第二章 不可计算性

华中科技大学计算机科学与技术学院
jrc@hust.edu.cn

§1 胜奕机之不存在性

定理 不存在下国际象棋每局必胜的机器。

假设存在一个计算机系统A，它下国际象棋能击败任何对手。

于是存在一个被另一个人掌握的与A同样的计算机系统B，它下国际象棋能击败任何对手。

现在让A与B下国际象棋。有如下三种可能：

i> A击败B；ii>下成平局；iii>B击败 A；

在任何情况下与假设矛盾。

因此，根本不存在胜奕机。

§ 2 不可计算函数的存在性

考虑从N到N的全函数，即映射f：$N\to N$。

引理 记 $N^N=\{f\mid f: N\to N\}$，则N^N与N之间不存在一一对应的关系。

定理 N^N中至少有一个函数不是可计算的。

引理 记 $2^N=\{A\mid A\subseteq N\}$，则$2^N$与$N$之间不存在一一对应的关系。

定理 2^N中至少有一个集合不是r.e.集。

§3 停机问题的不可解性

因为A^*与N是可以建立一一对应的，可以认为图灵机的输入都是N中的非负整数。

定义 N的子集$K=\{i|i\in w_i\}=\{i|T_i(i)$停机$\}=\{i|\varphi_i(i)\downarrow\}$

引理1 K是r.e.集。

证

可构造一台接受集合K的机器(图灵机)M：对任意输入$i\in N$，

（1）将i解码为图灵机T_i;

（2）模拟T_i以i为输入的运行。

故K是r.e.集。

§3 停机问题的不可解性

定义 $\theta=\overline{K}=\{i \mid i \notin w_i\}=\{i \mid \varphi_i(i)\uparrow\}$

引理2 θ不是r.e.集，K不是可计算集。

证

用反证法。假设θ是r.e.集，则存在常数p使得$\theta=w_p$，则$p\in w_p$当且仅当$p\in\theta$当且仅当$p\notin w_p$，矛盾。

因为可计算集的补集一定是可计算集，可计算集一定是r.e.集。但K的补集θ不是r.e.集，故K不是可计算集。□

§ 3 停机问题的不可解性

定义 停机判定函数H: $N\times N \rightarrow \{0,1\}$

$$H(x,y)=\begin{cases}1, & \text{若}\mathrm{T}_x(y)\text{停机;}\\ 0, & \text{否则}\end{cases}$$

如果此函数可计算，是非常有用的。比如，可以用来证明哥德巴赫猜想：

“所有大于等于 4 的偶数都是两个素数之和。”

设以下程序对应的图灵机为T_g：

```
i=4;
while(i是两个素数之和) i+=2;
```

只需计算$H(g,0)$，若$H(g,0)=0$，哥德巴赫猜想成立，否则不成立。

§ 3 停机问题的不可解性

定理 $H(x,y)$不是可计算函数。即停机问题不可计算。

证法1

$\chi_K(x)=1$ 当且仅当$x\in K$当且仅当$H(x,x)=1$.

若$H(x,y)$可计算，则K的特征函数χ_K也可计算，与K不是可计算集矛盾。□

§3 停机问题的不可解性

证法2

假设有程序实现$H(x,y)$的计算，则存在如下程序：

```
θ(x)
{
   if H(x,x)==1 then while(1){x=x;};//死循环
}
```

由Church-Turing论题，存在一台图灵机T_z与程序θ功能完全一样，则$H(z,z)=1$当且仅当$\theta(z)$永不停机当且仅当$T_z(z)$永不停机当且仅当$H(z,z)=0$，矛盾 □

因此假设是错的，即没有程序实现$H(x,y)$的计算。

§ 3 停机问题的不可解性

预言机（Oracle神谕） $A : \{0,1\}^* \rightarrow \{0,1\}$

- 可以访问预言机 A 的图灵机记为 M^A。其中M是多带图灵机，其中一个带是特殊的“预言机带”。 M可以在预言机带上问A一个问题，即写一些字符串 x，并且在下一步中，预言机将其答案 A(x) 写在预言机带上。
- 给定两个问题 P 和 Q，如果存在图灵机 M 使得 M^Q 解决 P，则 P可规约为 Q。我们将其写为 $P \leq_T Q$。
- 若$P \leq_T Q$，而Q可计算，则P可计算。

§ 3 停机问题的不可解性

- **费马大定理**

费马在丢番图的一本《算术》的空白处断言，三个正整数x、y和z不满足丢番图方程：

$$x^n + y^n = z^n$$

$$n \geq 3$$

$$xyz \neq 0$$

“一个立方数不可能是两个立方数的和，一个四次方数不可能是两个四次方数的和，一般来说，任何大于二次方的数都不可能是两个类似的幂的和。我发现了一个非常绝妙的例子来证明这个命题，即这个纸的边距太窄了，写不下。”——费马1637年

§3 停机问题的不可解性

希尔伯特的第十个问题：
给定一个含有任意数量的未知量和有理整数系数的丢番图方程，设计一个程序，根据这个程序，用有限的次数就可以确定这个方程是否可以用有理整数解。

这个问题在1970年被尤里·马蒂亚耶塞维奇（Yuri Matiyasevich）解决了，他在他的博士论文中证明了解决所有丢番图方程的一般算法是不存在的。他的解决方案蕴含了希尔伯特的第十题的不可解性。

§3 停机问题的不可解性

不可解性的直观理解

1。仿照图灵机编码方法，试给C语言程序编码，从而证明C语言所写的程序集是r.e.集。

2。试证明不可能用C语言写一个程序HALT(X,Y)，它能判断任意一个C语言写的程序X，当输入为Y时是否能最终正常退出。

3。试证明不可能用C语言写一个程序FORMATC(X,Y)，它能判断任意一个C语言写的程序X，当输入为Y时是否会格式化系统盘。

4。试证明不可能用C语言写一个程序ENDLESS(X)，用它来判断任何一个C语言写的程序X是否有死循环漏洞。或判断是否会做任何恶意的事情。

§4　Gödel不完备性定理

2002年美国《时代周刊》列出“20世纪震撼人类思想界的四大伟人”：

爱因斯坦（Albert Einstein）

图灵（Alan Turing）

哥德尔（Kurt Gödel）

凯恩斯（John Keynes）

Kurt Gödel, 1906－1978

§4　Gödel不完备性定理

1900年巴黎数学家会议上，希尔伯特遵从“世界上没有不可知”，“人类理性提出的问题人类理性一定能够回答”的哲学信念，提出**23**个数学问题，其中的第二个问题就是建立整个数学的一致性（即无矛盾性或称协调性）。

1920年代希尔伯特本人曾提出了一个使用有穷方法建立实数和分析的一致性的方案，称为希尔伯特元数学方案。

“Take any definite unsolved problem, such as … the existence of an infinite number of prime numbers of the form $2^n + 1$. *However unapproachable these problems may seem to us and however helpless we stand before them, we have, nevertheless, the firm conviction that their solution must follow by a finite number of purely logical processes…”*

“…This conviction of the solvability of every mathematical problem is a powerful incentive to the worker. We hear within us the perpetual call: There is the problem. Seek its solution. You can find it by pure reason, for in athematics there is no ignorabimus.”, David Hilbert, 1900.

“以任何明确的未解决问题为例，例如……存在无限数量的 $2^n + 1$ 形式的素数。无论这些问题在我们看来多么难以接近，无论我们站在它们面前多么无助，我们仍然坚信：他们的解决方案必须遵循有限数量的纯逻辑过程……”

“……这种对每个数学问题的可解性的信念是对数学工作者的强大激励。我们听到内心永恒的呼唤：问题在那里。寻求它的解决方案。你可以通过纯粹的理性找到它，因为在数学中没有无知的存在。”

----大卫希尔伯特1900年

§4 Gödel不完备性定理

1930年获得博士学位之后，哥德尔开始沿着希尔伯特方案的路线着手解决希尔伯特第二问题。

哥德尔最初是想寻此方案首先建立算术理论的一致性，然后再建立相对于算术而言更复杂的实数理论的一致性，但出乎意外的是，他得到了与希尔伯特预期完全相反的结果。

Gödel不完备性定理（非形式化版本）

任何一个可靠的能证明足够丰富的数学定理的证明系统，都存在至少一个为真的定理不能在此系统中得到证明。

§4 Gödel不完备性定理

- 什么是数学定理？数学定理就是一个陈述句。

例1：
2,696,635,869,504,783,333,238,805,675,613,588,278,597,832,162,617,892,474,670,798,113是一个素数.

例2：

以下Python程序对任意正整数n都会停机

```
def f(n):
    if n==1: return 1
    return f(3*n+1) if n % 2 else f(n//2)
```

§4 Gödel不完备性定理

- 什么是证明系统？

证明是一段佐证一个数学定理为真的文本（字符串）。

定义 **证明系统**：设T是某个“真”定理的集合。T的证明系统是一个算法V，满足：

1.有效性：给定定理x和证明w，V(x,w)都能停机并输出1(“真”)或0(“假”)（例如，通过逐行检查每一行是否都是前面某些行使用某条推理规则得到的）；

2.可靠性：若$x \notin T$，则对任意w， V(x,w)=0.

一个定理$x \in T$是**V不可证明的**，是指对任意w， V(x,w)=0.

若不存在定理$x \in T$是V不可证明的，则称V关于T是**完备的(complete)**

§4 Gödel不完备性定理

设H={"x∈K"| x∈*N*且"x∈K"为真}∪{"x∉K"| x∈*N*且" x∉K"为真}

定理 H不存在完备的证明系统。

证 反证法。若V是H的完备的证明系统，则可构造如下计算K的特征函数χ_K的算法。即K是可计算集，矛盾。

Boolean χ_K(x)
Input: x； Output: 1 若x∈K；0 若x∉K
for n = 0, 1, 2, 3, … do{
  for $w \in \{0, 1\}^n$ do{
    if V ("x∈K" , w) == 1 then return 1；
    if V ("x∉K" , w) == 1 then return 0；
  }
}

§4　Gödel不完备性定理

哥德尔不完备性定理有如下几种等价说法：

（1）没有定理证明机器（或机器程序）能够证明所有的数学真理。

（2）数学是算法不可完全的。

（3）数学是机器程序不可穷尽的。

（4）停机问题不可解，因此本质上，计算机的能力是有局限的。

§4 Gödel不完备性定理

1997年曾任美国数学会主席的斯梅尔（S.Smale）效仿数学家希尔伯特向全世界数学家提出了21世纪需要解决的24个数学问题，其中的第18个问题是，“人类智能的极限和人工智能的极限是什么”？并且指出，这个问题与哥德尔不完全性定理有关。

哥德尔定理说出了对于形式系统的局限，但是定理并没有给出人类理性的界限。

人脑是不是计算机？“人心”是不是计算机？图灵机的计算模型是否可以超越？

§4 Gödel不完备性定理

计算机科学研究受到的启发：

模拟人类智能，超越图灵机概念，探讨以自然为基础的生物计算、量子计算等新的计算模式，实施一种“算法+自然机制”的方法论策略。

对于能够归为传统算法解决的问题，依然使用算法手段解决，不能归为传统算法解决的问题，借助自然的生物、化学、物理的机制解决，并期望借助这种“半人工”手段制造出堪与人脑匹敌的所谓“半人工智能”来。如：

拟物拟人算法、人工神经网络、遗传进化算法、模拟淬火算法、蚁群算法