“计算”这个概念

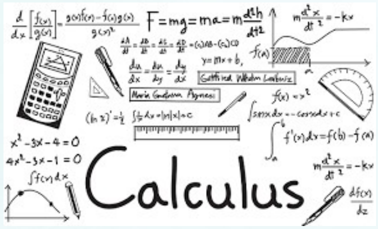
历史，定义与局限

毛昕渝
2021/10/10

第4周、第5周课程预告

CS1950：计算机科学的伟大思想

- 你或许会想：
  - 当我做数学分析作业的时候，我在做计算
  - 我面前的电脑（中的CPU）在计算
  - ……



VS.

- 想想下面这些问题：
  - 你体内的细胞、你的大脑是在“做计算”吗？
    - 你和一台计算机有什么区别？
    - 人的心智胜过机器吗？
  - 计算机无所不能吗？
    - 有没有什么是**不可计算**的？

计算机无所不能的世界……

莱布尼兹的梦想

"This is the best of all possible worlds."

Gottfried Wilhelm Leibniz

普遍文字（Universal Characteristic）

莱布尼兹
1646--1716

- 发明微积分：符号的重要性
- 在给 G.F.A.L’Hospital 的信中：代数“部分的秘密就在于文字，在于恰当地使用符号表达式的技艺”.
- “Calculus ratiocinator（推理演算）”
- 理性乐观主义

“如果产生了争议，哲学家们用不着像会计师一样相互争执，他们只需要掏出纸和笔，然后说：来，让我们算一下！”

I am convinced more and more of the utility and reality of this general science, and I see that very few people have understood its extent…

This characteristic consists of a certain script or language … that perfectly represents the relationships between our thoughts.

—— Leibniz wrote in his letter to Jean Galloys

后来者们

- 乔治·布尔（Gorge Boole）: 布尔代数
- 弗雷格：*Begriffsschrift(German for, roughly, "concept-script")* 第一个形式逻辑系统，基于朴素集合论
- 罗素悖论（理发师悖论）：第三次数学危机
  - "我只给那些不给自己刮胡子的人刮胡子"
- 希尔伯特计划（Hilbert's Program）：
  1. 形式化：用一种统一的严格形式化的语言来表达所有数学
  2. 完备性证明：证明数学是完备的
  3. 一致性证明：证明数学不会产生矛盾
  4. 可判定性：应该有一个算法能判定每个数学命题的真假

被哥德尔的不完备定理否证

"我不能被证明"

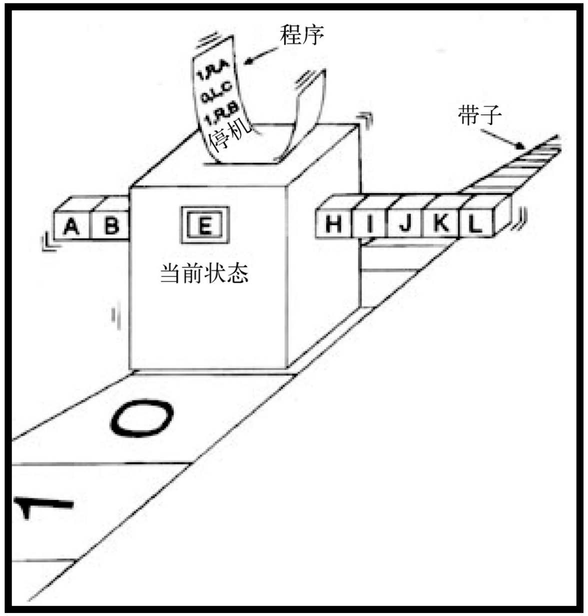
图灵机

*"A man provided with paper, pencil, and rubber, and subject to strict discipline, is in effect a **universal machine**."*

Alan Turing

图灵机：定义

- 一条无限长的纸带，被分成格子，每个格子上可以写一个符号，符号表 Σ 是有限的，例如 $\Sigma = \{0\,,1,\square\}$.
- 有限多个状态：$\{s_1, s_2, \ldots, s_k\}$, 其中有一个停机状态
- 有一个读写头
- 图灵机的工作方式：
  - 读取读写头所在格子的符号
  - 按照预先设定好的方式：
    1. 调整自己的状态
    2. 改变当前格子上的符号
    3. 决定左移一格，右移一格或者不移动



$\mathbb{M}(x)$ ≔ 输入为 x 时，$\mathbb{M}$ 停机时输出带上的内容.

图灵机的严格定义*

- 一台 单带图灵机 是一个三元组 (Σ, Q, δ)，其中
  - Σ是有限符号集，符号集至少包含 $0,1,\square,\triangleright$ 这四个符号；
  - Q 是有限状态集，状态集包含起始状态 q_{start} 和终止状态 q_{halt}；
  - $\delta: Q \times \Sigma \rightarrow Q \times \{\ll, \gg, \mathbf{STAY}\}$是迁移函数.
    - $\gg$ 表示读写头右移一格，$\ll$ 表示读写头左移一格， STAY表示读写头不动
- $\mathbb{M}(x) :=$ 输入为 x 时，$\mathbb{M}$ 停机时输出带上的内容。

问题的编码和判定问题

- 任何定义好的问题总是可以被编码成01字符串（记作$\{0,1\}^*$）
- 一个问题是一个函数 $f:\{0,1\}^* \to \{0,1\}^*$.
  - 如果$\mathbb{M}(x) = f(x), \forall x \in \{0,1\}^*$, 则称 $\mathbb{M}$ 计算了f.
  - 如果 f 的取值只有 0 和 1，则称f为判定问题.
- $\{0,1\}^*$ 的一个子集L 称为一个 语言.
  - 语言和判定问题一一对应.
  - 如果$\mathbb{M}$判定了$\mathbf{1}_L$, 则称 $\mathbb{M}$ 接受（或判定）了语言 L

Indicator function $\mathbf{1}_L(x) := \begin{cases} 1, \text{if } x \in L; \\ 0, \text{if } x \notin L. \end{cases}$

1 + 2 = 3

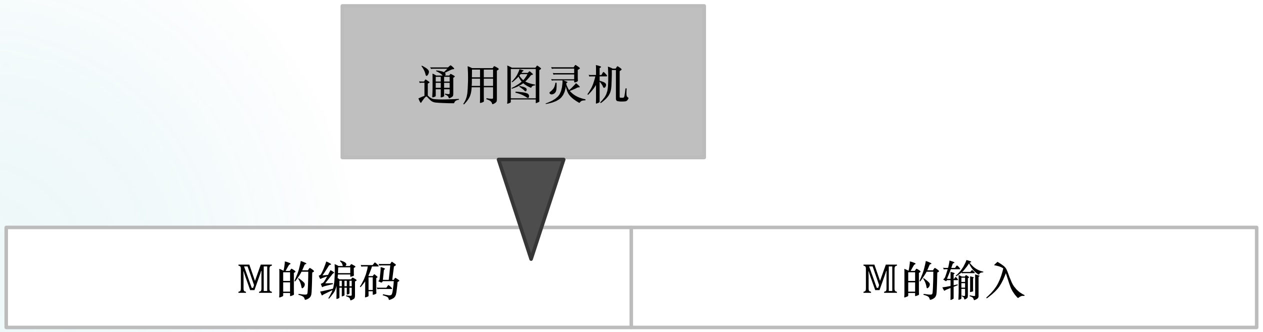
1#10#11

11101100101111

$0 \to 00$
$1 \to 11$
$\# \to 10$

图灵机的编码与通用图灵机

- 我们固定一个符号表，例如 $\Sigma = \{0,1,\square\}$.
- 一个图灵机可以在一张纸上写得清清楚楚，因此图灵机可以也被编码
- $\mathbb{M}_0, \mathbb{M}_1, \ldots$（规定：如果 i 不是合法编码，则 $\mathbb{M}_i := \mathbb{M}_0$）



枚举定理（UTM Theorem）：通用图灵机存在.

图灵超越巴贝奇的地方：
数据和程序在通用图灵机这里
被统一起来

停机问题与*Entscheidungsproblem*

- 停机问题 $\mathtt{HALT} \coloneqq \{\langle \alpha, x \rangle : \mathbb{M}_\alpha$ 在输入 x 时停机$\}$.
- *Entscheidungsproblem: German for 'decision problem'.*
  - 是否存在一个算法，输入一个一阶逻辑语句，判断其是否为真
  - 逻辑语句也可以被编码!
  - $\mathtt{TRUTH} \coloneqq \{[\phi] : \phi$为真$\}$ 是否可判定? （$[\phi]$ 表示 ϕ的编码）

定理. 图灵机的计算可以用一阶逻辑表达，因此：如果TRUTH可判定，则HALT可判定

图灵证明了：停机问题是不可判定的.

不可计算的“怪问题”：康托的对角线方法

- 停止集 $D_{\mathbb{M}} \coloneqq \{x: \ \mathbb{M} \text{ 在 } x \text{ 上停机}\}$.
- 康托的对角线方法
  - $|\mathbb{N}| \neq |\mathbb{R}|$.
  - Exercise: $|\mathcal{P}(A)| \neq |A|$ for any set A.
  - 存在不可计算的函数.
  - 存在不可判定的语言.

$$D^* \coloneqq \{x: \ \mathbb{M}_x \text{在 } x \text{ 上不停机}\}.$$

- D^*不是任何图灵机的停止集.
- 假设D^* 被图灵机$\mathbb{T}$判定，则可以利用$\mathbb{T}$ 定义另一台图灵机$\mathbb{S}$，使得 $\mathbb{S}$ 的停止集正好是D^* ，矛盾.
- D^* 不可判定.

ON COMPUTABLE NUMBERS, WITH AN APPLICATION TO THE ENTSCHEIDUNGSPROBLEM

By A. M. Turing.

[Received 28 May, 1936.—Read 12 November, 1936.]

The "computable" numbers may be described briefly as the real numbers whose expressions as a decimal are calculable by finite means. Although the subject of this paper is ostensibly the computable *numbers*, it is almost equally easy to define and investigate computable functions of an integral variable or a real or computable variable, computable predicates, and so forth. The fundamental problems involved are, however, the same in each case, and I have chosen the computable numbers for explicit treatment as involving the least cumbrous technique. I hope shortly to give an account of the relations of the computable numbers, functions, and so forth to one another. This will include a development of the theory of functions of a real variable expressed in terms of computable numbers. According to my definition, a number is computable if its decimal can be written down by a machine.

1936年，图灵关于*Entscheidungsproblem* 的论文

停机问题不可判定

$D^* := \{x: \ \mathbb{M}_x$在 x 上不停机$\}$. 我们证明了:

引理 1. D^* 不可判定

引理 2. 如果HALT 可判定，则D^* 可判定.

归约：如果算法 P 能判定问题A，那么有算法 Q 可以判定问题B，Q可以调用P.

- 根据 D^*的定义，判定 D^* 可以归约到停机问题:

$$x \in D^* \Leftrightarrow \langle \mathbb{M}_x, x \rangle \in \mathtt{HALT}.$$

定理. HALT 不可判定. 因此，TRUTH 不可判定，即：不存在一个算法可以判定数学定理的真假。

停机问题不可判定的另一种证明*

- 假设图灵机$\mathbb{H}$ 判定了HALT，这就是说：
$$\mathbb{H}(\alpha, x) = 1 \text{ 当且仅当 } (\alpha, x) \in \mathtt{HALT}.$$
- 考虑如下的图灵机$\mathbb{T}$：
  - 当输入α时， $\mathbb{T}$ 计算 FLAG $= \mathbb{H}(\alpha, \alpha)$;
  - 如果FLAG $= 1$,则死循环；如果FLAG $= 0$，则停机.
- 设$\mathbb{T}$的编码为β,根据定义
$$\mathbb{T}(\beta)\text{停机 当且仅当 } \mathbb{H}(\beta, \beta) = 0 \text{ 当且仅当 } \mathbb{T}(\beta)\text{不停机}$$

矛盾。

- 因此，不存在能够判定HALT的图灵机$\mathbb{H}$.

其他不可判定问题

- 希尔伯特的第十个问题
  - 丢番图方程(Diophantine Equation)是指有一个或者几个变量的整系数多项式方程，而且它们的求解仅仅在整数范围内进行.
  - 例子：$3x^2 + 9y^3 = 12, x \in \mathbb{Z}, y \in \mathbb{Z}$.
  - 费马大定理：某一类丢番图方程无解.
  - 是否存在解任意丢番图方程的算法?
  - Matiyasevich 于1970年证明了否定的结果. [Martin Davis, "Hilbert's Tenth Problem is Unsolvable," *American Mathematical Monthly*, vol.80(1973), pp. 233–269; reprinted as an appendix in Martin Davis, *Computability and Unsolvability*, Dover reprint 1982.]

Kolmogorov Complexity

- 一个二进制串有多“随机”？
  - 11111111111111111111111111111111111111111111
  - 10100100010000100000100000010000000100000000
  - 0110101001010001010001…1(总共19260817位)
  - 01010010110101001010101010100010101001001011
- Motivation: 越随机越难被描述清楚
- Kolmogorov Complexity: 最短描述的长度
  - “描述”的一种解释：打印这个二进制串的C++程序.
- $K(x)$不可计算：“最小的不能被少于二十汉字描述的自然数”.

Andrey Kolmogorov
苏联数学家

归约

- “如果能解决问题A， 那么就能解决问题B”

定义1. 设有两个语言L, L'. 如果存在可计算的函数f 满足

$$x \in L \text{ 当且仅当 } f(x) \in L', \forall x \in \{0,1\}^*$$

我们说L可以归约到L'.

C++的函数调用

定义2. 假设有两个语言L, L', 如果存在 oracle-图灵机$\mathbb{M}^{(\cdot)}$ 满足

$$\mathbb{M}^{L'} \text{判定 } L$$

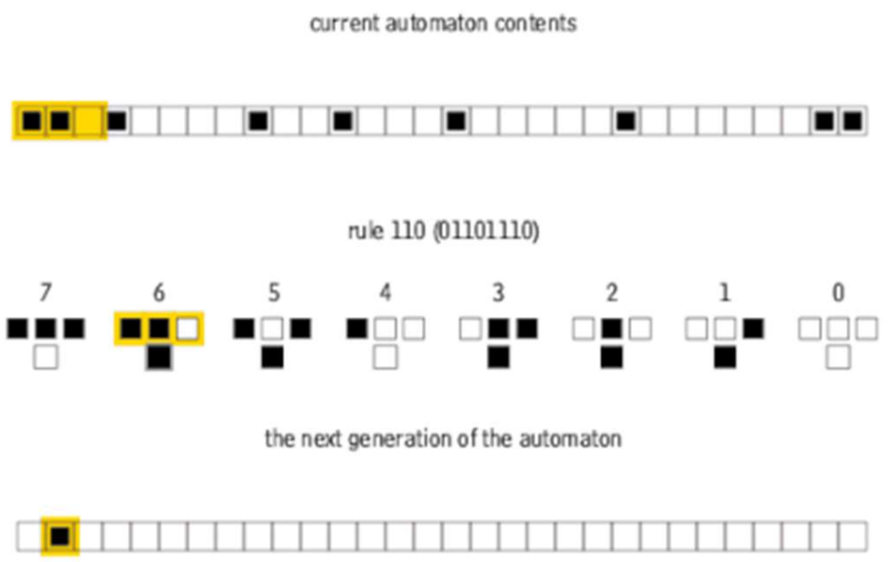
我们说L可以归约到L'.

计算：不止于此抑或仅仅如此

计算的其他定义

丘奇-图灵论题

元胞自动机



输入，输出

规则的编码

Rule 110 is universal.

"Computation is local."

> *Computation is the evolution process of some environment by a sequence of "simple, local" steps.*
>
> Avi Wigderson

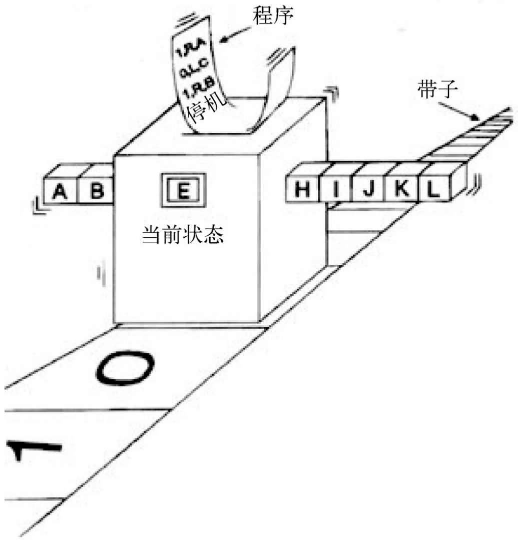
图灵机定义的鲁棒性

- 使用比$\Sigma = \{0\ , 1, \square\}$ 更多的符号能计算更多函数吗?
- 有≥ 2条纸带的图灵机能够计算更多函数吗?

RAM Machine: a sanity check

?

$\Leftrightarrow$



RAM Machine: a sanity check

- Von Neumann architecture.
- RAM计算机:
  - 内存单元：$M(1), M(2)$ ...
  - 有限个寄存器: $R_0, R_1, R_2, \ldots, R_n$.
  - Program counter(pc)
  - 指令集：GET/SET, ADD/MUL, LOAD/STORE, JUMP, BRANCH.
    - GET/SET: $R_0 \coloneqq n, \quad R_0 \coloneqq R_s, \quad R_s = R_0$.
    - ADD/MUL: $R_0 \coloneqq R_0 + n, \quad R_0 \coloneqq R_0 + R_s, \quad R_0 \coloneqq R_0 \times R_s$.
    - LOAD/STORE: $R_0 \coloneqq M(R_s), \quad M(R_s) \coloneqq R_0$.
    - JUMP/BRANCH: $pc \coloneqq n$; if $R_0 = 0$ then $pc \coloneqq n$; if $R_0 > 0$ then $pc \coloneqq n$.
- 用图灵机模拟RAM
  - 将RAM编码后写在纸带上：$\#1\$v_1\#2\$v_2\#3\$v_3\#4\$v_4 \cdots$
  - pc决定将符号解释为数据还是指令

丘奇-图灵论题（Church-Turing thesis）

丘奇-图灵论题. 任何物理上可实现的计算机都可以被图灵机模拟.

- 计算模型不胜枚举: λ-calculus, RAM Machine…
- 为什么是论题，而不是定理?
  - 直觉 vs. 定义
  - 道可道，非常道；名可名，非常名.
- 图灵机为什么是个好定义?
  - 显然"物理可实现"
  - 描述、研究图灵机很方便
  - "恰到好处的抽象"
- 可计算性理论（Theory of Computability）

心智与机器

> “或者人心胜过所有的机器，或者存在一些不能判定的数论问题。[不排除二者都真。]”
>
> 哥德尔

- 在每一个瞬间，心灵只能储存和感知有穷多款内容
  - 能从有穷大小的物理系统中复原出的信息总是有限的
- 心灵 = 大脑的物理运作？“物外无心？”
- 心灵的“状态”可能越来越多？心灵是不断发展的
  - 心灵的发展过程是不是可计算的？

Thanks for listening☺