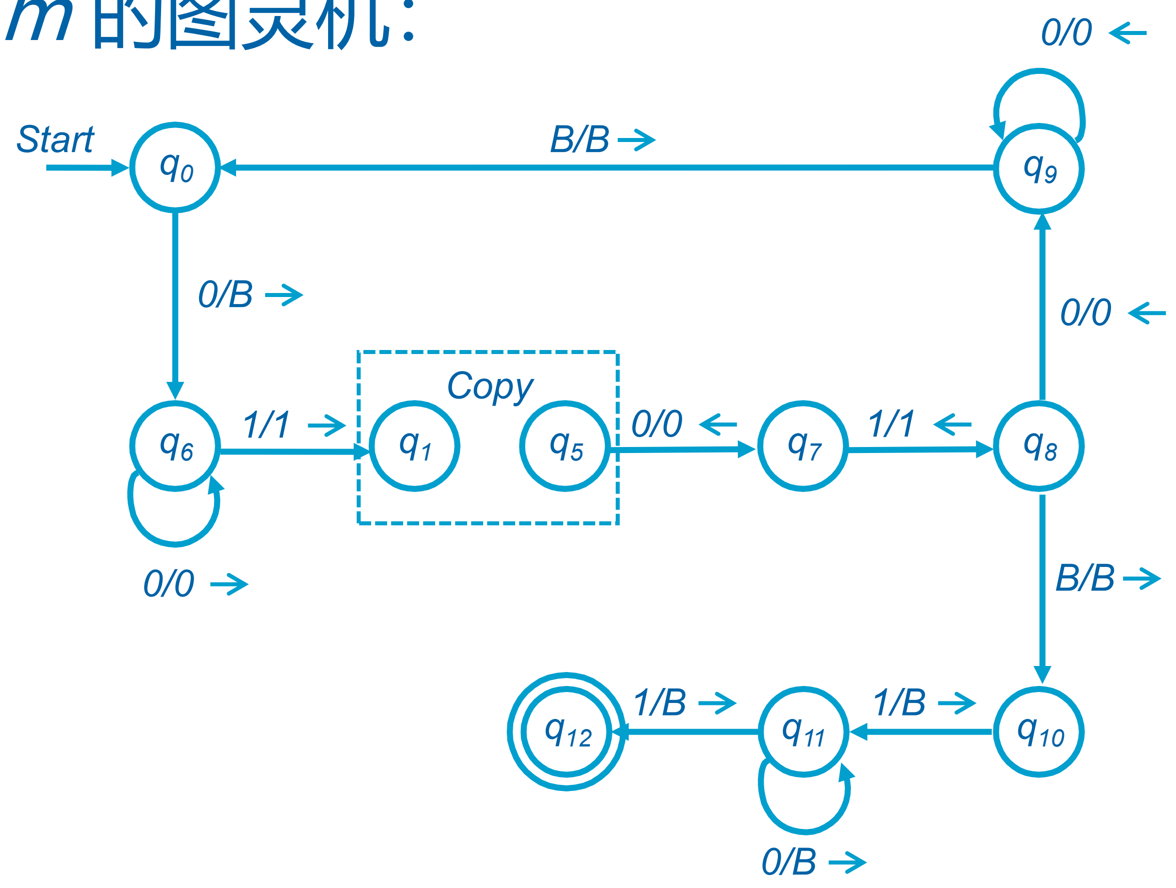
子例程的设计

复制数 0^n 的子程序（subroutines）：

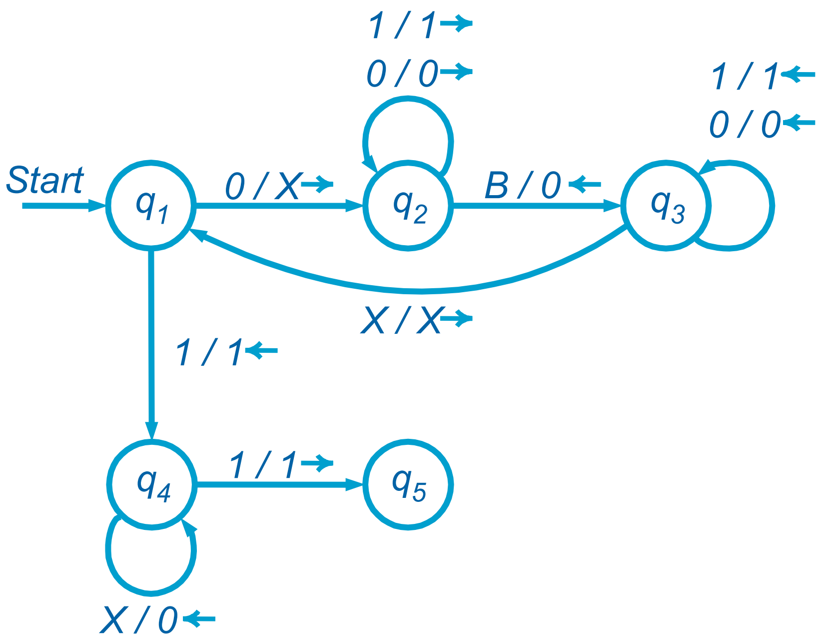


子例程的设计

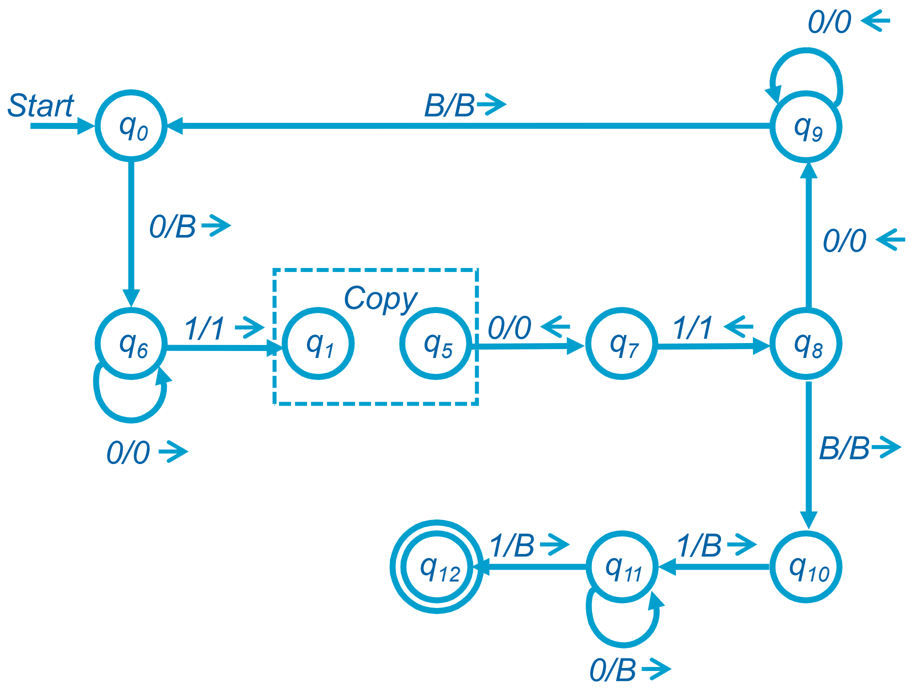
计算乘法 $n \times m$ 的图灵机：



子例程的设计



左图的图灵机表示子例程 *copy*， 下图的图灵机表示可以调用*copy* 的主程序。



完成两个正整数的乘法. 初始时，带上的符号串形如 0^m10^n1，而结束时，带上的符号串变为0^{mn}。

Turing 机

- 图灵机的计算
- 图灵机的编程技术
- 图灵理论
- 图灵机带的扩展
- 图灵机移动的扩展
- 受限图灵机
- 图灵机与其它自动机
- 图灵机与计算机

Turing理论

- Turing观点:

  能机械计算的一定能用图灵机实现 (1930’)

- 计算机科学理论:

  能机械计算是否一定能由图灵机实现?

  到此为止，没有比图灵机更强的计算模型.

- 算法的定义：

  函数$f(w)$的算法，定义为计算$f(w)$的图灵机。

冯 · 诺依曼

- 约翰 · 冯·诺依曼 （1903 - 1957）
  出生于匈牙利的美籍匈牙利数学家、计算机科学家、物理学家，是20世纪最重要的数学家之一，是现代计算机、博弈论、核武器和生化武器等领域内的科学全才之一，被后人称为“**现代计算机之父**”、“**博弈论之父**”。

- 冯·诺伊曼从小就以过人的智力与记忆力而闻名。冯·诺伊曼一生中发表了大约150篇论文，其中有60篇纯数学论文，20篇物理学以及60篇应用数学论文。他最后的作品是一个在医院未完成的手稿，后来以书名《计算机与人脑》发布，表现了他生命最后时光的兴趣方向。他先后任职于美国普林斯顿大学、美国普林斯顿高等研究院等机构。

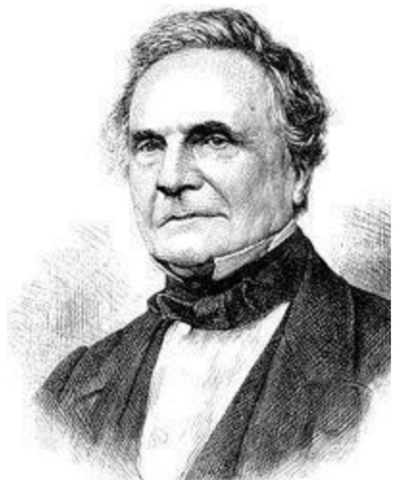
Alan Turing

曾担任过冯·诺依曼助手的美国学者弗兰克尔这样写道：“许多人都推举冯·诺依曼为‘计算机之父’，然而我确信他本人从来不会促成这个错误。或许，他可以被恰当地称为助产士，但是他曾向我，并且我肯定他也曾向别人坚决强调：如果不考虑巴贝奇、阿达和其他人早先提出的有关概念，计算机的基本概念属于图灵。”

正是冯·诺依曼本人亲手把“计算机之父”的桂冠转戴在图灵头上。

查尔斯 · 巴贝奇

- 查尔斯 · 巴贝奇 (1792—1871)
  科学管理的先驱者。巴贝奇出生于一个富有的银行家的家庭，曾就读于剑桥大学三一学院。巴贝奇在24岁时就被选为英国皇家学会会员。他参与创建了英国天文学会和统计学会，并且是天文学会金质奖章获得者。他还是巴黎伦理科学院、爱尔兰皇家学会和美国科学学院的成员。



- 由于提出了差分机与分析机的设计概念（并有部份实做机器），被视为计算机先驱。

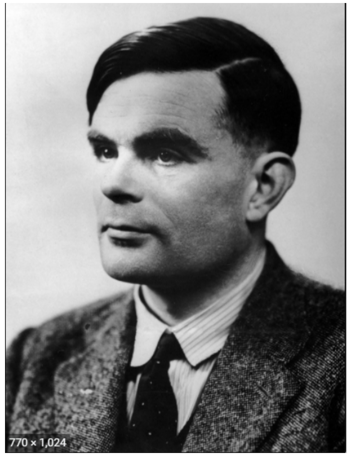
阿达 · 洛夫莱斯

- 阿达 · 洛芙莱斯（1815-1852）
  原名奥古斯塔·阿达·拜伦（Augusta Ada Byron），她的名字取自拜伦的异母的姊妹奥古斯塔·李，是著名英国诗人拜伦之女，数学家，计算机程序创始人，建立了循环和子程序概念。

- 其为计算程序拟定“算法”，写作的第一份“程序设计流程图”，被珍视为“第一位给计算机写程序的人”。为了纪念阿达·奥古斯塔对现代电脑与软件工程所产生的重大影响，美国国防部将耗费巨资、历时近20年研制成功的高级程序语言命名为Ada语言，它被公认为是第四代计算机语言的主要代表。

Alan Turing

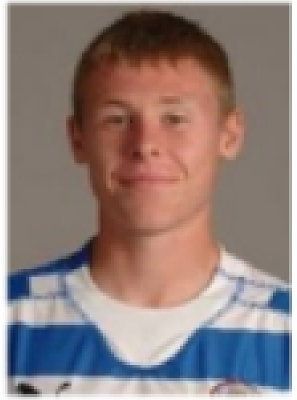
- 阿兰·麦席森·图灵（1912~1954），英国著名数学家、逻辑学家、密码学家，被称为计算机科学之父、人工智能之父。1912年6月23日生于英国帕丁顿，1931年进入剑桥大学国王学院，师从著名数学家哈代，1938年在美国普林斯顿大学取得博士学位，二战爆发后返回剑桥，曾协助军方破



他的英年早逝，像他横溢的才华一样，令世界吃惊与难以置信。生命虽然短暂，但那传奇的人生,丰富多彩的创造力和智慧而深邃的思想，使他犹如一颗耀眼的明星，持续地照耀着人间后世在科学的浩瀚太空里探索未来的人们。

Alonzo Church

- 阿隆佐 · 邱奇 Alonzo Church，1903年6月14日-1995年8月11日
  是美国数学家，1936年发表可计算函数的第一份精确定义，对算法理论的系统发展做出巨大贡献。邱奇在普林斯顿大学受教并工作四十年，曾任数学与哲学教授。1967年迁往加利福尼亚大学洛杉矶分校。



- 1936年9月，阿兰·麦席森·图灵应邀到美国普林斯顿高级研究院学习，并与丘奇一同工作。
- 1937年，阿兰·麦席森·图灵发表的另一篇文章“可计算性与1可定义性”则拓广了丘奇（Church）提出的“丘奇论点”，形成“丘奇-图灵论点”，对计算理论的严格化，对计算机科学的形成和发展都具有奠基性的意义。

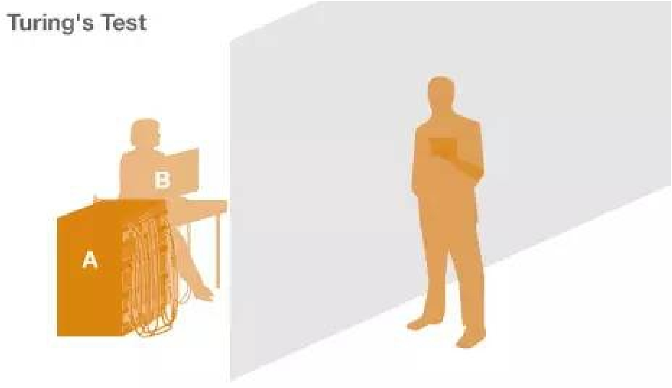
Turing test

- 1950年，图灵在那篇著名的论文《计算机械与智力》的开篇说：“我建议大家思考这个问题：‘机器能思考吗？’”
- 图灵提出了“模仿游戏”：
  - 一场正常的模仿游戏有ABC三人参与，A是男性，B是女性，两人坐在房间里；C是房间外的裁判，他的任务是要判断出这两人谁是男性谁是女性。但是男方是带着任务来的：他要欺骗裁判，让裁判做出错误的判断。
- 那么，图灵问：“如果一台机器取代了这个游戏里的男方的地位，会发生什么？这台机器骗过审问者的概率会比人类男女参加时更高吗？这个问题取代了我们原本的问题：‘机器能否思考？’”而这，就是图灵测试的本体。

Turing test

- 1952年，在一场BBC广播中，图灵谈到了一个新的具体想法：让计算机来冒充人。如果不足70%的人判对（也就是超过30%的裁判误以为在和自己说话的是人而非计算机），那就算作成功了。

- 2014 年，一个名为尤金·古斯曼的聊天机器人项目模拟了一名 13 岁的乌克兰男孩，在阅读大学组织的一次活动中，通过了图灵测试。聊天机器人虽然说服了伦敦皇家学会33% 的法官相信这是人类，但批评者很快也指出了测试的不足之处。



Turing理论

- Computer Science Law:

A computation is mechanical if and only if it can be performed by a Turing Machine

> There is no known model of computation more powerful than Turing Machines

- Definition of Algorithm:

An algorithm for function $f(w)$ is aTuring Machine which computes $f(w)$

Turing理论

算法就是图灵机

当我们说:

存在一个算法

意味着:

存在一个实现该算法的图灵机

图灵机的等价

- 图灵机 M_1 与图灵机 M_2 称为等价，如果

$$L(M_1)=L(M_2)$$

- 两类图灵机的功能相同是指: 两类图灵机接受相同的语言集合 $\Leftrightarrow$

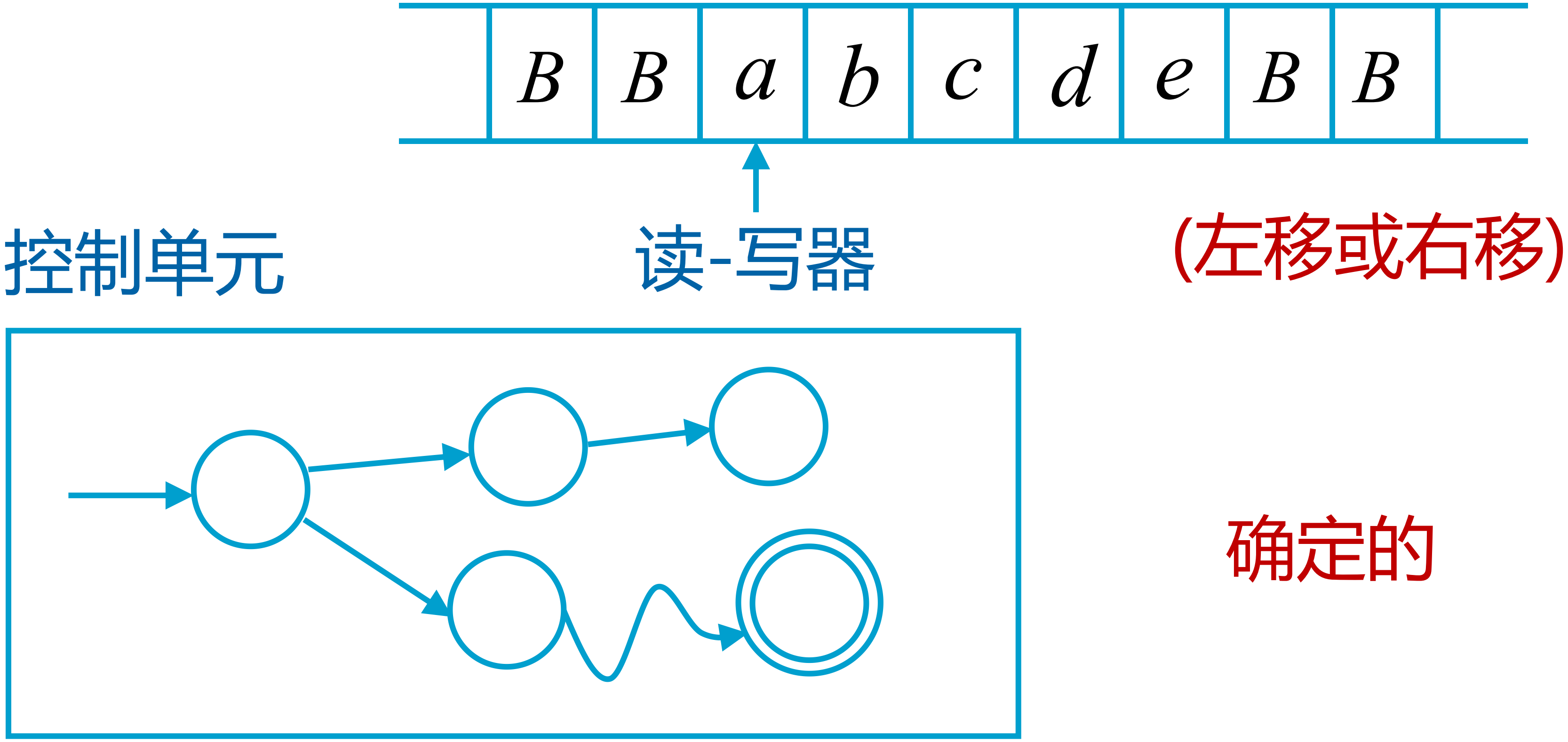
  第一类中任一TM M_1，第二类中一定存在 TM M_2，满足：

$$L(M_1)=L(M_2)$$

  反之亦然

标准图灵机

带两端无限：



图灵机的等价

将证明:

每一类型的图灵机与标准图灵机的功能相同的.

证明相同功能的方法— 模拟:

利用另一类中的TM模拟第一类中的TM.

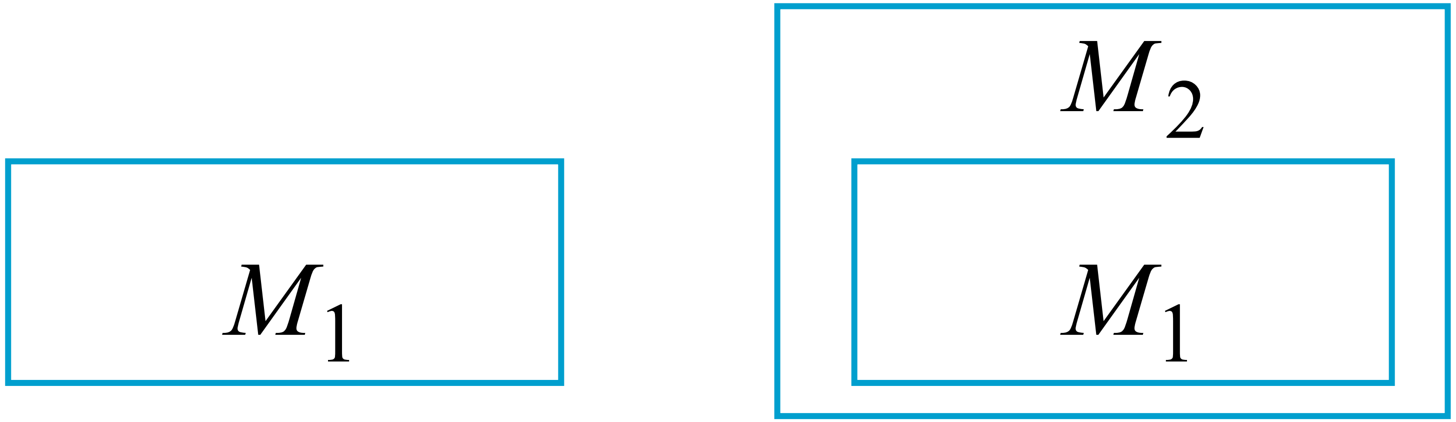
模拟方法

第一类

原TM

第二类

模拟TM



模拟方法

第一类TM结构对应模拟TM的结构

原TM　　　　最终结构

$$
\begin{array}{ccccccccc}
d_0 & \vdash & d_1 & \vdash & \ldots & \vdash & d_n & & d_f \\
\updownarrow & & \updownarrow & & & & \updownarrow & & \updownarrow \\
d_0' & \vdash^* & d_1' & \vdash^* & \ldots & \vdash^* & d_n' & & d'_f
\end{array}
$$

模拟TM

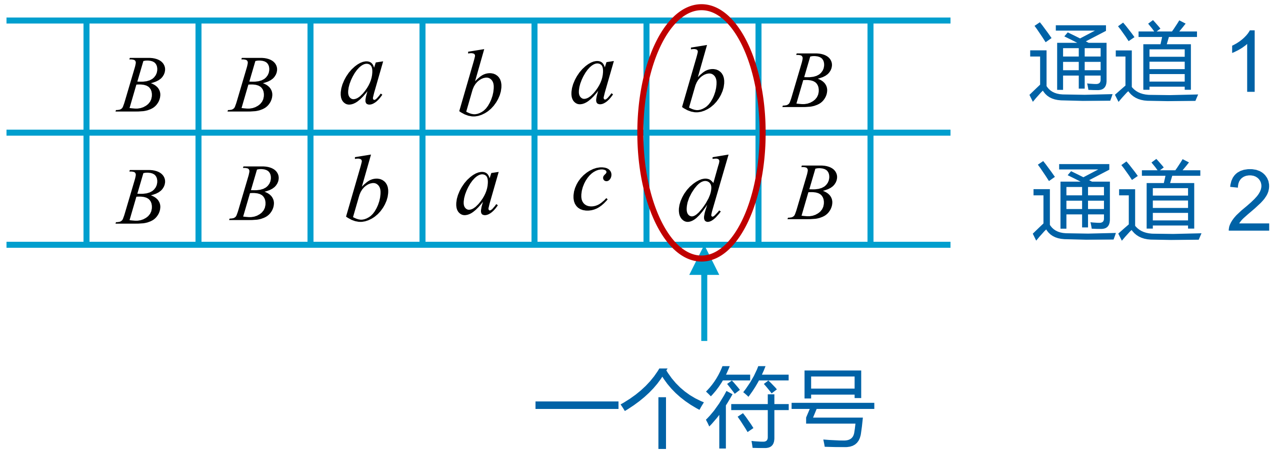
模拟TM与原TM接受相同的语言

Turing 机

- 图灵机的计算
- 图灵机的编程技术
- 图灵理论
- 图灵机带的扩展
- 图灵机移动的扩展
- 受限图灵机
- 图灵机与其它自动机
- 图灵机与计算机

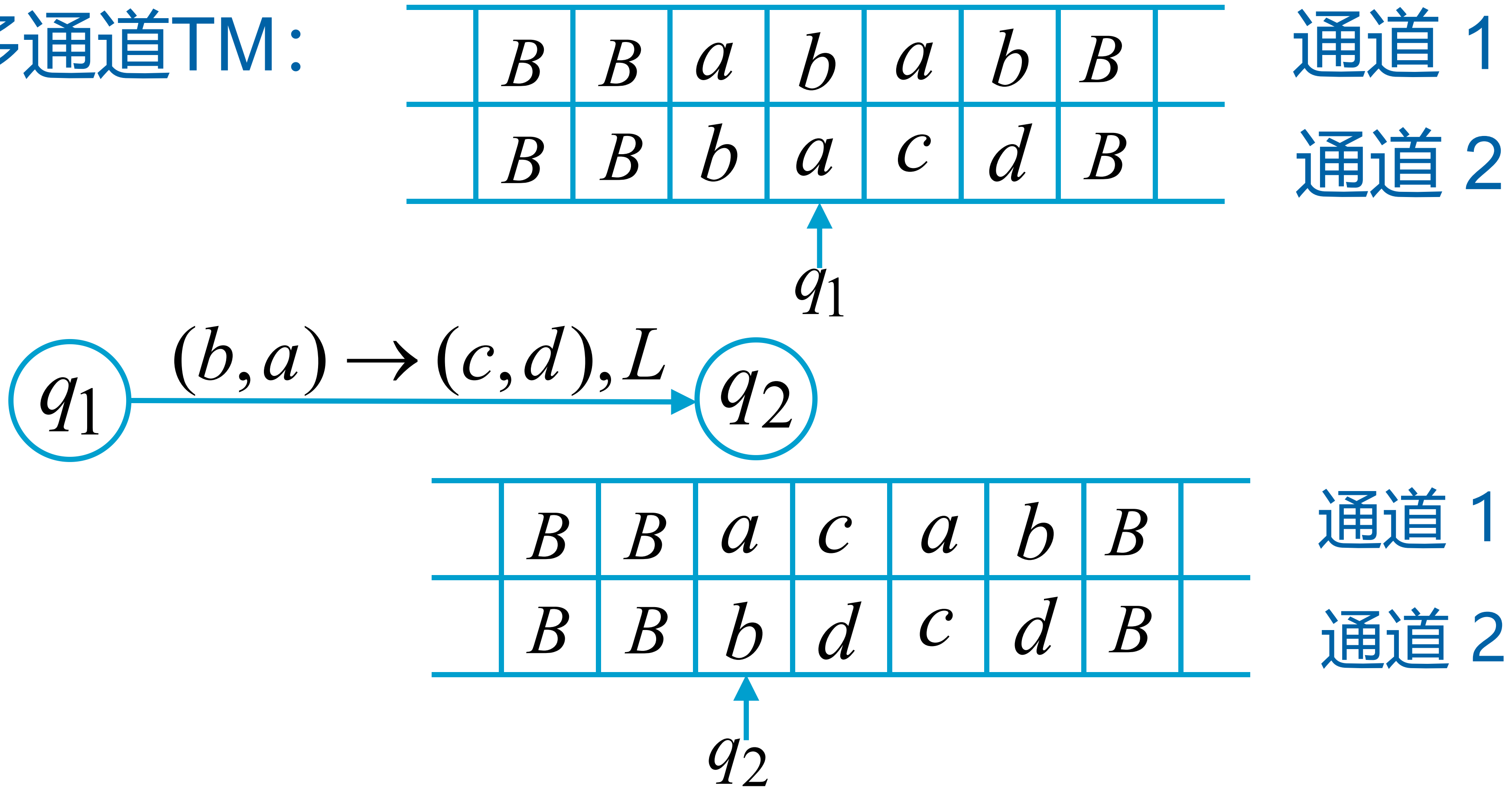
标准图灵机

多通道TM：



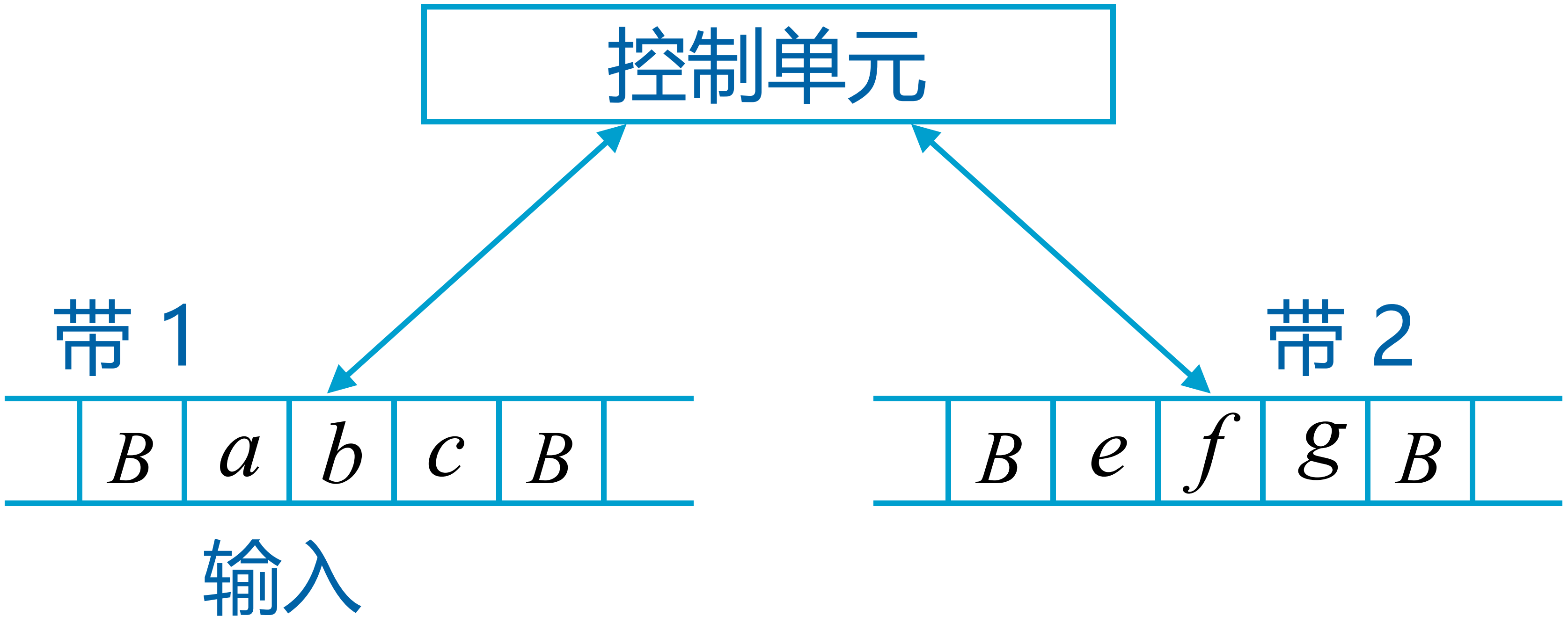
标准图灵机

多通道TM：



B B a b a b B
B B b a c d B
通道 1
通道 2
q_1
$(b,a) \to (c,d),L$
q_1
q_2
B B a c a b B
B B b d c d B
通道 1
通道 2
q_2

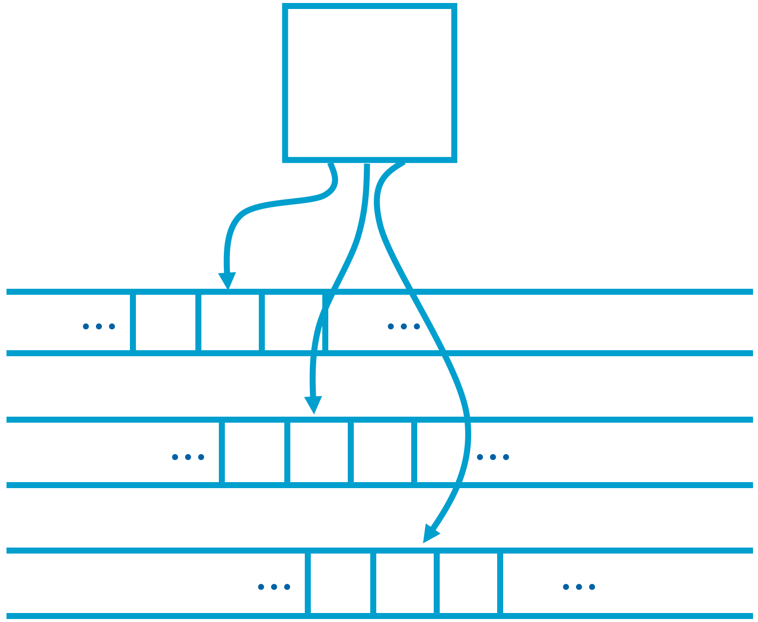
多带图灵机



多带图灵机

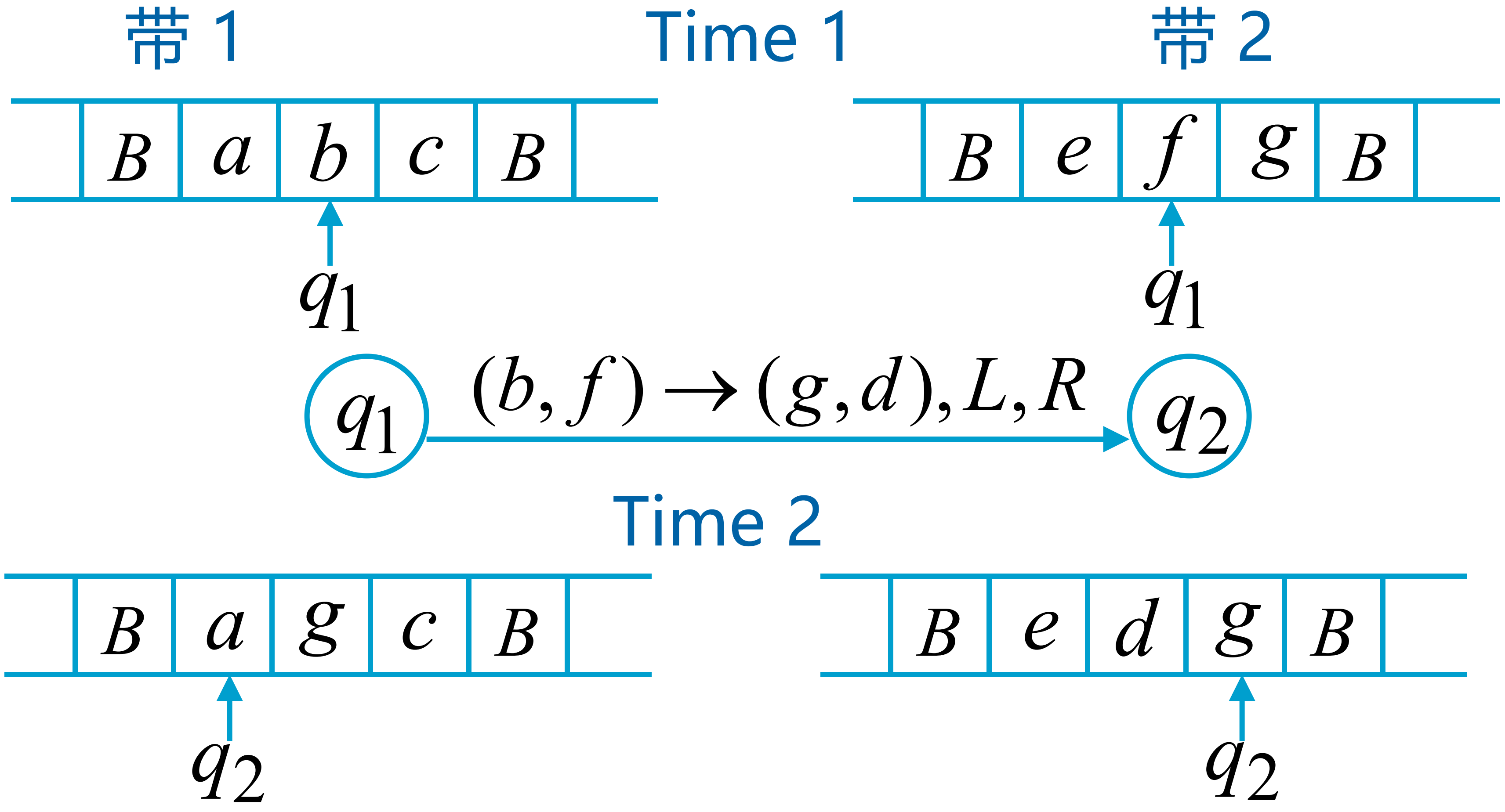
特点：

1. 开始时，有限控制的读写头 (读头) 处于第一条带初始状态，即第一条带输入符号串的最左端；其余各带的读头置于任何单元格上。



2. 每一步移动，控制单元进入新的状态，每条带上正被扫描的符号被替换为新的带符，每个带头独立地左移一格、右移一格或者不动。

多带图灵机



带 1
Time 1
带 2
B a b c B
B e f g B
q1
q1
q1
(b, f) → (g, d), L, R
q2
Time 2
B a g c B
B e d g B
q2
q2

多带图灵机

结论：多带TM可以模拟标准TM。

仅用一个带即可

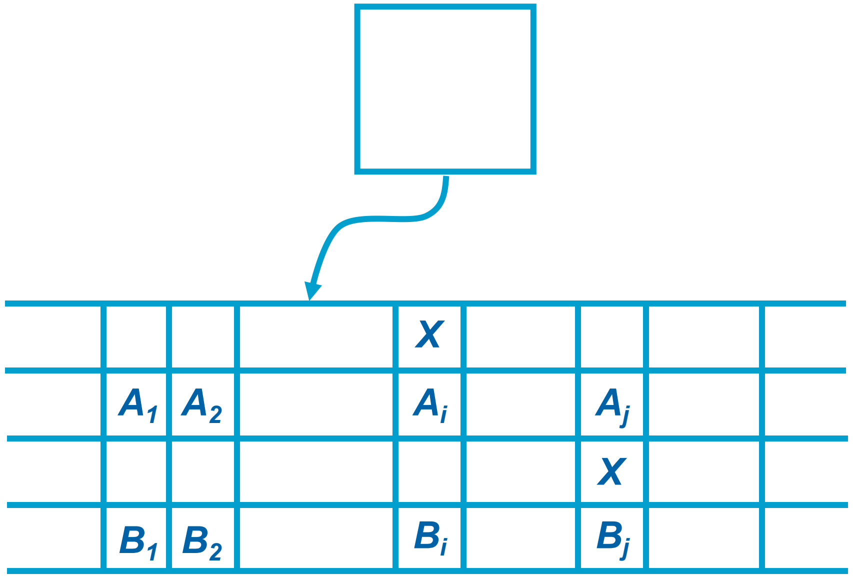
结论：标准TM也能模拟多带TM

标准TM:

- 采用多通道
- 多带TM中，每一带对应两个通道TM

多带图灵机

采用可带存储数据的状态和多通道图灵机模拟多带机。k 个带的图灵机可以用 $2k$ 个通道图灵机模拟。下图所示的 4 个通道图灵机模拟一个两带的图灵机。



多带图灵机

Tape 1

	B	a	b	c	B	

Tape 2

	B	e	f	g	h	B

具有4通道标准TM

		0	1	0				读头位置
		a	b	c				带 1
		0	0	1	0			读头位置
		e	f	g	h			带 2

多带图灵机

定理：多带TM与标准TM功能相同。

说明：相同的功能并不意味速度一致。

例：　　语言　　$L=\{a^n b^n\}$

	计算时间
标准TM	n^2
两带TM	n

多带图灵机

$$L=\{a^n b^n\}$$

标准TM:

来回次数 n^2

两带TM:

复制 b^n 到带2 （n 步）

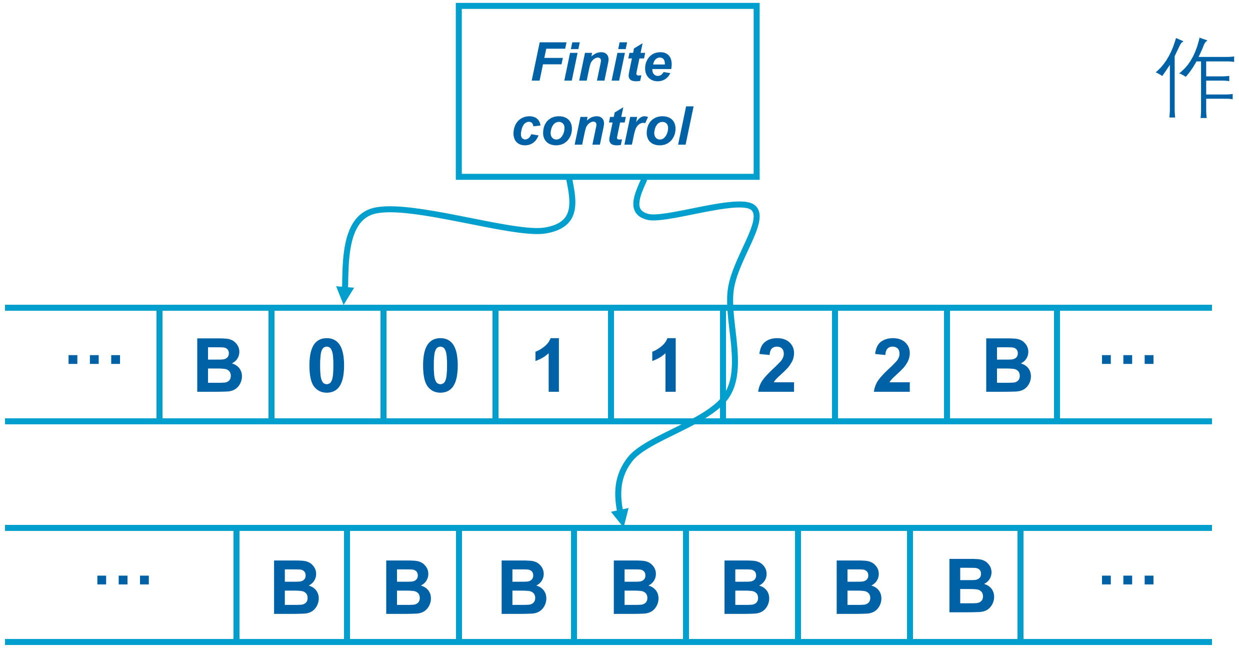
保留 a^n 在带1 （n 步）

比较带1与带2 （n 步）

多带图灵机

如何设计一个双带的图灵机接受语言

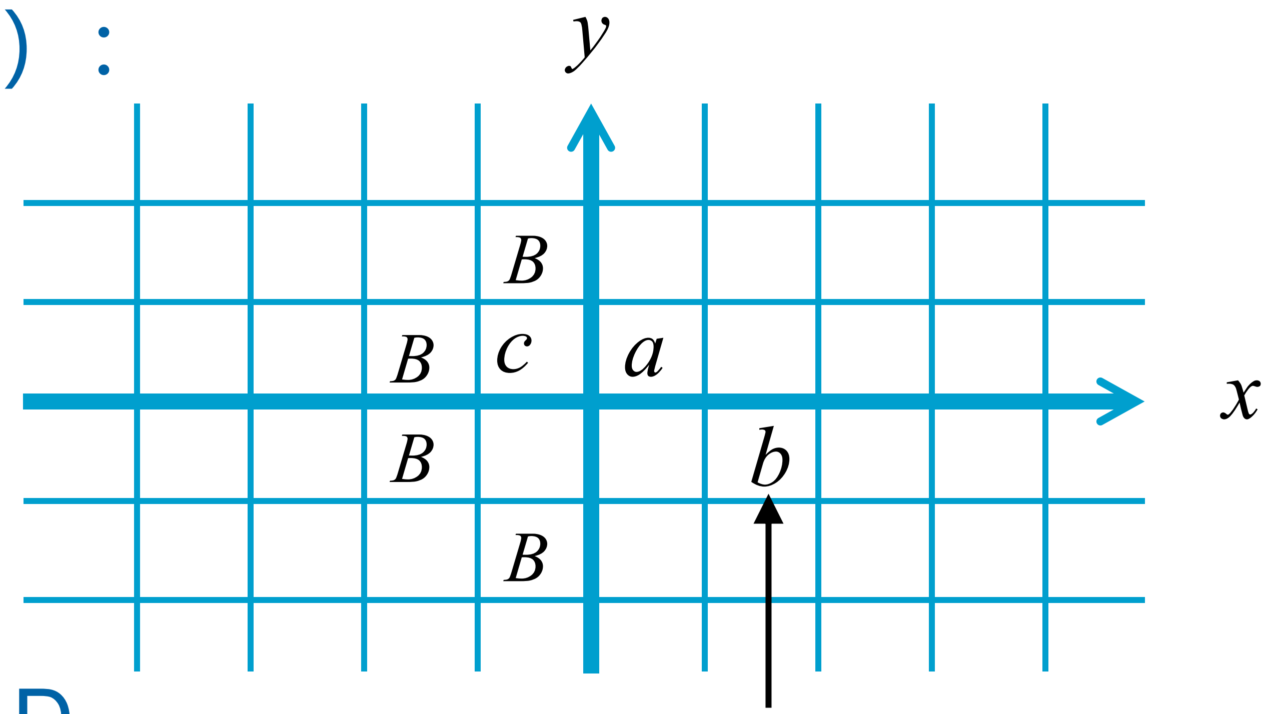
$$L = \{\, 0^n 1^n 2^n \mid n \geq 1 \,\}?$$



作为课后练习

多维图灵机

2-维带（面）：



移动: L,R,U,D

U: 上移，　D: 下移

位置: +2, -1

多维图灵机

结论：多维TM可以模拟标准TM。

使用一个维度TM即可。

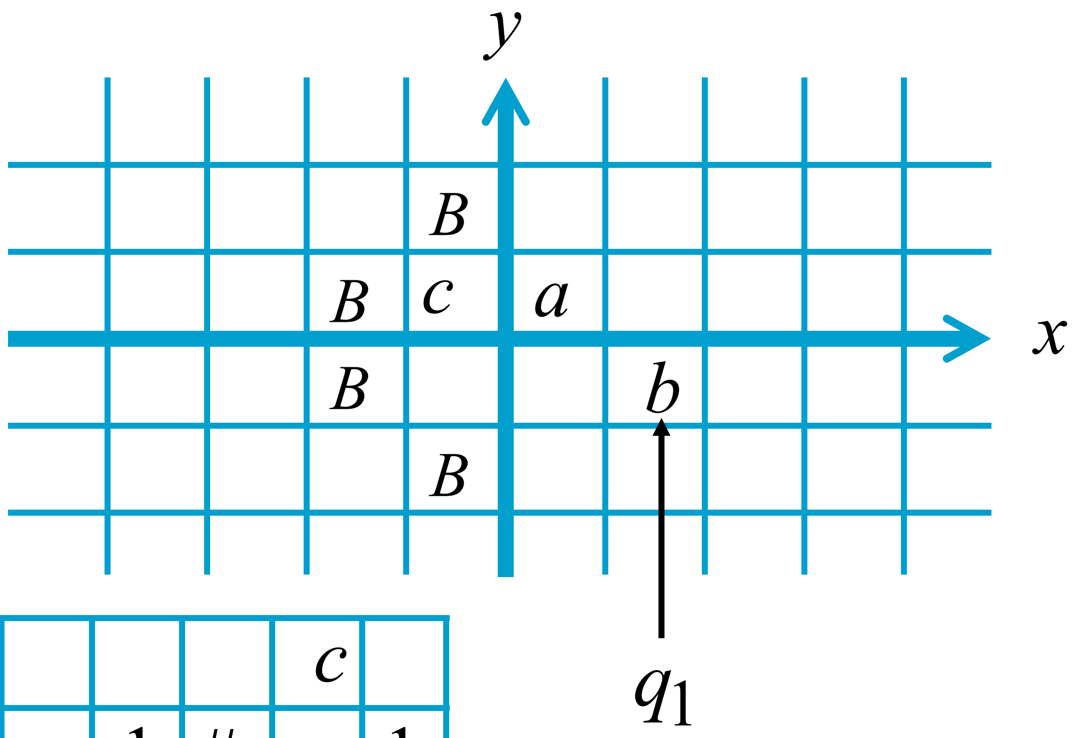
结论： 标准TM也可以模拟多维TM。

标准TM:

- 使用两个通道
- 符号存储在通道 1
- 对应坐标存储在通道 2

多维图灵机

2-维TM：



标准TM

符号	a				b					c	
坐标	1	#	1	#	2	#	−	1	#	−	1

q_1

多维图灵机

标准TM每次转移重复如下操作：

- 更新当前符号
- 计算后一位置的坐标
- 转移到新位置

定理: 多维TM与标准TM功能相同.

Turing 机

- 图灵机的计算
- 图灵机的编程技术
- 图灵理论
- 图灵机带的扩展
- 图灵机移动的扩展
- 受限图灵机
- 图灵机与其它自动机
- 图灵机与计算机

读头移动的扩展

停止功能的图灵机:

B	B	a	a	b	a	b	b	c	a	c	a	B	B	B

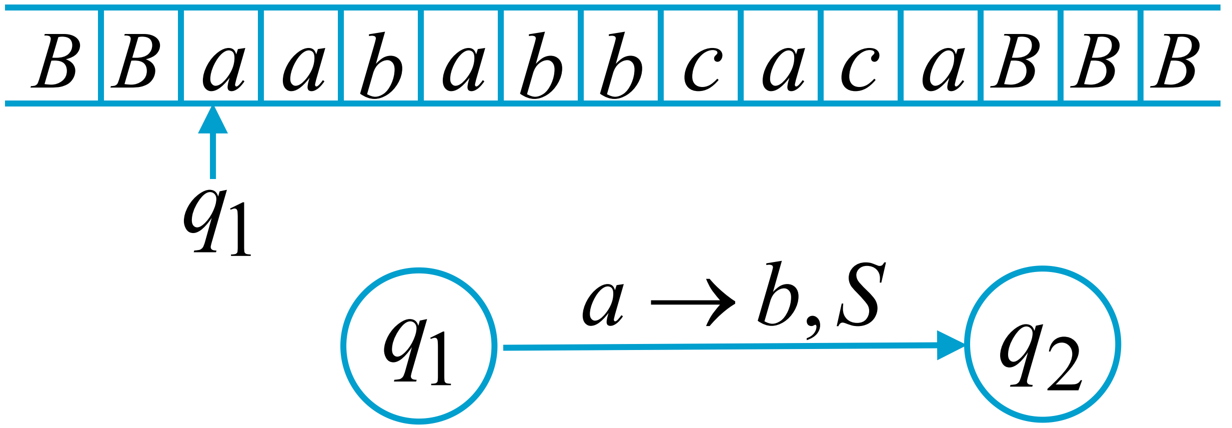
左移, 右移, 停止

L, R, S 读头可以停在原来位置

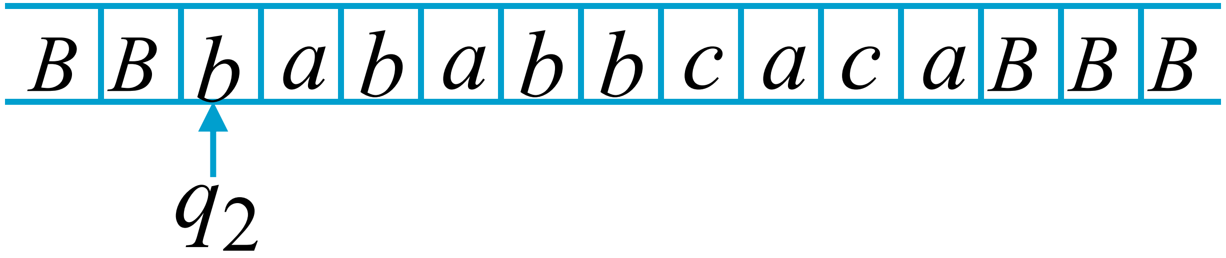
读头移动的扩展

例:

Time 1



Time 2



读头移动的扩展

定理: 停止功能TM与标准TM具有相同功能。

先证明:

1. 停止功能TM具有标准TM的功能。

证明:

显然，停止功能TM不使用S 移动时，与标准TM一致。

读头移动的扩展

再证明:

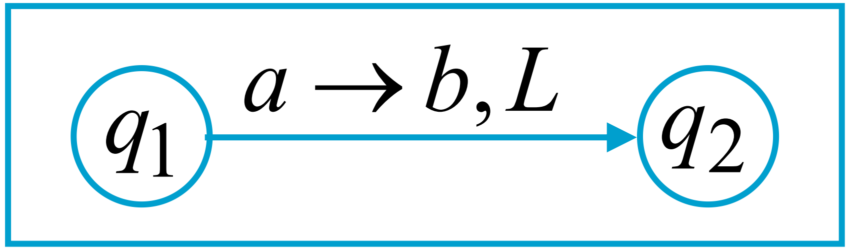
2. 标准TM也具有停止功能TM的功能。

证明:

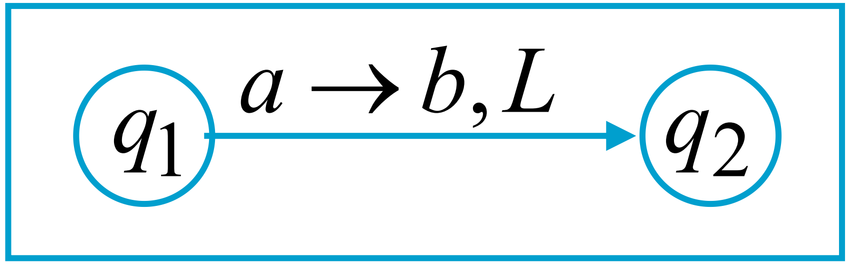
标准TM能模拟停止功能TM。

读头移动的扩展

停止功能TM：



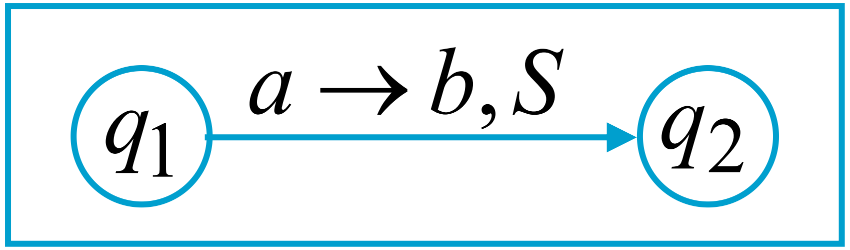
标准TM模拟：



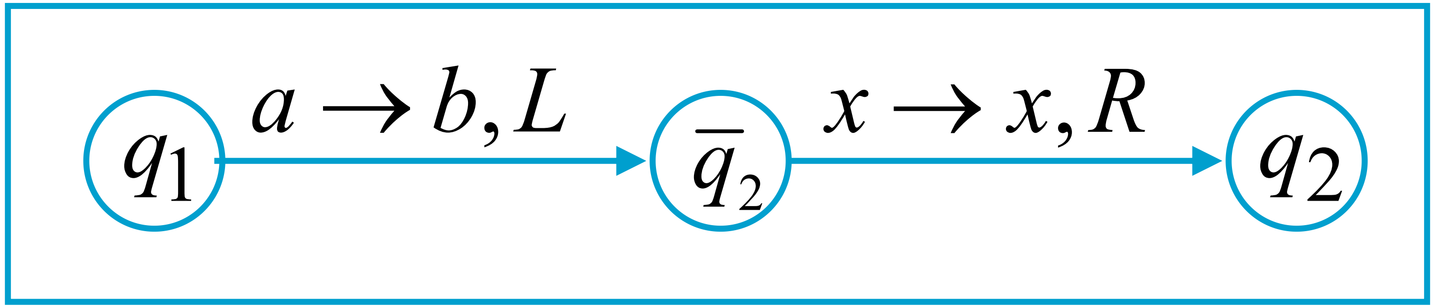
右移动结论类似。

读头移动的扩展

停止功能TM：



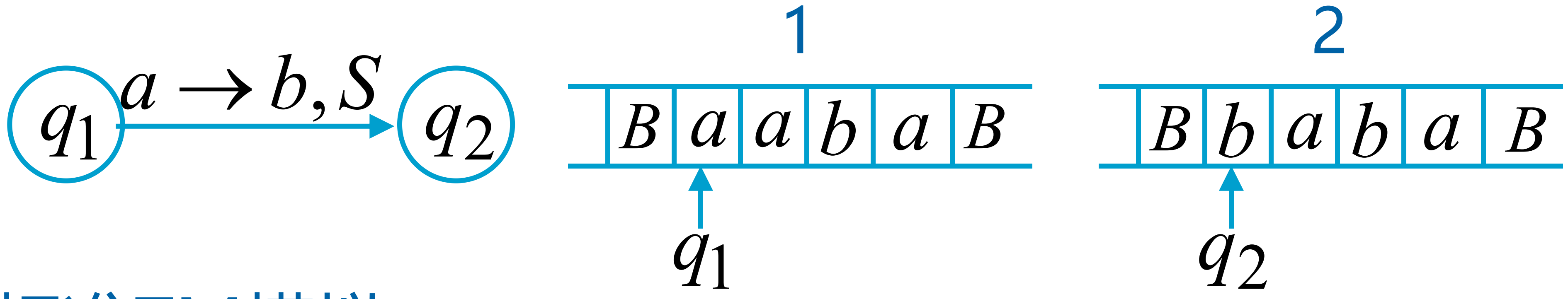
标准TM模拟：



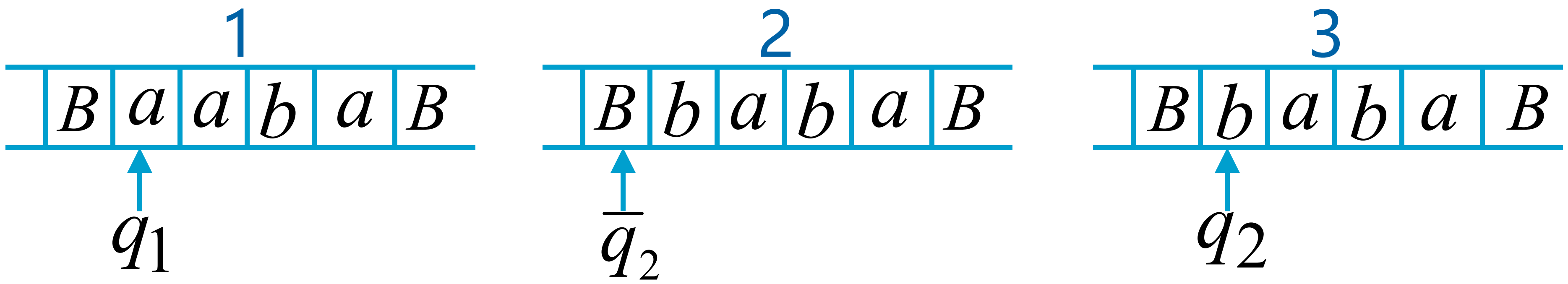
x 为任意带符号

读头移动的扩展

停止功能TM：



标准TM模拟：



状态移动的扩展

非确定图灵机：TM下一步移动有多种选择。

转移函数定义为：

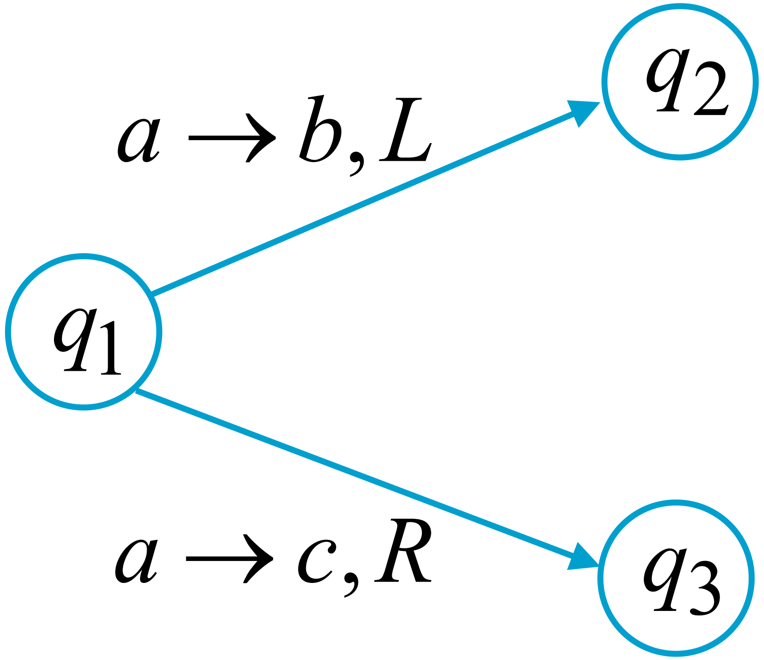
$$\delta: Q \times \Gamma \to 2^{Q \times \Gamma \times D}$$

其中 Q、Γ 和 D 分别为有限状态集、带符号集和读头的移动方向.

即 $\delta(q, X)$ 为三元组的集合：

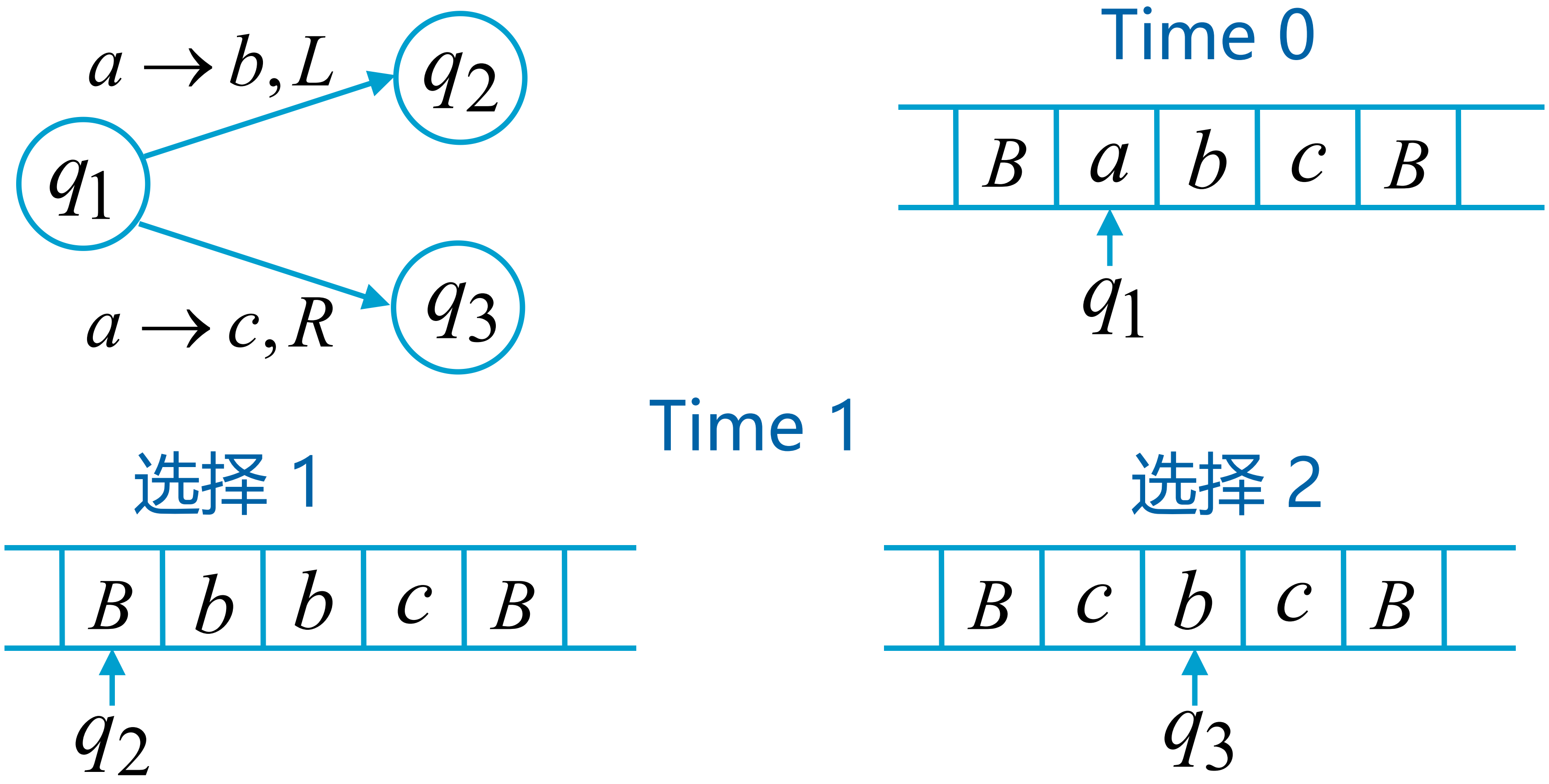
$$\{ (q_1, Y_1, D_1), (q_2, Y_2, D_2), \ldots , (q_k, Y_k, D_k) \}$$

非确定图灵机



状态转移是非确定

非确定图灵机



$a \rightarrow b, L$
$a \rightarrow c, R$
q_1
q_2
q_3
Time 0
B a b c B
q_1
Time 1
选择 1
B b b c B
q_2
选择 2
B c b c B
q_3

非确定图灵机

输入串 w 被接受，如果下列计算成立：

$$\underbrace{q_0 w}_{\text{初始ID}} \vdash^* \underbrace{x\, q_f\, y}_{\text{最终ID}}$$

终态

非确定图灵机

结论1：非确定TM能模拟标准TM。

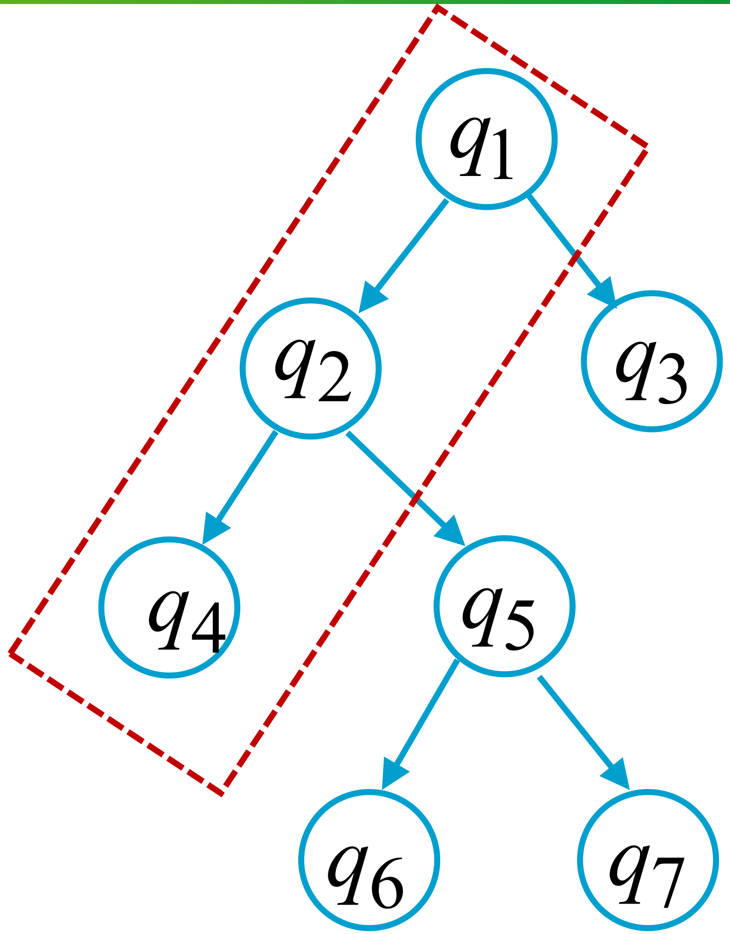
因为确定TM也是非确定TM中一类

结论2: 确定TM也能模拟非确定TM

对确定TM: 保存所有可能的计算路径。

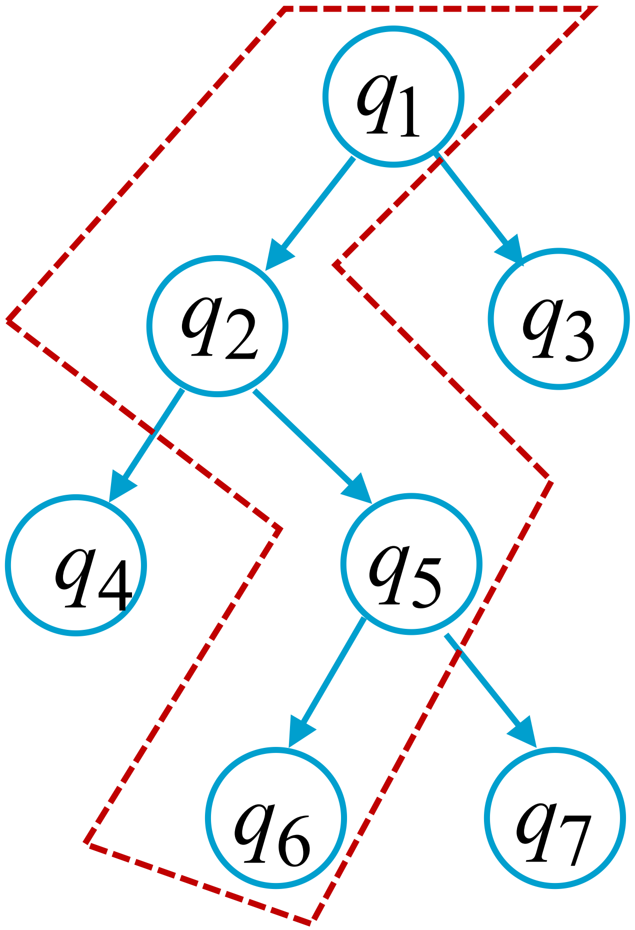
非确定图灵机

计算 1



非确定图灵机

计算 2



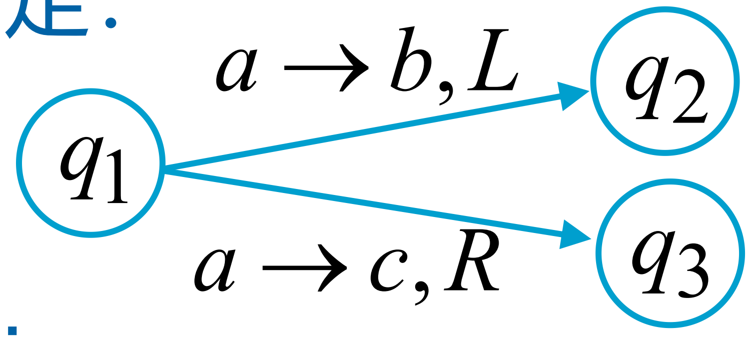
确定TM模拟
非确定TM:

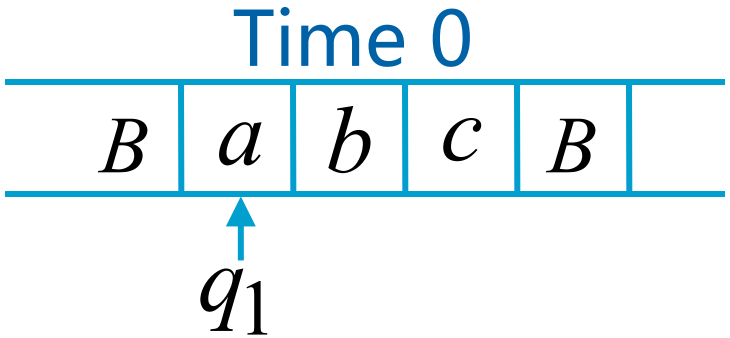
保存所有计算路径

每一计算存储在2维带中

模拟计算

非确定:





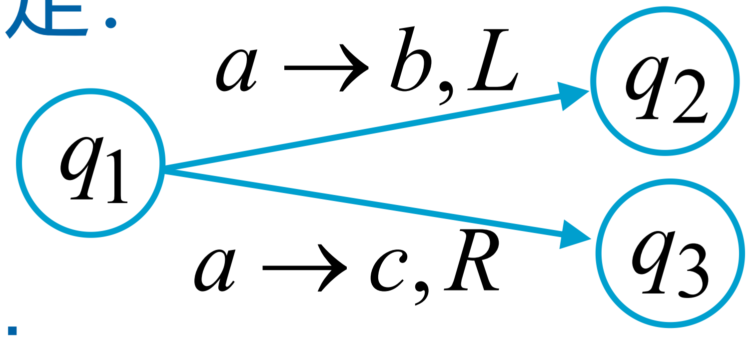
确定:

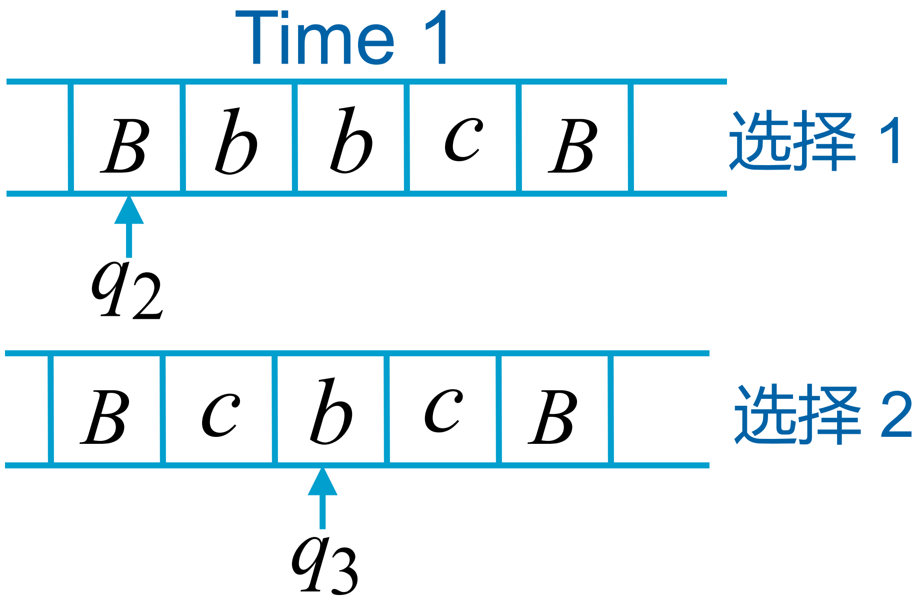
	#	#	#	#	#		
	#	a	b	c	#		
	#	q_1			#		
	#	#	#	#	#		

计算 1

模拟计算

非确定：





确定:

#	#	#	#	#	#		
#		b	b	c	#		
#	q_2				#		
#		c	b	c	#		
#			q_3		#		

计算 1

计算 2

模拟计算

重复

- 每一步执行一个计算
- 若当前的计算有两个或多个选择:
  1. 复制前结构
  2. 改变复制副本中的状态

模拟计算

定理:

非确定TM与标准（确定）TM具有相同的功能。

说明:

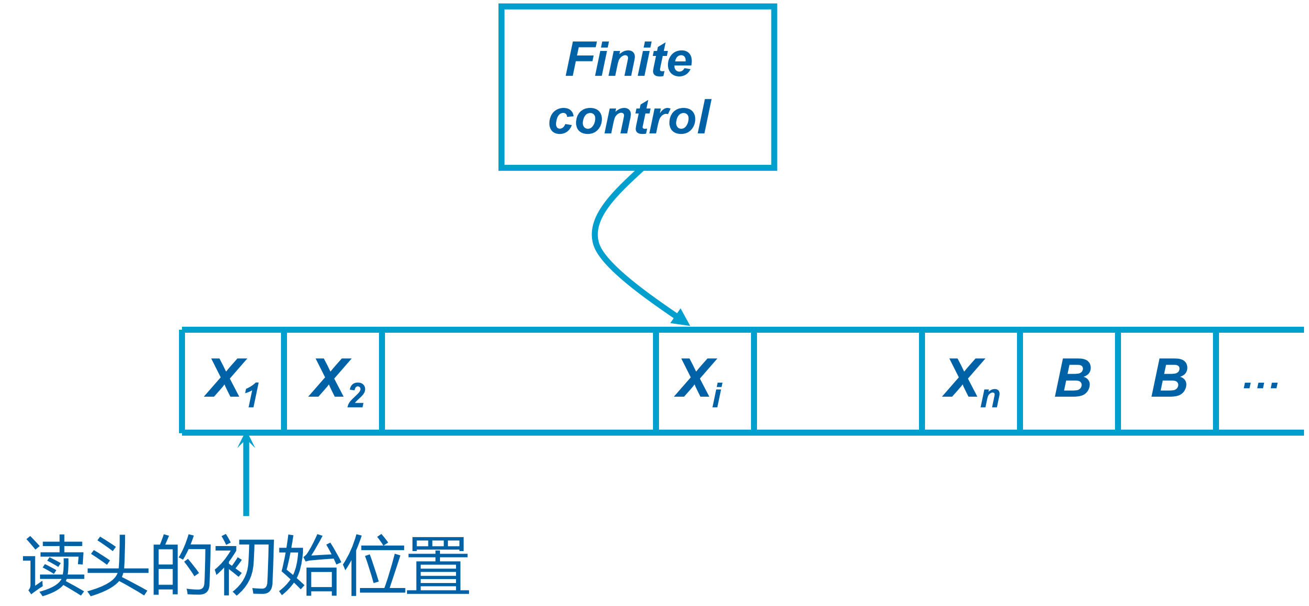
功能相同并不等同于复杂性相同。确定TM模拟非确定TM，时间为指数级

Turing 机

- 图灵机的计算
- 图灵机的编程技术
- 图灵理论
- 图灵机带的扩展
- 图灵机移动的扩展
- 受限图灵机
- 图灵机与其它自动机
- 图灵机与计算机

半无限带机

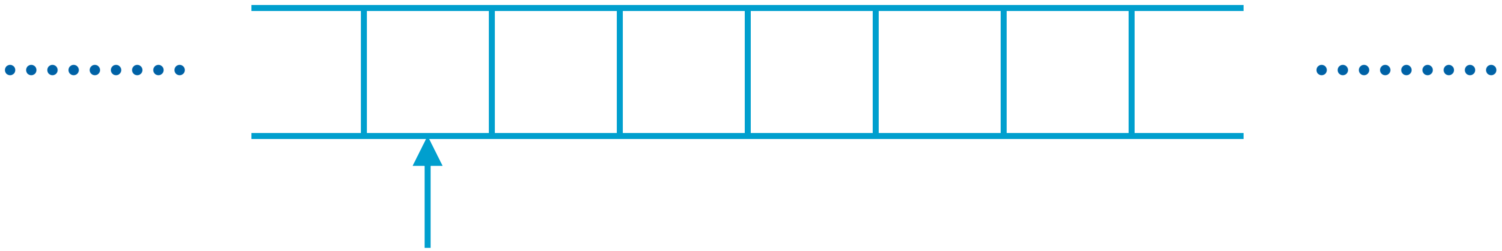
读头的初始位置左部没有单元格，读头移动受限于从读头初始位置向右无限延伸的范围。



半无限带机

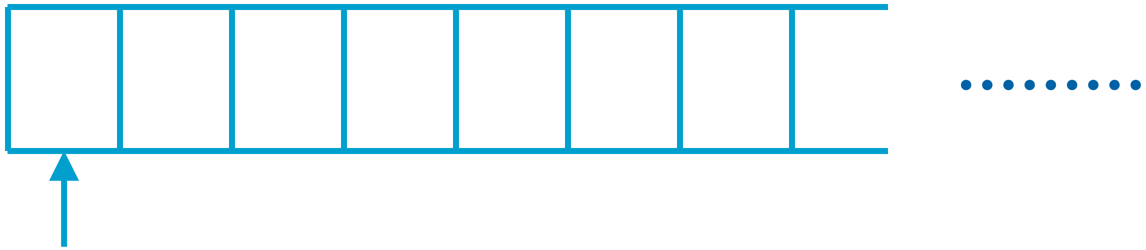
结论：标准图灵机可以模拟半无限带机。

标准图灵机



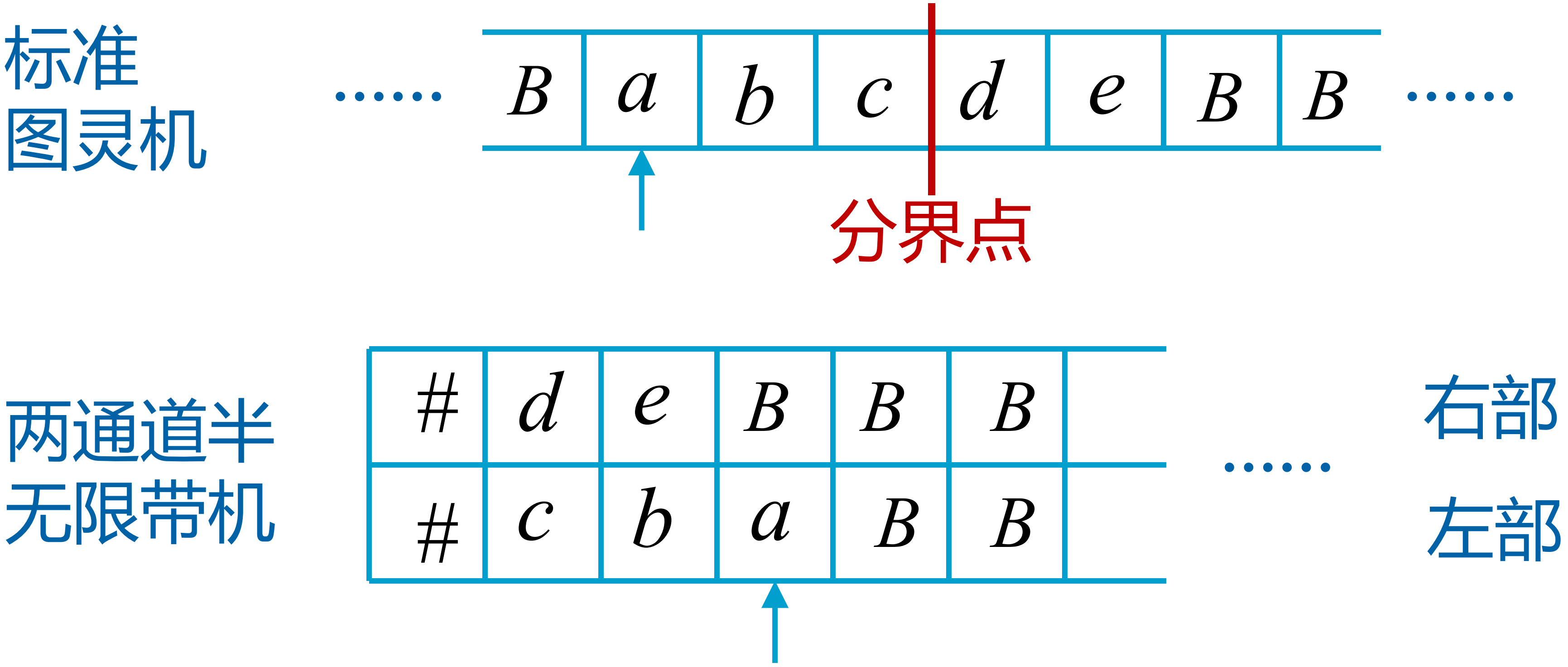
半无限带机

显然



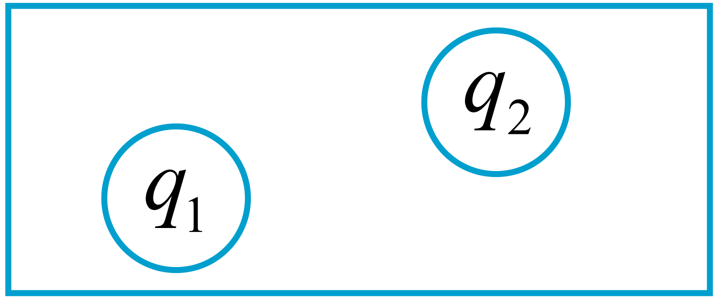
半无限带机

结论：半无限带机也能模拟标准图灵机。

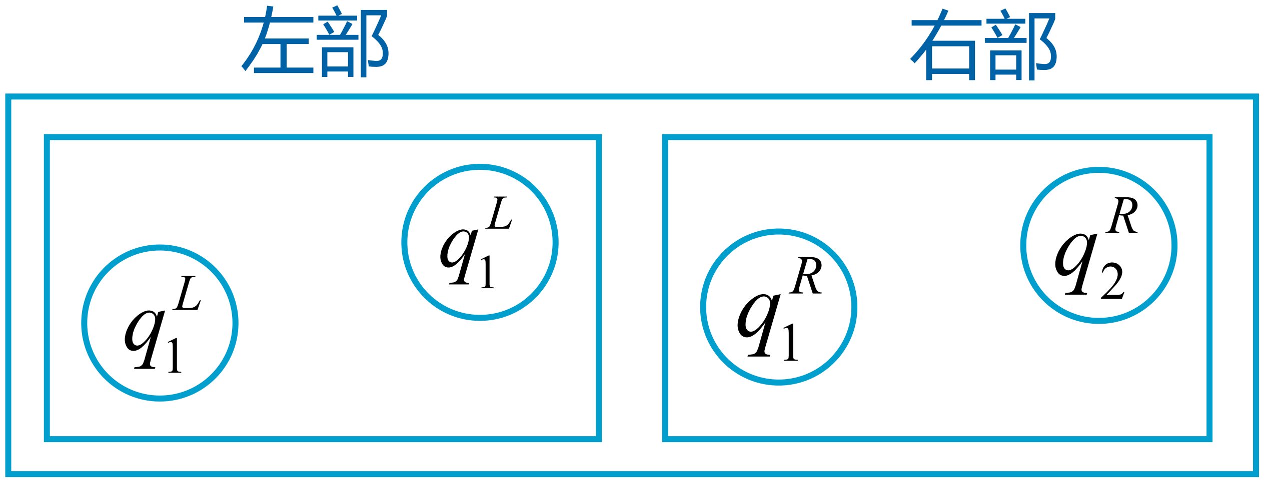


半无限带机

标准图灵机状态



半无限带机状态



半无限带机

标准图灵机：

$$q_1 \xrightarrow{a \to g, R} q_2$$

半无限带机：

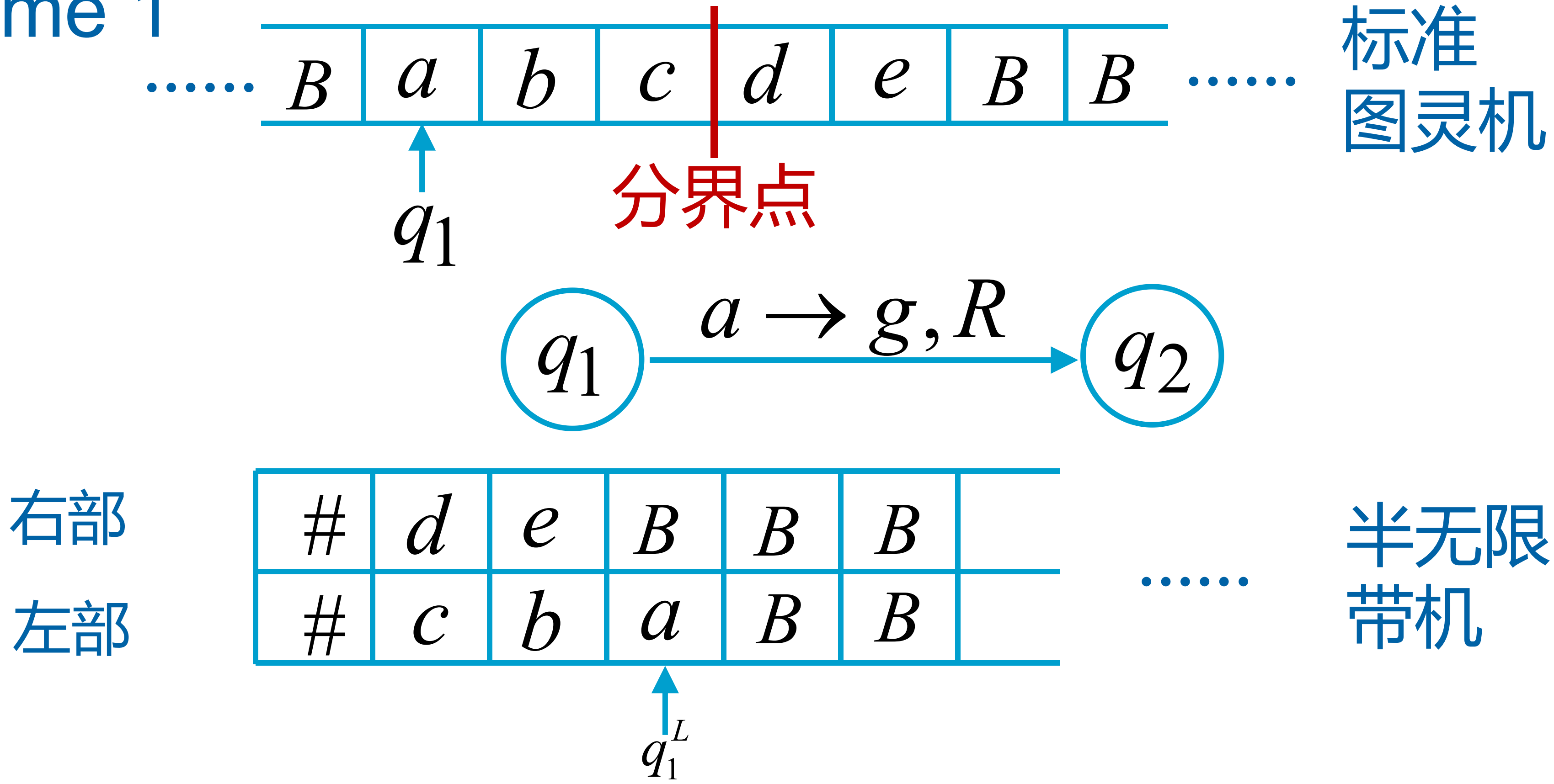
右部 $$q_1^R \xrightarrow{(a,x) \to (g,x), R} q_2^R$$

左部 $$q_1^L \xrightarrow{(x,a) \to (x,g), L} q_2^L$$

x 为任意符号

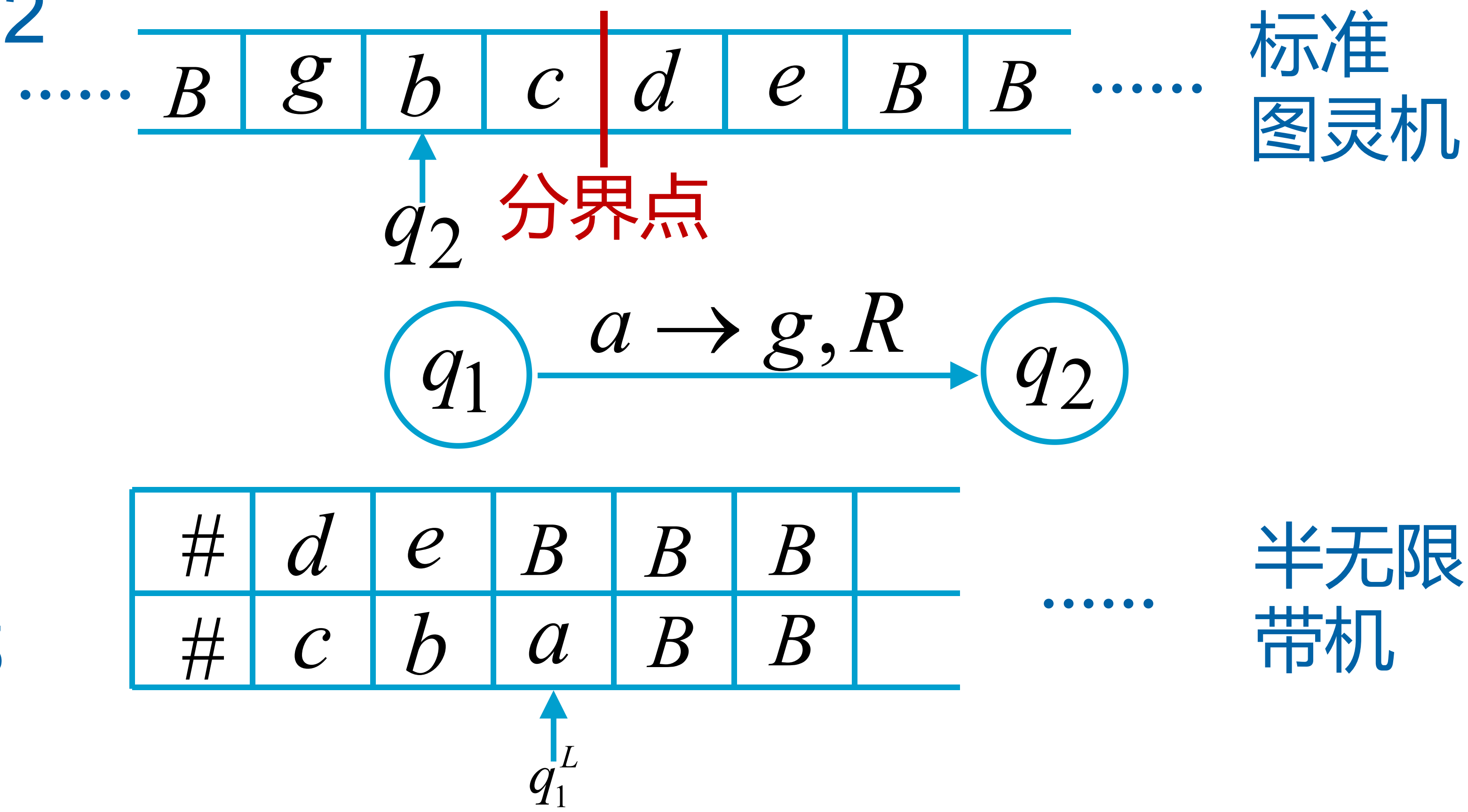
半无限带机

Time 1



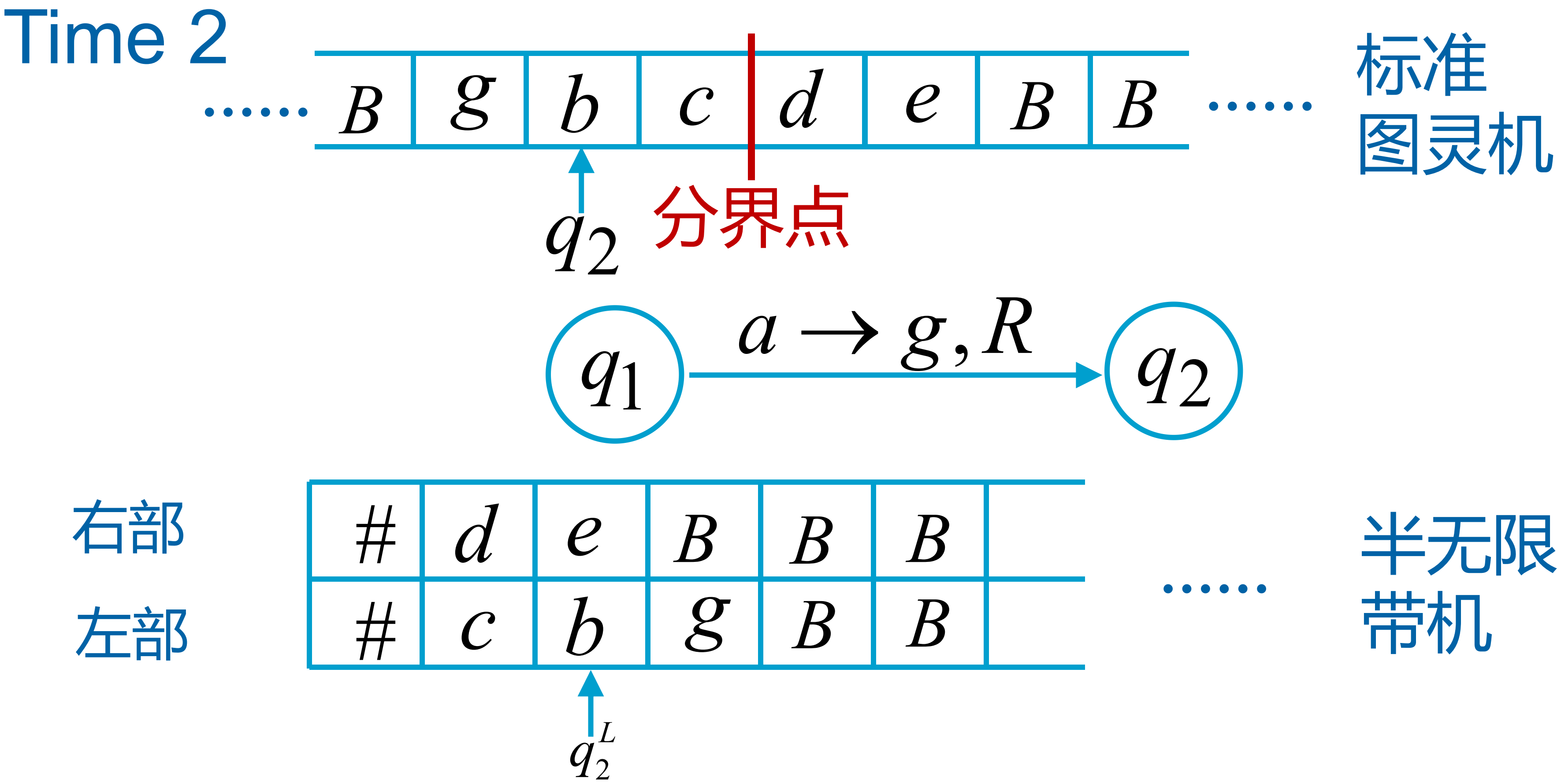
半无限带机

Time 2



B g b c d e B B
标准
图灵机
分界点
q_2
$q_1 \xrightarrow{a \to g, R} q_2$
右部
左部
d e B B B
c b a B B
半无限
带机
q_1^L

半无限带机


Time 2
标准图灵机
B g b c d e B B
q_2
分界点
q_1
$a \rightarrow g, R$
q_2
右部
左部
d e B B B
c b g B B
q_2^L
半无限带机

半无限带机

半无限带机在分界端:

右部 $$q_1^R \xrightarrow{(\#,\#)\to(\#,\#),R} q_1^L$$

左部 $$q_1^L \xrightarrow{(\#,\#)\to(\#,\#),R} q_1^R$$

传递函数是非确定的

半无限带机

例:

Time 1

右部	#	d	e	B	B	B	………
左部	#	c	b	g	B	B	………

q_1^L

Time 2

右部	#	d	e	B	B	B	………
左部	#	c	b	g	B	B	………

q_1^R

半无限带机

定理:

半无限带机与标准图灵机

具有相同的功能

Turing 机

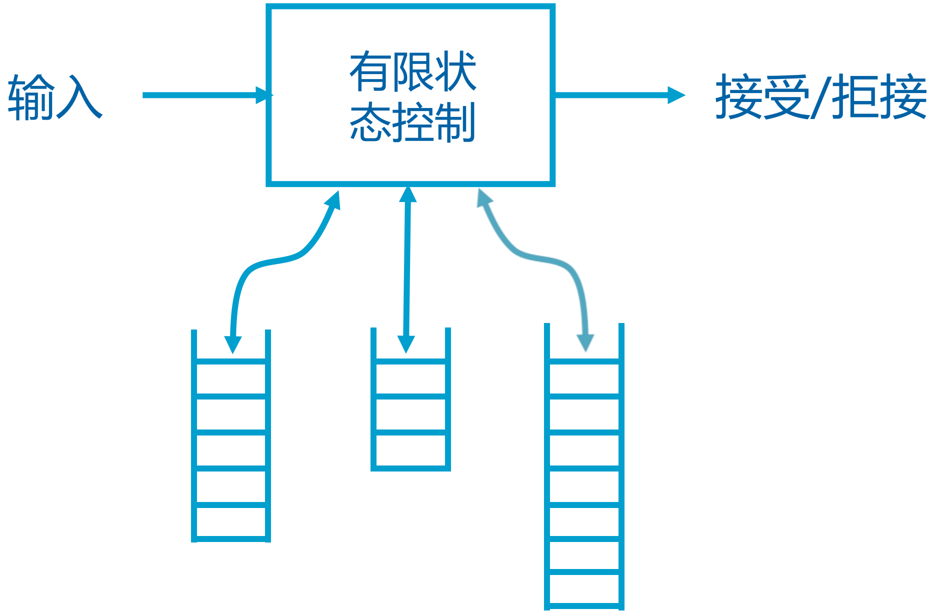
- 图灵机的计算
- 图灵机的编程技术
- 图灵理论
- 图灵机带的扩展
- 图灵机移动的扩展
- 受限图灵机
- 图灵机与其它自动机
- 图灵机与计算机

多栈机

包含 k 个堆栈PDA的状态转移函数为：

$$\delta(q, a, X_1, X_2, \ldots, X_k) = (p, \gamma_1, \gamma_2, \ldots, \gamma_k)$$

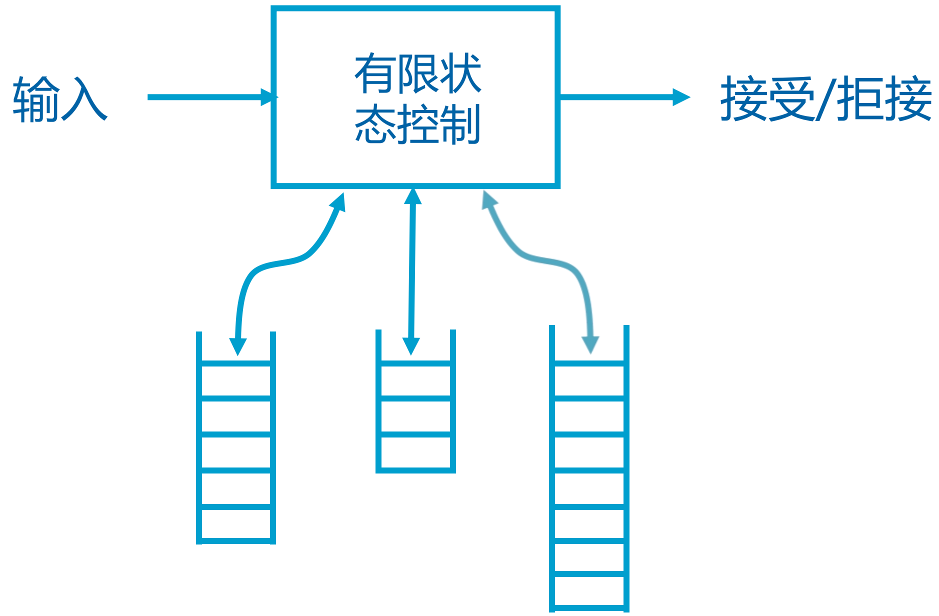
下图为 3 个堆栈的PDA：



多栈机

利用一个多带图灵机容易模拟多栈机。

下图多栈机可以采用 4 条带的多带图灵机模拟。一条带用于扫描输入，另 3 条带模拟堆栈。

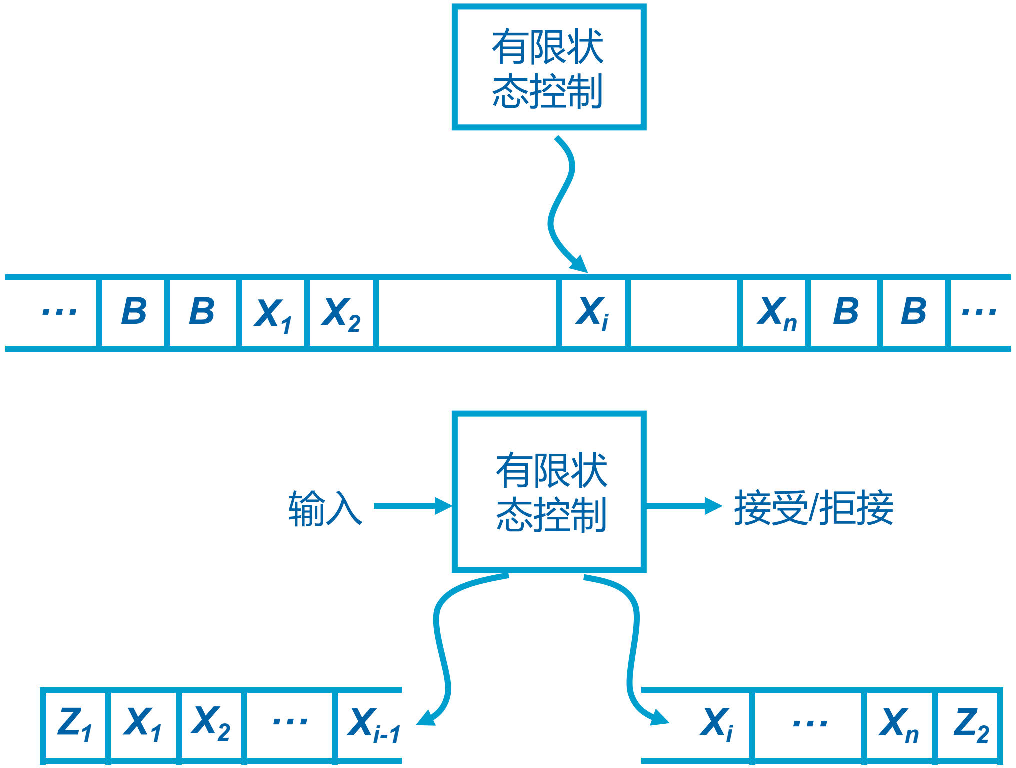


多栈机

一个双栈 PDA 可以模拟标准图灵机。

右图所示，第一个堆栈模拟当前读头位置左边的单元格；

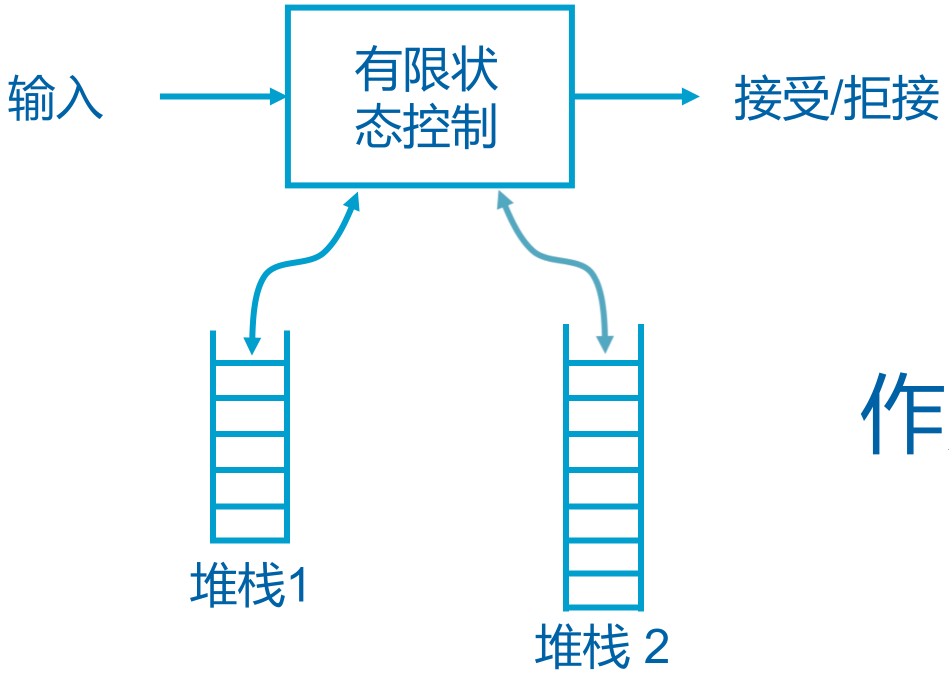
第二个堆栈模拟当前读头位置右边的单元格。



多栈机

如何设计一个双栈的 PDA 接受语言：

$$L = \{\, 0^n 1^n 2^n \mid n \geq 1 \}?$$

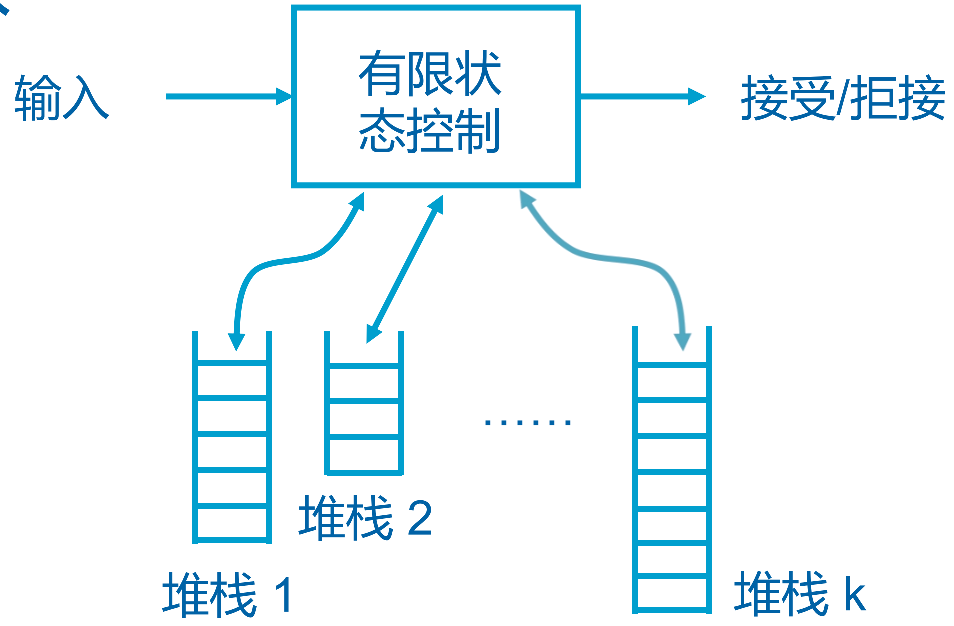


作为课后练习

计数器机

计数器机只有两个栈符号：{Z_0，X}，栈底符号为 Z_0，Z_0 只能被替换为 X^iZ_0 (i ≥0)，X 只能被替换为 X^i (i ≥0)栈顶为 Z_0 和 X 分别相当于计数器的值是0 或是非 0。

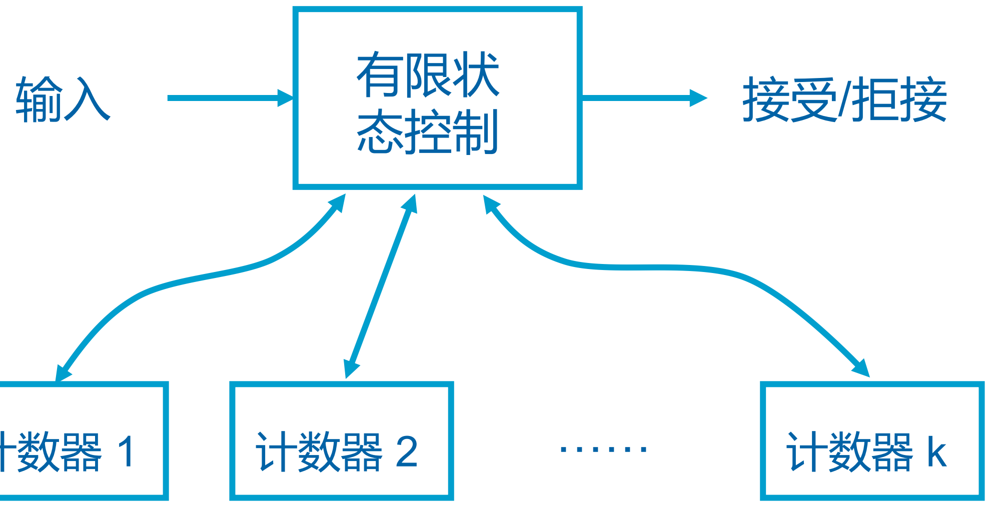
X 入栈相当于计数器加 1, X 出栈相当于计数器减 1, Z_0 不会出栈相当于计数器为 0 时不能减 1



计数器机

将多栈机的多个堆栈替换为多个计数器，每个计数器存放一非负整数，每个计数器的值是0或是非 0 。计数器机根据当前的状态，下一个输入符号以及每个计数器的当前值是否为 0 来确定下一步动作：

改变状态，计数器加 1 或减 1 ，但不允许对当前值为 0 的计数器减 1。



计数器机

上述计数器机相当于特殊的多栈机，因此计数器机接受的语言是递归可枚举的；

反之，所有递归可枚举语言是否都存在相应的计数器机接受？

关于计数器机有如下结论：

定理：具有一个计数器的计数器机的语言接受能力相当于确定的下推自动机.

计数器机

定理：具有两个（或以上）计数器的计数器机语言接受能力相当于图灵机，即可以接受递归可枚举语言。

证明 (分两步)：

（1）任何递归可枚举语言可以被具有三个计数器的计数器机接受；

（2）任何具有三个计数器的计数器机可以用一个具有两个计数器的计数器机来模拟。

Turing 机

- 图灵机的计算
- 图灵机的编程技术
- 图灵理论
- 图灵机带的扩展
- 图灵机移动的扩展
- 受限图灵机
- 图灵机与其它自动机
- 图灵机与计算机

图灵机与计算机

- 普通计算机能模拟图灵机

  采用适当的数据结构(如转移表)，可以编译普通的计算机程序，实现图灵机的有限状态控制机制。

- 存在问题：如何模拟无限延伸的带?

  设想可以无限扩充存储量的存储系统. 实际上，可装卸的外存系统并不严格规定存储量的上限，而且并非所有信息都需要在线存储。

图灵机与计算机

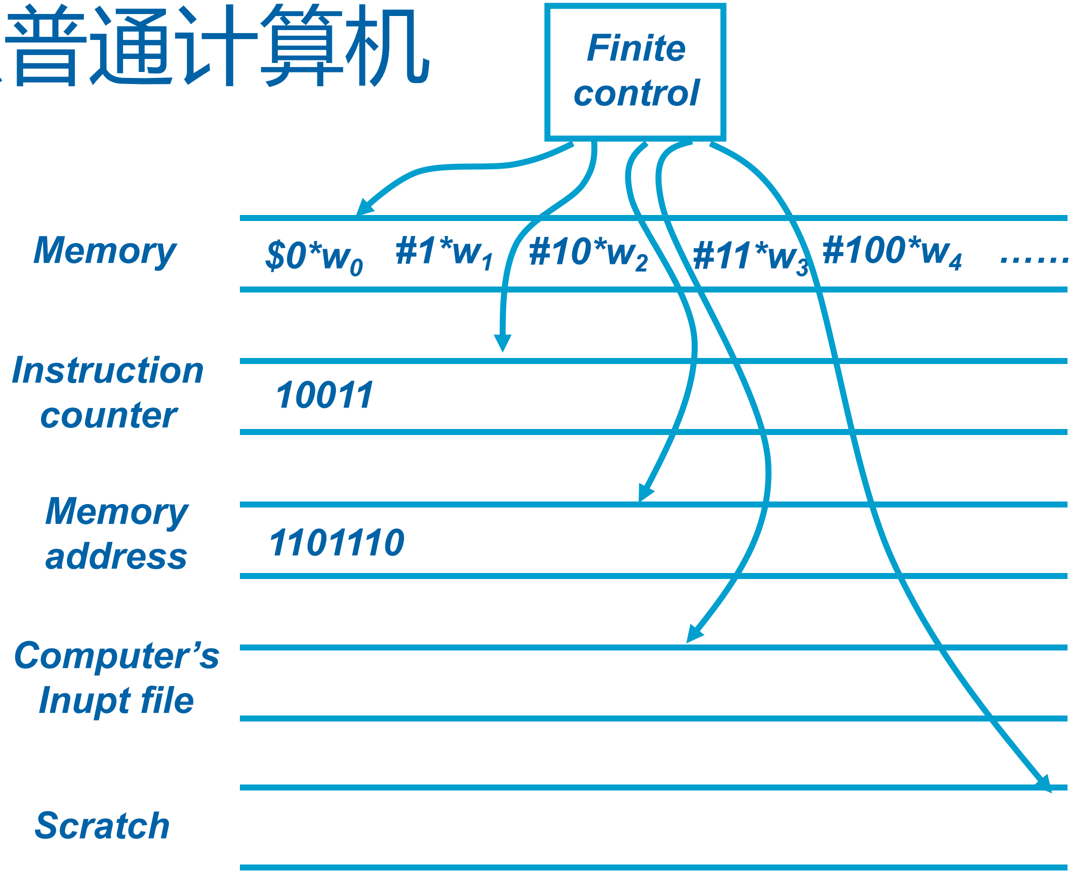
图灵机与计算机

多带图灵机可以模拟普通计算机

用多带图灵机模拟典型的存储程序式计算机。

参见右面示意图。

必要时，可增加更多的带。



课后作业

- 必做题：
  - P-350 Ex.8.4.2 (b)
  - P-350 Ex.8.4.3 (b)
  - P-351 Ex.8.4.6, Ex.8.4.7
  - P-361 Ex.8.5.1 (b) (c)
- 思考题:
  - P-361 Ex.8.5.1 (d)

Thank you