

# 图灵机、人工智能以及我们的世界

顾森

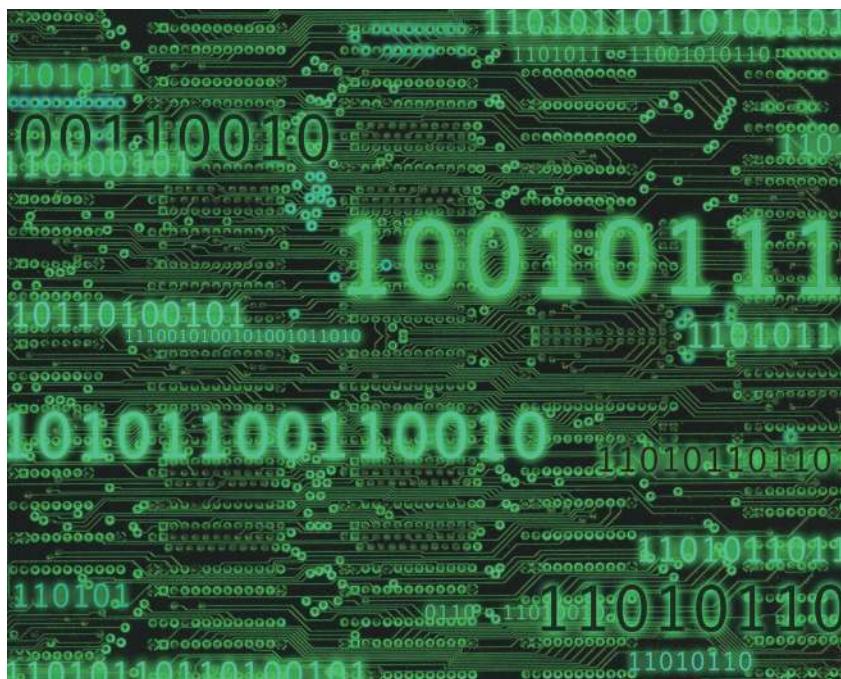
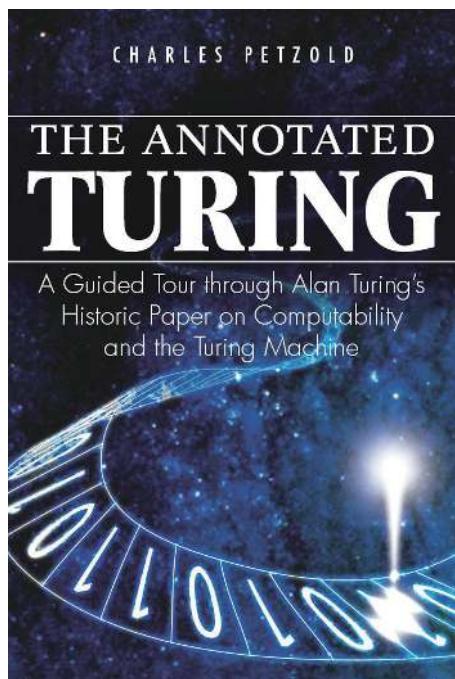
最近读完了 [The Annotated Turing](#) 一书，第一次完整地阅读了图灵最经典的那篇论文，理解了图灵机提出的动机和由此带来的一系列结论。不过，这本书的最大价值，则是让我开始重新认识和思考这个世界。在这里，我想把我以前积累的哲学观点和最近一些新的思考记下来，与大家一同分享。今年是阿兰·图灵诞辰 100 周年，图灵公司将推出这本书的中译本《图灵的秘密》，现在正在紧张的编辑排版中，不久之后就能和大家见面。

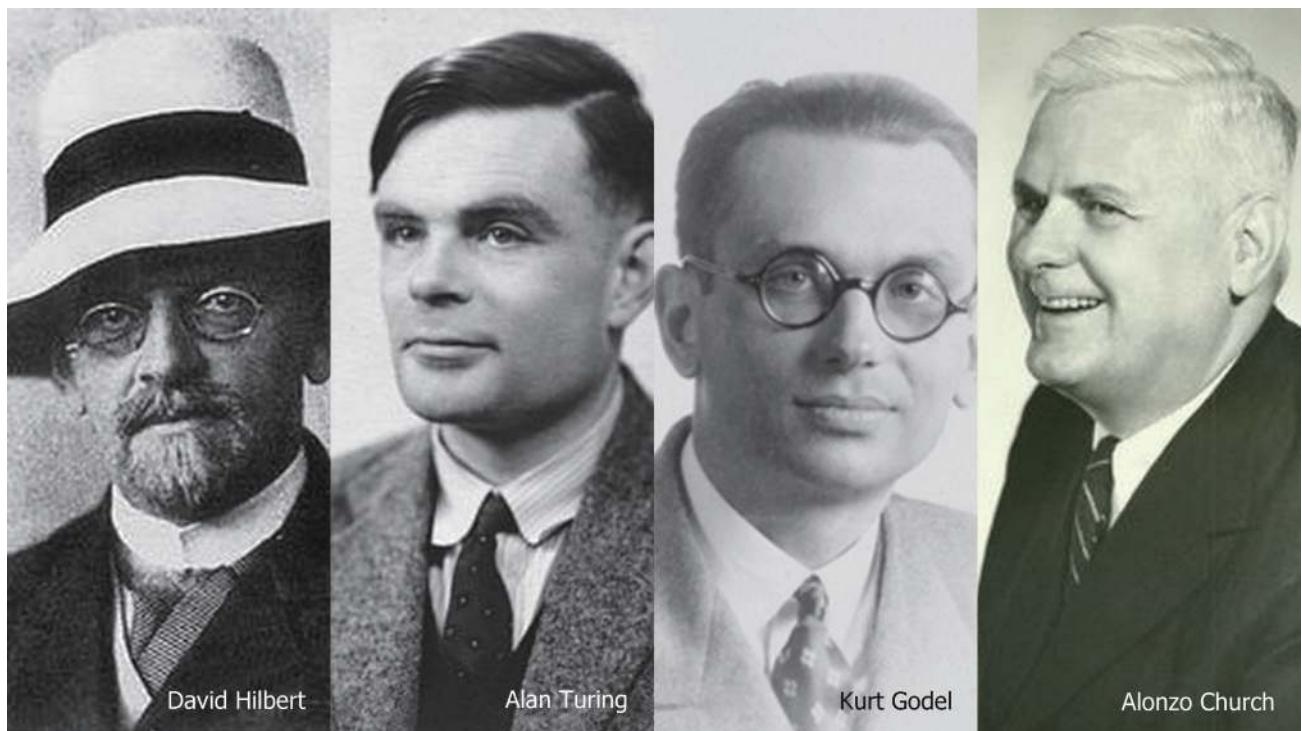
1928 年，大卫·希尔伯特 (David Hilbert) 提出了一个著名的问题：[是否存在一系列有限的步骤，它能判定任意一个给定的数学命题的真假？](#)这个问题就叫做 *Entscheidungsproblem*，德语“判定性问题”的意思。大家普遍认为，这样的一套步骤是不存在的，也就是说我们没有一种判断一个数学命题是否为真的通用方法。为了证明这一点，真正的难题是将问题形式化：什么叫做“一系列有限的步骤”？当然，现在大家知道，这里所说的“有限的步骤”指的就是由条件语句、循环语句等元素搭建而成的一个机械过程，也就是我们常说的

“算法”。不过，在没有计算机的时代，人们只能模模糊糊地体会“一个机械过程”的意思。1936 年，阿兰·图灵在其著名的论文 [On computable numbers, with an application to the Entscheidungsproblem](#) 提出了一种假想的机器，第一次给了“机械过程”一个确凿的含义。

图灵提出的机器非常简单。假设有一张无穷向右延伸的纸条，从左至右分成一个一个的小格子。每一个小格子里都可以填写一个字符（通常是单个数字或者字母）。纸条下方有一个用来标识“当前格子”的箭头，在机器运行过程中，箭头的位置会不断移动，颜色也会不断变化。不妨假设初始时所有格子都是空白，箭头的颜色是红色，并且指向左起第一个格子。为了让机器实现不同的功能，我们需要给它制定一大堆指令。每条指令都是由五个参数构成，格式非常单一，只能形如“如果当前箭头是红色，箭头所在格子写的是字符 A，则把这个格子里的字符改为 B，箭头变为绿色并且向右移动一格”，其中最后箭头的移动只能是“左移一格”、“右移一格”、“不动”中的一个。

精心设计不同的指令集合，我们就能得到功能不同





可计算理论研究的创始人（从左至右）：希尔伯特，图灵，哥德尔，邱奇

的图灵机。你可以设计一个生成自然数序列的图灵机，或者是计算根号 2 的图灵机，甚至是打印圆周率的图灵机。图灵本人甚至在论文中实现了这么一种特殊的图灵机叫做通用图灵机，它可以模拟别的图灵机的运行。具体地说，如果把任意一个图灵机的指令集用图灵自己提出的一种规范方式编码并预存在纸条上，那么通用图灵机就能够根据纸条上已有的信息，在纸条的空白处模拟那台图灵机的运作，输出那台图灵机应该输出的东西。

但是，图灵机并不是无所不能的。图灵证明了一个看似有些惊人的事实：不存在这样的一个图灵机，它能读取任意一个图灵机的指令集，并判断该图灵机是否将会在纸条上打印出至少一个 0。注意，简单地用通用图灵机做模拟并不是一个可行的方案，因为模拟到现在还没有打出 0，不意味着今后永远不会打出 0。这个定理有一个更深刻的含义，即没有一种通用的方法可以预测一台图灵机无穷远后的将来（后人把这个结论简化为了著名的停机问题）。正如 [The Annotated Turing](#) 封底上的一段文字所说：[在没有计算机的时代，图灵不但探索了计算机能做的事，还指出了计算机永远不能做到的事。](#)

在论文的最后一章，图灵给出了一种图灵机指令集和一阶逻辑表达式的转换规则，使得这个图灵机将会打出 0 来，当且仅当对应的一阶逻辑表达式为真。然而，

我们没有一种判断图灵机是否会输出 0 的算法，因此我们也就没有一种判断数学命题是否为真的通用办法。于是，Entscheidungsproblem 有了一个完美的解答。

有趣的是，图灵机本身的提出比 Entscheidungsproblem 的解决意义更大。计算机诞生以后，出现了五花八门的高级编程语言，一个比一个帅气，但它们的表达能力实际上都没有超过图灵机。事实上，再庞大的流程图，再复杂的数学关系，再怪异的语法规则，最终都可以用图灵机来描述。图灵机似乎是一个终极工具，它似乎能够表达一切形式的计算方法，可以描述一切事物背后的规律。在同一时代，美国数学家阿隆佐·邱奇（Alonzo Church）创立了  $\lambda$  算子（ $\lambda$ -calculus），用数学的方法去阐释“机械过程”的含义。后来人们发现，图灵机和  $\lambda$  算子是等价的，它们具有相同的表达能力，是描述“可计算性”的两种不同的模型。图灵机和  $\lambda$  算子真的能够描述所有直观意义上的“可计算数”、“可计算数列”、“可计算函数”吗？有没有什么东西超出了它们的表达能力？这个深刻的哲学问题就叫做邱奇-图灵论题（Church-Turing thesis）。当然，我们没法用形式化的方法对其进行论证，不过大家普遍认为，图灵机和  $\lambda$  算子确实已经具有描述世间一切复杂关系的能力了。人们曾经提出过一些 hypercomputer，即超出图灵机范围的假想机器，比如能