An Introduction to Turing Machines

An Historical Project

During the International Congress of Mathematicians in Paris in 1900 David Hilbert (1862–1943), one of the leading mathematicians of the last century, proposed a list of problems for the following generations to ponder. On the list was whether the axioms of arithmetic are consistent, a question which would have profound consequences for the foundations of mathematics. Continuing in this direction, in 1928 Hilbert proposed the decision problem (das Entscheidungsproblem), which asked whether there was a standard procedure that can be applied to decide whether a given mathematical statement is true. In a revolutionary paper "On Computable Numbers, with an Application to the Entscheidungsproblem" published in 1936, Alan Turing (1912–1954) proved that the decision problem had no solution, and in doing so he outlined the rudimentary ideas which form the basis for the modern programmable computer. Today his construction is known as a Turing machine.

The goal of this project is to read a few excerpts from Turing's original paper[1] and outline a machine that would verify the condition of set containment, a topic discussed in class. After carefully reading the attached pages, answer the following:

(a) Describe the workings of a Turing machine (referred to as a "computing machine" in the original paper).

(b) What is the precise output of the machine in Example 1. Be sure to justify your answer.

(c) Design a Turing machine which generates the following output. Be sure to justify your answer.

$$010010100101001\ \ldots$$

(d) Describe the behavior of the following machine, which begins with a blank tape, with the machine in configuration α.

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

Configuration		Behavior	
m-config.	symbol	operation	final m-config.
α	none	R P1	β
α	1	R P0	β
α	0	HALT	(none)
β	1	R P1	α
β	0	R P0	α

(e) Given finite, non-empty, sets A and B, design a Turing machine which tests whether $A \subseteq B$. Suppose that the first character on the tape is a 0, simply to indicate the beginning of the tape. To the right of 0 follow the (distinct, non-blank) elements of A, listed in consecutive positions, followed by the symbol &. To the right of & follow the (distinct, non-blank) elements of B, listed in consecutive positions, followed by the symbol Z to indicate the end of the tape:

0			. . .		&		. . .			Z

The symbols 0, &, Z are neither elements of A nor B. The machine starts reading the tape in the right most position, at Z. If $A \subseteq B$, have the machine erase all the elements of A and return a tape with blanks for every square which originally contained an element of A. You may use the following operations for the behavior of the machine:

- R: Move one position to the right.
- L: Move one position to the left.
- S: Store the scanned character in memory. Only one character can be stored at a time.
- C: Compare the currently scanned character with the character in memory. The only operation of C is to change the final configuration depending on whether the scanned square matches what is in memory.
- E: Erase the currently scanned square
- P( ): Print whatever is in parentheses in the current square.

You may use multiple operations for the machine in response to a given configuration. Also, for a configuration q_n, you may use the word "other"

to denote all symbols $\mathcal{S}(r)$ not specifically identified for the given q_n. Be sure that your machine halts.

ON COMPUTABLE NUMBERS, WITH AN APPLICATION TO THE ENTSCHEIDUNGSPROBLEM

By A. M. Turing.

1. *Computing Machines.*

We have said that the computable numbers are those whose decimals are calculable by finite means. This requires more explicit definition. No real attempt will be made to justify the definitions given until we reach §9. For 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 q_1, q_2, ..., q_R, which will be called the "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 $\mathcal{S}(r)$ 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 it has "seen" (scanned) previously. The possible behaviour of the machine at any moment is determined by the m-configuration q_n and the scanned symbol $\mathcal{S}(r)$. This pair q_n, $\mathcal{S}(r)$ 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 left. In addition to any of these operations the m-configuration may be changed. Some of the symbols written down 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 my contention that these operations include all those which are used in the computation of a number. The defense 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. *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 printed 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.

3. *Examples of computing machines.*

I. A machine can be constructed to compute the sequence $010101\ldots$. 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

If (contrary to the description §1) we allow the letters L, R to appear more than once in the operations column we can simplify the table considerably.

Configuration		Behaviour	
m-config.	symbol	operation	final m-config.
b	none	P0	b
b	0	R, R, P1	b
b	1	R, R, P0	b