UNIVERSITY OF WATERLOO

Faculty of Mathematics
Waterloo, Ontario N2L 3G1

Centre for Education in
Mathematics and Computing

Grade 7/8 Math Circles

Wednesday, March 31, 2021

Turing Machine

Introduction

Alan Turing (1912 – 1954) is a British mathematician and logician. He is among the founding fathers of modern computing and computer science with major contributions to cognitive science, artificial intelligence and artificial life. Additionally, he has made advancements in mathematics, cryptanalysis, logic, philosophy, and mathematical biology.

Turing is well-known for his paper, "On Computable Numbers, with an Application to the Entscheidungsproblem (Decision Problem)", published in 1936. In this article, he sought out to prove that any mathematical problem could potentially solved in an algorithmic or mechanical process.

Interpreting the solution to this problem as a machine, Turing began to design a computing machine that would be able to resolve all mathematical problems. He discovered that no such universal decision method exists and no formal system could reduce all of mathematics to computational problems. However, in the process of designing his machine, the Turing machine, he summarized the fundamental logistics of what would eventually be the digital computer.

War Hero

At the outbreak of the second world war, Turing joined an elite team of mathematician-cryptanalysts. Together, they created a code-breaking machine named the Bombe, to decipher German messages encoded by their cipher device, Enigma. From its inception in 1940 to the rest of the war, the cryptanalysts utilized the Bombe machine to decode intercepted messages and supply the Allies with military intelligence.

Parts to a Turing Machine

A Turing machine is a hypothetical device that very well defined computation and questioned the limits of computation and human thought. It is the simplest form of the computer and

the earliest conception of the idea of computers. These machines serve as an ideal computing device and model for mathematical calculation.

That means, that while we can theoretically understand and use the device that Turing designed, it is not possible to replicate in real-life due to the optimal conditions that the device works in. For example, the machine does not consider quantity of information, processing time, and materials.

There are a few parts to the Turing machine. First is a head that acts as a scanner and both reads and writes the data passing through it. The data is stored in the form of an infinitely long paper tape divided into squares with each square bearing a simple symbol. In this lesson, we will consider the symbols '0' and '1' in our explanations and examples.

Inside the head is a second form of working memory called the indicator. At any point in time, the indicator is set to one of a number of 'positions', specifying the machine's present state. For instance, it may track the last symbol that was encountered.

During a computation, the machine performs one of five different operations:

1. read/identify the symbol on the square under the head
2. edit the symbol by either i) writing a new one and replacing the first or ii) erasing the current symbol
3. move the tape one square to the left or right
4. halt/stop
5. change state

The computers that we use on a daily basis are designed to perform operations much more complex than the those listed above. Despite its simplicity, the Turing machine can perform the same computations as any modern computer due to its abstract nature. Furthermore, a Turing machine is an *ideal* device and therefore, not limited by real-world constraints, such as a finite amount of memory.

Examples

In this section, we will go over some examples to demonstrate how a Turing machine operates.

Example 1

The machine in this example will be given an endless blank tape to start. We want to set up the machine such that, when in motion, it prints alternating binary digits, 0 1 0 1 ...,

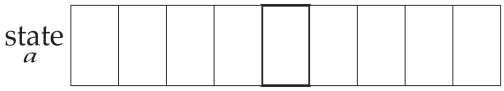
on the tape. It should work to the right from its starting place and leave a blank square in between each digit.

For this to occur, the machine will utilize four states. We will call them a, b, c, and d with the machine initially beginning in state a. Then, the instructions for the machine can be summarized in the following table.

State	Scanned Symbol	Print	Move Tape	Next State
a	blank	0	left	b
b	blank		left	c
c	blank	1	left	d
d	blank		left	a

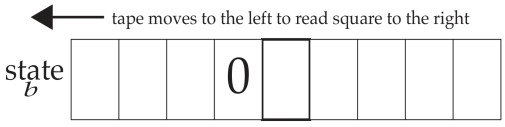
So, the first cycle of activity from the machine will occur as such.

1. Recall that the machine is given an infinitely long blank tape. The bold square represents the square under the head to start. This could be any square on the tape.

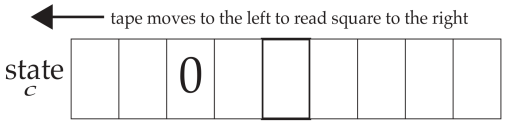


2. The machine starts in state a. It reads a blank symbol and thus, prints a 0.

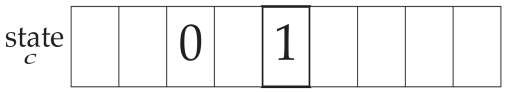
3. Now, the tape is moved to the left while the scanner stays in the same position so, it will next read the symbol one square to the right. The machine simultaneously changes to state b.



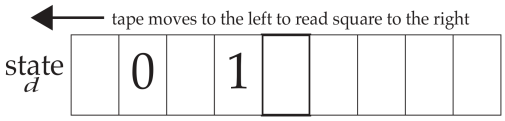
4. The machine is now in state b. According to the table of commands above, when a blank symbol is read while the machine is in state b, nothing is printed. Instead, the tape is moved to the left so the square to the right is under the scanner, and the machine changes to state c.



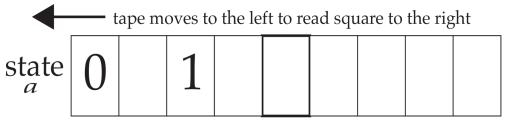
5. The machine is in state c. Therefore, it prints a 1 when it reads the blank symbol.



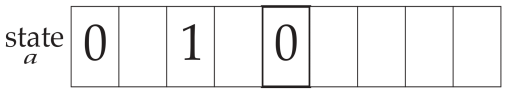
6. Then, the tape moves to the left and the square to the right is under the scanner. The machine changes to state d.



7. With the machine in state d, it reads a blank symbol and does not print anything. The tape moves to the left and the scanner is over the square to the right and returns to state a.



8. Finally, the machine is once again in state a. It reads a blank symbol and prints a 0. After, it will move one square to the right and change to state b.



This machine will continue on endlessly, printing the desired alternate sequence of digits. However, this infinite process does not show the halting capabilities of the machine and is not realistic for everyday purposes.

Example 2

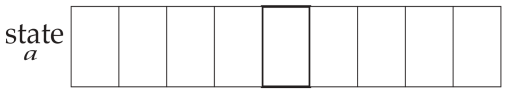
Let's take a look at a similar example as the program above. This time, the machine will simply print the sequence of digits "1 1 0". So, we get the following state table:

State	Scanned Symbol	Print	Move Tape	Next State
a	blank	1	left	b
b	blank	1	left	c
c	blank	0	left	stop state

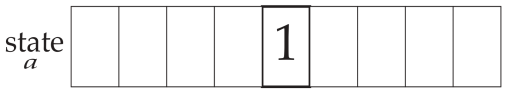
Then, the program will run as follows.

1. Recall that the machine is given an infinitely long blank tape. The bold square repre-

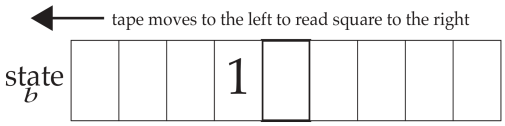
sents the square under the head to start. This could be any square on the tape.



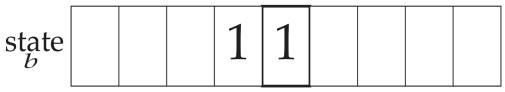
2. The machine starts in state a. It reads a blank symbol and thus, prints a 1.



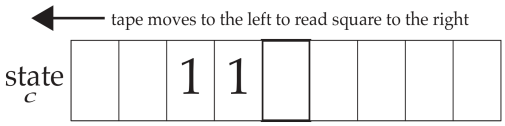
3. The tape is moved to the left while the scanner stays in the same position so, it will read the symbol in the square to the right. The machine now changes to state b.



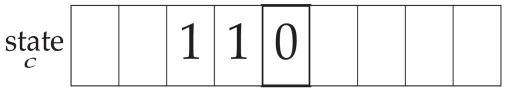
4. The machine is now in state b. When it reads a blank symbol in state b, it again prints a 1.



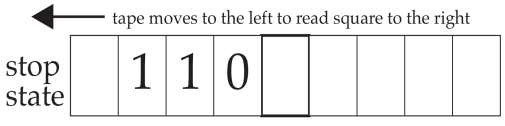
5. The tape is moved to the left so the square to the right is under the scanner, and the machine changes to state c.



6. The machine is in state c. Therefore, it prints a 0 when it reads the blank symbol.



7. Finally, the tape moves left, the scanner is over the square to the right. The machine changes to the stop state which halts the machine and stops running the instructions given.



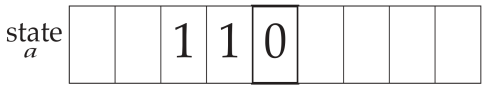
Example 3

So far, we have been printing symbols on blank tapes. In this final example, we are going to be inverting the 1s and 0s of the resulting tape from the previous example. Here is the instruction table:

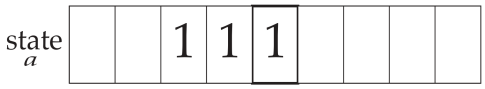
State	Scanned Symbol	Print	Move Tape	Next State
a	blank		right	b
	0	1	right	b
	1	0	right	a
b	blank			stop state
	0	1	right	b
	1	0	right	b

Observe that this time the tape is being moved to the right so, we are reading the tape from right-to-left. The machine starts in state a.

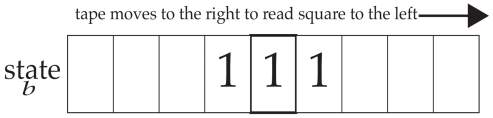
1. The machine is given the tape from the second example. The bold square represents the square under the head to start.



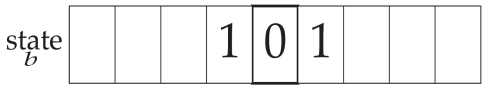
2. The machine starts in state a. The first symbol that's read is 0 and thus, a 1 is printed.



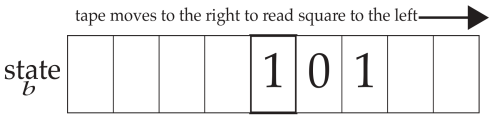
3. The tape is moved to the *right* while the scanner stays in the same position so, it will read the symbol in the square to the *left* next. The machine changes to state b.



4. The machine in state b reads a 1 and so, it prints a 0.

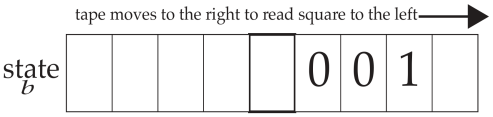


5. The tape is moved to the right so the square to the left is under the scanner, and the machine stays in state b.



6. The machine again reads a 1 while in state b. Thus, it prints another 0.

7. Then, the tape moves to the right so, the square to the left is under the scanner and the machine returns to state b.



8. Finally, the scanner reads a blank while in state b thus, nothing is printed, the machine does not move and it reaches the stop state.