Turing Machines - a formal definition

A Turing Machine is a 6-tuple

$$M = (Q, \Sigma, I, q_0, \delta, F)$$

where

Q finite, non-empty set of states

Σ finite set of at least 2 symbols: the alphabet. $\wedge \in \Sigma$

I non-empty subset of Σ; $\wedge \notin I$; input alphabet

q_0 $q_0 \in Q$; starting or initial state

d $\delta: (Q \backslash F) \times \Sigma \Rightarrow Q \times \Sigma \times \{-1, 0, 1\}$, a partial function, the instruction table

F $F \subseteq Q$, the set of final or halting states

Some notation:

A word is a string of symbols eg. lecture^room

..for an alphabet (set of symbols) $\sum$

a word of $\sum^*$ is a finite string of elements of $\sum$.

$\sum^*$ = all words of $\sum$

ε is the empty word; $\varepsilon \in \sum^*$ for any $\sum$.

We represent concatenation of words v and w by v.w

The value computed by the TM, M on input w is $\mathbf{f_M(w)}$.

If M does not Halt & Succeed on input w, then $f_M(w)$ is ***undefined***.

The TM tape contents:

The input alphabet is I; ^ $\notin$ I. TM Input is a word of I*

w_0	w_1	w_2	w_3	w_4	^	^	^	^

Eg. a tape with contents $w = w_0w_1w_2w_3w_4 \in I^*$ at start of TM run

There are no blanks (^) in the input w..

the input word is terminated by the first ^ on the tape when the TM starts.

If all the w_i are blank, we have the empty input, ε.

$\sum$ is the whole or internal alphabet of the TM.

The input alphabet I is a subset of $\sum$: $I \subseteq \sum$ and $\wedge \in \sum$.

the TM Output is a word w' of $\sum^*$..without blanks (^).
It starts at square 0 and
ends just before the first blank, ^, on the tape
when the TM halts and succeeds.
(otherwise it is not defined).

Not all words of $\sum^*$ are necessarily valid outputs.

Not all symbols of $\sum^*$ are necessarily valid outputs.

a Turing Machine example..

$M = (\{q_0, q_1, q_2\}, \{1, \wedge\}, \{1\}, q_0, \delta, \{q_2\})$

- Q — state set
- Σ — alphabet
- I — input alphabet
- q_0 — initial state
- δ — delta function
- $\{q_2\}$ — halting state

δ-function:

$\delta\,(q_0, 1) = (q_1, \wedge, 1)$

$\delta\,(q_0, \wedge) = (q_2, \wedge, 0)$

$\delta\,(q_1, 1) = (q_0, 1, 1)$

possible inputs:

11111^^^…

1111^^^…

111^^^…

what does this TM do?

..a TM example

$\delta(q_0, 1) = (q_1, \wedge, 1)$
$\delta(q_0, \wedge) = (q_2, \wedge, 0)$
$\delta(q_1, 1) = (q_0, 1, 1)$

W_2: 111^^^…

q_0	111^^
q_1	^11^^
q_0	^11^^
q_1	^1^^^

Halt & Fail

W_1: 1111^^

q_0	1111^^
q_1	^111^^
q_0	^111^^
q_1	^1^1^^
q_0	^1^1^^
q_2	^1^1^^

Halt & succeed

the TM M starts in state q_0 with the head over square 0..

if M is in state q_0: if M reads a 1 it
- writes a ^
- moves right 1 square on the tape
- goes into state q_1

otherwise, M reads a ^: it
- writes a ^
- does not move along the tape
- goes into state q_2

if M is in state q_1: if M reads a 1 it
- writes a 1
- moves right 1 square along the tape
- goes into state q_0

otherwise M **halts and fails**: no applicable instruction

if M is in state q_2 it
- **halts and succeeds**

another Turing Machine example..

design a TM, $M = (Q, \sum, I, q_0, \delta, F)$ which evaluates the function **head:**

$I = \{a,b\}$, $\sum = \{a, b, \wedge\}$

head(w) = s, where $s \in I$, $w_1 \in I^*$ and $w = s.w_1$

head(ε) is undefined

$\delta (q_0, a) = (q_1, a, 1)$

$\delta (q_0, b) = (q_1, b, 1)$

$\delta (q_1, a) = (q_2, \wedge, 0)$

$\delta (q_1, b) = (q_2, \wedge, 0)$

$\delta (q_1, \wedge) = (q_2, \wedge, 0)$

b	a	b	^	^	^	^ ....

q_0 bab^^^…

q_1 bab^^^…

q_2 b^b^^^…

Halt and Succeed

Summary

We have seen a formal definition of a Turing Machine:

$$M = (Q, \sum, I, q_0, \delta, F)$$

where

- Q finite, non-empty set of states
- $\sum$ finite set of at least 2 symbols: the alphabet. $\wedge \in \sum$
- I non-empty subset of $\sum$; $\wedge \notin I$; input alphabet
- q_0 $q_0 \in Q$; starting or initial state
- δ $\delta: (Q \backslash F) \times \sum \Rightarrow Q \times \sum \times \{-1, 0, 1\}$, a partial function, the instruction table
- F $F \subseteq Q$, the set of final or halting states

and examples of simple TMs to

- determine whether a number is odd/even
- return the Head of the input word.

Design a Turing Machine to implement the Tail function

$Q = \{q_0, \ldots \}$, $F = \{ \}$, $\Sigma = \{a,b,\wedge\}$.

a	**b**	**a**	**b**	^	^ ^ ^ ^