TURING MACHINE

Maria Thomas

October 10, 2023

- The finite-state automata studied earlier cannot be used as general models of computation.
- They have limitations.
  1. For example: Finite-state automata are not able to recognize many easy-to-describe sets, including $\{0^n 1^n : n \geq 0\}$.
  2. Finite-state automata to compute relatively simple functions such as the sum of two numbers
  3. It cannot use them to compute functions that computers can, such as the product of two numbers.
- Turing machine is a more powerful type of machine.
- named after Alan Turing, the famous mathematician and computer scientist who invented them in the 1930s.

Definition

- A Turing machine $T = (S, I, f, s_0)$ consists of a finite set S of states, an alphabet I containing the blank symbol B, a partial function f from $S \times I$ to $S \times I \times \{R, L\}$, and a starting state s_0.

Recall

- Partial Functions: a partial function is defined only for those elements in its domain of definition.
- This means that for some (state, symbol) pairs the partial function f may be undefined.
- But for a pair for which it is defined, there is a unique (state, symbol, direction) triple associated to this pair.
- The five-tuples corresponding to the partial function in the definition of a Turing machine is called the **transition rules** of the machine.

- What is the final tape when the Turing machine T defined by the seven five-tuples $(s_0, 0, s_0, 0, R)$, $(s_0, 1, s_1, 1, R)$, (s_0, B, s_3, B, R), $(s_1, 0, s_0, 0, R)$, $(s_1, 1, s_2, 0, L)$, (s_1, B, s_3, B, R) and $(s_2, 1, s_3, 0, R)$ is run on the following tape.

...	*B*	*B*	0	1	0	1	1	0	*B*	*B*	...

Exercise 1

- Let T be the Turing machine defined by the five-tuples: $(s_0, 0, s_1, 0, R)$, $(s_0, 1, s_1, 0, L)$, $(s_0, B, s_1, 1, R)$, $(s_1, 0, s_2, 1, R)$, $(s_1, 1, s_1, 1, R)$, $(s_1, B, s_2, 0, R)$ and $(s_2, B, s_3, 0, R)$. For each of these initial tapes, determine the final tape when T halts, assuming that T begins in initial position.

$\cdots$	B	B	0	0	1	1	B	B	$\cdots$

Exercise 2

- Let T be the Turing machine defined by the five-tuples: $(s_0, 0, s_1, 0, R)$, $(s_0, 1, s_1, 0, L)$, $(s_0, B, s_1, 1, R)$, $(s_1, 0, s_2, 1, R)$, $(s_1, 1, s_1, 1, R)$, $(s_1, B, s_2, 0, R)$ and $(s_2, B, s_3, 0, R)$. For each of these initial tapes, determine the final tape when T halts, assuming that T begins in initial position.

...	B	B	1	1	B	0	1	B	...

Exercise 3

- Let T be the Turing machine defined by the five-tuples: $(s_0, 0, s_1, 0, R)$, $(s_0, 1, s_1, 0, L)$, $(s_0, B, s_1, 1, R)$, $(s_1, 0, s_2, 1, R)$, $(s_1, 1, s_1, 1, R)$, $(s_1, B, s_2, 0, R)$ and $(s_2, B, s_3, 0, R)$. For each of these initial tapes, determine the final tape when T halts, assuming that T begins in initial position.

...	B	B	B	B	B	B	B	B	...

Language of a Turing Machine

- Let V be a subset of an alphabet I . A Turing machine $T = (S, I, f, s_0)$ recognizes a string x in V^* if and only if T, starting in the initial position when x is written on the tape, halts in a final state.

- When does a Turing machine T not recognize a string x in V^*?
- x is not recognized by T if T does not halt or halts in a state that is not final when it operates on a tape containing the symbols of x in consecutive cells, starting in the initial position.

Final State of a Turing Machine

- A final state of a Turing machine T is a state that is not the first state in any five-tuple in the description of T using five-tuples.

Exercise

- What does the Turing machine described by the five-tuples $(s_0, 0, s_1, 0, R)$, $(s_0, 1, s_1, 0, R)$, (s_0, B, s_2, B, R), $(s_1, 0, s_1, 0, R)$, $(s_1, 1, s_0, 1, R)$, (s_1, B, s_2, B, R) do when given
  1. 11 as input?
  2. a bit string consisting entirely of 1s as input?

Exercise

- What does the Turing machine described by the five-tuples $(s_0, 0, s_0, 1, R)$,$(s_0, 1, s_0, 1, R)$, (s_0, B, s_1, B, L),$(s_1, 1, s_2, 1, R)$, do when given
  1. 101 as input?
  2. an arbitrary bit string as input?

Exercise

- Construct a Turing machine for adding two non-negative integers.
- Solution:

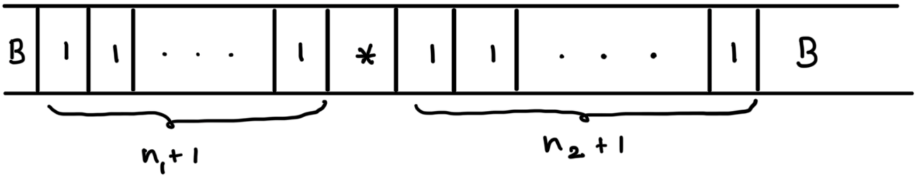
  Aim: To build a Turing machine, T, that computes the function

$$f(n_1, n_2) = n_1 + n_2.$$

  n_1 is represented by a string of 1s of length $n_1 + 1$.

  n_2 is represented by a string of 1s of length $n_2 + 1$.

  The pair (n_1, n_2) is represented by a string of $(n_1 + 1)$ 1s followed by an asterisk followed by $(n_2 + 1)$ 1s.



- The machine T should take this as input and produce as output a tape with $(n_1 + n_2 + 1)$ 1s.

  The machine starts at the left most 1 of the input string, and carries out steps to erase this 1.

  If $n_1 = 0$, then T should halt.

  Then replaces the asterisk with the leftmost remaining 1 and then halts.

- One possible solution is the following five-tuples: $(s_0, 1, s_1, B, R)$, $(s_1, *, s_3, B, R)$, $(s_1, 1, s_2, B, R)$, $(s_2, 1, s_2, 1, R)$ and $(s_2, *, s_3, 1, R)$.