**图灵1936年论文解读（1）：可计算性**

“可计算性（Computability）”是可计算性理论的核心概念，具有深刻的数学内涵和哲学底蕴，图灵、丘奇、哥德尔等前辈的工作为此概念打下了坚实的基础，应该说对此概念的理解已经不成问题了，然而从“NP是可计算的”流行观念看，此概念并未得到人们充分而正确的解读，这或许是造成千禧年难题“P versus NP”的最根本原因。

我们从回答“什么是可计算性？”说起，来解读学术界流行观点的回答，指出“可计算性”蕴含着“机器”与“问题”二个不同的层次：

一，学术界回答“什么是可计算性？”的典型答案

1，维基（[https://zh.wikipedia.org/wiki/可计算性](https://zh.wikipedia.org/wiki/%E5%8F%AF%E8%AE%A1%E7%AE%97%E6%80%A7" \t "_blank)）

可计算性（calculability）是指一个实际问题是否可以使用计算机来解决。

2，维基（[https://en.wikipedia.org/wiki/Computability](https://en.wikipedia.org/wiki/Computability" \t "_blank)）

Computability is the ability to solve a problem in an effective manner. It is a key topic of the field of computability theory within mathematical logic and the theory of computation within computer science. The computability of a problem is closely linked to the existence of an algorithm to solve the problem.

One goal of computability theory is to determine which problems, or classes of problems, can be solved in each model of computation.

3，Michael Sipser的书“Introduction to the Theory of Computation”

此书是计算理论领域的经典著作，常被大学选为教材。书中干脆就不谈“可计算性”，直接用“可计算函数”代替（p. 234）：

COMPUTABLE FUNCTIONS：A Turing machine computes a function by starting with the input to the function on the tape and halting with the output of the function on the tape.

4，Stanford Encyclopedia of Philosophy（<http://plato.stanford.edu/entries/computability/>）

A mathematical problem is computable if it can be solved in principle by a computing device.

二，解读“可计算性（Computability）”与“可计算的（computable）”

可见，上述回答实际上是在答“什么是可计算问题（computable problem）？”，而不是直接答“什么是可计算性（Computability）？”，那么，我们不禁要问：“computability”与“computable problem”是一回事吗？

对于“computable problem”的标准答案是由丘奇－图灵论题给出的（https://en.wikipedia.org/wiki/Church–Turing\_thesis）：

J. B. Rosser (1939) addresses the notion of "effective computability" as follows: "Clearly the existence of CC and RC (Church's and Rosser's proofs) presupposes a precise definition of 'effective'. 'Effective method' is here used in the rather special sense of a method each step of which is precisely predetermined and which is certain to produce the answer in a finite number of steps". Thus the adverb-adjective "effective" is used in a sense of "1a: producing a decided, decisive, or desired effect", and "capable of producing a result".

In the following, the words "effectively calculable" will mean "produced by any intuitively 'effective' means whatsoever" and "effectively computable" will mean "produced by a Turing-machine or equivalent mechanical device". Turing's "definitions" given in a footnote in his 1939 Ph.D. thesis Systems of Logic Based on Ordinals, supervised by Church, are virtually the same:

"† We shall use the expression 'computable function' to mean a function calculable by a machine, and let 'effectively calculable' refer to the intuitive idea without particular identification with any one of these definitions."

The thesis can be stated as follows:

Every effectively calculable function is a computable function.[8]

Turing stated it this way:

"It was stated ... that 'a function is effectively calculable if its values can be found by some purely mechanical process.' We may take this literally, understanding that by a purely mechanical process one which could be carried out by a machine. The development ... leads to ... an identification of computability† with effective calculability." († is the footnote above, ibid.)

“如果一个函数的值能被某个纯粹的机械过程求得，那么此函数就是能行可计算的”，可见，“可计算函数”是须要前提的，此前提就是存在着一台可以求解函数值的机器，换句话说，“可计算性”涉及到“问题”与“机器”二个层次，“机器的能力”为本，“问题的性质”为末。于是，用“可计算问题”代替“可计算性”，实际上是忽略了“可计算性”蕴含的二个层次的关系，从而造成认知混淆，。。。

所以，对于“可计算性”的理解应该回到对“机器的能力”的认知上，即回到对“图灵机”的认知上来。追本溯源，“图灵机”是在图灵1936年那篇重要论文《论可计算数及其在判定问题上的应用》（ On Computable Numbers, with an Application to the Entscheidungsproblem）中提出来的，让我们一起回到这篇具有历史性意义，但又令人望而生畏，至今使人无法深入的奠基性的论文。

三，《论可计算数及其在判定问题上的应用》中关于“图灵机”的论述

图灵在论文中开门见山指出，“按照我的定义，一个数是可计算的，如果它的十进制的表达能被机器写下来。”就是说，图灵强调预期的结果一定要被机器写下来，才能认为一个数是可计算的。

接下来，他开始设计这样的机器，将之称为“computing machine”。于“Computing machine”，图灵又区分了“circular machine”和“circle-free machine”，“circular machine”是因某些因素如“死循环”而无法写下计算结果的机器；而“circle-free machine”没有这样阻碍，能写下预期结果，换句话说， “circle-free”表达了图灵称之的“可计算性（computability）”。

然而，一个问题是否具有“computability”须要判断，这就是图灵这篇奠基性的论文的主题，回答“可计算性的判断”，进而回答著名的希尔伯特第十问题，。。。

让我们回到问题：“什么是可计算性？”，相对于流行答案“可计算性是指一个实际问题是否可以使用计算机来解决”，应该说：“可计算性是指算法（图灵机）具有解决一个实际问题的能力”，“可计算问题是指存在着具有可计算性的算法的问题”。。。

附：

1，图灵论文“On Computable Numbers, with an Application to the Entscheidungsproblem”原文：[https://www.cs.virginia.edu/~robins/Turing\_Paper\_1936.pdf](https://www.cs.virginia.edu/%7Erobins/Turing_Paper_1936.pdf" \t "_blank)

2，《论可计算数及其在判定问题上的应用》的前言，第一，二章部分译文：

“可计算数”简单说是，其十进制的表达用有限的手段可计算的实数。虽然本文的主题表面上讲可计算数，然而几乎可以同样容易定义和研究变量为整数或实数或可计算变量的可计算函数，可计算谓词等。在每种情况下，基本的问题是一样的，我选择可计算数来解释，是因为这样可以涉及最少的技术细节。不久我希望给出可计算数与可计算函数等之间的关系，这将包括用可计算数表达的实数变量的函数理论。按照我的定义，一个数是可计算的，如果它的十进制的表达能被机器写下来。

在§9,10，我给出一些论点，想指出可计算数包括所有的能自然看作可计算的数。比如，它们包括代数数的实数部分，Bessel函数的大小的实数部分，PI, e等等。然而可计算数并不包括所有可定义的数。

尽管可计算数类如此之大，在许多方面类似于实数类，它仍然是可枚举。在§8章，我调查某些证明似乎证明相反的论点。通过正确应用一个论点，可达成一个与哥德尔的结论表面上相似的结论。这些结果可有价值的应用，尤其是，指出（§11）Hilbertian Entscheidungsproblem没有解。

The “computable” numbers may be described briefly as the real numbers whose expressions as a decimal are calculable by finite means. Although the subject of this paper is ostensibly the computable numbers, it is almost equally easy to define and investigate computable functions of an integral variable or a real or computable variable, computable predicates, and so forth. The fundamental problems involved are, however, the same in each case, and I have chosen the computable numbers for explicit treatment as involving the least cumbrous technique. I hope shortly to give an account of the relations of the computable numbers, functions, and so forth to one another. This will include a development of the theory of functions of a real variable expressed in terms of computable numbers. According to my definition, a number is computable if its decimal can be written down by a machine.

In §§ 9, 10 I give some arguments with the intention of showing that the computable numbers include all numbers which could naturally be regarded as computable. In particular, I show that certain large classes of numbers are computable. They include, for instance, the real parts of all algebraic numbers, the real parts of the zeros of the Bessel functions, the numbers X, e, etc. The computable numbers do not, however, include all definable numbers, and an example is given of a definable number which is not computable.

Although the class of computable numbers is so great, and in many ways similar to the class of real numbers, it is nevertheless enumerable. In §8 I examine certain arguments which would seem to prove the contrary. By the correct application of one of these arguments, conclusions are reached which are superficially similar to those of Gödel [1] . These results {231}  have valuable applications. In particular, it is shown (§11) that the Hilbertian Entscheidungsproblem can have no solution.

1，计算机器（Computing machine）

我们已经说过，可计算数是那些可用有限的手段计算而得的表达为十进制的数，这需要较明确的定义。在第九章之前，不对定义作真正合理化的说明，目前我仅说理由是人类的记忆需要限制。

可将一个在计算实数的人与某机器比较，此机器具有有限数目的状态：q1;q2;…qk（m-格局，m-configurations）。给机器提供一条纸带（类似纸张），机器在上面运行，纸带被分为段（方格，squares），每个方格上放置一个符号。在任何时刻，只有一个方格，比如第r个方格，放置符号了S（r），“在机器里”，我们可以称此方格为“扫描方格”，“扫描方格”里的符号称为“扫描符号”，“扫描符号”是机器“直接感知”的唯一符号。然而，通过变化格局，机器可以有效记忆已经“见过（扫描）”的符号。机器在任何时刻可能的行为由m－格局qn，扫描符号S（r）决定，这对（qn，S（r））称为“格局”：于是，格局决定机器可能的行为。在有些格局，扫描方格是空的（即没有符号），机器就在扫描方格上写下一个新的符号；在别的格局，它擦掉扫描符号。机器也可能改变正在扫描的方格，但是仅仅向左右移动一个方格。此外，对任何一个这样的操作，m－格局可能变化。某些写下的符号将形成数字序列（ sequence of figures），它是正在计算的表达为十进制的实数，别 的符号只是“辅助记忆”的粗糙纪录，只有那些粗糙纪录可能被擦掉。

我的观点是，这些操作包括了所有用在计算一个数时所需的操作，为此论点辩护要在读者较熟悉机器理论后才会较容易。在下一节，我将着手发展此理论，假设读者已经知道“机器”，“纸带”，“扫描”等。

1. Computing machines.

We have said that the computable numbers are those whose decimals are calculable by finite means. This requires rather more explicit definition. No real attempt will be made to justify the definitions given until we reach §9. For the present I shall only say that the justification lies in the fact that the human memory is necessarily limited.

We may compare a man in the process of computing a real number to a machine which is only capable of a finite number of conditions q1, q2, ..., qR which will be called “m-configurations”. The machine is supplied with a “tape”, (the analogue of paper) running through it, and divided into sections (called “squares”) each capable of bearing a “symbol”. At any moment there is just one square, say the r-th, bearing the symbol S(r) which is “in the machine”. We may call this square the “scanned square”. The symbol on the scanned square may be called the “scanned symbol”. The “scanned symbol” is the only one of which the machine is, so to speak, “directly aware”. However, by altering its m-configuration the machine can effectively remember some of the symbols which it has “seen” (scanned) previously. The possible behaviour of the machine at any moment is determined by the m-configuration qn and the scanned symbol S(r). This pair qn, S(r) will be called the “configuration”: thus the configuration determines the possible behaviour of the machine. In some of the configurations in which the scanned square is blank (i.e. bears no symbol) the machine writes down a new symbol on the scanned square: in other configurations it erases the scanned symbol. The machine may also change the square which is being scanned, but only by shifting it one place to right or 1eft. In addition to any of these operations the m-configuration may be changed. Some of the symbols written down {232} will form the sequence of figures which is the decimal of the real number which is being computed. The others are just rough notes to “assist the memory”. It will only be these rough notes which will be liable to erasure.

It is my contention that these operations include all those which are used in the computation of a number. The defence of this contention will be easier when the theory of the machines is familiar to the reader. In the next section I therefore proceed with the development of the theory and assume that it is understood what is meant by “machine”, “tape”, “scanned”, etc.

2，定义

自动机（Automatic machines）

如果每个阶段机器的运动（在第一章的意义上）完全由格局确定，我们称此机器为«自动机»（a-machine）。

为了某种目的，我们可能使用一些机器（choice machines or c-machines），它的运动部分由格局决定。当这样的机器到达一个这样模棱两可的格局时，只有当外界的操作做某个任意的选择，它才能继续运行。如果我们用机器处理公理系统时会出现这样的情况，在本文中，我只处理自动机，因此往往忽略前缀a-。

计算机器（Computing machines）

如果一台机器打印两类符号，第一类（称为数字）全是0和1，其它被称为第二类符号，则机器将被称为“计算机器”。如果给机器装置一条空白纸带，让它运动起来，从正确的初始m－格局出发，机器打印的第一类符号的子序列称作“机器计算的序列”；在表达为二进制的十进制实数前放上小数点，称作“机器计算的数”。

在机器运动的任何阶段，扫描方格的数，在纸带上所有符号的完整序列，及m－格局，被说成是描述那时刻的“完全格局”。机器和纸带在一系列完整格局之间的变化被称作“机器的运动”。

循环和非循环机器（Circular and circle-free machines）

如果计算机不再写下有限数目的第一类符号，被称作“循环（Circular）”；否则，被称作“非循环（circle-free）”。

如果一台机器达到一个格局，从此不再运动，或者即使继续运动，只能打印第二类符号，而不能打印任何第一类符号，此机器则是“circular”。“circle-free”的意义将在第8章解释。

可计算序列和可计算数（Computable sequences and numbers）

一个序列被说成“可计算的”，如果能够通过一台“circle-free machine”计算而得。一个数是可计算的，如果它与由“circle-free machine”计算的数只差一个整数。

我们应该避免混淆，说可计算序列比说可计算数更经常。

2. Definitions.

Automatic machines.

If at each stage the motion of a machine (in the sense of §1) is completely determined by the configuration, we shall call the machine an “automatic machine” (or a-machine). For some purposes we might use machines (choice machines or c-machines) whose motion is only partially determined by the configuration (hence the use of the word “possible” in §1). When such a machine reaches one of these ambiguous configurations, it cannot go on until some arbitrary choice has been made by an external operator. This would be the case if we were using machines to deal with axiomatic systems. In this paper I deal only with automatic machines, and will therefore often omit the prefix a-.

Computing machines.

If an a-machine prints two kinds of symbols, of which the first kind (called figures) consists entirely of 0 and 1 (the others being called symbols of the second kind), then the machine will be called a computing machine. If the machine is supplied with a blank tape and set in motion, starting from the correct initial m-configuration, the subsequence of the symbols printed by it which are of the first kind will be called the sequence computed by the machine. The real number whose expression as a binary decimal is obtained by prefacing this sequence by a decimal point is called the number computed by the machine.

At any stage of the motion of the machine, the number of the scanned square, the complete sequence of all symbols on the tape, and the m-configuration will be said to describe the complete configuration at that stage. The changes of the machine and tape between successive complete configurations will be called the moves of the machine.{233}

Circular and circle-free machines.

If a computing machine never writes down more than a finite number of symbols of the first kind it will be called circular. Otherwise it is said to be circle-free.

A machine will be circular if it reaches a configuration from which there is no possible move, or if it goes on moving, and possibly printing symbols of the second kind, but cannot print any more symbols of the first kind. The significance of the term “circular” will be explained in §8.

Computable sequences and numbers.

A sequence is said to be computable if it can be computed by a circle-free machine. A number is computable if it differs by an integer from the number computed by a circle- free machine.

We shall avoid confusion by speaking more often of computable sequences than of computable numbers.

# 图灵1936年论文解读（2）：可计算数

图灵在1936年的论文（《论可计算数及其在判定问题上的应用》）的前言、第一、二章中提出了几个基本概念（见博文：可计算性——图灵1936年论文解读（1）），这些概念与当时数学和数理逻辑中一些观念不同，也与以前关于“算法”的直觉概念有别，图灵突出了机器计算的直观性，这些都是以后称之为“图灵机”的基础，这里我们先分析“可计算数（computable number）”。

一，可计算数

在可计算性理论中，人们一般讨论“可计算函数”，然而图灵的论文谈及的是“可计算数”。图灵之所以谈“可计算数”，是因为“虽然本文的主题表面上讲可计算数，然而几乎可以同样容易定义和研究变量为整数或实数或可计算变量的可计算函数，可计算谓词等。在每种情况下，基本的问题是一样的，我选择可计算数来解释，是因为这样可以涉及最少的技术细节”［1］。

图灵说：“可计算的”数，简单地说就是其十进制形式可用有限手段计算出来的实数（The “computable” numbers may be described briefly as the real numbers whose expressions as a decimal are calculable by finite means.）。这个定义实质上就把“可计算的”用“有限手段的计算”定义了，后者就指机器结构和过程，也就是论文以下的“图灵机”的思想实现。

图灵进一步强调：按照我的定义，一个数是可计算的，如果它的十进制的表达能被机器写下来（According to my definition, a number is computable if its decimal can be written down by a machine.）。这样就明确地把数学意义上的“可计算的”数表达为“可计算的”机械过程，揭示了算法的“能行性”本质（见博文：NP理论的基本概念（2）：“判定问题”与“停机问题”）。

图灵强调可计算数一定要由机器写下来，这是因为机器能得到的最后结果一定是确定的，以此倒推机器的算法步骤关系，这是图灵的思想实验最关键的一步，由此揭示了“机械步骤”的实时（the Actual Time）性本质。这正是“算法”这个直觉概念所隐含的一个本质，也是“图灵机”的本质特征。

二，例子

图灵在论文的第三章给出一个可计算数（序列）：010101...，计算此序列的机器就是在第一、第二章定义Computing Machine：

此机器有四个状态（m-configuration）：“ b”,  “c”, “z”, “e” ，能打印“0”和“1”。机器的行为被描述在下面的表中，其中“R” 表示机器移动去扫描当前方格的右边方格。“P” 表示“打印”。这个表被理解成，机器在头二列表达的格局下，执行第三列的操作，进入最后一列的状态。当第二列是空白时，可理解为第三第四列的行为应用于任何符号和无符号。

格局             行为

b     None     P0,R    c

c     None     R         e

e     None     P1,R    z

z     None     R         b

机器开始于状态b和一条空白纸带，打印完01又回到状态b，再重复打印01，形成无限长的字符串010101...，这就是一个最简单的机器计算步骤过程。

参考文献

［1］《论可计算数及其在判定问题上的应用》的第三章部分原文（[https://www.cs.virginia.edu/~robins/Turing\_Paper\_1936.pdf](https://www.cs.virginia.edu/%7Erobins/Turing_Paper_1936.pdf" \t "_blank)）：

3. Examples of computing machines.

I. A machine can be constructed to compute the sequence 010101.... The machine is to have the four m-configurations “ b”, “c”, “z”, “e” and is capable of printing “0”, and “1”. The behaviour of the machine is described in the following table in which “R” means “the machine moves so that it scans the square immediately on the right of the one it was scanning previously”. Similarly for “L”. “E” means the scanned symbol is “erased” and “P” stands for “prints”. This table (and all succeeding tables of the same kind) is to be understood to mean that for a configuration described in the first two columns the operations in the third column are carried out successively, and the machine then goes over into the m-configuration described in the last column. When the second column is left blank, it is understood that the behaviour of the third and fourth columns applies for any symbol and for no symbol. The machine starts in the m-configuration b

with a blank tape.

Configuration Behaviour

b     None     P0,R     c

c     None     R          e

e     None     P1,R     z

z     None     R          b

# 图灵1936年论文解读（3）：“不停机”的circle－free

在图灵1936年论文第二章，图灵用circle-free来定义“可计算数（Computable sequences and numbers）”，这个定义实际上是由二部份构成的，首先由circle-free来定义“可计算序列”，进一步由“可计算序列”定义“可计算数”。

按照图灵的定义：

A sequence is said to be computable if it can be computed by a circle-free machine. 一个序列被说成“可计算的”，如果能够通过一台“circle-free machine”计算而得。

A number is computable if it differs by an integer from the number computed by a circle-free machine. 一个数是可计算的，如果它与由“circle-free machine”计算的数只差一个整数。

我们以图灵所举的一个可计算序列：010101...为例来理解circle-free与“可计算序例”与“可计算数”的定义关系。图灵的意思是：一个可计算的数（A number is computable）只与circle-free计算的数（the number computed by a circle-free machine）相差一个整数，这就是指“小数”。因此，可计算序列“010101...”对应一个（可计算的）二进制无限循环小数“.010101...”，就是十进制小数：2^(-2)+2^(-4)+2^(－6)＋ ...，即0.333...。

由这个实例的递进定义来理解图灵的“可计算数”是理解图灵论文的关键之一。无限循环小数表达了机器意义的“可计算数”的最大能力，这个最大能力是由（不停机的）circle－free的机器过程实现的。得到确定性的结果而“停机”只是circle-free的特殊情况，因此图灵并不是想由机器直接写出一个可计算数（而停机），“停机”是对已经存在的一台具体可计算机器而言，这正是把“停机问题”顶替“判定问题”所发生的对图灵1936年论文误读的原因。所以图灵并不关心“停机”，图灵所做的是揭示机器自身的机械步骤过程的算法性质，即机器化的“算法”——图灵机。

因此这和一般人观念中的机器的“可计算性”完全不同，通常认为，一台计算机器如果不输出最终结果而总是处于计算之中，则这台机器（算法）就进入了“死循环”，被认为是没有用的。产生这种差别的原因在于普通人只是把计算机当作一台现成的计算工具使用，而图灵是计算机器的建造师，他只有首先保证机器具有无限的运行能力（circle-free），然后才有可能让机器“停机”而得到某个解。