

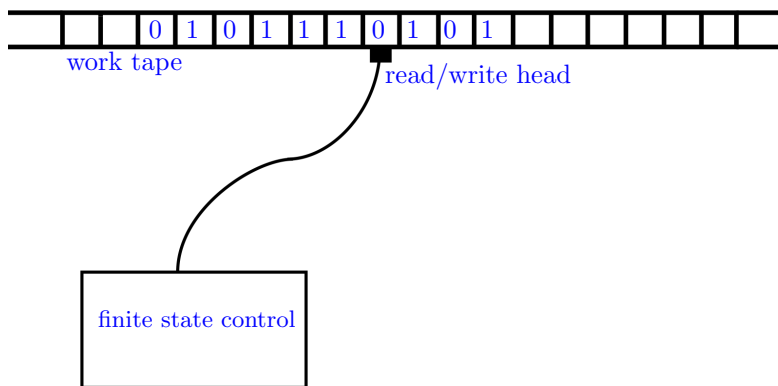
Summary on Lecture 21, May 16th, 2016

Turing Machines: First Examples.

A Turing Machine is an abstract mathematical object which models a “human computer”. Here we describe a simplest version of a Turing Machine.

First we casually describe the pieces of a Turing Machine and then give several examples.

- A **work tape**, i.e. an infinite to both sides strip of paper subdivided into **cells**. Each cell contains a single letter; all but finitely many cells are blank.
- A **read/write head** that is positioned at a particular cell. That head can move left and right.
- A **finite state control** that directs the head: symbols are read and written, and the head is moved around and the internal state of the machine changes.



Conventions:

- The alphabet $\Sigma = \{0, 1\}$. We represent natural numbers in the unary system. In other words, if the tape consists of all zeros, then it represents 0, while a string of n 1's represents n . Numbers are interspaced by a single 0's. For example, the string

01111110111111111100111011111111111111111110

represents the sequence 7,10,0,3,21.

- The set of states $S = \{0, 1, \dots, k - 1\}$. A “non-existent” state k denotes a halting state
- We assume that if the tape has at least one 1, then the Turing Machine starts at the left-most 1.
- The Turing Machine starts at the state 0.

Example 1. We would like to design a Turing Machine to compute very simple function: $n \mapsto f(n) = 1$. We start with the situation

$$0 \downarrow \underbrace{11 \dots 11}_n 0$$

and we would like to end up with

$$0 \downarrow 10$$

Here the vertical arrow denotes the position of the head of the Turing Machine.

This Turing Machine has to be designed to change 1's to 0's as it moves right, and then stop when it will read 0, then replace it with a single 1 and move to a halting state. Here is the table describing the machine:

	0	1
0	1L1	0R0
1	0R2	

Here the left vertical column describes the states. We check the sequence of actions when we start with 01110:

$$\begin{array}{ccccccc} \downarrow 0 & & \downarrow 0 & & \downarrow 0 & & \downarrow 0 & & \downarrow 1 & & \downarrow 2 \\ 01110 & \xrightarrow{0R0} & 00110 & \xrightarrow{0R0} & 00010 & \xrightarrow{0R0} & 000000 & \xrightarrow{1L1} & 000010 & \xrightarrow{0R2} & 000010 \end{array}$$

Example 2. Let us design a Turing Machine to compute the function $f(n) = n + 1$ for a nonnegative integer n , i.e., we need a machine which performs as follows:

$$0 \underbrace{11 \dots 11}_n 0 \mapsto 0 \underbrace{11 \dots 11}_{n+1} 0$$

Here is the table describing the machine:

	0	1
0	1L1	1R0
1	0R2	1L1

We check the sequence of actions when we start with 01110:

$$\begin{array}{cccccccccccccccc} \downarrow 0 & & \downarrow 0 & & \downarrow 0 & & \downarrow 0 & & \downarrow 1 & & \downarrow 1 & & \downarrow 1 & & \downarrow 1 & & \downarrow 1 & & \downarrow 2 \\ 001110 & \xrightarrow{1R0} & 001110 & \xrightarrow{1R0} & 001110 & \xrightarrow{1R0} & 0011100 & \xrightarrow{1L1} & 0011110 & \xrightarrow{1L1} & 0011110 & \xrightarrow{1L1} & 0011110 & \xrightarrow{1L1} & 0011110 & \xrightarrow{1L1} & 0011110 & \xrightarrow{0R2} & 0011110 \end{array}$$

Example 3. Now we describe two Turing Machine which compute the function $f(n, m) = n + m$ for nonnegative integers n, m , i.e., a machine which performs as follows:

$$0 \underbrace{11 \dots 11}_n 0 \underbrace{11 \dots 11}_m 0 \mapsto 0 \underbrace{11 \dots 11}_{n+m} 0$$

(a) This Turing Machine moves right keeping 1's as 1's, when it reads 0, replaces it with 1 and then moves left, replaces the left-most by 0 and then moves to the halting state. Here is the design:

	0	1
0	1L1	1R0
1	0R2	1L1
2		0R3

We check the sequence of actions when we start with 001101110:

$$\begin{array}{cccccccccccccccc} \downarrow 0 & & \downarrow 0 & & \downarrow 0 & & \downarrow 0 & & \downarrow 1 & & \downarrow 1 & & \downarrow 1 & & \downarrow 1 & & \downarrow 1 \\ 00110111 & \xrightarrow{1R0} & 001110111 & \xrightarrow{1R0} & 0011110111 & \xrightarrow{1R0} & 0011110111 & \xrightarrow{1L1} & 0011111111 & \xrightarrow{1L1} & 0011111111 & \xrightarrow{1L1} & 0011111111 & \xrightarrow{1L1} & 0011111111 & \xrightarrow{1L1} & 0011111111 \\ \downarrow 1 & & \downarrow 2 & & \downarrow 3 \\ 0011111111 & \xrightarrow{0R2} & 0011111111 & \xrightarrow{0R3} & 0001111111 \end{array}$$

(b) That Turing Machine changes first 1 to 0 then moves right keeping 1's as 1's, when it reads 0, replaces it with 1 and then moves left to the left-most 1. Here is the design:

	0	1
0		0R1
1	1L2	1R1
2	0R3	1L2

Again, we check the sequence of actions when we start with 001101110:

$$\begin{array}{cccccccccccccccc} \downarrow 0 & & \downarrow 1 & & \downarrow 1 & & \downarrow 1 & & \downarrow 2 & & \downarrow 2 & & \downarrow 2 & & \downarrow 2 \\ 00110111 & \xrightarrow{0R1} & 000110111 & \xrightarrow{1R1} & 000110111 & \xrightarrow{1R1} & 000110111 & \xrightarrow{1L2} & 000111111 & \xrightarrow{1L2} & 000111111 & \xrightarrow{1L2} & 000111111 & \xrightarrow{1L2} & 000111111 \\ \downarrow 2 & & \downarrow 3 \\ 000111111 & \xrightarrow{0R3} & 000111111 \end{array}$$