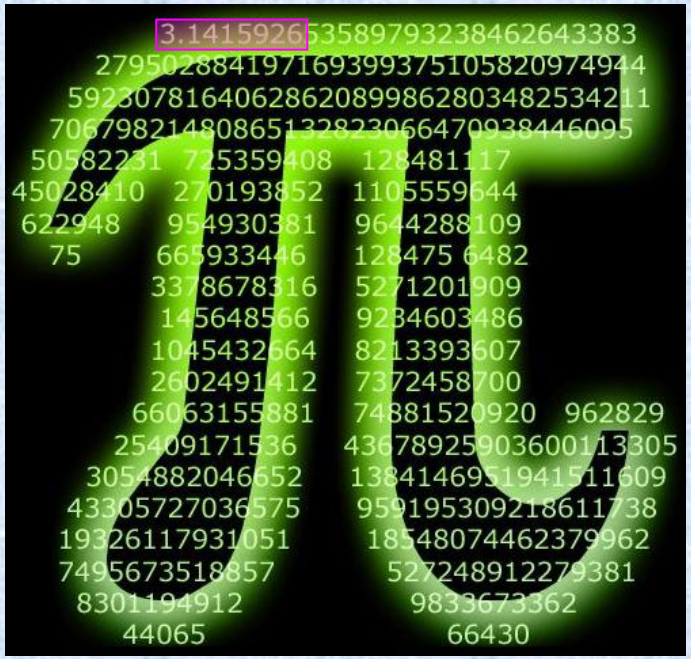
第二章
计算理论与计算模型

2.1 计算的几种视角

一、计数与计算

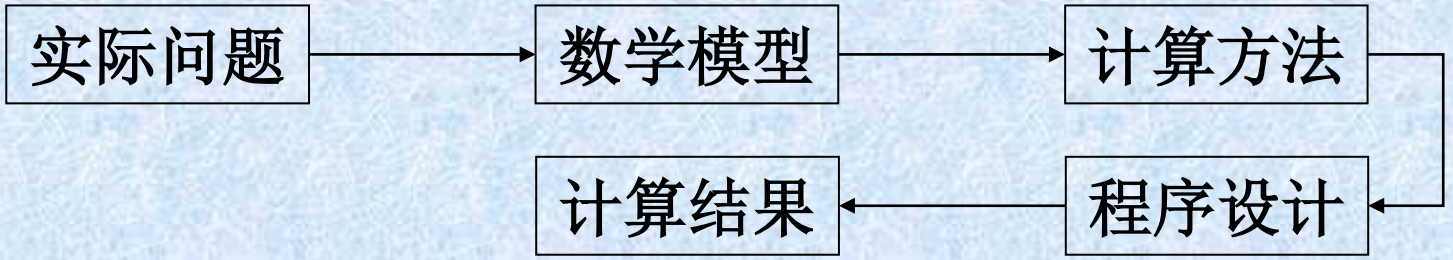
手指、石头、结绳计数，算筹计算



2.1 计算的几种视角

许多计算领域的**求解问题**，如计算物理学、计算力学、计算化学和计算经济学等都可以归结为数值计算问题，而**数值计算方法**是一门与计算机应用紧密结合的、实用性很强的数学课程。

科学计算的过程：



如对气象资料的汇总、加工并生成天气图像，其计算量大且时限性强，要求计算机能够进行高速运算，以便对天气做出短期或中期的预报。

2.1 计算的几种视角

二、逻辑与计算

逻辑学有三大源泉：①以亚里士多德的词项逻辑和斯多亚学派的命题逻辑为代表的古希腊逻辑。

②以先秦名辩学为代表的古中国逻辑。

③以正理论和因明学为代表的古印度逻辑。

逻辑是研究推理的学科，人们可以把推理看成是对符号的操作，即符号演算。

利用数学方法来研究推理的规律称为数理逻辑。为什么要研究数理逻辑呢？我们知道要使用计算机，就要有程序。

程序＝算法＋数据结构，而算法＝逻辑＋控制

2.1 计算的几种视角

三、算法与计算

从不同角度看，算法的定义有多种：

从哲学角度看：算法是解决一个问题的抽象行为序列。

从抽象层次看：算法是一个将输入转化为输出的计算步骤序列

从技术层面看：算法是接收输入并产生输出的计算过程。

简而言之，算法就是计算的办法或法则。

算法无处不在，每个人每天都在使用不同的算法来活出自己的人生。比如你去食堂买饭会选择一个较短的队列，而有人则可能选择一个推进速度更快的队列。

2.1 计算的几种视角

算法：为解决一个特定的问题所采取确定的有限步骤。

计算机用于解决数值计算，如科学计算中的数值积分、解线性方程等计算方法，就是数值计算的算法。

计算机用于解决非数值计算，如用于管理、文字处理、图像图形等的排序、分类和查找，就是非数值计算的算法。

算法的组成：操作、数据。

这些操作包括加、减、乘、除和判断等，并按顺序、分支、循环等控制结构所规定的次序执行。

数据是指操作对象和操作结果，包括布尔值、字符、整数和实数等；以及向量、记录、集合、树和图以及声音等。

为什么学习算法：①算法是计算机的灵魂；②算法是数学机械化的一部分，能够帮助我们解决复杂的计算问题；③算法作为一种思想，能锻炼我们的思维，使思维变得更清晰、更有逻辑。

2.2 计算理论

计算理论：关于计算和计算机械的数学理论，它研究计算的过程与功效。

计算理论主要包括算法、算法学、计算复杂性理论、可计算性理论、自动机理论和形式语言理论等等。

2.2 计算理论

一、计算与问题求解

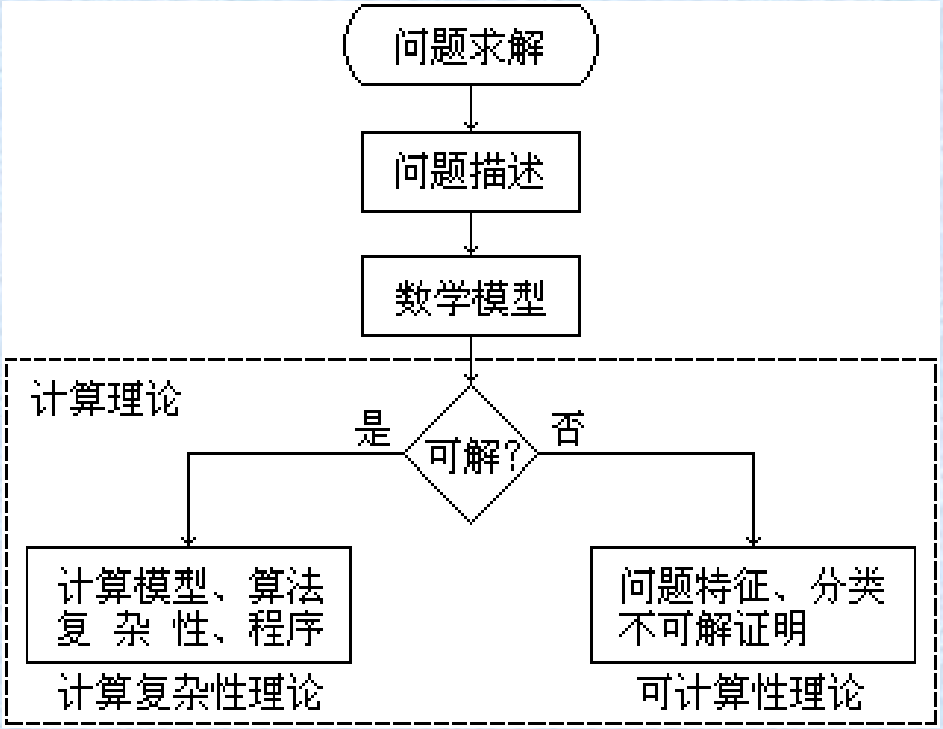
许多重大的科技问题：无法求得理论解
难以应用实验手段
可以进行计算模拟

计算是依据一定的法则对有关符号串的变换过程。抽象地说，计算的本质就是递归。

直观描述：计算是从已知符号开始，一步一步地改变符号串，经过有限步骤，最终得到一个满足预定条件的符号串的过程。这样一种有限的符号串变换过程与递归过程是等价的。

2.2 计算理论

问题求解：可能找到不同的方法或算法，是否可以计算仍取决于算法的存在性和计算的复杂性，即取决于是否存在可求解的算法。



2.2 计算理论

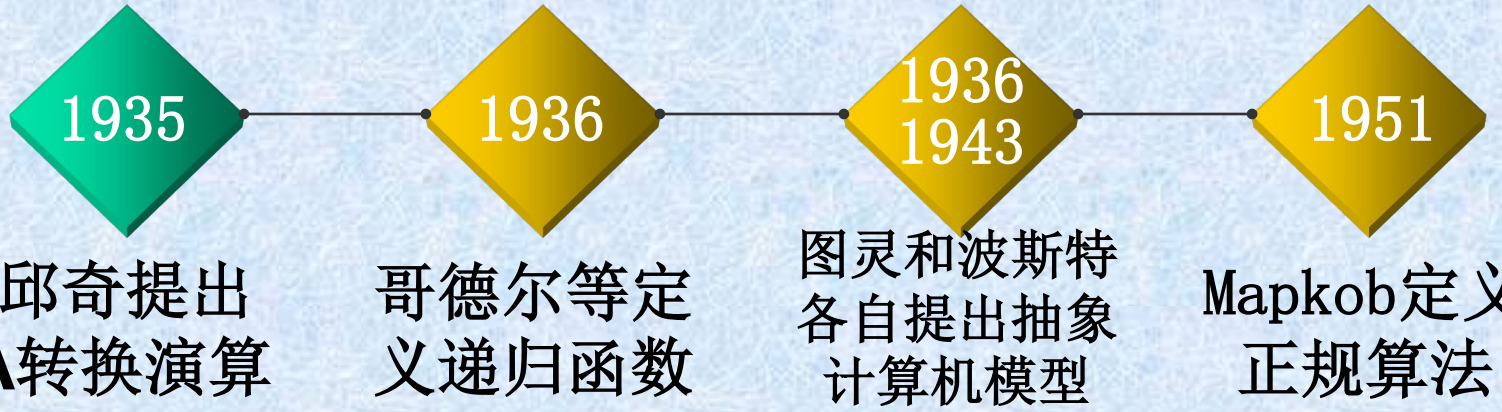
二、可计算性理论

- **可计算性理论**：研究计算的一般性质的数学理论。
- **可计算理论的中心课题**：将算法这一直观概念精确化，建立计算的数学模型，研究哪些是**可计算**的，哪些是**不可计算**的，以此揭示计算的实质。
- 由于计算与算法联系在一起，因此，可计算性理论又称**算法理论**。

2.2 计算理论

1.可计算理论的发展

可计算理论起源于对数学基础问题的研究。从20世纪30年代开始，为了讨论所有问题是否都有求解的算法，数学家和逻辑学家从不同角度提出了几种不同的算法概念精确化定义。



陆续证明，上述这些不同计算模型(算法精确化定义模式)的计算能力都是一样的，即它们是等价的。

2.2 计算理论

2.可计算性的定义和特性

- 可计算性的定义应算是一个哲学定义。
- 如果存在一个机械的过程，对给定的一个输入，能在有限步内给出答案，那么这个问题是可计算性的。
- 定义：凡可用某种程序设计语言描述的问题都是可计算性问题。
- 特性：确定性、有限性、机械性、可执行性、终止性。

2.2 计算理论

2.可计算性的定义和特性

- 图灵给出的可计算性定义：能够在图灵机上执行的过程（通常又称算法的过程）。
- 图灵之所以能取得成功，是他采用了算法思维来研究计算的过程，从而揭示可计算性的概念。
- 算法思维与目前在计算机上运行的程序之间有着密切的关系，从而使他的理论受到重视并被广泛使用。

2.2 计算理论

3.可计算性理论的主要内容

图灵机：用于精确描述算法的特征。可用一个图灵机来计算其值的函数是可计算函数，找不到图灵机来计算其值的函数是不可计算函数。

λ演算：引进λ记号以明确区分函数和函数值，并把函数值的计算归结为按照一定规则进行一系列转换，最后得到函数值。

丘奇-图灵论题：λ可定义函数类与直观可计算函数类相同，图灵机可计算函数类与直观可计算函数类相同。

原始递归函数：定义少量直观可计算的函数为原始递归函数，原始递归函数的合成仍是原始递归函数。

2.2 计算理论

4.可计算理论的意义

- 可计算性理论的基本思想、概念和方法被广泛应用于计算科学的各个领域。
- 数学模型的建立方法在科学、工程、技术领域中被广泛采用。
- 递归的思想被用于程序设计、数据结构和计算机体系结构。
- λ演算被用于研究程序设计语言的语义。

2.2 计算理论

4.可计算理论的意义

- 计算学科的一个基本结论是不可计算的函数要比可计算的函数多得多。
- 虽然许多问题是可判定的，但更多的问题是不可判定的。
- 例如：停机问题和波斯特对应问题都是不可判定的。

2.2 计算理论

三、停机问题

停机问题是目前逻辑数学的焦点和第三次数学危机的解决方案，它是重要的不可判定问题。

1936年，Turing发表“论可计算数及其在判定问题中的应用”论文中提出一般性停机问题的不可判定性，并用形式化方法证明其为一个不可计算问题。

一般性的停机问题：对于任意的图灵机和输入，是否存在一个算法，用于判定图灵机在接收初始输入后可达停机状态。若能找到这种算法，停机问题可解；否则不可解。

2.2 计算理论

通俗地说，停机问题就是判断任意一个程序是否在有限的时间内结束运行的问题。

例如：
```
main()
{  int i=1;
   while ( i<10 )
   {  i=i+1;
   }
   return;
}
```
程序可终止

又如：
```
main()
{  int i=1;
   while ( i>0 )
   {  i=i+1;
   }
   return;
}
```
程序死循环

程序简单时容易做出判断，当示例复杂时会遇到较大的困难，而在某些情况下则无法预测。

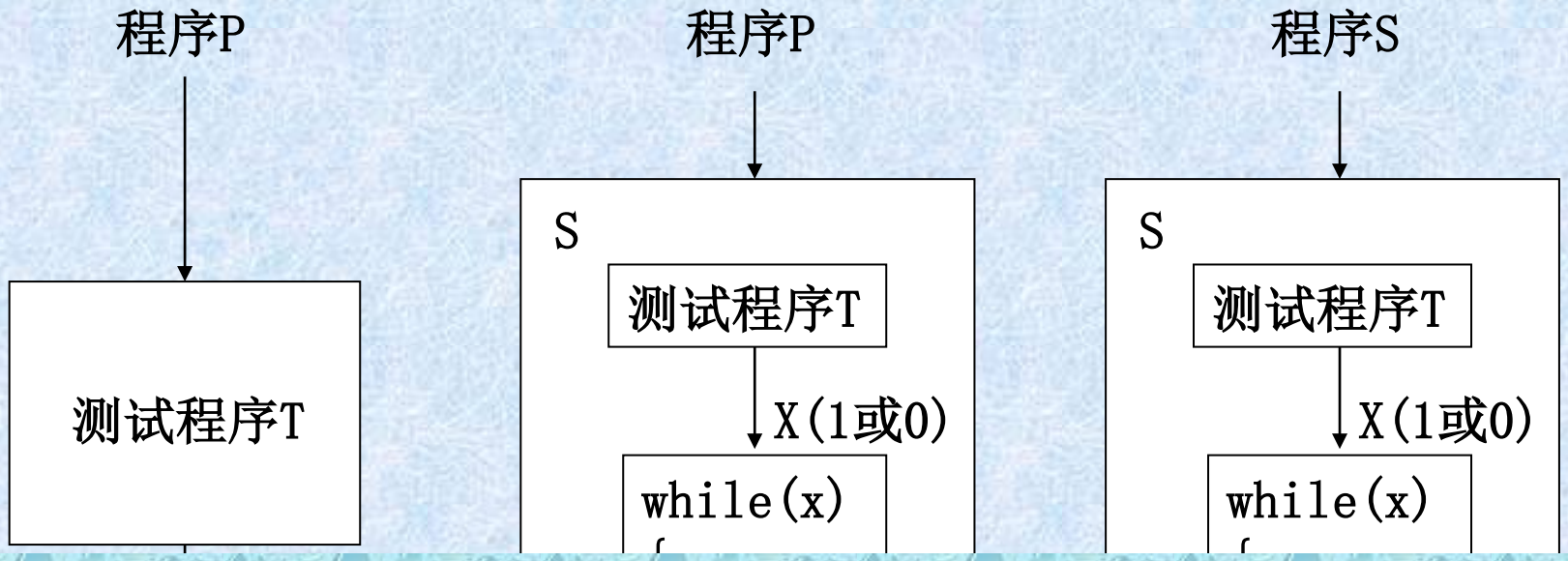
2.2 计算理论

停机问题的关键：能否找到一个测试程序，这个测试程序能判定任何一个程序在给定的输入下能否终止。

数学反证法证明：先假设存在这样的测试程序，然后再构造一个程序，该测试程序不能测试

假设存在一个测试程序T，它能接受任何输入。
当输入程序P能终止，输出1；
P不能终止，输出0。

2.2 计算理论



结论：若S终止，则S不终止；若S不终止，则S终止，结论矛盾

故可以确定这样的测试程序不存在，从而证明停机问题不可解

P能终止, X→1	P终止, X→1, S→不终止	S终止, X→1, S→不终止
P不终止, X→0	P不终止, X→0, S→终止	S不终止, X→0, S→终止

2.2 计算理论

[例题]理发师悖论。一个理发师的招牌：城里所有不自己刮脸的男人都由我给他们刮脸，我也只给这些人刮脸。

- 谁给这位理发师刮脸呢？如果他自己刮脸，那他就属于自己刮脸的那类人。但是，他的招牌说明他不给这类人刮脸，因此他不能自己来刮脸。
- 如果另外一个人来给他刮脸，那他就是不自己刮脸的人。但是，他的招牌说他要给所有这类人刮脸。因此，其他任何人也不能给他刮脸。
- 从本质上看，理发师问题和停机问题是一样的。

2.2 计算理论

四、计算复杂性理论

计算复杂性理论：用数学方法研究各类问题的计算复杂性的学科。

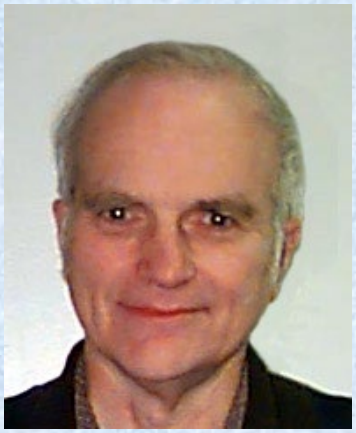
计算复杂性理论研究各种可计算问题在计算过程中资源(如时间、空间等)的耗费情况，以及在不同计算模型下，使用不同类型资源和不同数量的资源时，各类问题复杂性的本质特性和相互关系。

2.2 计算理论

1.计算复杂性的发展

1993年的图灵奖授予合作奠定了计算复杂性理论基础的两位学者J.Hartmanis和R.E.Stearns。

在此以前，已有M.O.Rabin、S.A.Cook、R.M.Karp等学者因在计算复杂性理论研究中做出先驱性工作而分别在1976、 1982和1985年获得图灵奖。Hartmanis和Stearns则在前人工作的基础上，比较完整地提出了计算复杂性的理论体系，并首次正式命名了计算复杂性(**computational complexity**)，因而被公认为计算复杂性理论的主要创始人。

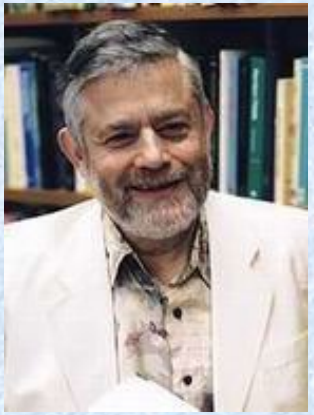


2.2 计算理论

1995年度的图灵奖授予加州大学伯克利分校的计算机科学家Manuel Blum，他是计算复杂性理论的主要奠基人之一。

Blum与前述两人互相独立地进行着相关问题的研究，并完成了他的博士论文：**A machine independent theory of the complexity of recursive functions**（与机器无关的递归函数复杂性理论），论文提出了有关计算复杂性的4个公理，被称为布卢姆公理系统。目前，可计算理论的绝大部分结果都可以从这个公理系统推导出来。

计算复杂性理论应用于计算机安全(密码学)、软件工程的程序正确验证，以及算法博弈论。



2.2 计算理论

2.计算复杂性

算法复杂性→针对特定算法

计算复杂性→针对特定问题，反映问题的固有难度

计算复杂性＝最佳的算法复杂性

计算复杂性：用计算机求解问题的难易程度。

度量标准：

①时间复杂度→计算所需的步数或指令条数；

②空间复杂度→计算所需的存储空间大小。

2.2 计算理论

假设一个问题有两种算法：

①算法复杂性是n^3（0.2s）

②算法复杂性是3^n（$4*10^{28}$s，1千万亿年）

（用每秒百万次的计算机，n=60）

如果一个问题没有多项式时间复杂性的算法，则被称为难解型问题。

复杂性函数	问题规模n			
	10	30	50	60
n	0.01ms	0.03ms	0.05ms	0.06ms
n^3	1ms	27ms	125ms	216ms
n^5	100ms	24.3s	5.2min	13min
2^n	1ms	17.9min	35.7年	366世纪

2.2 计算理论

3.P类问题和NP类问题

按复杂性把问题分成不同的类。

P类问题：由确定型图灵机在多项式时间内可解的一切判定问题所组成的集合。

P类问题包含了大量的已知自然问题，如计算最大公约数、计算π值、排序问题、二维匹配问题等。

NP类问题：由非确定型图灵机在多项式时间内可计算的判定问题所组成的集合。

也就是说，如果一个问题的潜在解答可以在多项式时间内被证实或证伪，则该问题属于NP。NP类问题数量巨大，如完全子图问题、图的着色问题、汉密尔顿回路问题、以及旅行销售员问题等。

2.2 计算理论

对于NP来说，一个常见的误解是人们认为NP问题不存在多项式时间解。这是否意味着P＝NP呢？或者说，P类集合是否与NP类问题集合完全重合呢？这个问题是21世纪数学界和计算机科学理论界面临的一个重大问题。

所有P类问题都是NP类问题：因为确定性图灵机能够解决的问题当然能够被非确定性图灵机解决。

是否所有NP问题都是P问题：凭直觉NP应该不属于P，因为非确定性图灵机比确定性图灵机强大得多，很难相信一个强大得多的机器所能够解决的问题都可以被一个功能更弱的机器解决！

必须拿出证据来说明NP不属于P。要证明这一点，只需证明某个NP问题不属于P即可。但遗憾的是，目前尚没有人证明NP不属于P。当然也没有人证明了NP属于P。也就是说，P与NP是否等价是一个既没有证实也没有证伪的问题！

2.2 计算理论

五、公钥密码学

- NP问题和NP完全问题与密码学关系最为密切。
- 密码学研究的主要内容之一是对不同的密码技术的计算复杂性进行比较，以便确定其安全性。
- 例如，给定的一个整数要求找到它的因数，如果这样的解存在，那么就要为这一问题找到有效的解。
- 对许多数学家来说，至今还没有找到一种有效的方法来确定大整数的因数。
- 密码学利用这种情况用来产生一种对报文进行加密和解密的方法，俗称RSA算法。

2.3 计算模型

计算模型是刻画计算的抽象的形式系统或数学系统。在计算科学中，计算模型是指具有状态转换特征，能够对所处理对象的数据或信息进行表示、加工、变换和输出的数学机器。

1936年，图灵发表“论可计算数及其在判定问题中的应用”论文，给“可计算性”下了严格的数学定义，并提出著名的图灵机(Turing Machine)的设想。

图灵机是一种十分简单但运算能力很强的理想计算装置，它描述了一种假想的可实现通用计算的机器。

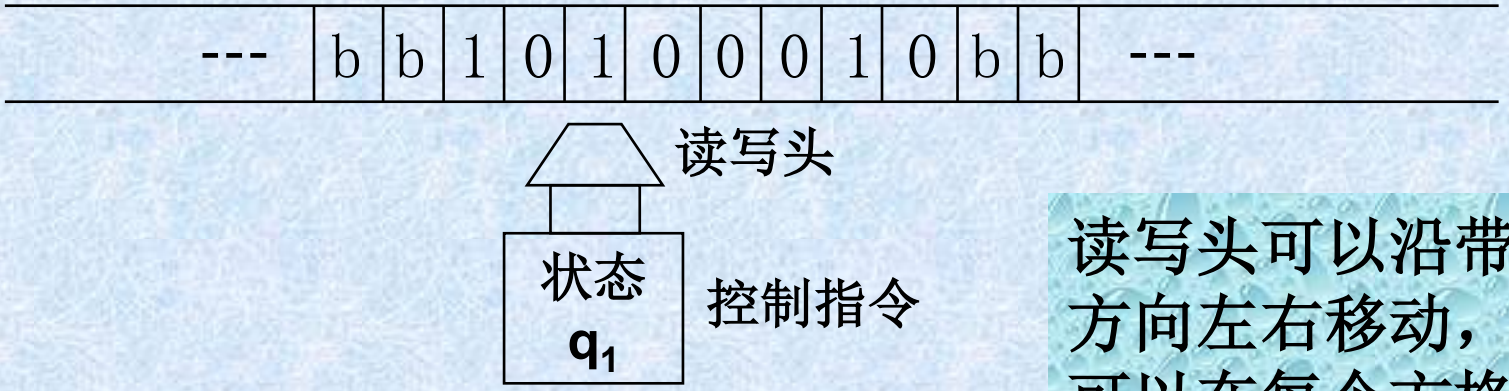
1912 –1954

2.3 计算模型

一、图灵机

1.直观描述

①图灵机的计算装置：一条两端可无限延长的带子，一个读写头，一组控制指令。



读写头可以沿带子方向左右移动，并可以在每个方格上进行读写。

2.3 计算模型

②带子上的符号为一个有穷字母表：

$\{S_0, S_1, S_2, \cdots, S_p\}$

通常仅有S_0、S_1两个字符，其中：

$S_0 \rightarrow 0$，$S_1 \rightarrow 1$

这可加深对布尔值、二进制机器的理解。

③机器的控制状态：

$\{q_1, q_2, \cdots, q_n\}$

图灵机的初始状态设为q_1,结束状态设为q_n

2.3 计算模型

④五元组指令集合：

$$(q_iS_jS_kR(LN)q_n)$$

q_i--机器目前所处的状态
S_j--机器从方格中读入的符号
S_k--机器用来代替S_j写入方格的符号
R,L,N--右移一格,左移一格,不移动
q_n--下一步机器的状态

一个给定机器的程序是机器内的五元组形式的指令集，它定义了机器在特定状态下读入一个特定字符时所采取的动作。

2.3 计算模型

2.工作原理

机器从给定带子上的某起点出发，其动作完全由其初始状态值及机内五元组指令集来决定。计算结果是从机器停止时带子上的信息得到。

指令死循环：$q_1S_2S_2Rq_3$

$q_3S_3S_3Lq_1$

指令二义性：$q_3S_2S_2Rq_4$

$q_3S_2S_4Lq_6$

2.3 计算模型

3.应用实例

[例]假设：b表示空格

q_1表示机器的初始状态

q_4表示机器的结束状态

如果带子上的输入信息为10100010，读写头位对准最右边第一个为0的方格，且状态为q_1。

按照以下五元组指令集执行后，输出正确的计算结果是什么?

2.3 计算模型

指令集

$q_1 01 L q_2$

$q_1 10 L q_3$

$q_1 bb N q_4$

$q_2 00 L q_2$

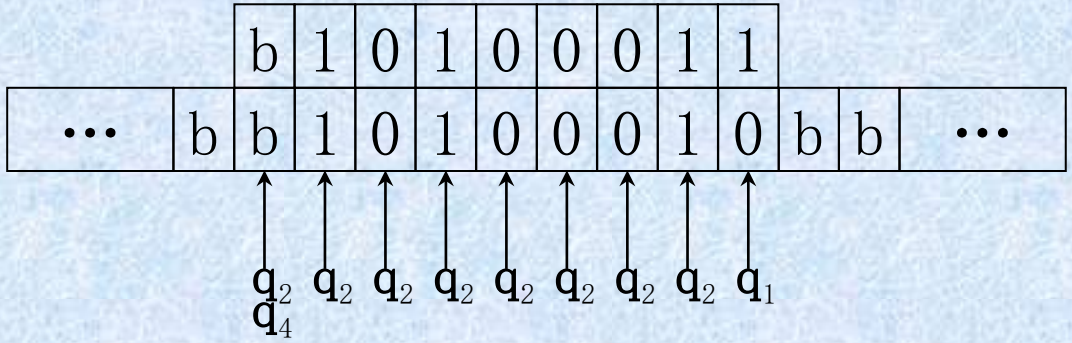
$q_2 11 L q_2$

$q_2 bb N q_4$

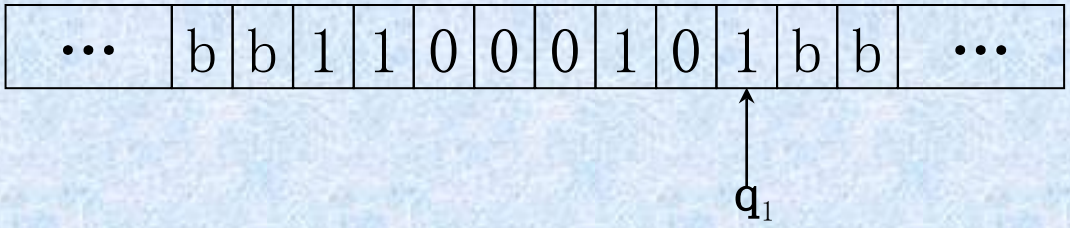
$q_3 01 L q_2$

$q_3 10 L q_3$

$q_3 bb N q_4$



计算函数是：$S(x)=x+1$



2.3 计算模型

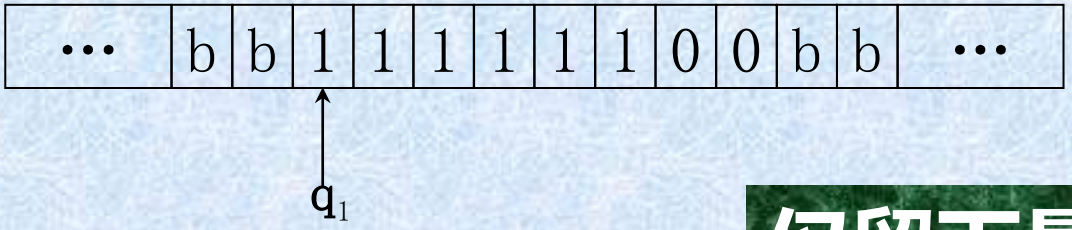
[例]图灵机Mz：其中Q={q_1,q_2,q_f}

五元组指令集为：$q_1$10Rq_1

$q_1$00Lq_2

$q_2$01Nq_f

求Mz对任何一串“1”的作用是什么?



仅留下最后一个“1”

2.3 计算模型

图灵机：$S(x)=x+1$ 后继函数

$N(x)=0$　零函数

$U_i^{(n)}=x_i$　投影函数

任何原始递归函数都是从这3个初始递归函数经有限次的复合、递归和极小化操作得到。

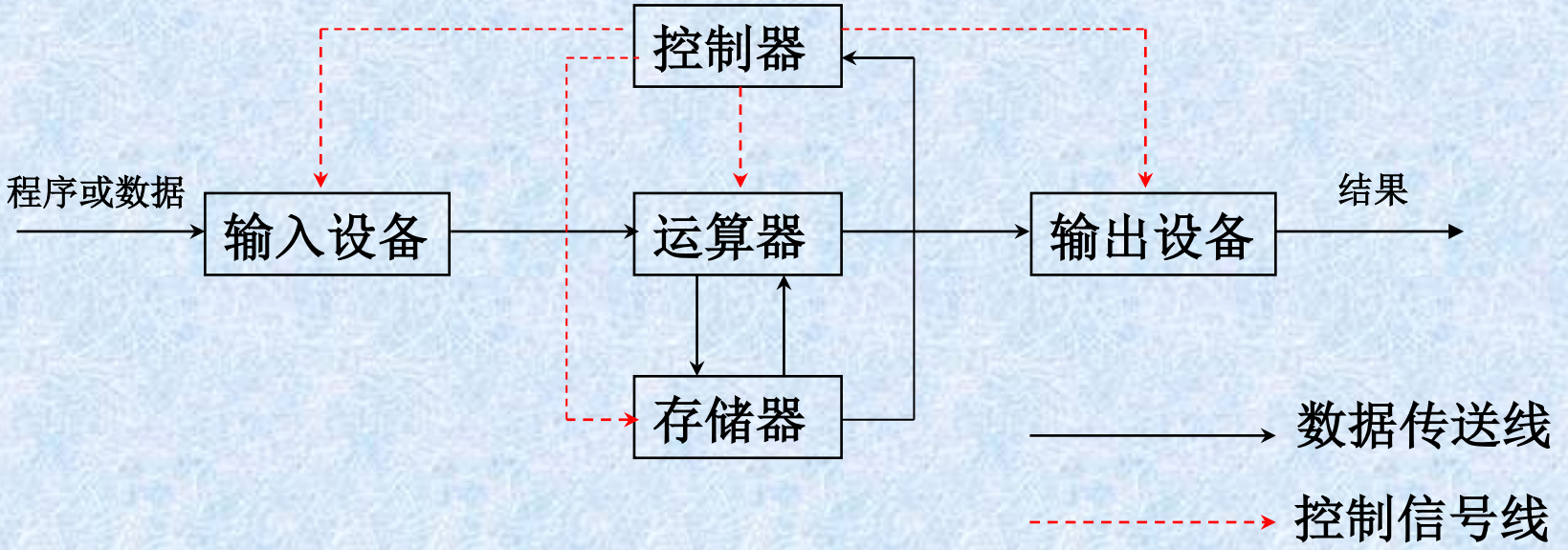
非确定性图灵机与确定性图灵机的区别是：在给定状态和输入时，其行为将不是唯一确定的。也就是说，对应同一个状态和输入，非确定性图灵机可以有多种行为来选择。

2.3 计算模型

二、冯·诺依曼机

重要思想：存储程序、二进制

1.冯·诺依曼机模型



2.3 计算模型

运算器：对数据进行加工处理的部件。

在控制器的操纵下，它与内存交换数据，负责算术运算、逻辑运算和移位运算等。

控制器：负责对指令进行分析和判断，发出控制信号，使计算机各部件协调工作，确保系统的自动运行。

运算器 + 控制器 = 中央处理单元(CPU)

2.3 计算模型

存储器：存放大量程序和数据的部件，其分类是内存储器和外存储器。

输入设备：用来接受用户输入的原始数据和程序，并将它们转变为计算机能够识别的形式存放在内存中，如键盘、鼠标、扫描仪等。

输出设备：将计算机处理的信息以人们所能接受的形式表示出来，如显示器、打印机等

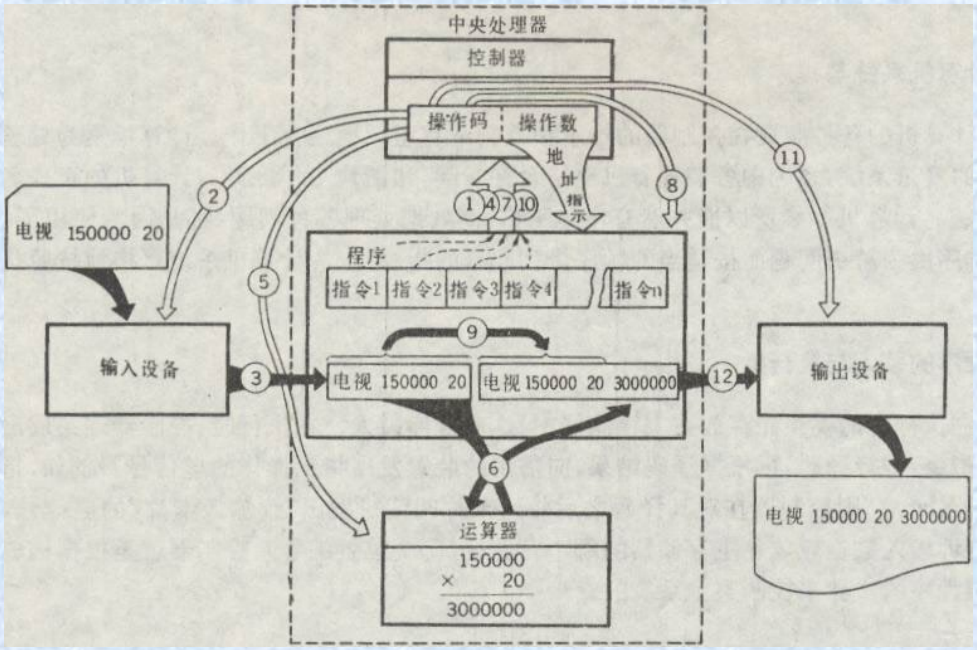
运算器＋控制器＋内存储器→主机

输入设备、输出设备、外存储器→外部设备

2.3 计算模型

2.冯·诺依曼机工作原理

先将程序(一组指令)和数据存入计算机，启动程序就能按照程序指定的逻辑顺序把指令读取并逐条执行，自动完成指令规定的操作。



2.3 计算模型

3.冯·诺依曼机的特点

①机器以运算器为中心，输入、输出设备与存储器之间的数据传送都要经过运算器。

②采用存储程序原理。

③指令是由操作码和地址码组成。

④数据以二进制表示，并采用二进制运算。

⑤硬件与软件完全分开，硬件在结构和功能上是不变的，完全靠编制软件来适应用户需要。

2.3 计算模型

4.冯·诺依曼机结构的局限性

冯·诺依曼瓶颈：存储器与中央处理单元之间的通路太狭窄，每次执行一条指令，所需的指令和数据都必须经过这条通路。

从本质上讲，冯·诺依曼机是采取串行顺序处理的工作机制，即使有关数据已经准备好，也必须逐条执行指令序列，而提高计算机性能的根本方向之一是并行处理。

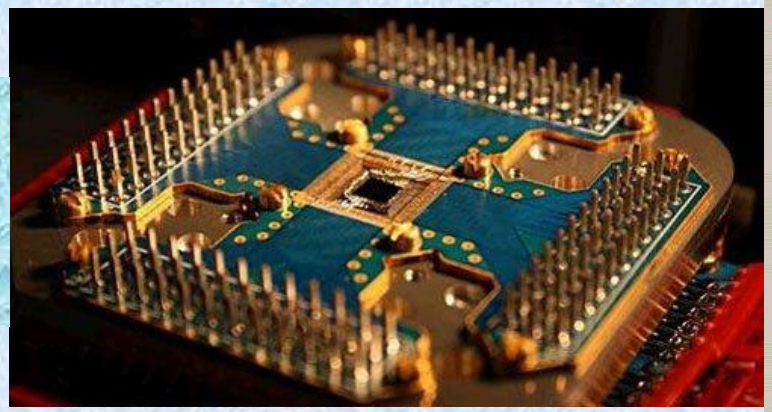
2.3 计算模型

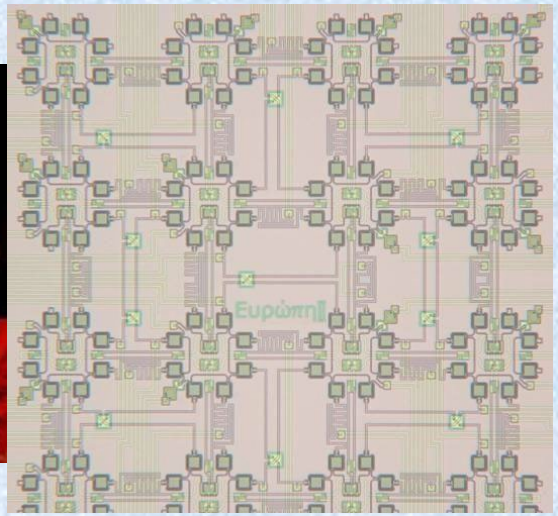
三、量子计算机

量子计算机：一类遵循量子力学规律进行高速数学和逻辑运算、存储及处理量子信息的物理装置。量子计算机处理和计算的是量子信息，运行的是量子算法。

由于量子态具有相干叠加性质，特别是量子纠缠特性，使得量子计算机具有天然的大规模并行计算的能力，其并行规模随芯片上集成量子位数目指数增加。

承载16个量子位的硅芯片

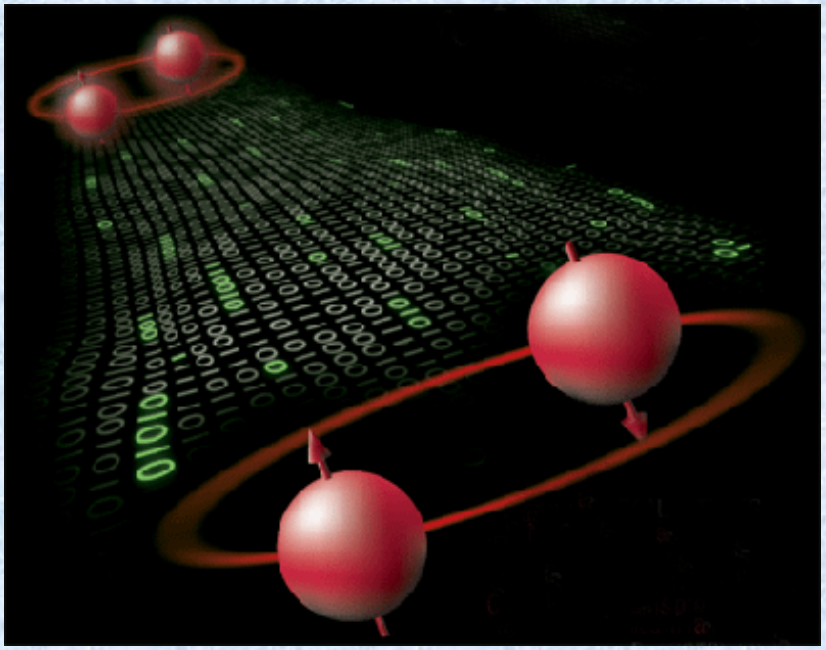




2.3 计算模型

基本原理：量子计算机以量子态为记忆单元、开关电路和信息存储形式，以量子动力学演化为信息传递与量子通信，其硬件的各种元件的尺寸达到原子或分子的量级。

量子计算的信息存储单位是量子比特，其两态的表示常用以下两种方式：①利用电子自旋方向。如向左自转状态代表1，向右自转状态代表0。②利用原子不同能级。原子有基态和激发态两种能级，规定原子基态为0，激发态为1。



2.3 计算模型

量子算法：Shor大数质因子分解算法

Grover数据库搜索算法

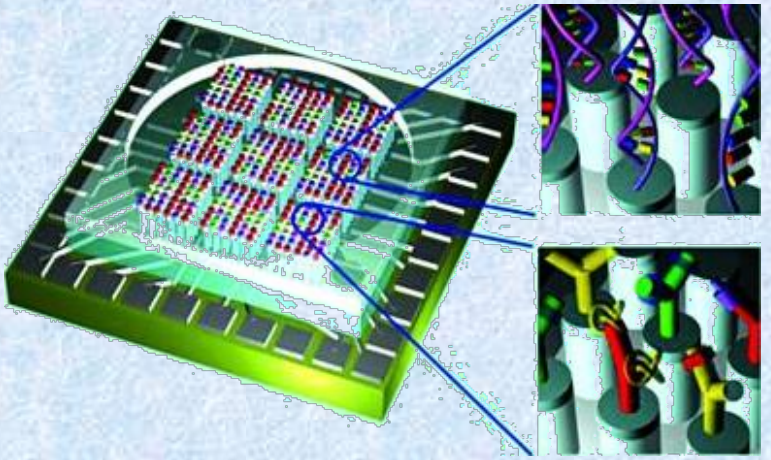
量子智能算法

2.3 计算模型

四、生物计算机

生物计算机是指用生物芯片制成的计算机，这种生物芯片是由蛋白质和其他有机物质的分子组成，它是以分子电子学为基础研制的一种新型计算机。

生物芯片又称DNA芯片，其结构大致是每个芯片的基质面上都可划分出数百甚至数百万个小区，在指定的小区内可固定大量具有特定功能、长约20个碱基组成的核酸分子，也叫分子探针。这一技术所派生出来的蛋白质芯片是生物计算机的基本结构单元。



2.3 计算模型

主要特点：①强大的记忆功能
②运算速度快
③能耗低
④具有自愈特性
⑤具有模仿人脑的思考机制
⑥具有超高密度

研究方向：①研制分子计算机，即制造有机分子元件去替代半导体逻辑元件和存储元件；
②深入研究人脑结构和思维规律，再构想生物计算机的结构。

本章小结

- 从计数、逻辑、算法角度，看计算问题
- 专业术语：计算理论，计算，计算过程
- 可计算理论
  定义，特性，主要内容(图灵机、λ转换演算、丘奇-图灵论题、原始递归函数)，意义
- 停机问题(不可判定性、理发师悖论)
- 计算复杂性理论
  算法/计算复杂性(时间、空间)，P类问题，NP问题
- 计算模型
  图灵机，冯·诺依曼机，量子计算机，生物计算机