Turing Machines

A definition of computation is needed to study computation mathematically. A Turing machine is a primitive, yet general, computer with an infinite tape. In each "cycle":

- the control unit reads the current tape symbol,
- writes a symbol on the tape,
- moves one position to the left or right, and
- switches to the next state.

The last three actions depend on the current state and tape symbol.

Formally, a Turing machine T consists of:

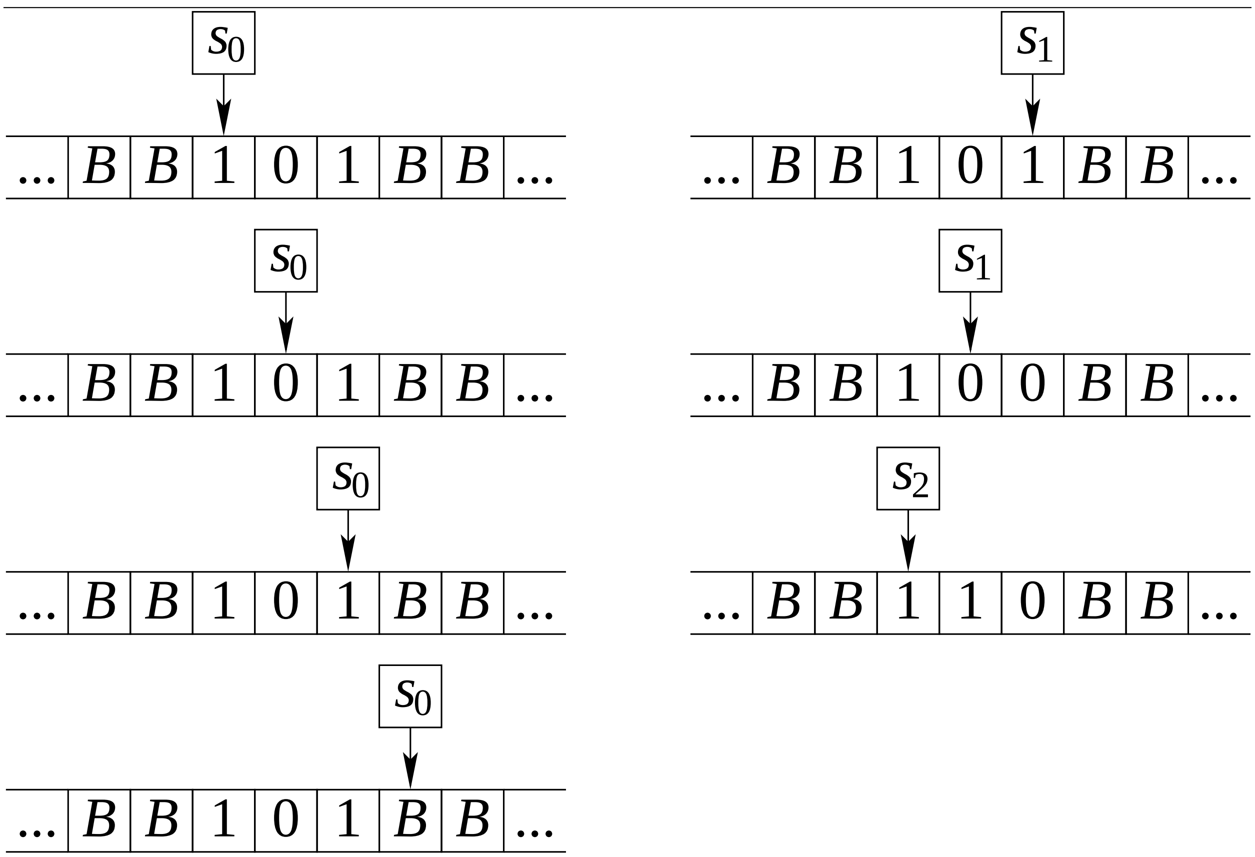
- S, a finite set of states, including a start state s_0,
- I, an alphabet, which is a finite set of symbols including the blank symbol B, and
- f, a state transition function, which is a partial function from $S \times I$ to $S \times I \times \{R, L\}$.

The Turing machine starts in state s_0 with the control unit reading the first symbol of the input string. There are an infinite number of blanks to the left and right of the input.

Turing Machine: Example 1

Here is a Turing machine for incrementing a binary string.

$f(s_0, 0) = (s_0, 0, R)$	In state s_0, move to the right
$f(s_0, 1) = (s_0, 1, R)$	until you reach a blank, and then
$f(s_0, B) = (s_1, B, L)$	switch to state s_1.
$f(s_1, 1) = (s_1, 0, L)$	In state s_1, move to the left
$f(s_1, 0) = (s_2, 1, L)$	changing 1s to 0s until you reach
$f(s_1, B) = (s_2, 1, L)$	a 0 or blank, changing it to a 1.
	There are no transitions from s_2, so this is where you halt.



Turing Machine: Example 2

On the next page is a Turing machine for recognizing binary strings that have equal numbers of 0s and 1s.

On each pass to the right, one 0 and one 1 are changed to an M, then the machine goes back to the left of the string.

The machine accepts the string when the whole input has been changed to Ms.

$f(s_0, 0)$	$= (s_1, M, R)$	In state s_0, change the first
$f(s_0, 1)$	$= (s_2, M, R)$	0 or 1 to an M, and then switch
$f(s_0, M)$	$= (s_0, M, R)$	to state s_1 or s_2.
$f(s_0, B)$	$= \textit{accept}$	Accept if all symbols are M.
$f(s_1, M)$	$= (s_1, M, R)$	In state s_1, change the first
$f(s_1, 0)$	$= (s_1, 0, R)$	1 to an M, and then switch to
$f(s_1, 1)$	$= (s_3, M, L)$	state s_3.
$f(s_2, M)$	$= (s_2, M, R)$	In state s_2, change the first
$f(s_2, 1)$	$= (s_2, 1, R)$	0 to an M, and then switch to
$f(s_2, 0)$	$= (s_3, M, L)$	state s_3.
$f(s_3, 0)$	$= (s_3, 0, L)$	In state s_3, move back to the
$f(s_3, 1)$	$= (s_3, 1, L)$	beginning of the string, and then
$f(s_3, M)$	$= (s_3, M, L)$	switch to state s_0.
$f(s_3, B)$	$= (s_0, B, R)$	

Properties of Turing Machines

- A Turing machine can recognize a language iff it can be generated by a phrase-structure grammar.
- The Church-Turing Thesis: A function can be computed by an algorithm iff it can be computed by a Turing machine.