The Decision Problem

Das Entscheidungsproblem

An Historical Project

Alan Turing's 1936 paper "On Computable Numbers with an Application to the Entscheidungsproblem" [3] proved most influential not only for mathematical logic, but also for the development of the programmable computer, and together with work of Alonzo Church (1903–1995) [1, 2] inaugurated a new field of study, known today as computability. Recall that Turing's original motivation for writing the paper was to answer the decision problem of David Hilbert (1862–1943), which asked whether there is a standard procedure that can be applied to decide whether an arbitrary statement (within some system of logic) is provable. A previous project examined the construction of Turing's "universal computing machine," which accepts the instructions of any other machine M in standard form, and then outputs the same sequence as M. The concept of a universal machine has evolved into what now is known as a compiler or interpreter in computer science, and is indispensable for the processing of any programming language. The question then arises, does the universal computing machine provide a solution to the decision problem? The universal machine is the standard procedure for answering all questions that can in turn be phrased in terms of a computer program.

First, study the following excerpts from Turing's paper [3, p. 232–233]

Automatic machines.

If at each stage the motion of a machine is *completely* determined by the configuration, we shall call the machine an "automatic machine" (or a-machine). . . .

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 symbols printed by it which are of the first kind will called the *sequence computed by the machine*. ...

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 cannnot print any more symbols of the first kind.

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. ...

(a) Consider the following machine, T_1, which begins in m-configuration a with a blank tape, reading the blank at the far left. Is T_1 circle-free? Justify your answer.

T_1 :

Configuration		Behavior	
m-config.	symbol	operation	final m-config.
a	blank	$R,\ P(1)$	b
a	0	R	b
b	1	$R,\ R,\ P(0)$	a
b	blank	(none)	a

(b) Consider the following machine, T_2, which begins in m-configuration a with a blank tape, reading the blank at the far left. Is T_2 circle-free? Justify your answer.

T_2 :

Configuration		Behavior	
m-config.	symbol	operation	final m-config.
a	blank	$R,\ P(1)$	b
a	0	R	b
b	1	$R,\ R,\ P(0)$	a
b	0	R	a

(c) Describe in your own words the key feature which distinguishes a circle-free machine from a circular machine.

(d) Is the sequence 101001000100001 ... computable? If so, find a circle-free machine (with a finite number of m-configurations) that computes this sequence on every other square (the F-squares) of a tape which is originally blank. If not, prove that there is no circle-free machine that computes the above sequence.

Turing's insight into the decision problem begins by listing all computable sequences in some order:

$$\phi_1, \ \phi_2, \ \phi_3, \ \ldots, \ \phi_n, \ \ldots,$$

where ϕ_n is the n-th computable sequence. Moreover, let $\phi_n(k)$ denote the k-th figure (0 or 1) of ϕ_n. For example, if

$$\phi_2 = 101010 \ldots,$$

then $\phi_2(1) = 1$, $\phi_2(2) = 0$, $\phi_2(3) = 1$, etc. Turing then considers the sequence β' defined by $\beta'(n) = \phi_n(n)$. If the decision problem has a solution, then:

we can invent a machine D which, when supplied with the $S.D$ [standard description] of any computing machine M will test this $S.D$ and if M is circular will mark the $S.D$ with the symbol "u" [unsatisfactory] and if it is circle-free will mark it with "s" [satisfactory]. By combining the machines D and U [the universal computing machine] we could construct a machine H to compute the sequence β' [3, p. 247].

(e) Is the number of computable sequences finite or infinite? If finite, list the computable sequences. If infinite, find a one-to-one correspondence between the natural numbers, **N**, and a subset of the computable sequences. Use the result of this question to carefully explain why H must be circle-free.

(f) Since H is circle-free, the sequence computed by H must be listed among the ϕ_n's. Suppose this occurs for $n = N_0$. In a written paragraph, explain how $\beta'(N_0)$ should be computed. Is it possible to construct a machine H that computes β'? If so, find the configuration table for H. If not, what part of H, i.e., D or U, cannot be constructed? Justify your answer.

(g) Does the universal computing machine solve the decision problem? Explain.

(h) By what name is the decision problem known today in computer science? Support your answer with excerpts from outside sources.

REFERENCES

[1] Church, A., "An Unsolvable Problem of Elementary Number Theory," *American Journal of Math.*, **58** (1936), 345–363.

[2] Church, A., "A Note on the Entscheidungsproblem," *Journal of Symbolic Logic*, **1** (1936), 40–41.

[3] Turing, A., "On Computable Numbers with an Application to the Entscheidungsproblem," *Proceedings of the London Mathematical Society*, **42** (1936), 230–265.