Turing Machines, Induction and Recursion

An Historical Project

The logic behind the modern programmable computer owes much to Turing's "computing machines," discussed in the first project, which the reader should review. Since the state of the machine, or m-configuration as called by Turing, can be altered according to the symbol being scanned, the operation of the machine can be changed depending on what symbols have been written on the tape, and affords the machine a degree of programmability. The program consists of the list of configurations of the machine and its behavior for each configuration. Turing's description of his machine, however, did not include memory in its modern usage for computers, and symbols read on the tape could not be stored in any separate devise. Using a brilliant design feature for the tape, Turing achieves a limited type of memory for the machine, which allows it to compute many arithmetic operations. The numbers needed for a calculation are printed on every other square of the tape, while the squares between these are used as "rough notes to 'assist the memory.' It will only be these rough notes which will be liable to erasure" [1, p. 232].

Turing continues: The convention of writing the figures only on alternate squares is very useful: I shall always make use of it. I shall call the one sequence of alternate squares F-squares, and the other sequence E-squares. The symbols on E-squares will be liable to erasure. The symbols on F-squares form a continuous sequence. ... There is no need to have more than one E-square between each pair of F-squares: an apparent need of more E-squares can be satisfied by having a sufficiently rich variety of symbols capable of being printed on E-squares [1, p. 235].

Let's examine the Englishman's use of these two types of squares. Determine the output of the following Turing machine, which begins with the tape

X			...			

and the scanner at the far left, reading the symbol X.

Configuration		Behavior	
m-config.	symbol	operation	final m-config.
a	X	R	a
a	1	R, R	a
a	blank	P(1), R, R, P(1), R, R, P(0)	b
b	X	E, R	c
b	other	L	b
c	0	R, P(X), R	a
c	1	R, P(X), R	d
d	0	R, R	e
d	other	R, R	d
e	blank	P(1)	b
e	other	R, R	e

Recall the meaning of the following symbols used for operations.

- R: Move one position to the right.
- L: Move one position to the left.
- E: Erase the currently scanned square
- P( ): Print whatever is in parentheses in the current square.

(a) What is the precise output of the machine as it just finishes configuration a and enters configuration b for the first time? Justify your answer.

(b) What is the precise output of the machine as it just finishes configuration a and enters configuration b for the second time? Justify your answer.

(c) What is the precise output of the machine as it just finishes configuration a and enters configuration b for the third time? Justify your answer.

(d) Guess what the output of the machine is as it just finishes configuration a and enters configuration b for the n-th time. Use induction to prove that your guess is correct. Be sure to carefully write the details of this proof by induction.

(e) Design a Turing machine, which when given two arbitrary natural numbers, n and m, will compute the product $n \cdot m$. Suppose that the machine begins with the tape

A	1		1		...		1		1	B	1		1		...		1		1	C

where the number of ones between A and B is n, the number of ones between B and C is m, and the machine begins scanning the tape at the far left, reading the symbol A. The output of the machine should be:

A	1		...	1	B	1		...	1	C	1		1		...		1	D

where the number of ones between C and D is $n \cdot m$. Be sure to justify your answer.

Letting T denote the Turing machine which multiplies n and m together, so that the value of $T(n, m)$ is $n \cdot m$, design T so that for $n \in \mathbf{N}$,

$$T(n, 1) = n$$

and for $m \in \mathbf{N}$, $m \geq 2$, we have

$$T(n, m) = T(n, m - 1) + n.$$

Such an equation provides an example of a recursively defined function, an important topic in computer science. In our case, the algorithm for multiplication, T, is defined in terms of addition, a more elementary operation.

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.