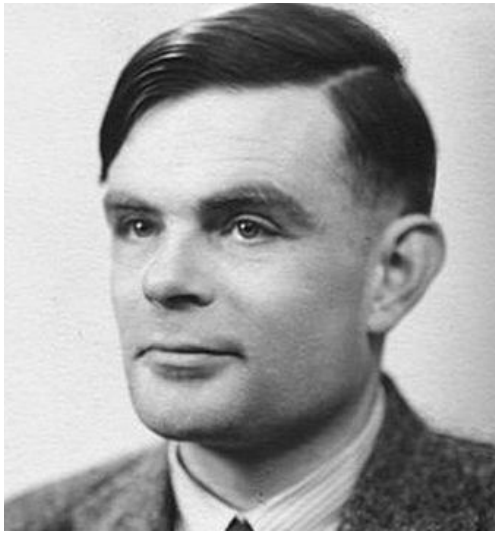
Turing Machines

Lisa & Mika & Logan

Alan Turing (1912-1954)

English computer scientist, mathematician, logician, cryptanalyst, and theoretical biologist

Considered to be the father of theoretical computer science and artificial intelligence



The Turing Machine

"The Turing machine is an abstract machine that manipulates symbols on a tape according to a table of rules" (1)

The idea is that given a tape of symbols and sets of instructions, you could use a Turing machine to solve any computational problem possible.

A turing machine involve three important components: **tape**, **symbols**, and **states**

[1] Wikipedia: Turing Machine

So, how does it work? (decoding)

5-tuple representation: (s, x, s', x', d)

current state (s, x): control unit is in state **s**, and reads current tape symbol **x**
instructions (s', x', d): control unit changes to state **s'**, writes **x'** in place of x, and moves right if **d=R**, or moves left if **d=L**

Rules:

initial state: s0
initial position: leftmost non-blank cell
final position: control unit halts when there is no five-tuple for the given cell and state

So, how does it work? (decoding)

Five-tuples:
(s0, 0, s1, 1, R), (s0, 1, s1, 0, R), (s0, B, s1, 0, R), (s1, 0, s2, 1, L), (s1, 1, s1, 0, R), (s1, B, s2, 0, R)

Starting string:
… | B | B | 0 | 0 | 1 | 1 | B | B | …

… | B | B | 0 | 0 | 1 | 1 | B | B | …	(s0, 0, s1, 1, R)
… | B | B | 1 | 0 | 1 | 1 | B | B | …	(s1, 0, s2, 1, L)
… | B | B | 1 | 1 | 1 | 1 | B | B | …	s2 → no tuple; halt.

So, how does it work? (decoding)

Five-tuples:

(s0, 0, s1, 0, R), (s0, 1, s1, 0, L), (s0, B, s1, 1, R),

(s1, 0, s2, 1, R), (s1, 1, s1, 1, R), (s1, B, s2, 0, R), (s2, B, s3, 0, R)

Starting string:

… | B | B | 0 | 1 | 0 | 1 | B | B | …

Try it out!

So, how does it work? (decoding)

Five-tuples:
(s0, 0, s1, 0, R), (s0, 1, s1, 0, L), (s0, B, s1, 1, R), (s1, 0, s2, 1, R), (s1, 1, s1, 1, R), (s1, B, s2, 0, R), (s2, B, s3, 0, R)

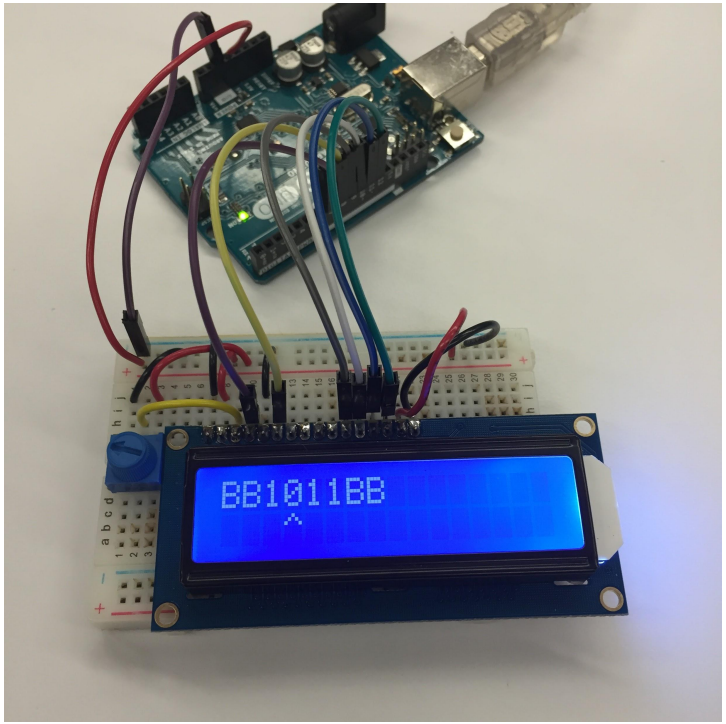
Starting string:
… | B | B | 0 | 1 | 0 | 1 | B | B | …

… | B | B | 0 | 1 | 0 | 1 | B | B | …	(s0, 0, s1, 0, R)
… | B | B | 0 | 1 | 0 | 1 | B | B | …	(s1, 1, s1, 1, R)
… | B | B | 0 | 1 | 0 | 1 | B | B | …	(s1, 0, s2, 1, R)
… | B | B | 0 | 1 | 1 | 1 | B | B | …	s2, 1 → no tuple; halt.

A Turing machine in Arduino

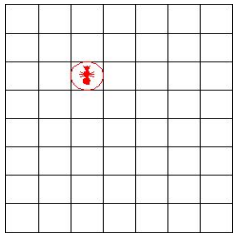
We implemented the previous example in Arduino!

The top row on the LCD screen indicates the tape, and the bottom row shows what point on the tape is being edited.

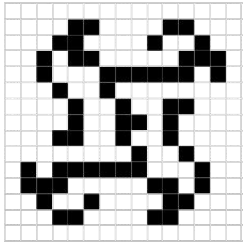


The Artistic Turing: Langton's Ant

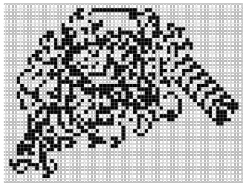
Langton's Ant[1] is a four-state Turing machine invented in the 80s that is much simpler than it sounds!



50 moves



386 moves

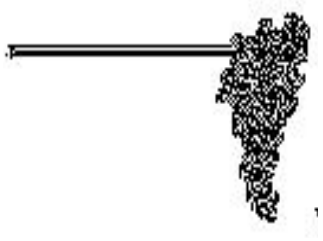


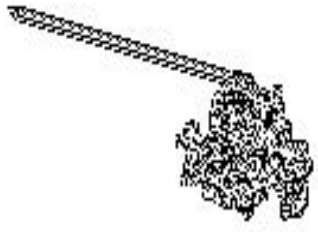
10,647 moves

→ Turns right on black, left on white

→ Flips the color as soon as it leaves

(s_N, B, s_E, W, R) (s_N, W, s_W, B, L)
(s_E, B, s_S, W, R) (s_E, W, s_N, B, L)
(s_S, B, s_W, W, R) (s_S, W, s_E, B, L)
(s_W, B, s_N, W, R) (s_W, W, s_S, B, L)

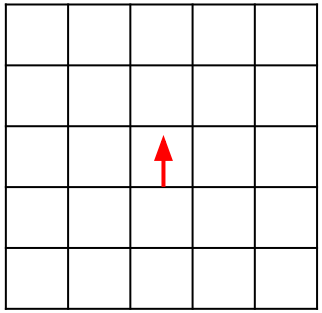
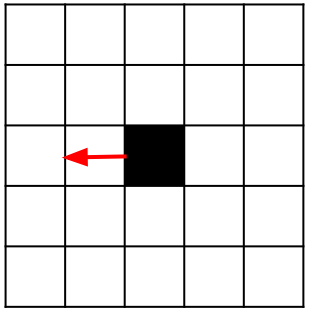
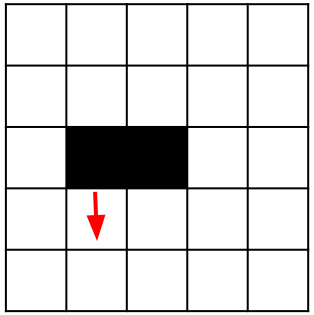
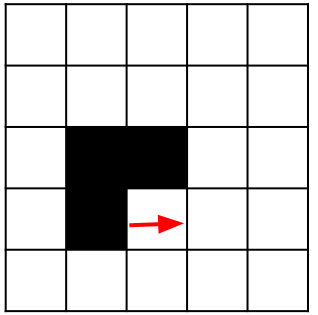
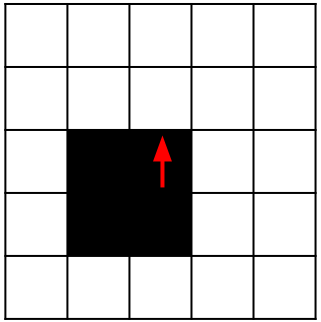
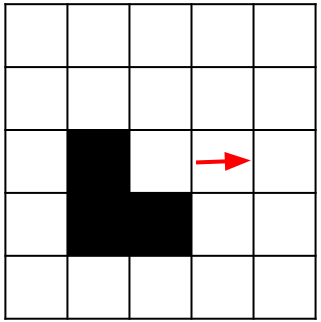
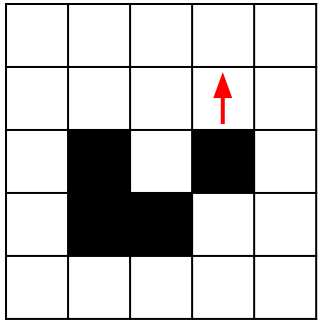
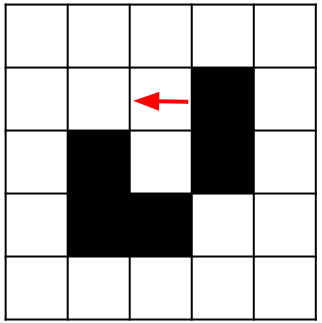




→ Believed to always create a "highway"

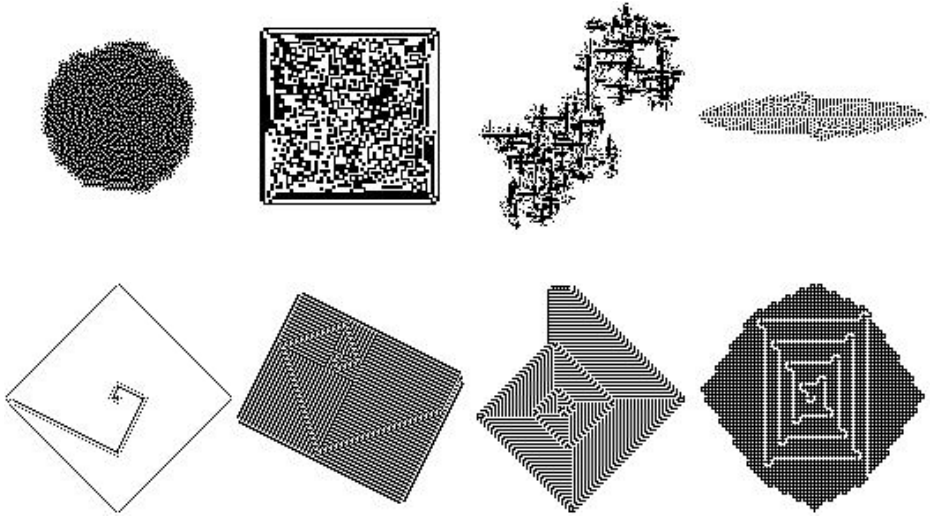
[1]http://mathworld.wolfram.com/LangtonsAnt.html

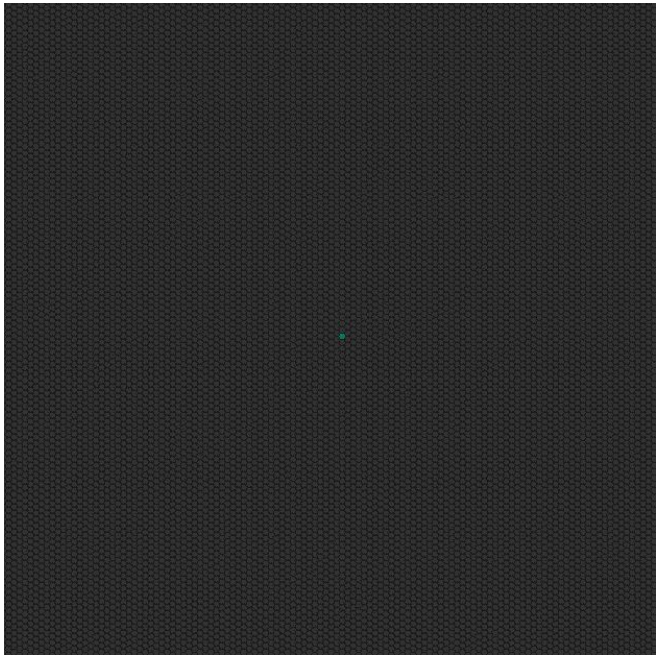
Progression of Langton's Ant

The Artistic Turing: Other Turmites

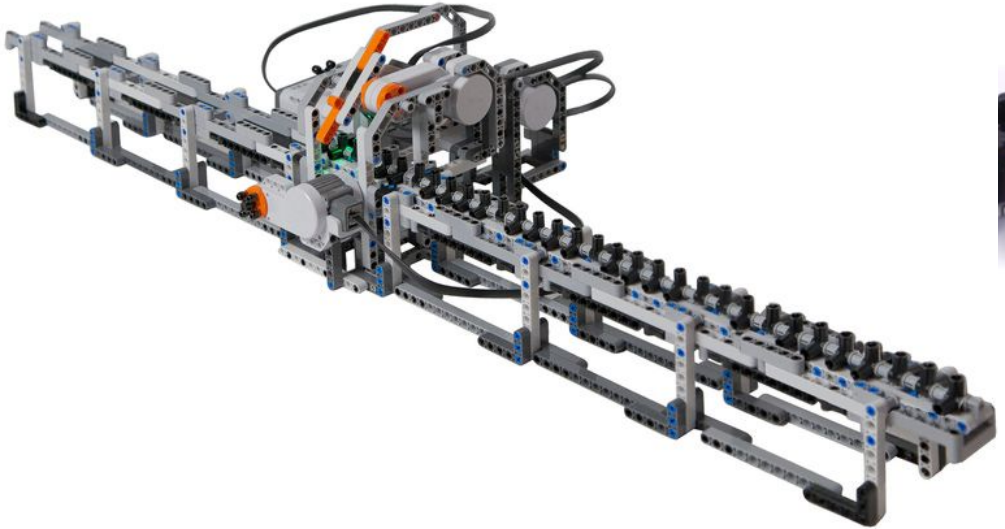
Turmites[1] are similar to Langton's Ant, but with more states.

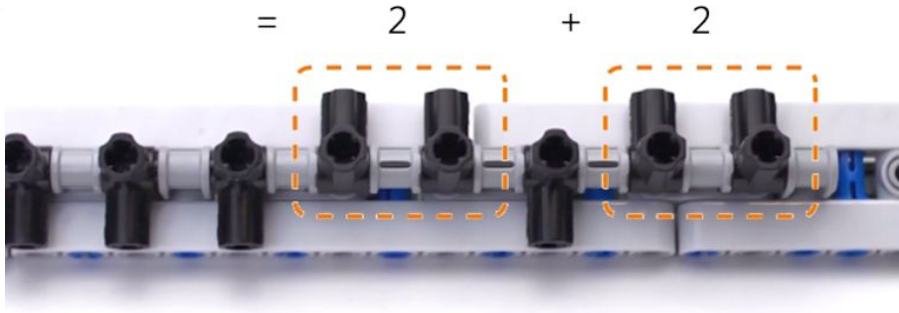


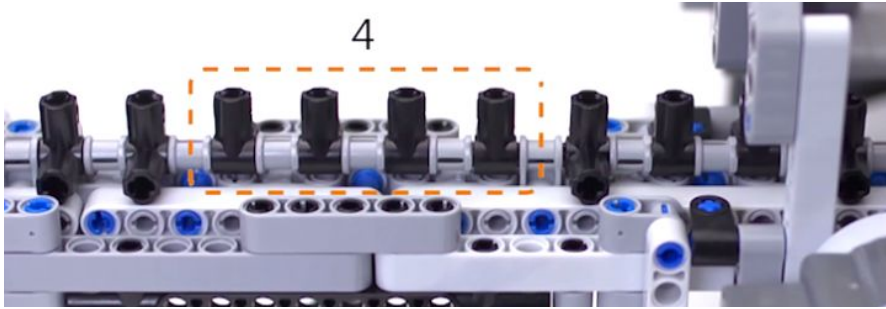


[1] http://mathworld.wolfram.com/Turmite.html
[2] Wikipedia user Maxter315 https://en.wikipedia.org/wiki/Langton's_ant#/media/File:CA50338_animation.gif

LEGO example







Is it used in real life?

For the most part, Turing machines are used as theoretical models of computational algorithms, as they can be used to solve any algorithm

A true universal Turing machine would require infinite tape, and therefore is physically impossible

Turing machines can be used to determine the possible functionalities of a computer[1]

[1] *Automata and Computability*

The Turing machine as a theoretical exercise

In the earlier half of the 20th century, people were pondering things similar to:

What are computable numbers?

What is a computing machine?

What does it mean for a function on the natural numbers to be computable? [1]

The Turing Machine poses answers to these questions.

[1] Wikipedia: Computability Theory

Computability theory terms

Halting problem[1]: With a certain input, will this algorithm run forever or reach an answer?

Circular and circle-free machines[3]: A machine will be circular if it reaches a configuration from which there is no possible move. A machine will be circle-free if it continues to print for infinity.

[1]Wolfram MathWorld: Halting problem
[2] Wolfram Math World: Tag systems
[3] Turing: On computable numbers

Theoretical applications

- Computer (back in the 1930's)
- Check spelling
- Predictive text
- encode/decode messages
- Rubix Cube

Can you think of any more?

HALT.

Questions?