The Universal Computing Machine

An Historical Project

The decision problem of David Hilbert (1862–1943) asked whether there is a standard procedure, an algorithm in modern terminology, which can be invoked to decide whether an arbitrary statement (within some system of logic) is valid. In answering this question, Alan Turing (1912–1954) introduced several fundamental concepts, certainly of importance to logic, but also pivotal to the development of the modern programmable computer. The first of these is a "computing machine," called a "Turing machine" today, which is the forerunner of a modern computer program. The next concept is the "universal computing machine," which is in fact a particular type of Turing machine that accepts the instructions of some other machine M in standard form, and the outputs the same sequence as M. Turing writes: "It is possible to invent a single machine which can be used to compute any computable sequence. If this machine U is supplied with a tape on the beginning of which is written the $S.D$ [standard description] of some computing machine M, then U will compute the same sequence as M" [1, p. 241–242]. After reading the excerpts at the end of the project from Turing's original paper [1], answer the questions:

(a) What is the output of the following machine, T, if T begins in configuration a with a blank tape, scanning the blank at the far left?

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

(b) Rewrite the output of machine T using the "standard description" for the output symbols, i.e., S_0 = blank, $S_1 = 0$, $S_2 = 1$, and then replace each S_j with D followed by C repeated j times.

(c) What is the standard description ($S.D$) of machine T? Be sure that every instruction, including the last one, is followed by a semi-colon. Make sure that you have the correct answer to this before continuing.

(d) Suppose that the number of configurations for a given machine M is limited to nine, while the number of symbols which M can recognize or write is limited to four. The machine M begins in its first listed configuration (a) with a blank tape, reading the blank at the far left. Now suppose that the standard description of M is written on every other square (in fact the F-squares) of a second tape, with each instruction followed by a semi-colon (on an F-square as well), with the last semi-colon followed by the symbol "::" on an F-square. Initially all other squares on this second tape are blank. If the standard description for machine I of §3 in Turing's paper (see attachment) is entered on tape in this way, what is the output of the following machine U, which begins in configuration one reading the tape at the far left? Note that 20R is shorthand for move 20 squares to the right, and similarly for 10R. How does this compare with the actual output of machine I, §3?

Configuration		Behavior	
m-config.	symbol	operation	final m-config.
1	D	R	1
1	A	R	2
1	::	none	none
1	other	R	1
2	blank	R	2
2	A	R	4
2	D	R, R	3
3	D	R, P(X)	5
3	C	R	4
4	;	R	1
4	::	none	none
4	other	R	4
5	::	20R, P(Y), R, R, P(D), 10R, P(Z)	6
5	other	R	5
6	X	E, R	7
6	other	L	6
7	C	R, P(X)	8
7	R	R, P(X)	10
7	L	none	none
7	N	R, P(X)	11
8	Y	4R	9
8	other	R	8
9	blank	P(C)	6
9	C	R, R	9
10	Z	E, P(T)	12
10	other	R	10
11	Y	R, P(T)	12
11	other	R	11
12	X	E, 4R, P(X)	13
12	other	L	12
13	::	R, R	14
13	other	R	13
14	blank	P(A)	15
14	Y	none	none
14	other	R, R	14
15	X	E, R	16
15	other	L	15
16	;	none	17
16	A	R, P(X)	13

(e) If the standard description of machine T from part (a) is written on tape as described in (d), what is the output of U when applied to this tape? How does this compare to the actual output of T?

(f) Describe in words the operation of configuration one from machine U. Describe separately the operations of configurations two, three, and four.

(g) Note that only the first 16 configurations of U are listed. From this point, if U was originally supplied with the standard description of a machine M (as specified in (d)), then U should output the same sequence as M, except coded according to the standard description of the output symbols. Moreover, U keeps track of the current configuration of M in the first 18 squares immediately following "::". The position of the scanner for M is recorded via the symbol "T" on the tape which U processes. Beginning with configuration 17, outline the remaining operations of U. You may use phrases such as "match m-configuration," "match scanned symbol," or " move scanner for M," etc. Be sure to include a written explanation of these and any other operations you decide to use in your outline. For this part, you do not need to design an actual Turing machine to perform these tasks. Does recursion occur in your outline? How?

(h) Beginning with configuration 17, find the actual machine instructions of U so that U finds a match between an arbitrary configuration stored on the 18 squares to the right of "::" and a configuration at the beginning of a coded instruction to the left of "::". Suppose that the standard description entered on tape is that of a machine M for which there is always a well-defined configuration to follow every move of M. Use only the operations "R," "L," "E," "P( )," "none," and be sure to explain the new steps of U. What is the present-day terminology used to describe U?

EXTRA-CREDIT: Write a computer program for a universal computing machine (in the language of your choice). Demonstrate with several examples that your universal machine functions properly.

For this project, the relevant excerpts from Turing's original paper are, first [1, p. 233]:

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", "f", "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 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. (Example 1).

Configuration		Behaviour	
m-config.	symbol	operation	final m-config.
b	none	P0, R	c
c	none	R	e
e	none	P1, R	f
f	none	R	b

Next, [1, p. 239–241]:

5. *Enumeration of computable sequences.*

A computable sequence γ is determined by a description of a machine which computes γ. Thus the sequence 001011011101111... is determined by the table on p. 234, and, in fact, any computable sequence is capable of being described in terms of such a table.

It will be useful to put these tables into a kind of standard form. In the first place let us suppose that the table is given in the same form as the first table, for example, 1 on p. 233. That is to say, that the entry in the operations column is always one of the form E: E, R: E, L: $P\alpha$: $P\alpha, R$: $P\alpha, L$: R: L: or no entry at all. The table can always be put into this form by introducing more m-configurations. Now let us give numbers to the m-configurations, calling them $q_1, \ldots, q_R$ as in §1. The initial m-configuration is always to be called q_1. We also give nubers to the symbols $S_1, \ldots, S_m$ and, in particular, blank $= S_0$, $0 = S_1$, $1 = S_2$. The lines of the table are now of form

m-config.	*Symbol*	*Operations*	*Final m-config.*	
q_i	S_j	PS_k, L	q_m	(N_1)
q_i	S_j	PS_k, R	q_m	(N_2)
q_i	S_j	PS_k	q_m	(N_3)

Lines such as

$$q_i \quad S_j \quad E, R \quad q_m$$

are to be written as

$$q_i \quad S_j \quad PS_0, R \quad q_m$$

and lines such as

$$q_i \quad S_j \quad R \quad q_m$$

to be written as

$$q_i \quad S_j \quad PS_j, R \quad q_m$$

In this way we reduce each line of the table to a line of one of the forms (N_1), (N_2), (N_3).

From each line of form (N_1) let us form an expression $q_i\, S_j\, S_k\, L\, q_m$; from each line of form (N_2) we form an expression $q_i\, S_j\, S_k\, R\, q_m$; and from each line of form (N_3) we form an expression $q_i\, S_j\, S_k\, N\, q_m$.

Let us write down all expressions so formed from the table for the machine and separate them by semi-colons. In this way we obtain a complete description of the machine. In this description we shall replace q_i by the letter "D" followed by the letter "A" repeated i times, and S_j by "D" followed by "C" repeated j times. This new description of the machine may be called the *standard description* (S.D). It is made up entirely from the letters "A", "C", "D", "L", "R", "N", and from ";". ...

Let us find a description number for the machine I of §3. When we rename the m-configurations its table becomes:

m-config.	Symbol	Operations	Final m-config.
q_1	S_0	PS_1, R	q_2
q_2	S_0	PS_0, R	q_3
q_3	S_0	PS_2, R	q_4
q_4	S_0	PS_0, R	q_1

Other tables could be obtained by adding irrelevant lines such as

$$q_1 \quad S_1 \quad PS_1, R \quad q_2$$

Our first standard form would be

$$q_1\, S_0\, S_1\, R\, q_2;\;\; q_2\, S_0\, S_0\, R\, q_3;\;\; q_3\, S_0\, S_2\, R\, q_4;\;\; q_4\, S_0\, S_0\, R\, q_1;\; .$$

The standard description is

$$DADDCRDAA; DAADDRDAAA; DAAADDCCRDAAAA; DAAAADDRDA;$$

Continuing from [1, p. 243]:

Each instruction consists of five consecutive parts

(i) "D" followed by a sequence of letters "A". This describes the relevant m-configuration.

(ii) "D" followed by a sequence of letters "C". This describes the scanned symbol.

(iii) "D" followed by another sequence of letters "C". This describes the symbol into which the scanned symbol is to be changed.

(iv) "L", "R", or "N", describing whether the machine is to move to left, right, or not at all.

(v) "D" followed by a sequence of letters "A". This describes the final m-configuration.

REFERENCES:

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