Turing Machine

Prof. (Dr.) K.R. Chowdhary
Email: kr.chowdhary@iitj.ac.in

Formerly at department of Computer Science and Engineering
MBM Engineering College, Jodhpur

Monday 12th October, 2015

Turing Machine: Agenda

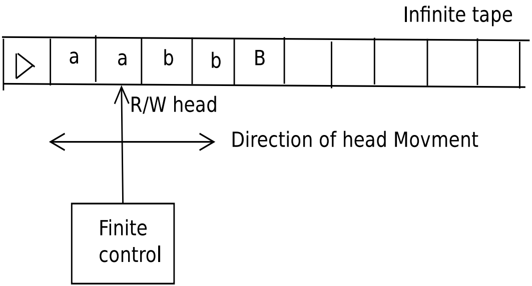
- Alan M. Turing
- Church-Turing Thesis
- Definitions
- Computation
- TM Configurations

Alan M. Turing

- Alan Turing was one of the founding fathers of CS.
- His computer model - the Turing Machine (TM) - was inspiration /premonition of the electronic computer that came two decades later to the TM model
- He was instrumental in cracking the Nazi Enigma cryptosystem in WW-II
- Invented the Turing Test used in AI
- Legacy: The Turing Award, eminent award in Theoretical CS research
- **Church-Turing Thesis**

  TM is ultimate model for compuation. Any thing, which is solvable, i.e., has an algorithm, what ever the computation model is used to compute that algorithm, it is ultimately the TM model.

Turing Machine Model for computation



$M = (Q, \Sigma, \Gamma, \delta, s, H)$, $\Gamma = \Sigma \cup \{B, \triangleright\}$, $H \subseteq Q$

where,

Q is set of states

H is set of Halting states

Σ is set of input symbols

δ is transition function (a partial function)

$\delta : (Q - H) \times \mathit{\Gamma} \rightarrow Q \times \mathit{\Gamma} \times \{L, R\}$

language acceptability:

$L = \{w | w \in \Sigma^*, q_0 w \vdash^* \alpha p \beta, p \in H; \alpha, \beta \in \Gamma^*\}$

A Thinking Machine

- A Turing Machine (TM) is a device with a finite amount of read-only hard memory (states), and an unbounded amount of read/write tape-memory. There is no separate input. Rather, the input is assumed to reside on the tape at the time when the TM starts running.
- Just as with Automata, TM's can either be input/output machines (compare with Finite State Transducers), or yes/no decision machines.
- 1936(Alan M. Turing): Given any logical/arithmetic computation, for which complete instructions for carrying out this are supplied, it is always possible to design a TM that can perform this computation.
- TM v/s Human: TM model is based on human problem solving process using pencil and paper. As we do this, our mental state changes, for every smallest step. Correspondingly, TM has tape (=paper), R/W head (=pencil), and state (=state of our mind).
- $\{a^n b^n | n \geq 0\}$ v/s $\{a^n b^n c^n | n \geq 0\}$
- powers of TM: Power is problem solving capability, and not about how fast or slow it can do.

- **Turing Machine Criteria:**

1. These are Automata
2. As simple as possible - to define formally, describe and reason about them
3. As general as possible (any computation can be represented by them)

- **Acceptability by Turing machine:**

  A string *w* is accepted by *M* if after being put on the tape with the Turing machine head set to the left-most position, and letting *M* run, *M* eventually enters the halting state state. In this case *w* is an element of L(M),the language accepted by *M*:-

$$L(M) = \{w | w \in \Sigma^* \wedge q_0 w \vdash^* \alpha y \beta\}$$

where, y is halting configuration, and $\alpha, \beta \in \Gamma^*$

Turing Machine solves a Problem: Erase entire tape

Consider a TM $M = (Q, \Sigma, \Gamma, \delta, s, H), Q = \{q_0, q_1\}$, $\Sigma = \{a\}, \Gamma = \{a, B, \triangleright\}, s = q_0$, B is blank character, $\triangleright$ is left end marker.

$\delta(q_0, a) = (q_0, B, R)$

$\delta(q_0, B) = (q_1, B, L)$

Let $w = aaaa$

$q_0 aaaaB$

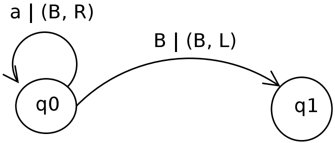
$\vdash Bq_0 aaaB$

$\vdash BBq_0 aaB$

$\vdash BBBq_0 aB$

$\vdash BBBBq_0 B$

$\vdash BBBq_1 B$



Representation of a Configuration

If $i - 1 = n$ then $X_i = \#$. If $i = 1$ then $X_i = \triangleright$ and the head will move to right, else it will fall off the tape or we say it crashes. If $i > 1$ and $i = n$ then for $d =$ L, we write a move as

$$X_1 X_2 \ldots X_{i-1}\, q\, X_i X_{i+1} \ldots X_n \vdash_M X_1 X_2 \ldots X_{i-2}\, p\, X_{i-1}\, Y\, X_{i+1} \ldots X_n.$$

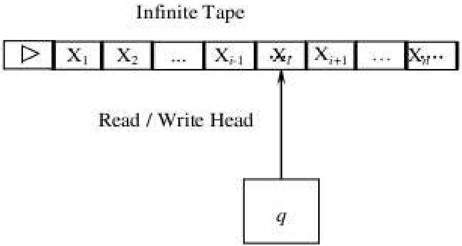


Figure 12.2: Turing Machine representing
ID = $X_1 X_2 \ldots X_{i-1}\, q\, X_i X_{i+1} \ldots X_n$

Alternatively, for $i > 1$ and $d =$ R, a move is written as

$$X_1 X_2 \ldots X_{i-1}\, q\, X_i X_{i+1} \ldots X_n \vdash_M X_1 X_2 \ldots X_{i-1}\, Y p\, X_{i+1} \ldots X_n.$$

Configuration-1

- A configuration of a TM:
  - Current state
  - Symbols on tape
  - position of RW head
- A formal specification of configuration:
  - *uqv*, where *u,v* are strings on Σ, and *uv* is current content on tape, *q* is current state, and head is at first symbol of *v*.

    For example, $00101q_5011$ where read head points at *0* (third digit from end) and state is q_5.

Configuration-2

- For Two configurations:

  uaq_ibv and uq_jacv, where , $a, b, c \in \Sigma$ and $u, v \in \Sigma^*$

$$uaq_ibv \vdash uq_jacv \text{ if } \delta(q_i, b) = (q_j, c, L)$$
$$uaq_ibv \vdash uacq_jv \text{ if } \delta(q_i, b) = (q_j, c, R)$$

- Two special cases:

- The left most cell:

$$q_ibv \vdash q_jcv \text{ for } \delta(q_i, b) = (q_j, c, L)$$
$$q_ibv \vdash cq_jv \text{ for } \delta(q_i, b) = (q_j, c, R)$$

- On the cell with blank symbol:

  uaq_i is equivalent to uaq_iB

Example: of language recognition

Design TM to accept: $\{a^n b^n, n \geq 0\}$

Let $M = (Q, \Sigma, \Gamma, \delta, s, H)$. The algorithm can be specified as:

1. M replaces left most a by X, and then head moves to right until it encounters left most b
2. Replaces this b by Y, and then moves left to find the right most X. Then moves one step right to left most a
3. Repeat Step 2 and 3 in order, i.e., 2, 3, 2, 3, ...
4. When searching for b, if finds a blank character B (i.e., $|a^n| > |b^n|$), then M does not accept w.
5. If a is not found but it finds b, then also M does not accept, (i.e., $|a^n| < |b^n|$).
6. After changing last b to Y, if M finds no more a then it checks that no more b remains. If this is true then $a^n b^n$ is accepted by M i.e., $|a^n| = |b^n|$

Example: of language recognition

$Q = \{q_0, q_1, q_2, q_3\}$

$\Sigma = \{a, b\}$

$\Gamma = \{\triangleright, a, b, X, Y, B\}$

$s = q_0$

$H = \{q_3, q_0\}$

- Design TM to accept: $\{a^n b^n, n \geq 0\}$

1. $\delta(q_0, a) = (q_1, X, R)$

   $\delta(q_1, a) = (q_1, a, R)$; skip through $a's$ and

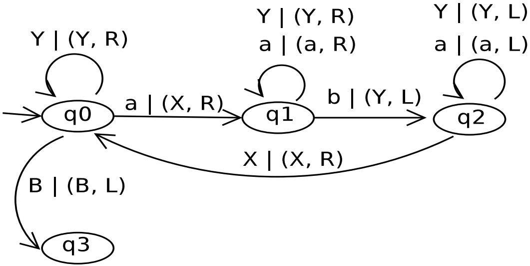
   $\delta(q_1, Y) = (q_1, Y, R)$; then $Y's$

   $\delta(q_1, b) = (q_2, Y, L)$

   $\delta(q_2, Y) = (q_2, Y, L)$, traverse through Y's and then

   $\delta(q_2, a) = (q_2, a, L)$, traverse a's

TM to accept: $\{a^n b^n, n \geq 0\}$



- Move from R to L until X is found and start back:

  $\delta(q_2, X) = (q_0, X, R)$, right most X is found. Now repeat from 1 else from 2.

2. $\delta(q_0, Y) = (q_0, Y, R)$, scan through Y's

   $\delta(q_0, B) = (q_3, B, L)$, accept w, and halt.

Example: of language recognition Dry Run

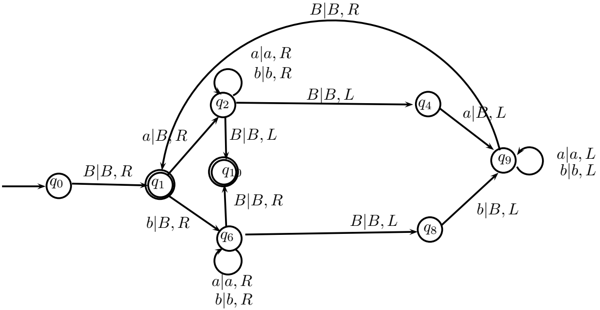
- TM to accept: $\{a^n b^n, n \geq 0\}$, Let $w = aabb$

$$q_0 aabbB \vdash Xq_1 abbB \vdash Xaq_1 bbB \vdash Xq_2 aYbB \vdash Xq_2 aBbB$$
$$\vdash q_2 XaYbB \vdash Xq_0 aYbB \vdash XXq_1 YbB \vdash XXYq_1 bB$$
$$\vdash XXq_2 YYB \vdash Xq_2 XYYB \vdash XXq_0 YYB \vdash XXYq_0 YB$$
$$\vdash XXYYq_0 B \vdash XXYq_3 YB \text{ (accept the input)}$$

Total number of transitions for $|w| = n$ are: $n/2$ forward and $n/2$ in backward, in each to and fro round, i.e., n. Since, in each trip, two symbols are marked, therefore, there will be total $n/2$ trips, making total transitions: $n \times n/2 = n^2/2$. Time complexity, $O(n^2/2) = O(n^2)$, which is polynomial (P) time complexity.

Example: Construct TM to recognize $L = \{ww^R \mid w \in \{a, b\}^*\}$, as well odd palindromes in a, b.

Approach: We can traverse w; each time if a/b is found in begin, replace it by B and also, replace a/b at end by B.



Time complexity: Let $|w| = n$. Total of trips $= n/2$. Number of transitions in successive trips are: $n + (n-2) + (n-4) + \cdots + 4 + 2 = n^2/2 - n/2$ in even palindromes, and $n^2/2 - n/2 + 2$ in odd palindromes, which is $O(n^2)$ is polynomial time complexity.

Acceptors v/s Deciders

- Let M is TM.
- Three possibilities occur on a given input w:
- M eventually enters q_{acc} and therefore halts and accepts. $w \in L(M)$
- M eventually enters q_{rej} or crashes somewhere. M rejects w, i.e., $w \notin L(M)$
- M never halts its computation and is caught up in an infinite loop. In this case w is neither accepted nor rejected. However, any string not explicitly accepted is considered to be outside the accepted language. $w \notin L(M)$
- Decider: M never enters infinite loop(A recursive language).