第七章：可计算性，不可解性，不可判定性

沈榆平
yuping.shen.ilc@gmail.com

中山大学逻辑与认知研究所
2013年12月

算法

在1900及1928年世界数学家大会上，Hilbert分别提出了一些著名的问题,比如:

- 有一个机械过程可以判定任意多项式丢番图方程有整数解吗？(1900)
- 数学是完全的吗？即，是否任何数学命题都能被证明或者证否？（1928）
- 数学是一致的吗？即，是否从正确的证明过程不会得到错误的结论？(1928)
- 数学是可判定的吗？即，对任一个数学命题，是否有一机械过程可以处理并给出正确判断？(1928)

讨论

除了第三个问题答案为"是"，其它答案都为“否”。回答这些问题与算法(又称"能行过程,effective procedure")的概念紧密相关。那么，算法(能行过程)，是如何定义的？

算法

定义7.1

一个算法是一个计算过程的明确可行的指令集合，可用于解答给定问题类中的任一实例。

讨论

和集合的定义类似，算法的定义不是一个严格数学定义，而更接近一个哲学定义。我们可以注意到它的概念很难再进一步细分下去。比如，什么是过程？什么是明确可行？等等.

问题类例子

- $\{$f(n)=2n的值是什么？$|n \in \mathbb{N}\}$
- $\{$2是n的一个因子吗？$|n \in \mathbb{N}\}$
- $\{$n是一个素数吗?$|n \in \mathbb{N}\}$

算法

问题的算法

1. {f(n)=2n的值是什么？ $|n \in \mathbb{N}\}$
   对给出的n用乘法算出2n即可。
2. {2是n的一个因子吗？ $|n \in \mathbb{N}\}$
   将n连续除于2，如果最终余数是0，那么回答是，否则回答否。
3. {n是一个素数吗?$|n \in \mathbb{N}\}$
   辗转相除法....

为了更方便讨论，我们假设所有问题都有问题1的形式。

算法

如何描述算法？

- 用一种理论上精确定义的抽象的计算机器，如图灵机
- 描述计算过程的形式系统,如λ演算
- 给出函数类的形式构造，如递归函数
- 前两者是等价的。

Church论题(Church's Thesis)

所有可以由算法计算的函数，就是递归函数；所有递归函数，也存在计算的算法。

Church'Thesis不是定理，只是一个哲学论断。但绝大多数数学家都接受这一论断。我们接下来介绍一种描述算法的机器—图灵机。

图灵机

数学家、逻辑学家、计算机科学家阿兰图灵

Figure: Alan Mathison Turing (1912-1954)

图灵机

图灵机构造

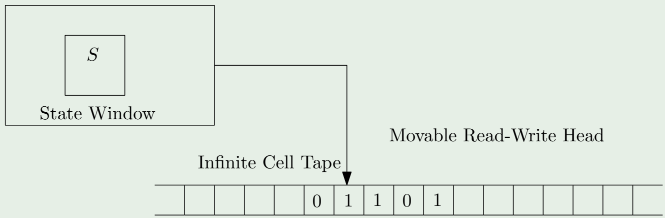


Figure: A Turing Machine

图灵机读写头操作

- 读/写一个单元格的字符；
- 向左/右移动一个单元格；

但如何决定读写头的操作呢？由图灵机的指令给出。

图灵机

一条指令告诉图灵机根据:

- 当前状态窗口中的字符(即当前状态)
- 与当前读取的字符

  来决定:

- 下一状态窗口中的字符(即下一状态)
- 读写头操作(左移/右移/写入字符)

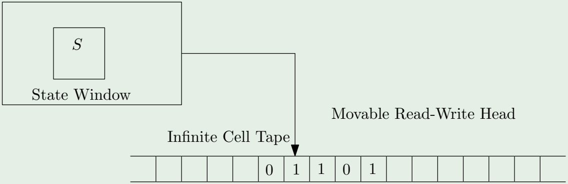
图灵机的指令

图灵机的一条指令是一个四元组(q, w, A, q'),其中:

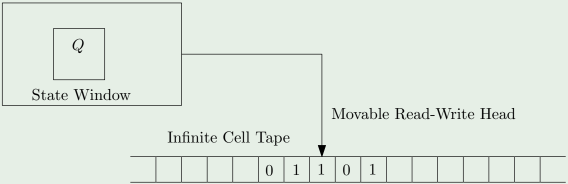
- q是当前状态
- w是当前字符
- $A \in \{L, R, w'\}$是一个操作，其中L/R表示读写头向左/右移，w'是一个被写入的字符
- q'是一个操作后的状态

图灵机

图灵机TM

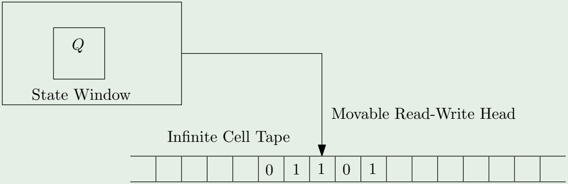


图灵机TM执行(S,1,R,Q)

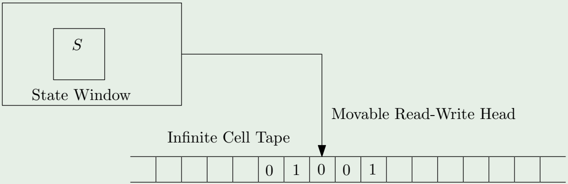


图灵机

图灵机TM



图灵机TM执行(Q,1,0,S)



图灵机

图灵机定义

一个图灵机是一个三元组$\langle\Sigma,\Omega,\Delta\rangle$,其中:

- Σ是一个有穷字符集合;
- Ω是一个有穷状态集合;
- Δ是一个有穷指令集合.

停机状态

特别的，我们可以指定某个状态，如S_{end}作为图灵机停机状态，也就是规定图灵机遇到此状态时就停止。此外，图灵机在没有指令可执行时，也停止。

图灵机

图灵机TM

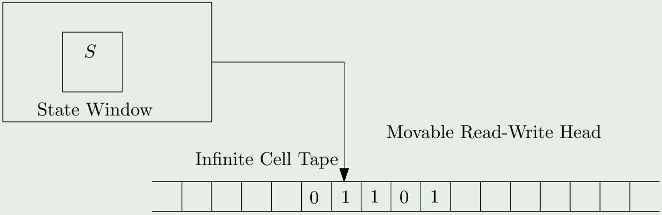


Figure: A Turing Machine

TM的形式定义

这前讨论的图灵机可以表示为$\langle\{0,1\},\{Q,S\},\{(Q,1,0,S),(S,1,R,Q)\}\rangle$。注意，纸带的输入并没有被包含在图灵机的定义中。纸带事实上是一个计算的输入。

图灵机

图灵机的歌德尔编码

既然图灵机也可以用符号定义给出，那么我们可以很自然地想到可以利用歌德尔编码的方法，对图灵机进行编码。事实上，每个图灵机都可以编码成一个自然数，而一个自然数(如果存在对应)也可以被解码成一个图灵机。从这个角度说，**程序(图灵机)就是数，数就是程序(图灵机)**。

命题7.20

所有图灵机(或其编码)是能行可枚举的。即，我们有方法把所有图灵机(理论上)排成一列：$T_0, T_1, T_2, \ldots$。

图灵机

讨论

- 所有图灵机都存在一个等价的只使用两个字符$\{0,1\}$的图灵机(0可以看成是空字符B)。
- 所有图灵机都存在一个等价的只使用两个状态$\{Q,S\}$的图灵机。
- 但并非所有图灵机都能转换成只有两个字符及两个状态的图灵机。

前二者的原因在于任意有穷的字符集都可以被二进制编码表示，但只有两个状态和两个字符的图灵机个数是有穷的，所以不足于表示所有图灵机(无穷可数多个)。

图灵机

图灵论题

所有算法可以计算的函数，都可以由图灵机计算,反之亦然。

图灵可计算函数

此处图灵可计算函数的含义是图灵机在计算这个函数时最终将停机。以下几个概念是等价的：

- 图灵可计算的
- 可计算的
- 递归的
- 可判定的
- 可解的

一个**判定问题**即答案为是或者否的问题。

图灵机

L的判定问题

对任意意L的公式A，问A是否为L的定理？。

L的可判定性

这个问题是可判定的，因为可以构造一个验证真值表的图灵机，如果公式$\mathscr{A}$是个重言式，它可以打印出1，反之打印0.

不可判定性

虽然图灵机能够计算很多问题，但也存在图灵机不能解决的问题(即根据实际情况回答是或者否)，即不可判定的问题。

图灵机

停机问题(Halting Problem)

任给一个图灵机T_m和一个输入n，它可以停机吗？(注意，停机意味着运算正常终止)

停机问题不可判定

这个问题是不可判定的。也就是说，任意给一个图灵机及输入，我们没有一个通用的办法确定它是否可以正常结束(停机)。

K的判定问题

任给一个$\mathscr{L}$的公式$\mathscr{A}$，问$\mathscr{A}$是否为K的定理？

K的不可判定性

这个问题是不可判定的，因为可以证明K可判定当且仅当停机问题可判定。

图灵机

不可判定性讨论

事实上，存在一些图灵机对可停机的图灵机及K的定理作出正确回答（是），但对不可停机的图灵机或者不是K的定理的公式则无法回答。因此，停机问题与系统K定理证明问题又称**半可判定的**，或者部分可判定的。

另一种理论计算机

我们已经知道图灵机刻画所有可计算的函数，但图灵机有时不够灵活，表示不够直观，我们介绍一种等价于图灵机的抽象机器，称为Unlimited Register Machine(无限制寄存机)。

URM机器

URM寄存器

3	1	5	0	1											
r_0	r_1	r_2	r_3	r_4	r_5	r_6									

URM直观上是一条一端无限延升的纸带，上面每个单元格可用于存储自然数，被称为寄存器。第一个寄存器(单元格)被标为r_0,第二个被标为r_1，等等。

URM机器

URM程序

URM程序是一个以下指令组成的序列:

Z(n)	将第n个单元格清零
S(n)	将第n个单元格值加1
T(m,n)	将第m个单元格的内容放到第n个单元格
J(m,n,q)	如果第m,n个单元格相同, 则转至第q条指令，否则执行下一指令

Instruction Examples

$$\begin{array}{|c|c|c|c|c}\hline 4 & 5 & 2 & 7 & \\ \hline\end{array} \Rightarrow Z(2) \Rightarrow \begin{array}{|c|c|c|c|c}\hline 4 & 0 & 2 & 7 & \\ \hline\end{array}$$

$$\begin{array}{|c|c|c|c|c}\hline 4 & 0 & 2 & 7 & \\ \hline\end{array} \Rightarrow S(3) \Rightarrow \begin{array}{|c|c|c|c|c}\hline 4 & 0 & 3 & 7 & \\ \hline\end{array}$$

$$\begin{array}{|c|c|c|c|c}\hline 4 & 0 & 3 & 7 & \\ \hline\end{array} \Rightarrow T(1,2) \Rightarrow \begin{array}{|c|c|c|c|c}\hline 4 & 4 & 3 & 7 & \\ \hline\end{array}$$

URM

URM程序的执行

- 除非有跳转，URM按顺序执行指令。
- 如果没有合适的指令可被执行，则停止运行。

URM程序例子

I_1.	J(1,2,6)
I_2.	S(2)
I_3.	S(3)
I_4.	J(1,2,6)
I_5.	J(1,1,2)
I_6.	T(3,1)

9	7	0	0	I_1
9	7	0	0	I_2
9	8	0	0	I_3
9	8	1	0	I_4
9	8	1	0	I_5

9	8	1	0	I_2
9	9	1	0	I_3
9	9	2	0	I_4
9	9	2	0	I_6
2	9	2	0	I_7

这个程序计算了减法$9-7=2$，为方便，我们在纸带后部标出下一步指令I_n。

URM

练习计算URM程序

I_1.	J(1,2,6)
I_2.	S(2)
I_3.	S(3)
I_4.	J(1,2,6)
I_5.	J(1,1,2)
I_6.	T(3,1)

2	3	0	0	0

结果：不停机！

URM与图灵机

URM程序计算sg函数

$$f(x) = \begin{cases} 0 & \text{if } x = 0, \\ 1 & \text{otherwise} \end{cases}$$

1. $J(1,2,4)$
2. $Z(1)$
3. $S(1)$

URM与图灵机

URM程序计算$x = y$特征函数

$$f(x,y) = \begin{cases} 0 & \text{if } x = y, \\ 1 & \text{otherwise} \end{cases}$$

1. $J(1,2,5)$
2. $Z(1)$
3. $S(1)$
4. $J(1,1,6)$
5. $Z(1)$

URM与图灵机

URM程序计算$x+y$

I_1.	J(3,2,5)
I_2.	S(1)
I_3.	S(3)
I_4.	J(1,1,1)

x+k	y	k	0	...

URM与图灵机

URM与图灵机等价

每一个URM机器都有一图灵机与之等价，反之亦然。

可计算函数

一个自然数域上的函数f称为可计算的(递归的)，如果存在一个UMR程序计算f (输出结果并正常停机).

URM与图灵机

作业：

URM程序计算$x \leq y$特征函数

$$f(x,y) = \begin{cases} 0 & \text{if } x \leq y, \\ 1 & \text{otherwise} \end{cases}$$

URM程序计算$x \dot{-} 1$

$$f(x) = \begin{cases} x-1 & \text{if } x > 0, \\ 0 & \text{if } x = 0 \end{cases}$$

URM程序计算$x/3$

$$f(x) = \begin{cases} x/3 & \text{if } x \text{ is a multiple of } 3\text{ ,} \\ \text{undefined} & \text{otherwise} \end{cases}$$