1. 引言

丘奇－图灵论题（Church–Turing Thesis）断言：图灵机是最广义的计算装置。丘奇－图灵论题有时也称图灵论题，是计算机科学的基石，它之于计算机科学，宛如公理之于几何学，牛顿定律之于物理学。计算理论的缘起就是丘奇－图灵论题的形成过程。费曼说微积分是上帝的语言。如果我们把第一次工业革命，归因于机械和能量，贴上牛顿的标签，而把当下正在经历的第四次工业革命归因于信息和计算；那么，上帝的语言该改成图灵机了。至少，英国50英镑钞票的头像刚刚从瓦特换成了图灵。

丘奇－图灵论题的根据就是所知的最广义的计算机制都是等价的，这些机制包括：广义递归函数、λ－演算、珀斯特（Post）系统、图灵机等。这些机制中的任意两个之间都可以互相模仿。这个结论有时也称为图灵定理，因为这些等价性是数学上可证明的。图灵论题是把图灵定理推广到所有可能的、潜在的机制，这个跳跃使得“定理”变成了“论题”，因为没法在现在确定的已知框架中去证明未来的不确定的潜在未知。这需要一种信念（conviction），信念就是论题。论题既非数学定理也非物理定律，而是一种实用主义的共识。当然，如果把这种共识公理化，也有可能证明丘奇－图灵论题，这本是哥德尔（Kurt Gödel）的期望。逻辑学家德肖维茨和古列维奇近来曾做过努力，他们按照哥德尔指明的方向，试图提出一套可以被广泛接受的公理，在此之上证明丘奇－图灵论题[1]。但是公理本身也需要某种直觉上的共识，况且，今天对公理的共识也无法消解明天潜在的非议，共识了两千多年的欧几里得公理体系也可以被掰弯。

[1] Dershowitz, N. & Y. Gurevich (2008), A Natural Axiomatization of Computability and Proof of Church's Thesis, The Bulletin of Symbolic Logic, vol 14, no 3.

如果公理的根据还没有图灵机更加直觉上可靠的话，其上的证明也是徒劳。

有人说数学中只有定义和定理，没听说过“论题”。此言差矣。极限和连续的 ε–δ 定义，其实就是某种“论题”，只不过我们习惯于用定义来指称它，而不说这是“柯西 – 维尔斯特拉斯论题”（Cauchy–Weierstrass Thesis），我们的这种习惯恰是因为它符合我们对极限和连续的直觉。

对数学或者理论计算机科学的外行人，我们甚至可以采用一种人类学或社会学的标准：所有最聪明的人想到的机制都是等价的，那么这个论题肯定是靠谱的。为了避免过多地偏离主线，本文中，我们从此以共识作为“论题”的解释。这个共识的结论之一就是不存在比图灵机更强的计算装置。

丘奇 – 图灵论题是可计算性理论的基础。在计算复杂性理论里，还有“强丘奇 – 图灵论题”，它假设了更多：所有计算装置之间不仅可以互相模拟，而且模拟的成本是多项式的。姚期智认为丘奇 – 图灵论题和强丘奇 – 图灵论题可能更会受到物理定律的约束[2]。很明显，强丘奇 – 图灵论题受到的约束更大。肖尔（Peter Shor）提出素数分解的量子算法之后，量子计算就是对强丘奇 – 图灵论题的明确且现实的挑战，如果素数分解问题被证明不能在多项式时间内求解，那么量子计算对强丘奇 – 图灵论题的挑战就成功了。若果真如此，将来某一天我们不得不修正强丘奇 – 图灵论题。姚期智聪明地指出，否定强丘奇 – 图灵论题，甚至都不一定需要量子力学，经典力学中的 N– 体问题，可能就没法找到多项式时间的解。但即使如此，大家仍然会认可丘奇 – 图灵论题。

丘奇 – 图灵论题的主要论点是图灵机捕捉到了直觉上的计算概念，尽管它不是定理。它有如下几种表述方式：

1）所有计算装置都与图灵机等价。
2）人按照算法执行的计算和图灵机等价。
3）人的智能和图灵机的能力等价。

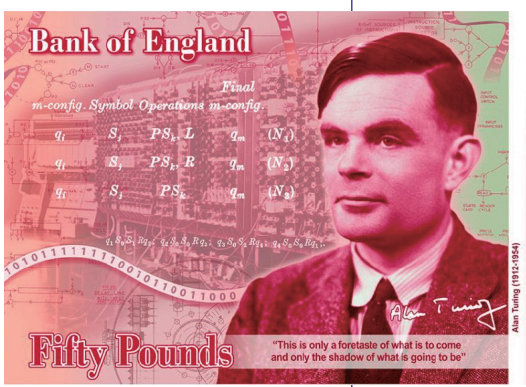
这几种表述代表了不同的哲学观点，一个比一个强。1）虽不是定理，但目前我们知道的所有计算装置都和图灵机等价，例如哥德尔 – 厄布朗（Gödel–Herbrand）广义递归函数，珀斯特（Emil Post）的各种产生式系统，丘奇（Alonzo Church）的 λ– 演算，乔姆斯基（Noam Chomsky）0 型文法等。第一种表述目前是几乎所有理论计算机科学家的共识。第二种和第一种表述有很强的关联，丘奇、珀斯特和图灵的初衷都是为了能够找到一个机械装置可以完整地刻画人施行计算的过程。如果认为算法就是机械过程的话，头两种表述是等价的。哥德尔认可前两种表述，但对第三种表述不买账。认可第三种表述就意味着认可“人是机器”，这是人工智能的终极哲学问题。对丘奇 – 图灵论题的态度决定了如何应答这个问题。

[2] Yao, A. C.-C. (2003), Classical physics and the Church-Turing thesis, Journal of the ACM 50, 100-105.

目前没有证据表明存在着某种自然过程可以展现比图灵机更强的能力，但即使某一天真的发现了这种自然存在（就像有些人期望的那样：可能会有某种生物过程超越图灵机），也不影响图灵机是最强的机械可计算装置。哥德尔就认为可能存在不可机械计算的心理过程，但他也毫不含糊地认可图灵机是最强、最广义的机械计算装置。哥德尔在 1946 年为他 1931 年不完全定理文章写的后记中说：是否存在不等价于任何算法的非机械的过程，与“形式系统”或者“机械过程”的定义的充足性没啥关系。

2. 图灵与图灵机

图灵在剑桥读书时的兴趣是数论和量子力学，但为什么会想到图灵机呢？按照图灵唯一的学生和恋人罗宾 • 甘地（Robin Gandy）1954 年在《自然》杂志上为图灵写的讣告里的说法：图灵本科毕业后在剑桥国王学院做研究员（Fellow）的第一年曾对计算黎曼 zeta 函数的零点感兴趣，由此想到要造一台机器。事实上，图灵怀疑黎曼猜想不成立，于是他一直试图找一个反例。图灵机实际是图灵研究黎曼猜想的副产品。



图灵（1912 年 6 月 23 日— 1954 年 6 月 7 日），英国数学家，计算机科学的奠基者。和当时人才的迂徙和鄙视链相反，他剑桥国王学院毕业后做了一年研究员，到美国普林斯顿大学跟随逻辑学家丘奇，得到博士学位。毕业后回到英国不久即被征召为军方从事密码学工作，现在已经解密的材料表明，图灵为破解德国密码做出了杰出贡献。他因为性取向得到不公正的待遇，1954 年不到 42 岁就去世了。关于他的死因，他的学生和爱人甘地曾经激动地说：有些事太深太私密不宜深究。

图灵一生最重要的工作有三项，第一，1936 年的可计算数（Computable Numbers），提出了后来被丘奇称为图灵机的装置，奠定了计算机科学；第二，1950 年在哲学杂志 Mind 上发表的“计算机与智能”，这篇文章为人工智能划定了边界；第三，1952 年的“Chemical Basis Morphogenesis”，这篇文章近来开始得到越来越多的复杂性研究者的关注。

图灵到美国留学回到英国不久就被军方征召做加密工作。他关于黎曼 zeta 函数零点的计算方法的文章拖到 1943 年才在《伦敦数学会会刊》上发表[3]。但图灵最早设计的计算黎曼 zeta 函数的机器是一个可以进行离散近似的模拟装置。这与甘地所说不知是否一回事。二战结束后，图灵在曼彻斯特大学时，又

[3]Turing, A. M. (1943), A method for the calculation of the zeta-function, Proc. Lond. Math. Soc. (2) 48 180-197.

短暂恢复了他对数论的兴趣，他 1949-1950 年间写了两个程序在英国最早的计算机曼彻斯特 Mark-I 上运行，一个力图找出更大的梅森素数，图灵和合作者验证了当时已知的梅森素数，但没有找到新的；另一个计算黎曼 zeta 函数的零点。其实图灵是想找黎曼猜想的反例，自然无果。图灵的计算结果在他去世前一年才在《伦敦数学会会刊》上发表[4]，现在所有计算黎曼 zeta 函数零点的算法都是由图灵的原始算法衍生出来的，于是有人称图灵是计算数论的开拓者[5]。图灵和哥德尔的初衷不一样，哥德尔的起点和终点都是逻辑，而图灵的起点是数论，终点是计算。

马克斯·纽曼（Max Newman）是英国当时的领袖级数学家，主攻拓扑，他 1928 年参加了在意大利博洛尼亚召开的国际数学家大会，现场听到身体欠佳的希尔伯特讲到判定问题。两年后判定问题就被哥德尔负面地解决了，颇令希尔伯特不爽。哥德尔的结果激发了纽曼对逻辑的兴趣。他 1935 年春季学期为剑桥高年级学生开讲“数学基础”，图灵也去听讲。图灵和纽曼深入讨论，理解了希尔伯特期望的是某种机械过程。

Dr. A. M. Turing, O.B.E., F.R.S.

ALAN TURING was born on June 23, 1912, and was educated at Sherborne and King's College, Cambridge. He was made a Fellow of King's in 1934; he submitted his fellowship dissertation—a version of the central limit theorem for the normal distribution—four months after being placed as Wrangler in the Mathematical Tripos. During his first years of research he worked on a number of subjects, including the theory of numbers and quantum mechanics, and started to build a machine for computing the Riemann ζ-function, cutting the gears for it himself. His interest in computing led him to consider just what sort of processes could be carried out by a machine: he described a 'universal' machine, which, when supplied with suitable instructions, would imitate the behaviour of any other; he was thus able to give a precise definition of 'computable', and to show that there are mathematical problems the solutions of which are not computable in this sense. The paper which contains these results is typical of Turing's methods: starting from first principles, and using concrete illustrations, he builds up a general and abstract argument. Many years later he used an elaboration of the same ideas to prove the unsolvability of the word problem in semi-groups with cancellation.

In 1936 he went to Princeton for two years; he worked on group theory and logic, receiving his Ph.D. for a dissertation on "Ordinal Logics". In this he showed that when transfinite induction is used in logic for proofs and definitions, it is not the ordinal up to which induction runs that has significance, but rather the particular way in which that ordinal is described.

He was awarded the O.B.E. for the work he did during the War, and after it he was invited by the National Physical Laboratory to direct the design of an electronic digital computor (which he christened "The Automatic Computing Engine"). Although the theoretical aspects of its design were his chief concern, he was also keenly interested in its electronics; and while the final construction was in progress

536 NAT

he turned to long-term problems, considering how machines might be made to learn by trial and error and the ways in which they could be compared with human brains.

In 1949 he was made deputy director of the Computing Laboratory at the University of Manchester, where work on an electronic computor had just started. He was elected Fellow of the Royal Society in 1951. His last work was a mathematical theory of morphogenesis; the main idea was to show how an originally uniform distribution of interacting substances may, as a result of diffusion, develop a strongly marked pattern. He had already published a version of the theory for distributions around a ring, and was at work on the case of a cylinder; using the machine to solve the appropriate differential equations, he was hoping to be able to exhibit the spiral patterns based on the Fibonacci series which are so frequently found in plants.

The marks of Turing's genius were his originality, his ability to control abstract thought by concrete illustration, and his preference for always working things out for himself. The freshness of his mind, his love of inquiry, and his relish for the comic, made him a lively and stimulating companion.

R. O. GANDY

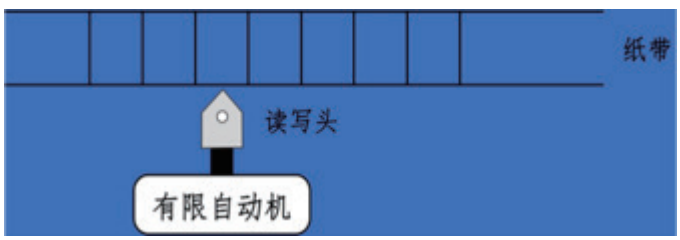
[4] Turing, A. M. (1953), Some calculations of the Riemann zeta-function, Proc. Lond. Math. Soc.

[5] Booker, Andrew (2016), Turing and Primes, in Cooper and Hodges, 2016.

图灵利用图灵机否定了可判定性。1936 年丘奇在创办《符号逻辑杂志》（*JSL*）首刊上登了自己关于可判定性问题的两页短文。纽曼收到丘奇寄来的 *JSL* 首刊时，已经看过图灵的初稿，他马上把丘奇的结果告诉图灵，并给丘奇写信告知自己的学生图灵也在研究同一问题并取得进展，推荐图灵给丘奇做学生。图灵知道丘奇的结果后，略有失望和焦急，他在给母亲的信中提到纽曼认为虽然他的结果和丘奇类似，但方法不同也可发表[6]。事实上，图灵稍晚发表文章，从某种意义上帮助了他，让他有机会完善结果。图灵研究了丘奇的 λ– 演算，很快证明了图灵机和 λ– 演算是等价的，把证明过程增补到原文作为附录。恰是这个附录增加了所有人的信心。纽曼利用自己在伦敦数学会的影响力，帮助图灵把论文投给该会的会刊，这就是那篇无论怎样赞誉都不过分的经典[7]。

图灵机

图灵机是一个超级简单的计算装置，它包含一条纸带，上面有无限多个格子，有个有限自动机，可以在当下的格子上写 0 或 1；左移右移。



假设我们想利用图灵机计算二进制的 5*3 = 15

$$\begin{array}{r} 101 \\ 11 \\ \hline 101 \\ 1010 \\ \hline 1111 \end{array}$$

我们用最朴素的算法，并且除了 0 和 1 之外，还可以利用符号：“*”和“=”。以上结果在图灵机的纸带上可表示为：

1	0	1	*	1	1	=	1	0	1	+	1	0	1	0	=	1	1	1	1

[6] Turing, Dermot (2015), Alan Turing Decoded.

[7] Turing, A. M. (1936), On computable numbers, with an application to the Entscheidungsproblem, Proc. Lond. Math. Soc. (2) 42, 230-265 19. A correction (1937), Proc. Lond. Math. Soc. (2) 43, 544-546.