The two most famous and--in the view of many--most important examples of conceptual analysis in twentieth century logic were Alfred Tarski’s definition of truth and Alan Turing’s definition of computability. In both cases a prior, extensively used, informal or intuitive concept was replaced by one defined in precise mathematical terms. (p. 1)

Feferman, Solomon Tarski’s Conceptual Analysis of Semantical Notions 2008 Patterson, Douglas (Ed.) New Essays on Tarski and PhilosophyNew Essays on Tarski and Philosophyp. 72-93

"instead of arguing continually over this point it is usual to have the polite convention that everyone thinks"

it is impossible to extract philosophical juice from a piece of pure mathematics without invoking some philsophical thesis(p320)

The only philosophical impetus which Suszko mentions for his scepticism, though, is an analogy with computability. And that analogy misses its mark. Theformal equivalence between different notions of computability shows that we can happily employ any of several different formalisms. But this does not indicate that every formalism is philosophically on a par. Indeed, when presenting arguments infavour of the Church–Turing Thesis, many people have found Turing’s formal notion of computability more helpful than Church’s(on the grounds, for example, that it offers some kind of perspicuous decomposition of what computors could do inprinciple).(p320)

引自 13: Boolean-valued structures

这一点（图灵机在哲学上的明晰性）为许多人所提到。

书名: Philosophy and Model Theory

作者: Tim Button/Sean Walsh

出版社: Oxford University Press

“A. M. Turing produced a neat model of idealized computers and gave convincing arguments to show that it captures the intuitive concept of computability.” （Hao Wang,Beyond Analytic Philosophy: Doing Justice to What We Know,MIT,1986,p180）

我对计算机科学的发展历程不是太了解。我就说些计算机科学的史前史吧。

1931年[哥德尔](https://www.zhihu.com/search?q=%E5%93%A5%E5%BE%B7%E5%B0%94&search_source=Entity&hybrid_search_source=Entity&hybrid_search_extra=%7B%22sourceType%22%3A%22answer%22%2C%22sourceId%22%3A2272685195%7D)证明了不完备性定理，其中已经出现了[递归函数](https://www.zhihu.com/search?q=%E9%80%92%E5%BD%92%E5%87%BD%E6%95%B0&search_source=Entity&hybrid_search_source=Entity&hybrid_search_extra=%7B%22sourceType%22%3A%22answer%22%2C%22sourceId%22%3A2272685195%7D)的雏形，而我们知道，递归函数和图灵机、lambda演算都是等价的；

1930年代初，Alonzo Church（"[**A set of postulates for the foundations of logic**](https://link.zhihu.com/?target=https%3A//doi.org/10.2307/1968337)"）和Haskell Curry（“**[Grundlagen der kombinatorischen Logik](https://link.zhihu.com/?target=https%3A//doi.org/10.2307/2370619" \t "_blank)**“）为数学的基础提出了新的逻辑系统。两者都在1935年被Kleene和Rosser证明是不一致的（就Curry而言，只有[**equality axioms proposed in 1934**](https://link.zhihu.com/?target=https%3A//doi.org/10.2307/1968498)导致了矛盾）。丘奇提取的一致子系统是--你猜对了--λ演算（详见斯坦福哲学百科[**相关条目**](https://link.zhihu.com/?target=https%3A//plato.stanford.edu/entries/paradoxes-contemporary-logic/%23IncoCertFormLogi)）。另一方面，Curry对不一致性的分析给了我们[**Curry悖论**](https://link.zhihu.com/?target=https%3A//plato.stanford.edu/entries/curry-paradox/)。（[**Famous logicians and their inconsistent theories**](https://link.zhihu.com/?target=https%3A//richardzach.org/2021/06/23/famous-logicians-and-their-inconsistent-theories/)）

根据与Herbrand的通信，哥德尔提出了general recursive function。并认为凡是可由递归函数组合出来的都是可计算的，但他并没有认同这个论题的逆，即：凡是可计算的都是递归函数的。

1936年，church提出了church's thesis，主张凡是能行可计算的都是lambda演算的。可是哥德尔认为lambda演算并没有很好地刻画“可计算”这一概念。Turing晚几个月成就了他的著名论文，并提出了同样的论题，他当时用的词是“purely mechanical”，在听闻church的工作后他在自己的论文后加了一个appendix证明图灵机和lambda演算的等价性。

正是在图灵之后，哥德尔认为图灵机很好地刻画了“可计算”这一概念。

尽管图灵机和lambda演算是等价的。但这里的要点在这里：能行可计算或纯粹机器可计算的是一个非形式化的概念，那么用一个[形式化](https://www.zhihu.com/search?q=%E5%BD%A2%E5%BC%8F%E5%8C%96&search_source=Entity&hybrid_search_source=Entity&hybrid_search_extra=%7B%22sourceType%22%3A%22answer%22%2C%22sourceId%22%3A2272685195%7D)的概念来为一个非形式化的概念进行刻画，（）。

图灵机是很清晰地符合我们的日常的直观的，同时这是因为Turing为他的论题作了一个很好的论证，可以参考图灵的原论文，或Charles Petzold的《[图灵的秘密](https://www.zhihu.com/search?q=%E5%9B%BE%E7%81%B5%E7%9A%84%E7%A7%98%E5%AF%86&search_source=Entity&hybrid_search_source=Entity&hybrid_search_extra=%7B%22sourceType%22%3A%22answer%22%2C%22sourceId%22%3A2272685195%7D)》或杨睿之老师的[为什么能行可计算的就是图灵可计算的（递归的）？](https://www.zhihu.com/question/267184153/answer/320101790)

如果我没理解错的话，题主想要问的实际上就是[丘奇-图灵论题](https://www.zhihu.com/search?q=%E4%B8%98%E5%A5%87-%E5%9B%BE%E7%81%B5%E8%AE%BA%E9%A2%98&search_source=Entity&hybrid_search_source=Entity&hybrid_search_extra=%7B%22sourceType%22%3A%22answer%22%2C%22sourceId%22%3A320101790%7D)（Church-Turing Thesis）为什么成立。丘奇-图灵论题简单地说就是：

一个[自然数](https://www.zhihu.com/search?q=%E8%87%AA%E7%84%B6%E6%95%B0&search_source=Entity&hybrid_search_source=Entity&hybrid_search_extra=%7B%22sourceType%22%3A%22answer%22%2C%22sourceId%22%3A320101790%7D)上的函数 是*能行可计算的*（effectively computable），当且仅当它是*图灵可计算的*（Turing computable）。

其中，“能行可计算”不是一个数学概念，而“图灵可计算”是有严格定义的数学概念。我们知道，与其等价的数学概念还有哥德尔定义的[*递归函数*](https://www.zhihu.com/search?q=%E9%80%92%E5%BD%92%E5%87%BD%E6%95%B0&search_source=Entity&hybrid_search_source=Entity&hybrid_search_extra=%7B%22sourceType%22%3A%22answer%22%2C%22sourceId%22%3A320101790%7D)，和丘奇定义的 *-可演算*。丘奇-图灵论题要表达的就是“图灵可计算”、“递归函数”、“ -可演算”这些数学概念正确刻画了人们关于“能行可计算”的理解。因而，丘奇-图灵论题不是一个数学命题。但我们仍然可以期待对它的一个论证。

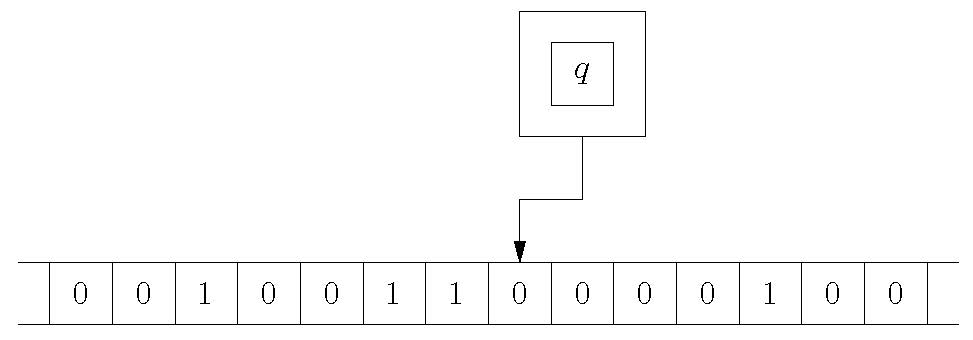
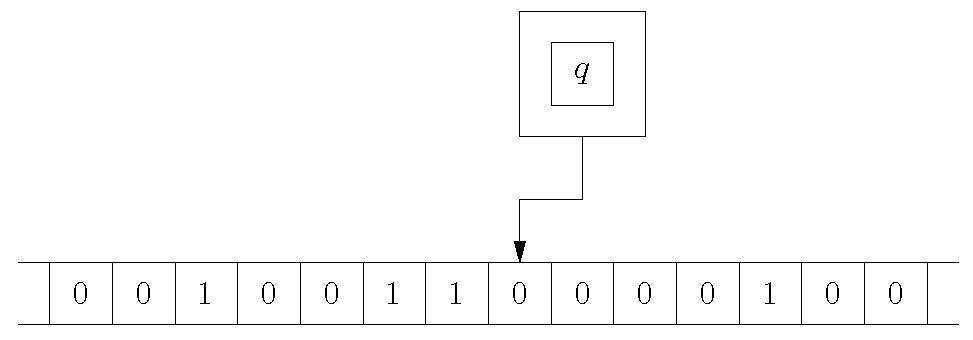
图灵本人在他1936年的文章[On Computable Numbers, with an Application to the Entscheidungsproblem](https://link.zhihu.com/?target=https%3A//doi.org/10.1112/plms/s2-42.1.230)中对丘奇论题的论证在我看来就很有说服力，甚至可以说是概念分析的典范。根据王浩[（Wang, 1996, 7.3.1）](https://link.zhihu.com/?target=https%3A//mitpress.mit.edu/books/logical-journey" \t "_blank)的报道，[哥德尔](https://www.zhihu.com/search?q=%E5%93%A5%E5%BE%B7%E5%B0%94&search_source=Entity&hybrid_search_source=Entity&hybrid_search_extra=%7B%22sourceType%22%3A%22answer%22%2C%22sourceId%22%3A320101790%7D)高度评价图灵对能行可计算的刻画（相对于更早的递归函数和 -可演算）及其论证：

我们之前就已经有了关于机械过程的精确概念，但直到了解了图灵的工作之后，我们才清晰地感知 (perceive) 到它。

接下来我们就看看图灵的论证。

**图灵机（Turing Machine）**

图灵机的定义很多人都很熟悉了，我就不具体码了：[https://en.wikipedia.org/wiki/Turing\_machine](https://link.zhihu.com/?target=https%3A//en.wikipedia.org/wiki/Turing_machine)

图灵机

关键是：双向无限纸带，有穷字母表，有穷内部状体，有穷指令集。

一个函数 是*图灵可计算*，当且仅当存在一个图灵机（你也可以理解成用任何图灵完全的语言写的一个程序），输入 ，有穷步内停机并输出 。

根据上述定义。人们一般不会怀疑图灵可计算的都是能行可计算的（其实还是有质疑的，又懒得码了）。所以关键是论证所有能行可计算的都存在一个图灵机来算。图灵本人也自觉地意识到这本质上不是一个数学的证明，而是诉诸于我们直观理解的论证：

【本文】到此为止尚未试图证明“【图灵】可计算的”数囊括了所有会被自然地认为是可计算的数【指实数等价于自然数上的函数】。对此，所有可以被给出的论证都注定是从根本上诉诸直观的，也因此是在数学上不令人满意的。

图灵区分了三种“从根本上诉诸直观的”论证：

1. 直接诉诸直观的论证；
2. 证明两种不同的定义等价；
3. 给出一大类“被自然地认为是可计算的数”，并证明它们是可计算的。

其中，后两条都是间接地诉诸直观的，即考虑人们对于那些直观的理解所形成的经验。第二条是对丘奇-图灵论题最常用的论证，即人们基于各自的直观给出了关于“能行”或“机械”的定义，却发现它们互相等价，以此作为所有可能的定义全都刻画了同一个直观的经验证据。第三条是指收集大量人们通过各自直观认定为可计算的函数并证明它们是图灵可计算的，从而经验地使人相信图灵可计算概念确实囊括了所有可能“被自然地认为是可计算的”东西。这两种归纳都是仅涉及正面证据，可以加强我们对于丘奇-图灵论题的信心，但仅此而已。我们这里要谈论的图灵的“直接诉诸直观的论证”

图灵理解的“能行可计算”即**原则上人能完成的计算**。图灵从对一个理想的计算者（computer。*现在，computer往往被用来指称计算机或电脑。而在当时，computer指的是那些被雇佣以科学研究或其他目的做计算的人，且往往是女性*）的计算过程的分析出发，试图论证**她**能计算的也可以找个图灵机来算。

图灵的基本想法是将她的操作分解为基本的“简单操作”以至于很难想象再怎么继续划分了。

首先，一个计算者总是需要使用一些纸张来工作。其次，需要有一套符号（字母表）供计算者在纸张上记录。我们可以假设纸上画着方格，符号必须写在方格中，正如儿童的写字练习簿一样。由于我们可以证明使用方格二维展开的纸张（无论是一片有无穷个格子二维展开的纸，还是无穷张有穷格子的纸张的序列）与使用一维的纸带（即纸带被分为格子向左右无穷展开）工作没有本质的区别。不妨假设计算者总在**一维纸带**上工作，并且可以假设**字母表总是有穷的**（图灵对字母表有穷的论证基于相应[度量空间](https://www.zhihu.com/search?q=%E5%BA%A6%E9%87%8F%E7%A9%BA%E9%97%B4&search_source=Entity&hybrid_search_source=Entity&hybrid_search_extra=%7B%22sourceType%22%3A%22answer%22%2C%22sourceId%22%3A320101790%7D)的紧致性，比那些单纯基于[物理主义](https://www.zhihu.com/search?q=%E7%89%A9%E7%90%86%E4%B8%BB%E4%B9%89&search_source=Entity&hybrid_search_source=Entity&hybrid_search_extra=%7B%22sourceType%22%3A%22answer%22%2C%22sourceId%22%3A320101790%7D)的有穷主义者高吧）。

如果字母表总是有穷的话，具体限制符号个数也变得不重要了。因为，“总是可以用[符号序列](https://www.zhihu.com/search?q=%E7%AC%A6%E5%8F%B7%E5%BA%8F%E5%88%97&search_source=Entity&hybrid_search_source=Entity&hybrid_search_extra=%7B%22sourceType%22%3A%22answer%22%2C%22sourceId%22%3A320101790%7D)来代替一个符号”。计算者在某一时刻会观察到一些纸带上的符号。然而，她在同一时刻至多观察到一定数量的符号。若如此，假设她**每个时刻只能观察到一个符号**也不会有本质的差别。

此外，计算者某一时刻的行为不仅仅取决于她所观察到的符号，还关系到她当下的[**心灵状态**](https://www.zhihu.com/search?q=%E5%BF%83%E7%81%B5%E7%8A%B6%E6%80%81&search_source=Entity&hybrid_search_source=Entity&hybrid_search_extra=%7B%22sourceType%22%3A%22answer%22%2C%22sourceId%22%3A320101790%7D)（state of mind）。图灵意识到这是一个复杂的对象，想要从生理学等角度来分析它还不太现实。这里，图灵聪明地设想了心灵状态的一种外部对应，而不是像当代一些认知科学家、脑科学家一样妄图用扫描大脑活跃片区等方法来解释心灵（图灵无疑是[功能主义](https://www.zhihu.com/search?q=%E5%8A%9F%E8%83%BD%E4%B8%BB%E4%B9%89&search_source=Entity&hybrid_search_source=Entity&hybrid_search_extra=%7B%22sourceType%22%3A%22answer%22%2C%22sourceId%22%3A320101790%7D)的先驱，但这里要分析的仅仅是计算功能，不涉及功能主义的各种争议）。

计算者总是可以暂停她的工作然后继续这项工作。如果她暂停了她的工作，她必须留下一条**指示笔记**（note of instructions），以某个标准形式书写，说明这项工作如何继续下去。这条笔记就是“心灵状态”的对应。我们假设这个计算者工作得如此断断续续以至于她每次坐下只能工作至多一步。指示笔记必须让她可以进行一步并且写出下一条笔记。

基于类似符号有穷的理由，人的**“心灵状态”或其对应的“指示笔记”也必须是有穷的**。否则，“其中的某些会变得‘任意接近’以至于被混淆”。只是这里我们没法像字母表那样把心灵状态的个数限制为某个特定的自然数。

计算者的每一步操作“完全取决于”她的当时的心灵状态（或指示笔记）以及观察到的纸带上的符号。她能做的操作，当被拆分得尽量简单时，无非是：(a) 改变纸带上的一个符号；(b) 改变关注的方格。我们可以假设计算者在每一步观察的方格和可能改变其中符号的方格都只有一个，而进一步假设这两个方格是同一个也不会带来本质的不同。此外，有理由假设当我们将所关注的方格切换到另一个时，这两个方格的距离总是在某个特定范围之内，如不超过 格。而这与假设每次**只能将注意力移至相邻方格上**是没有本质区别的。此外，当执行这些操作时，计算者的**心灵状态也可能会改变**。因此，“最一般的简单操作总是以下两者之一”：

1. 改变(也可能不改变)当前关注的方格里的符号，同时改变（也可能不改变）心灵状态；
2. 改变关注的方格为左右某个方格，同时改变（也可能不改变）心灵状态。

计算者每一步简单操作总是取决于她当前所注意的方格中的字符以及她当前的心灵状态，这种对应关系即所谓的[**指令集**](https://www.zhihu.com/search?q=%E6%8C%87%E4%BB%A4%E9%9B%86&search_source=Entity&hybrid_search_source=Entity&hybrid_search_extra=%7B%22sourceType%22%3A%22answer%22%2C%22sourceId%22%3A320101790%7D)。由于字母表、心灵状态和可能的操作都是有穷的，指令集也是有穷的。

分析至此，已经可以看出图灵的“计算者原则上可计算的”就是图灵可计算的。

严格来说，图灵的论证仍然有可商榷的地方。例如，假设能行可计算的必须是原则上人可计算的；假设计算者的操作总是完全取决于当前的心灵状态和观察到的符号；以及假设心灵状态总是有穷的。哥德尔就批评图灵关于只有有穷可辨别的“心灵状态”的论证是一个**哲学错误**（philosophical error）。他说：“图灵完全忽视了的是，心灵在发挥作用时不是静止的，而是不断发展的……因此，尽管在每一个阶段我们能够处理的抽象词项的数量和精度可能都是有穷的，但两者（因而也包括图灵的**可辨别心灵状态**）都可能在运用这套程序的过程中**收敛于无穷**。”哥德尔这些评价背后是他的实在论，这又是另一个故事了。

参考：

Why Gödel didn't have church's thesis, MartinDavis, [https://doi.org/10.1016/S0019-9958(82)91226-8](https://link.zhihu.com/?target=https%3A//doi.org/10.1016/S0019-9958%2882%2991226-8)

[SEP](https://link.zhihu.com/?target=https%3A//plato.stanford.edu/index.html)