Turing Machines

Intro to Turing Machines

- A *Turing Machine* (TM) has finite-state control (like PDA), and an infinite read-write *tape*. The tape serves as both input and unbounded storage device.

- The tape is divided into *cells*, and each cell holds one symbol from the *tape alphabet*.

- There is a special *blank* symbol B. At any instant, all but finitely many cells hold B.

- *Tape head* sees only one cell at any instant. The contents of this cell and the current state determine the next move of the TM.

The tape extends to infinity in both directions with all “B” s

B	B	B	B	B	B	B	1	1	0	2	B	B	B	B	B	B	B

state	0	1	2
1	L,2,3		
2			
3			
4			

State = 1

Moves

- A *move* consists of:
  - replacing the contents of the scanned cell
  - repositioning of the tape head to the nearest cell on the left, or on the right
  - changing the state

- The *input alphabet* is a subset of the tape alphabet. Initially, the tape holds a string of input symbols (the *input*), surrounded on both sides with in infinite sequence of blanks. The initial position of the head is at the first non-blank symbol.

Formal Definition

- A TM is a septuple $M=(Q,\Sigma,\Gamma,\delta,q_0,B,F)$, where
  - Q is a finite set of states
  - Γ is the tape alphabet, and $\Sigma \subseteq \Gamma$ is the input alphabet
  - $B \in \Gamma - \Sigma$ is the blank symbol
  - $q_0 \in Q$ is the start state, and $F \subseteq Q$ is the set of accepting states
  - $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L,R\}$ is a partial function. The value of $\delta(q,X)$ is either undefined, or is a triple consisting of the new state, the replacement symbol, and direction (left/right) for the head motion.

Example

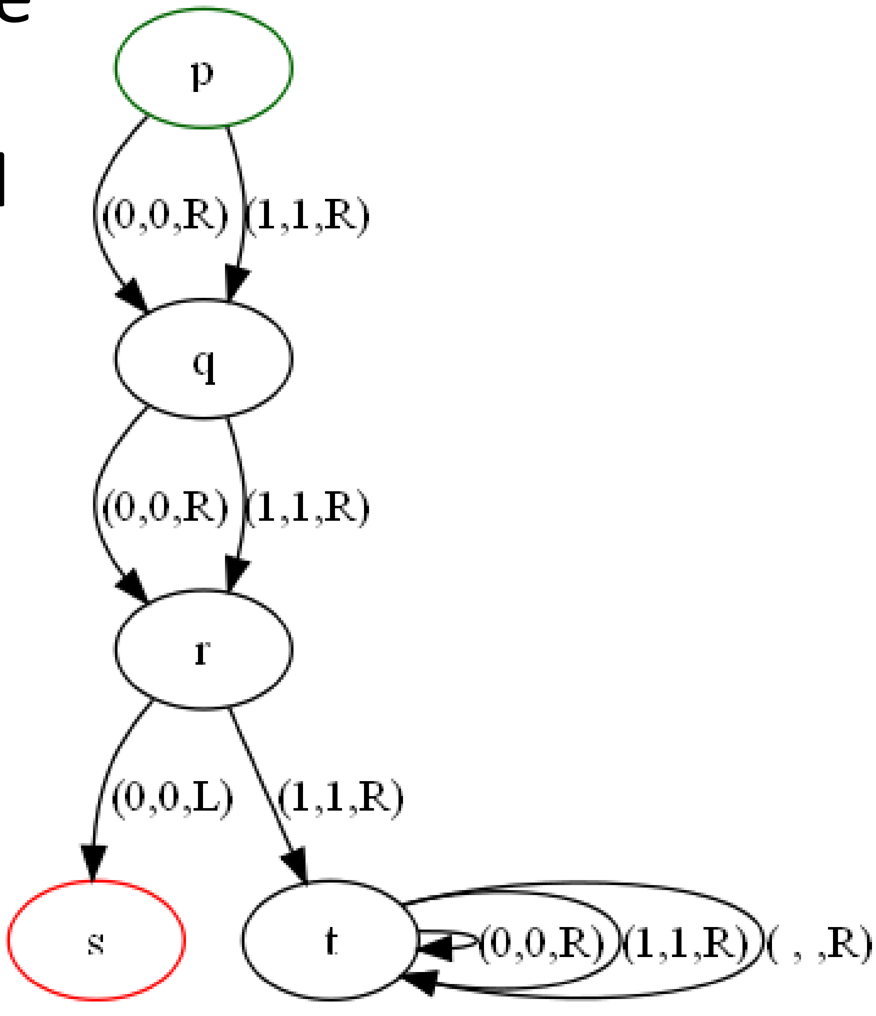
- Here is a TM that checks its third symbol is 0, accepts if so, and runs forever, if not.

- M=({p,q,r,s,t},{0,1,},{0,1,B},p,B,{s})

- $\delta(p,X) = (q,X,R)$ for X=0,1
- $\delta(q,X) = (r,X,R)$ for X=0,1
- $\delta(r,0) = (s,0,L)$
- $\delta(r,1) = (t,1,R)$
- $\delta(t,X) = (t,X,R)$ for X=0,1,B

Transisition Diagrams for TM

- Pictures of TM can be drawn like those for PDA's. Here's the TM of the example below.



$\delta(p,X) = (q,X,R)$ for X=0,1
$\delta(q,X) = (r,X,R)$ for X=0,1
$\delta(r,0) = (s,0,L)$
$\delta(r,1) = (t,1,R)$
$\delta(t,X) = (t,X,R)$ for X=0,1,B

Implicit Assumptions

- Input is placed on tape in contiguous block of cells
- All other cells are blank: ‘B’
- Tape head positioned at Left of input block
- There is one start state

- The text uses a single Halt state, an alternative is to have many final states. These are equivalent, why?

Example 2: { $a^n b^m$ | n,m in Nat}

states = 0,1,H

tape alphabet = a,b,^

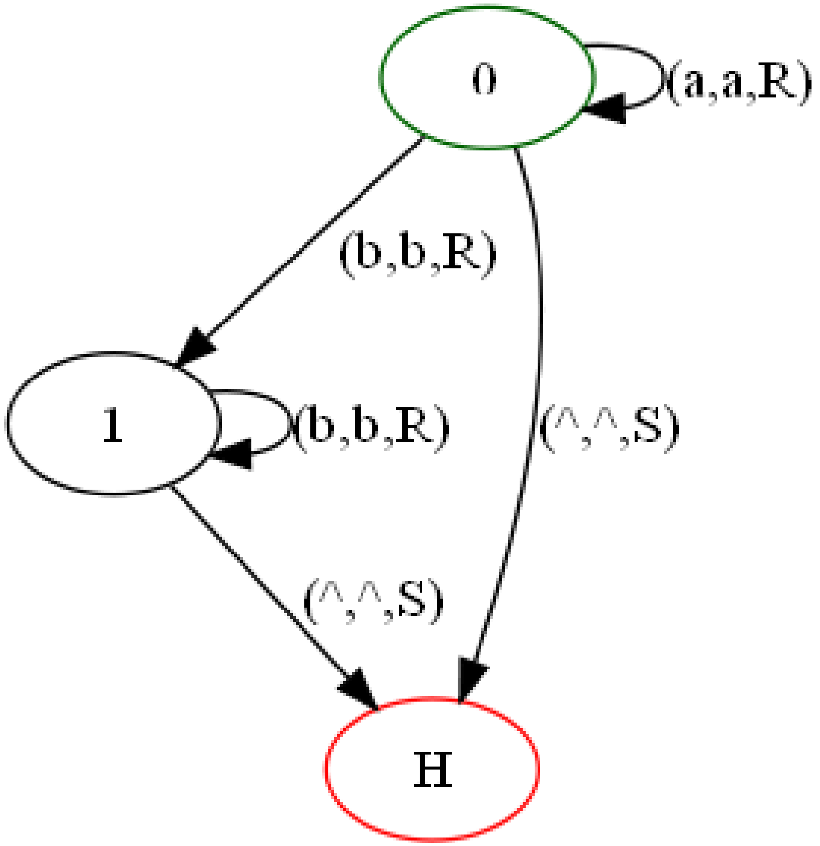
input alphabet = a,b

start = 0

blank = ‘^'

final = H

delta = (0,^,^,S,H)
(0,a,a,R,0)
(0,b,b,R,1)
(1,b,b,R,1)
(1,^,^,S,H)

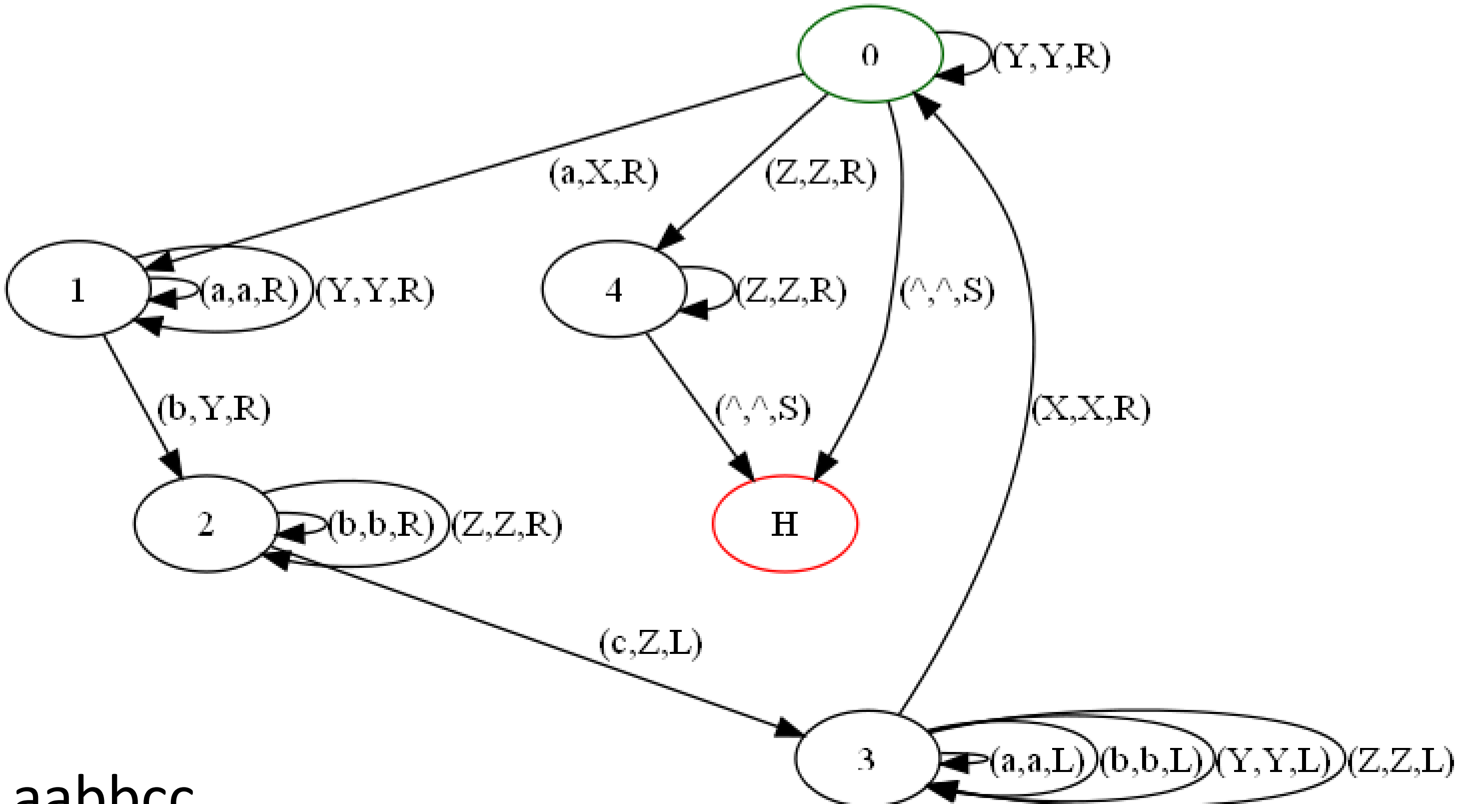


Example 3: { $a^n b^n c^n$ | n in Nat}

delta =

(0,a,X,R,1) Replace a by X and scan right
(0,Y,Y,R,0) Scan right over Y
(0,Z,Z,R,4) Scan right over Z, but make final check
(0,^,^,S,H) Nothing left, so its success
(1,a,a,R,1) Scan right looking for b, skip over a
(1,b,Y,R,2) Replace b by y, and scan right
(1,Y,Y,R,1) scan right over Y
(2,c,Z,L,3) Scan right looking for c, replacxe it by Z
(2,b,b,R,2) scan right skipping over b
(2,Z,Z,R,2) scan right skipping over Z
(3,a,a,L,3) scan left looking for X, skipping over a
(3,b,b,L,3) scan left looking for X, skipping over b
(3,X,X,R,0) Found an X, move right one cell
(3,Y,Y,L,3) scan left over Y
(3,Z,Z,L,3) scan left over Z
(4,Z,Z,R,4) Scan right looking for ^, skip over Z
(4,^,^,S,H) Found what we're looking for, success!

tape alphabet = a,b,c,^,X,Y,Z
input alphabet = a,b,c
start = 0
blank = '^ '
final = H



aabbcc

Xabbcc

XaYbcc

XaYbZc

XXYbZc

XXYYZc

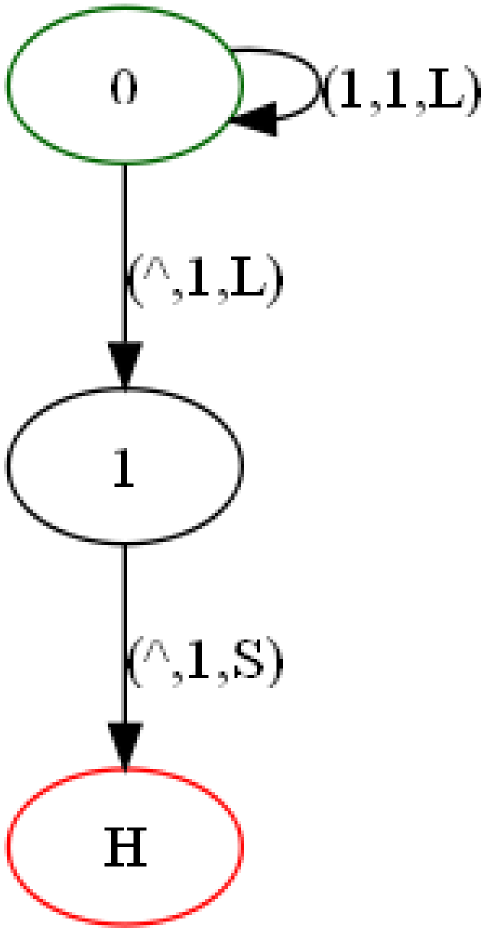
XXYYZZ

Turing machines with output

- A Turing machine can compute an output by leaving an answer on the tape when it halts.
- We must specify the form of the output when the machine halts.

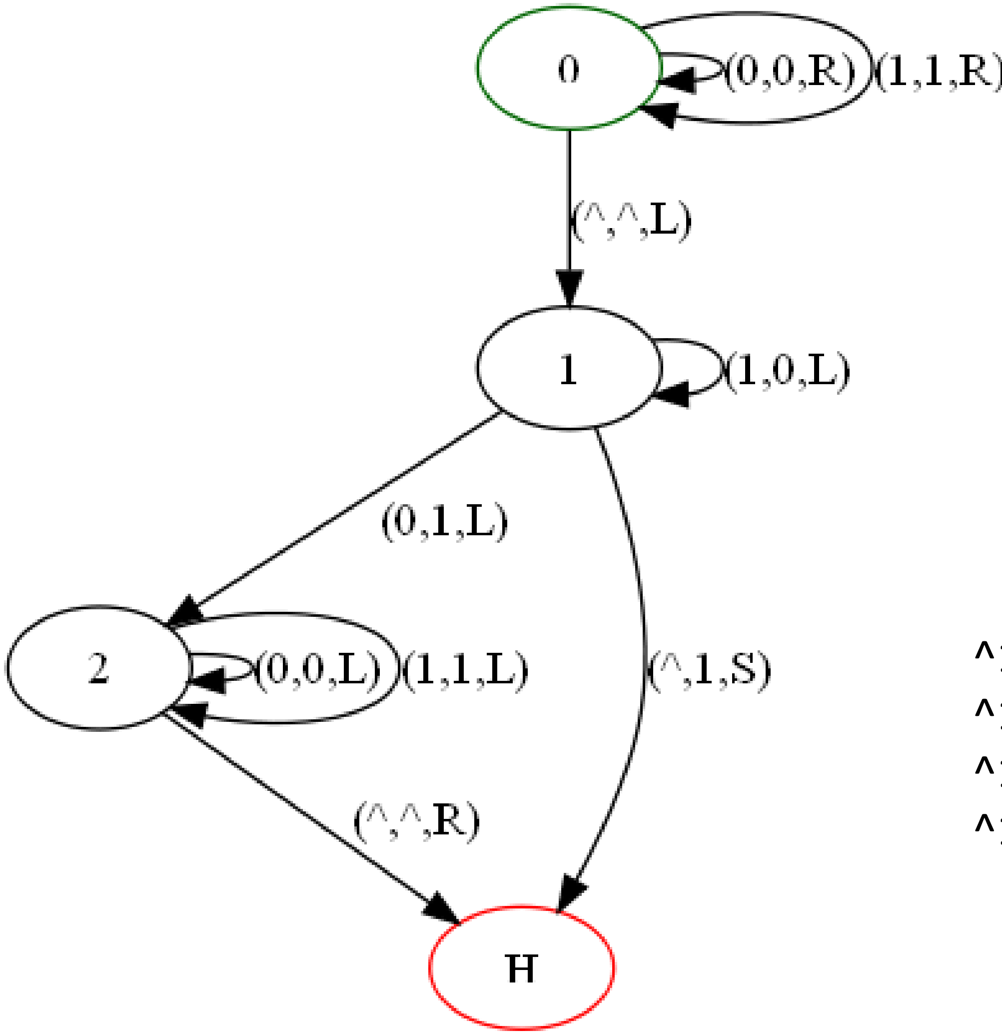
Adding two to a number in unary

states = 0,1,H
tape alphabet = 1,^
input alphabet = 1
start = 0
blank = '^'
final = H
delta =
(0,1,1,L,0)
(0,^,1,L,1)
(1,^,1,S,H)



Adding 1 to a Binary Number

states = 0,1,2,H
tape alphabet = 1,0,^
input alphabet = 1,0
start = 0
blank = '^ '
final = H
delta =
(0,0,0,R,0)
(0,1,1,R,0)
(0,^,^,L,1)
(1,0,1,L,2)
(1,1,0,L,1)
(1,^,1,S,H)
(2,0,0,L,2)
(2,1,1,L,2)
(2,^,^,R,H)



^1011^
^1010^
^1000^
^1100^

An equality Test

states = 0,1,2,3,4,H
tape alphabet = 1,0,#,^
input alphabet = 1,0,#
start = 0
blank = ‘^'
final = H

delta =
(0,1,^,R,1)
(0,^,^,R,4)
(0,#,#,R,4)
(1,1,1,R,1)
(1,^,^,L,2)
(1,#,#,R,1)
(2,1,^,L,3)
(2,#,1,S,H)
(3,1,1,L,3)
(3,^,^,R,0)
(3,#,#,L,3)
(4,1,1,S,H)
(4,^,^,S,H)
(4,#,#,R,4)