丘奇-图灵论题
与
丘奇-图灵定理

Miner

2011.03.13

http://www.douban.com/group/swarmagents_ai/

http://groups.google.com/group/swarmagents_ai/

希尔伯特第十问题

- 1900 年，巴黎第二届国际数学家大会，希尔伯特“数学问题”演讲：23个他认为最具重要意义的数学问题
- 第十问题：判定丢番图方程的可解性
  - 对包含任意个未知数的丢番图方程，给出一个有效的算法，通过有限次的计算，能够判定该方程在有理数整数上是否可解。
    - 丢番图方程：整系数代数多项式方程
    - x2+y2=z2 有整数解 (勾三股四弦五)
    - 2x-2y=1 没有整数解

- 30年没有进展

- 什么是算法？

- 可计算性和计算复杂性理论
  - 研究计算和可计算性概念
  - 研究各种计算模型及其等价性
  - 研究不可计算性
  - 研究P和NP问题

Alonzo Church
(1903-1995)

Kurt Godel
[1906-1978]

Alan Turing
(1912-1954)

可以有效计算的函数

- 数学上，算法是（通过有限多的步骤）对数学函数进行有效计算的方法。
- 因此算法研究的一个重要的切入点，是寻找可以有效计算的函数。
- 开始只知道一些最简单的函数，以及用这些函数通过若干简单规则组合出的函数是可以有效计算的。数学家们把这类函数叫做递归函数（Recursive Function）。

- 1931 年， Herbrand(1908-1931，登山时遇难) 对递归函数进行了研究， 并给哥德尔写信叙述了自己的研究结果。
- Church(1903-1995) 命题：所有可以有效计算的函数都可以用 λ-calculus来定义。
- 1934 年， 丘奇向哥德尔介绍了这一猜测， 但哥德尔不认同。 于是丘奇请哥德尔给出一个他认为合适的描述。
- 哥德尔在Herbrand结果的基础上提出了一般递归函数的概念，并指出：凡算法可计算函数都是一般递归函数，反之亦然．（但哥德尔当时并没想到他的递归概念包含了所有可能的递归 ）
- 丘奇与克林证明了这两种看上去完全不同的描述方式实际上是彼此等价的。丘奇相信已经找到了可以有效计算的函数的普遍定义。 但哥德尔并不赞成，在他看来，在还没有找到一组公理刻划算法可计算性概念所包含的公认特性之前，不可能有完全令人满意的严格的数学定义。
- 在此期间，图灵独立思考了可计算性问题，最终以通用图灵机概念刻划了算法可计算性，并发现丘奇与哥德尔所定义的那些函数与他的抽象计算机可以计算的函数等价，获得哥德尔的承认（图灵当时研究图灵机的目的是要研究希尔伯特于 1928 年提出的有关一阶逻辑的判定问题 ）
- 哥德尔不认同丘奇论题的原因可能是他认为可计算性不应依赖于形式系统的选择。而图灵机可计算这个概念的第一次成功地给出有意义的认识论观念上的一种绝对的定义。

丘奇论题

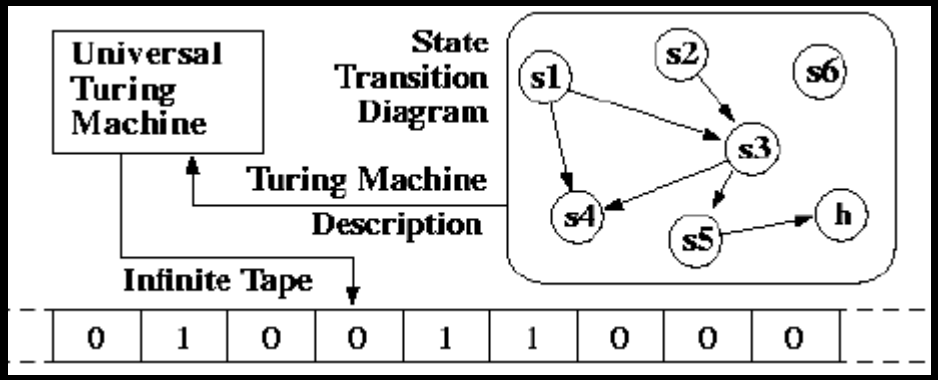
- 丘奇论题： 1936年，丘奇证明了他提出的λ可定义函数与一般递归函数是等价的，并提出算法可计算函数等同于一般递归函数或λ可定义函数
- 用一般递归函数虽给出了可计算函数的严格数学定义，但在具体的计算过程中，就某一步运算而言，选用什么初始函数和基本运算仍有不确定性

图灵“论可计算数及其在判定问题中的应用”

- 1936 完成，1937 发表
- 从一个全新的角度定义了可计算函数
- 全面分析了人的计算过程，把计算归结为最简单、最基本、最确 定的操作动作，从而用一种简单的方法来描述那种直观上具有机械性的基本计算程序，使任何机械(能行)的程序都可以归约为这些动作。这种简单的方法是以一个 抽象自动机概念为基础的，其结果是：算法可计算函数就是这种自动机能计算的函数．这不仅给计算下了一个完全确定的定义，而且第一次把计算和自动机联系起 来。

图灵机

- 由以下几个部分组成：
- 1.一条无限长的纸带 **TAPE**。纸带被划分为一个接一个的小格子，每个格子上包含一个来自有限字母表的符号，字母表中有一个特殊的符号 表示空白。纸带上的格子从左到右依此被编号为 0, 1, 2, ... ，纸带的右端可以无限伸展。
- 2.一个读写头 **HEAD**。该读写头可以在纸带上左右移动，它能读出当前所指的格子上的符号，并能改变当前格子上的符号。
- 3.一套控制规则 **TABLE**。它根据当前机器所处的状态以及当前读写头所指的格子上的符号来确定读写头下一步的动作，并改变状态寄存器的值，令机器进入一个新的状态。
- 4.一个**状态寄存器**。它用来保存图灵机当前所处的状态。图灵机的所有可能状态的数目是有限的，并且有一个特殊的状态，称为停机状态。参见停机问题。



丘奇-图灵论题
Church-Turing Thesis

- 1937年，图灵在他的“可计算性与λ可定义性”一文中证明了图灵机可计算函数与λ可定义函数是等价的，从而拓广了丘奇论题，得出：算法(能行)可计算函数等同于一般递归函数(Herbrand-Gödel 递归函数)或λ可定义函数或图灵机可计算函数。

- 可计算就是图灵可计算

为什么是论题(thesis)，而不是定理？

- Church-Turing Thesis 无法被证明
  - “可有效计算”本身是一个不存在精确定义的概念， 它本质上取决于人们对“有效”及“计算”这样的非精确概念的理解。
  - 如果有一个方法能被普遍接受为一个有效的算法但却无法在图灵机上实现，则该论题不成立。
  - 目前普遍认为 Church-Turing Thesis 是正确的

希尔伯特第十问题 – 判定问题

- “判定问题”：指判定大量问题(某个问题包含无限种情况)是否具有算法解，或者是否存在可行的方法使得对该问题类的每一个特例都能在有限步骤内机械地判定它是否具有某种性质(如是否真，是否可满足或是否有解等，随大量问题本身的性质而定)的问题
- 1928,希尔伯特提出判定问题(*Entscheidungs* Problem):狭谓词演算(亦称一阶逻辑)公式的可满足性的判定问题.
- 1937年，图灵发表的“论可计算数及其在判定问题中的应用”解决了希尔伯特判定问题。他用一阶逻辑中的公式对图灵机进行编码，再由图灵机停机问题的不可判定性推出一阶逻辑的不可判定性。
- 1944，Post 首先猜测，对于第十问题，应寻求不可解的证明。
- 1970，希尔伯特第十问题被证明是不可判定问题

图灵“停机问题”

- 图灵机停机问题: 是否存在一台"万能的"图灵机 H, 把任意一台图灵机 M 输入给 H, 它都能判定 M 最终是否停机, 输出一个明确的 "yes" 或 "no" 的答案?
  - 假定存在一个能够判定任意一台图灵机是否停机的万能图灵机 H(M), 如果 M 最终停机, H 输出 “halt”; 如果 M 不停机, H 输出 “loop”. 我们把 H 当作子程序, 构造如下程序 P:

    ```
    function P(M) {
    if (H(M)==“loop”) return “halt”;
    else if (H(M)==“halt”) while(true); // loop forever
    }
    ```

    因为 P 本身也是一台图灵机, 可以表示为一个字符串, 所以我们可以把 P 输入给它自己, 然后问 P(P) 是否停机. 按照程序 P 的流程, 如果 P 不停机无限循环, 那么它就停机, 输出“halt”; 如果 P 停机, 那么它就无限循环, 不停机

- “停机问题”可作为研究许多判定问题的基础，一般地，把一个判定问题归结为停机问题：“如果问题A可判定，则停机问题可判定．”从而由“停机问题是不可判定的”推出“问题A是不可判定的”。

丘奇定理 Church‘s theorem
丘奇-图灵定理 Church–Turing Theorem

- 丘奇定理: 不存在一个判定程序来确定谓词演算中的任意公式是否为演算
- 1935~1937, Church 和 Turing 独立证明“ it is impossible to decide algorithmically whether statements in arithmetic are true or false”
- Church 用 λ 演算，Turing用图灵机，两个证明等价
- 都受到了 Gödel证明不完备性定理的影响，尤其是Gödel 配数（通过给逻辑公式编号把逻辑转化为算术）.

Appendix

半可判定

- 对于一个判定问题，如果能够编出一个程序*P*,以域中任意元素作为输入，当相应的个别问题的解答是肯定的时候,*P* 的执行将终止并输出“是”,否则*P* 的执行不终止，就称该判定问题为半可判定的。可判定的问题总是半可判定的。集合是递归可枚举集的充分必要条件为对应的判定问题是半可判定的。

哥德尔不完备性定理

- 1931, incompleteness theorems
  - “On formally undecidable propositions of Principia Mathematica and related systems 数学原理中的形式不可判定命题及有关系统”
  - for any computable axiomatic system that is powerful enough to describe the arithmetic of the natural numbers (e.g. the Peano axioms or ZFC), that:
    - 1. If the system is consistent, it cannot be complete. (This is generally known as the incompleteness theorem.)
    - 2. The consistency of the axioms cannot be proved within the system.
    - 第一不完全定理:设系统S包含有一阶谓词逻辑与初等数论,如果S是一致的(无矛盾),存在则下文的T与非T在S中均不可证.
    - 第二不完全定理:如果系统S含有初等数论，当S无矛盾时，它的无矛盾性不可能在S内证明。

对“哥德尔不完备性定理”的理解

- 哥德尔本人认为“不完全性定理并未给出人类理性的极限，而只揭示了数学形式主义的内在局限”
- 图灵认为“AI能否实现”不受哥德尔不完备性定理限制：因为并不需要机器永远输出永真的答案
  - there might be men cleverer than any given machine, but then again there might be other machines cleverer again, and so on.
  - 改进的图灵机：可以中断计算来寻求外部信息

Church–Turing–Deutsch principle (CTD principle)

- physical form of the Church–Turing thesis formulated by David Deutsch in 1985
- a universal computing device can simulate every physical process