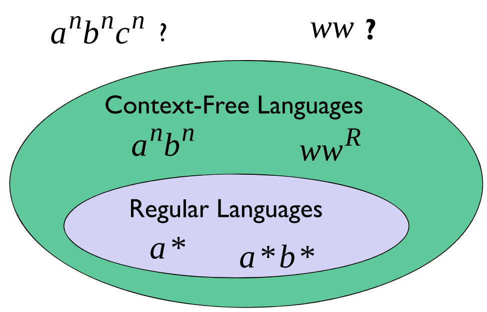
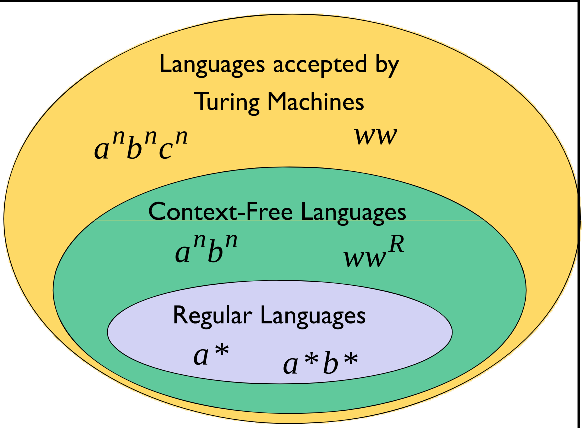
Department of Computer & Information Sciences
Pakistan Institute of Engineering and Applied Sciences

Turing machines

Umar Faiz
http://www.pieas.edu.pk/umarfaiz/cis317

The Language Hierarchy





Turing Machine

- Finite set of states
- Unbounded tape
  - A tape that extends infinitely in both directions
  - The tape is divided into cells, each cell can carry
    - A finite string (input) on the tape
    - blank characters, except
  - The symbols comes from a finite set of symbols called the **alphabet**

Turing Machine

- Tape head
  - Always scanning one cell
  - initially scanning leftmost character of input
- Movement
  - moves left or right
  - reads the current character on the tape
  - writes a character symbol

Turing Machine

$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$, where
- Q is the set of states
- Σ is the set of input symbols
- Γ is the set of tape symbols ($\Sigma \subseteq \Gamma$)
- δ is the transition function
  - $\delta: (Q \times \Gamma) \rightarrow (Q \times \Gamma \times \{left, right\})$
- q_0 is the starting state
- B is the blank symbol ($B \in \Gamma - \Sigma$)
- F is the set of final states

The Turing Machine (TM)

Turing Machine actions depend on two inputs:

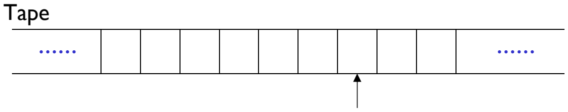
1. The current state of the machine
2. content of cell currently being read (input)

A Turing machine can do only one operation at a time. Each time an operation is done, three actions may take place:

3. write a symbol to cell
4. go into a new state
5. move one cell left or right

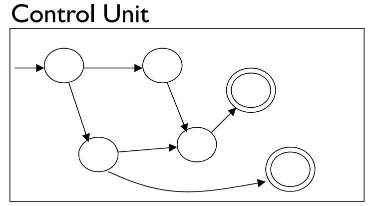
The Turing Machine (TM)

Tape



Read-Write head

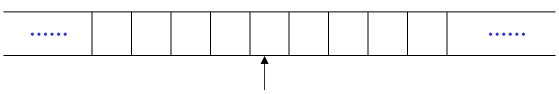
Control Unit



The Turing Machine (TM)

The Tape:

No boundaries -- infinite length

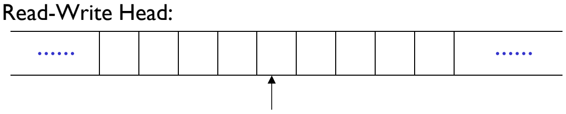


Read-Write head

The head moves Left or Right

The Turing Machine (TM)

Read-Write Head:



Read-Write head

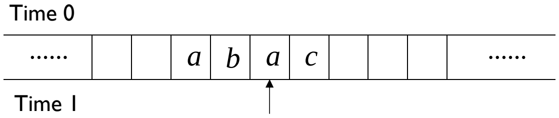
The head at each time step:

1. Reads a symbol
2. Writes a symbol
3. Moves Left or Right

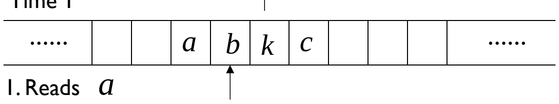
The Turing Machine (TM)

Example:

Time 0



Time 1

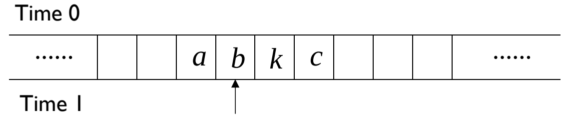


1. Reads a
2. Writes k
3. Moves Left

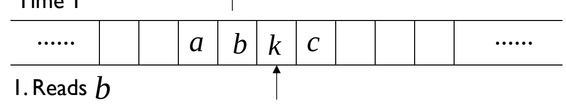
The Turing Machine (TM)

Example:

Time 0



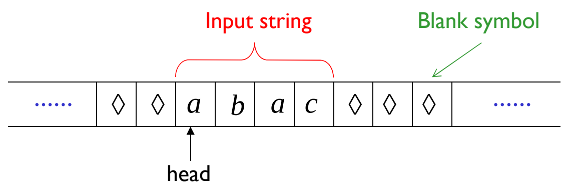
Time 1



1. Reads b
2. Writes f
3. Moves Right

Turing Machine (TM)

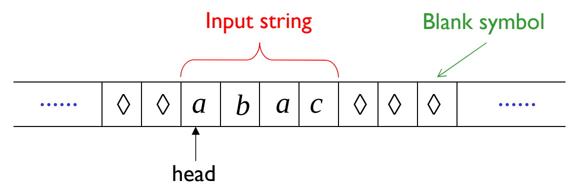
The Input String:



Head starts at the leftmost position of the input string

Turing Machine (TM)

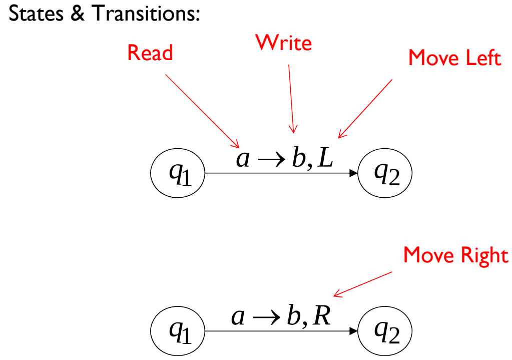
The Input String:



Remark: the input string is never empty

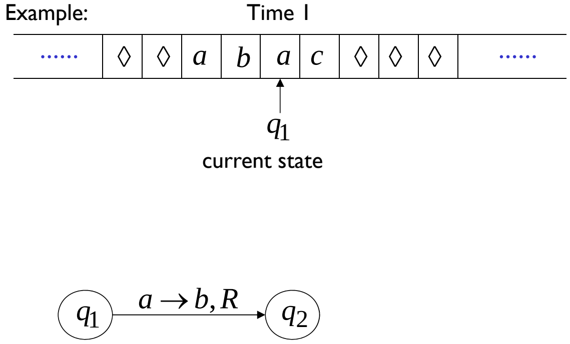
Turing Machine (TM)

States & Transitions:



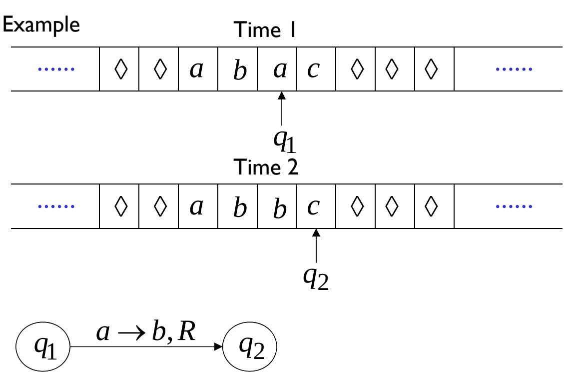
Turing Machine (TM)

Example: Time 1



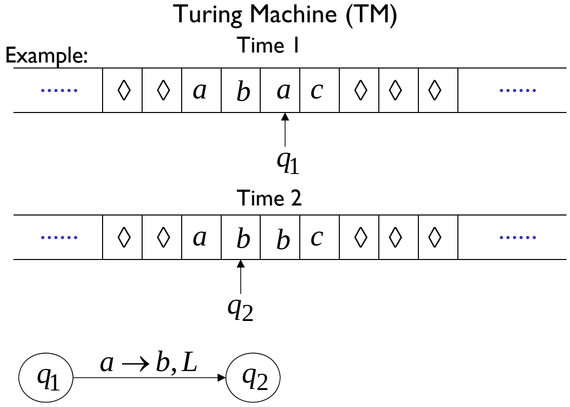
Turing Machine (TM)

Example



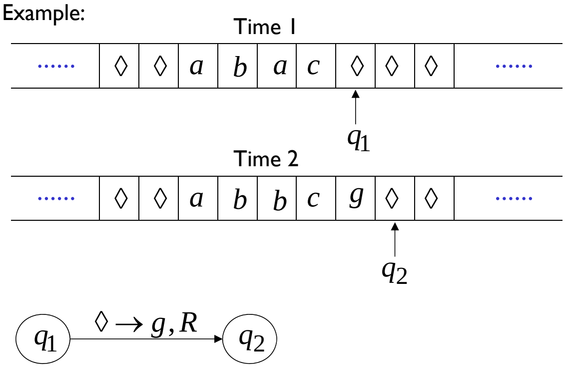
Turing Machine (TM)

Example:



Turing Machine (TM)

Example:

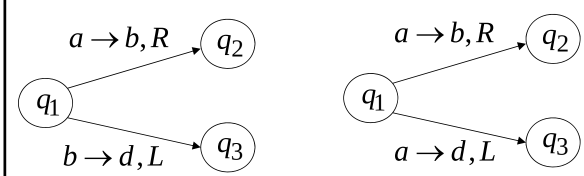


Turing Machine (TM)

Determinism

Turing Machines are deterministic

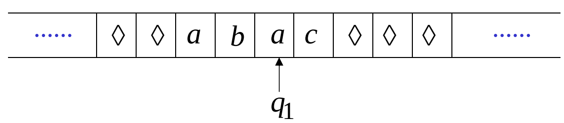
Allowed | Not Allowed

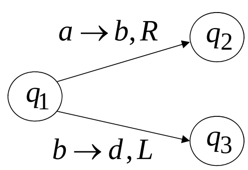


No lambda transitions allowed

TM - Partial Transition Function

Example:





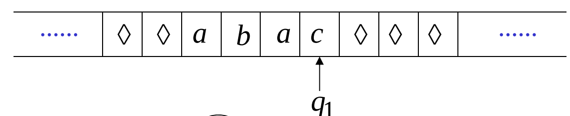
Allowed:

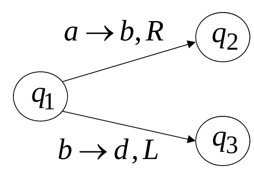
No transition for input symbol c

TM - Halting

The machine ***halts*** if there are no possible transitions to follow

Example:





No possible transition

HALT!!!

TM - Final States

Allowed

Not Allowed

- Final states have no outgoing transitions
- In a final state the machine halts

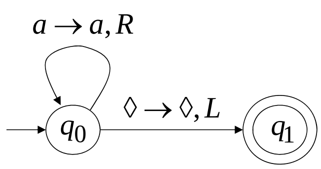
TM - Acceptance

Accept Input → If machine halts in a final state

Reject Input → If machine halts in a non-final state or If machine enters an infinite loop

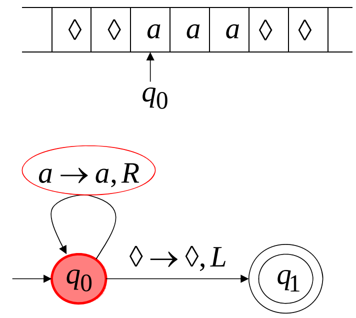
Turing Machine Example

A Turing machine that accepts the language: aa^*



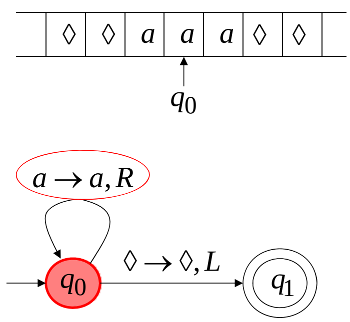
Turing Machine Example

Time 0



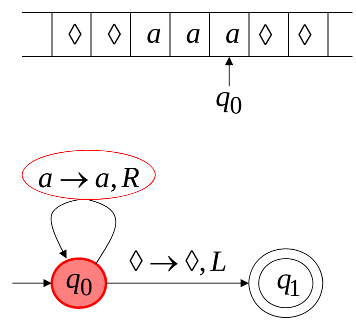
Turing Machine Example

Time 1



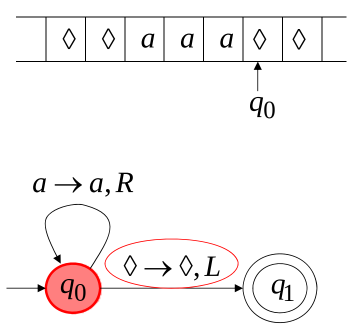
Turing Machine Example

Time 2



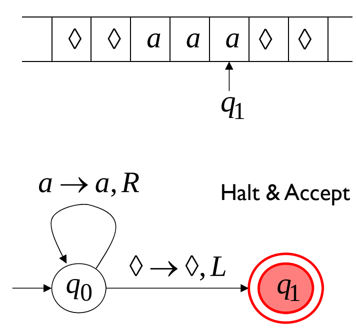
Turing Machine Example

Time 3



Turing Machine Example

Time 4

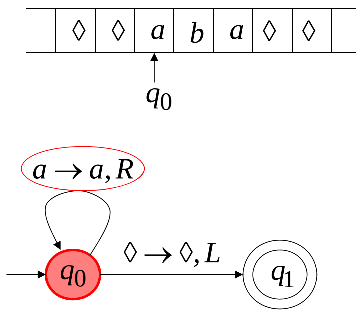


Halt & Accept

Turing Machine Example

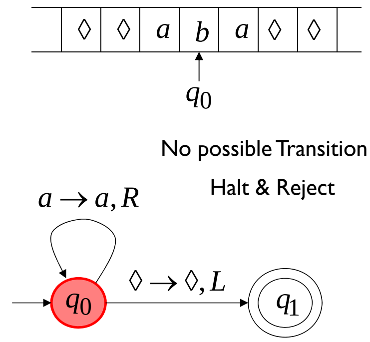
Rejection Example:

Time 0



Turing Machine Example

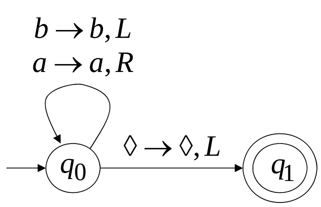
Time 1



Turing Machine Example

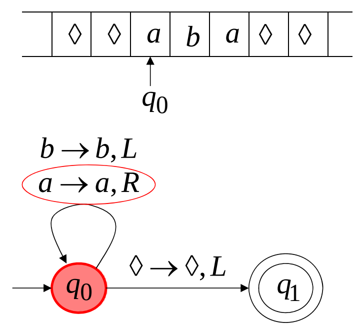
Infinite Loop Example:

- A Turing machine for language $aa^* + b(a+b)^*$



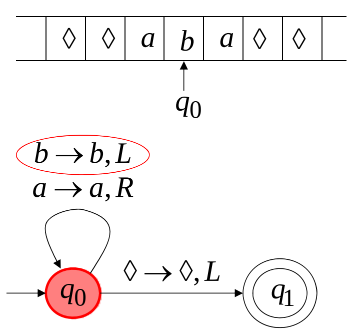
Turing Machine Example

Time 0



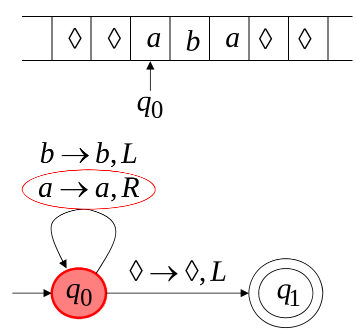
Turing Machine Example

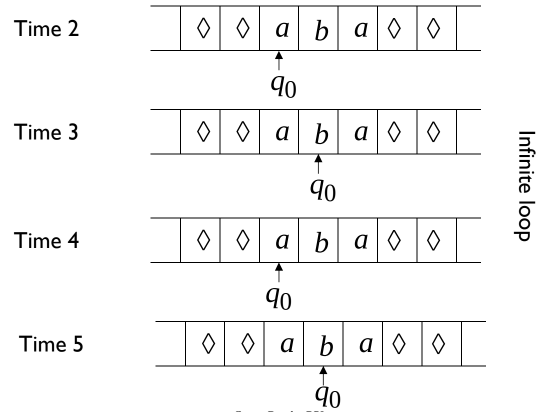
Time 1



Turing Machine Example

Time 2





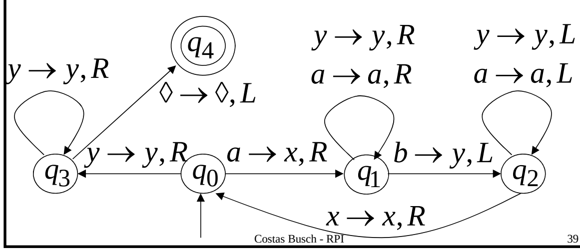
Turing Machine Example

Because of the infinite loop:

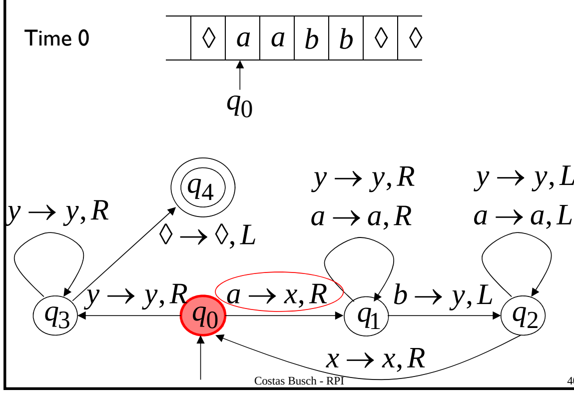
- The final state cannot be reached
- The machine never halts
- The input is NOT ACCEPTED

Turing Machine Example

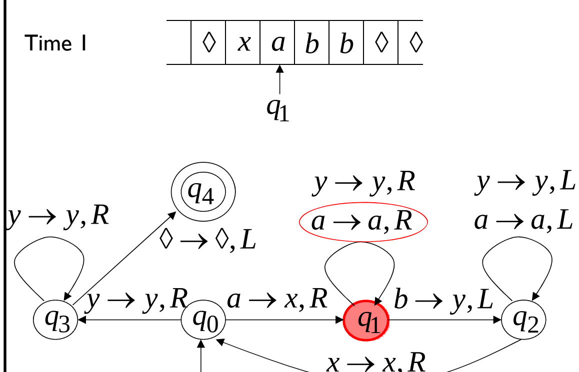
Turing machine for the language $\{a^n b^n\}$



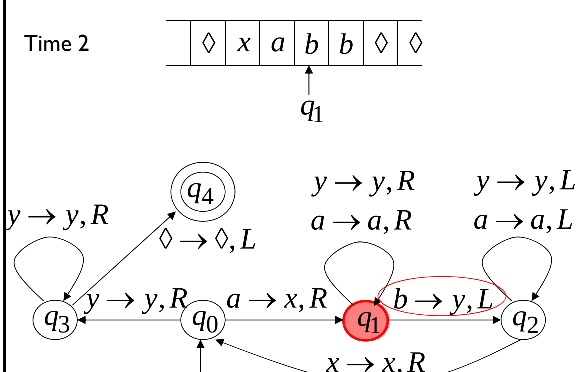
Turing Machine Example

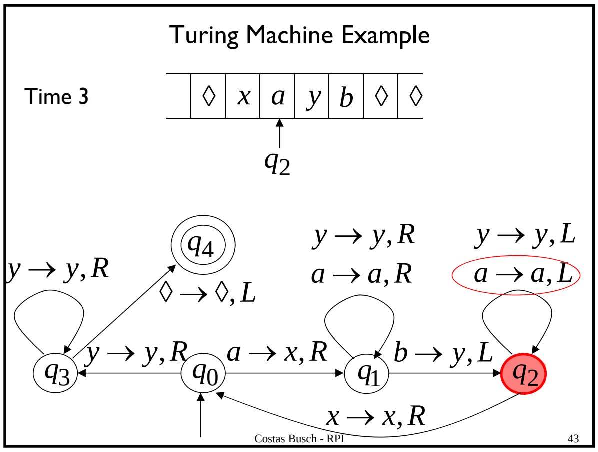


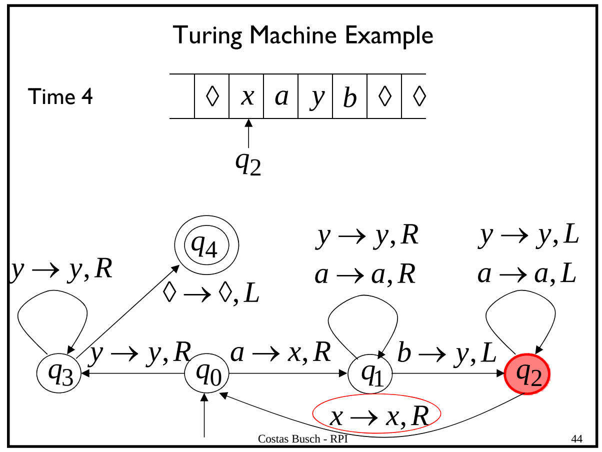
Turing Machine Example

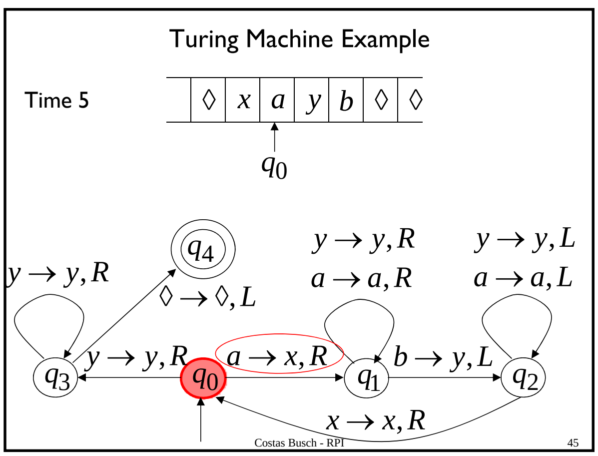


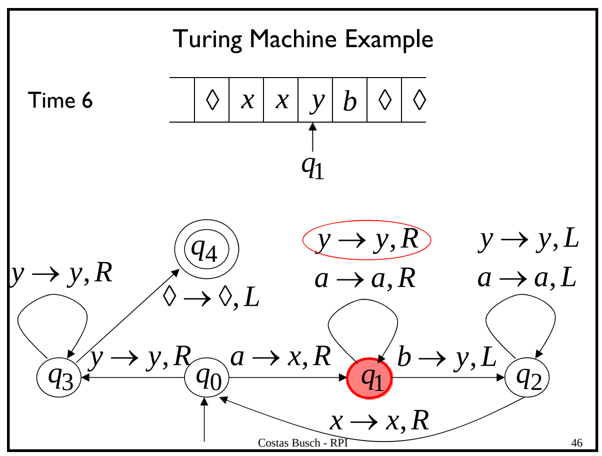
Turing Machine Example

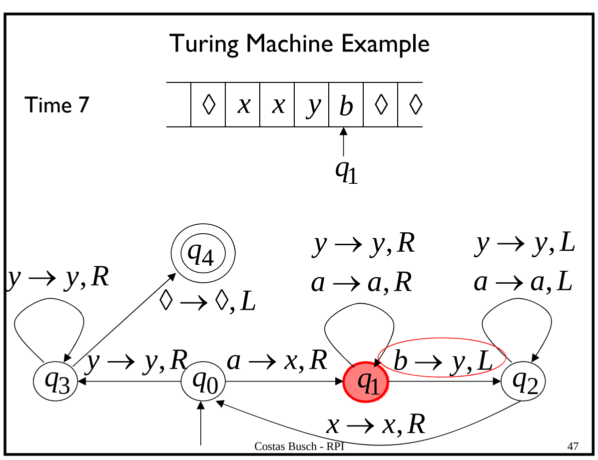


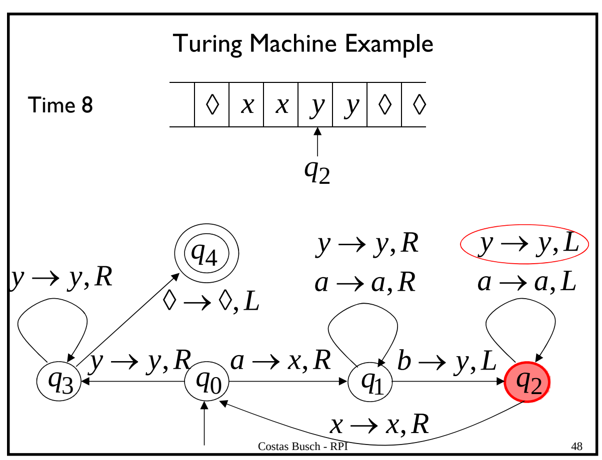

Turing Machine Example
Time 3
◊ x a y b ◊ ◊
q2
y → y,R
◊ → ◊,L
y → y,R
a → a,R
y → y,L
a → a,L
y → y,R
a → x,R
b → y,L
x → x,R
q0 q1 q2 q3 q4
Costas Busch - RPI
43


Turing Machine Example
Time 4
◊ x a y b ◊ ◊
q2
y → y,R
◊ → ◊,L
y → y,R
a → a,R
y → y,L
a → a,L
y → y,R
a → x,R
b → y,L
x → x,R
q0 q1 q2 q3 q4
Costas Busch - RPI
44


Turing Machine Example
Time 5
◊ x a y b ◊ ◊
q0
y → y,R
◊ → ◊,L
y → y,R
a → a,R
y → y,L
a → a,L
y → y,R
a → x,R
b → y,L
x → x,R
q0 q1 q2 q3 q4
Costas Busch - RPI
45


Turing Machine Example
Time 6
◊ x x y b ◊ ◊
q1
y → y,R
◊ → ◊,L
y → y,R
a → a,R
y → y,L
a → a,L
y → y,R
a → x,R
b → y,L
x → x,R
q0 q1 q2 q3 q4
Costas Busch - RPI
46


Turing Machine Example
Time 7
◊ x x y b ◊ ◊
q1
y → y,R
◊ → ◊,L
y → y,R
a → a,R
y → y,L
a → a,L
y → y,R
a → x,R
b → y,L
x → x,R
q0 q1 q2 q3 q4
Costas Busch - RPI
47

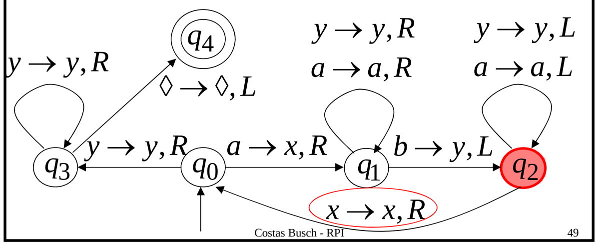

Turing Machine Example
Time 8
◊ x x y y ◊ ◊
q2
y → y,R
◊ → ◊,L
y → y,R
a → a,R
y → y,L
a → a,L
y → y,R
a → x,R
b → y,L
x → x,R
q0 q1 q2 q3 q4
Costas Busch - RPI
48

Turing Machine Example

Time 9

◊	x	x	y	y	◊	◊

q_2

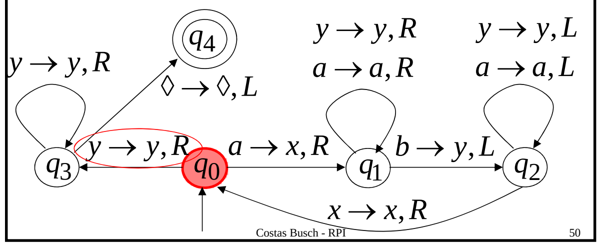


Turing Machine Example

Time 10

◊	x	x	y	y	◊	◊

q_0

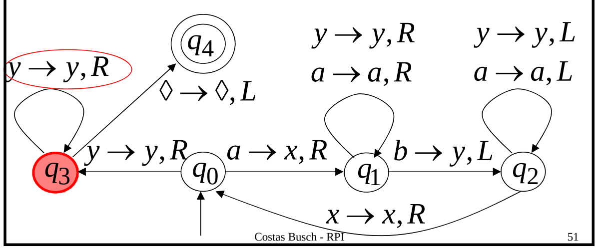


Turing Machine Example

Time 11

◊	x	x	y	y	◊	◊

q_3

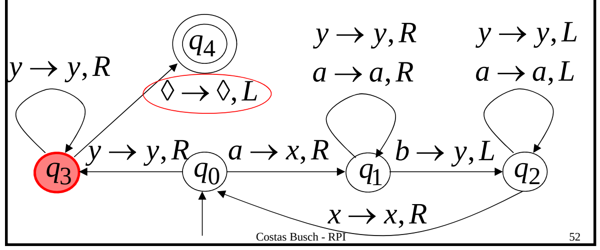


Turing Machine Example

Time 12

◊	x	x	y	y	◊	◊

q_3



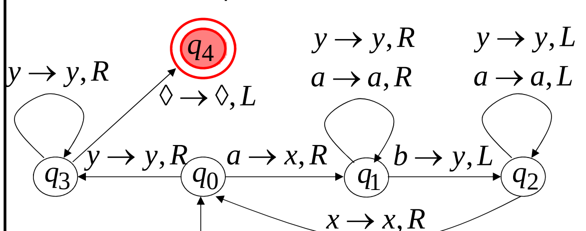
Turing Machine Example

Time 13

◊	x	x	y	y	◊	◊

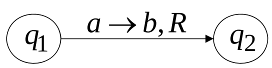
q_4

Halt & Accept



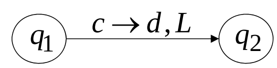
Formal Definitions for Turing Machines

Transition Function



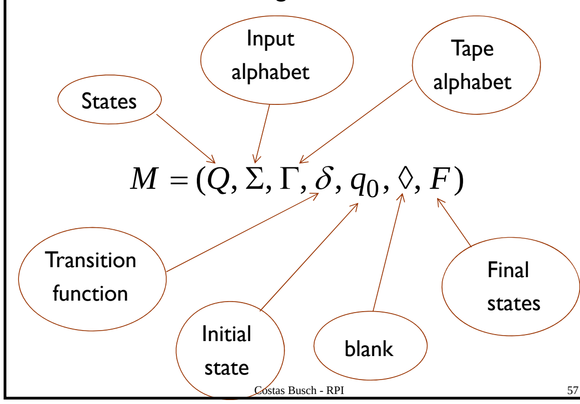
$$\delta(q_1, a) = (q_2, b, R)$$

Transition Function



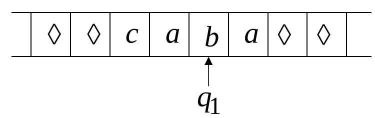
$$\delta(q_1, c) = (q_2, d, L)$$

Turing Machine



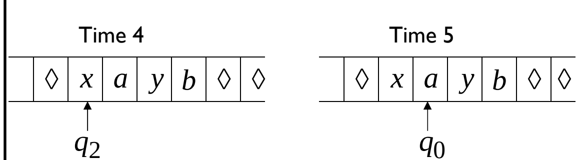
$$M = (Q, \Sigma, \Gamma, \delta, q_0, \lozenge, F)$$

Configuration



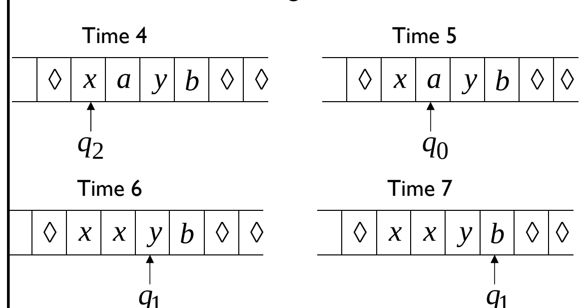
Instantaneous description: $ca\ q_1\ ba$

Configuration



A Move: $q_2\ xayb \succ x\ q_0\ ayb$

Configuration



$$q_2\ xayb \succ x\ q_0\ ayb \succ xx\ q_1\ yb \succ xxy\ q_1\ b$$

Configuration

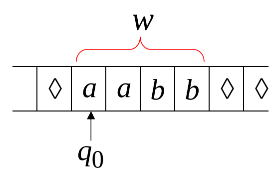
$$q_2\ xayb \succ x\ q_0\ ayb \succ xx\ q_1\ yb \succ xxy\ q_1\ b$$

Equivalent notation: $q_2\ xayb \overset{*}{\succ} xxy\ q_1\ b$

Configuration

Initial configuration: $q_0\ w$

Input string



The Accepted Language

For any Turing Machine M

$$L(M) = \{w : \quad q_0\ w \overset{*}{\succ} x_1\ q_f\ x_2\}$$

Initial state — Final state

Standard Turing Machine

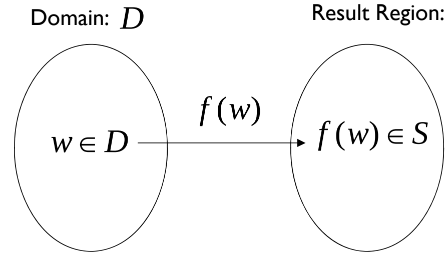
- The machine we described is the standard:
  - Deterministic
  - Infinite tape in both directions
  - Tape is the input/output file

Computing Functions with Turing Machines

Computing Functions with Turing Machines

A function $f(w)$ has:

Domain: D — Result Region: S



Computing Functions with Turing Machines

A function may have many parameters:

Example: Addition function

$$f(x, y) = x + y$$

Computing Functions with Turing Machines

Integer Domain:

Decimal: 5

Binary: 101

Unary: 11111

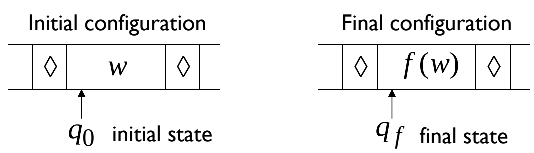
We prefer **unary** representation:

easier to manipulate with Turing machines

Computing Functions with Turing Machines

Definition:

A function f is computable if there is a Turing Machine M such that:



For all $w \in D$ Domain

Computing Functions with Turing Machines

Definition:

A function f is computable if there is a Turing Machine M such that:

$$q_0\, w \overset{*}{\succ} q_f\, f(w)$$

Initial Configuration → $q_0\, w$; Final Configuration → $q_f\, f(w)$

For all $w \in D$ Domain

Computing Functions with Turing Machines

Example:

The function $f(x, y) = x + y$ is computable

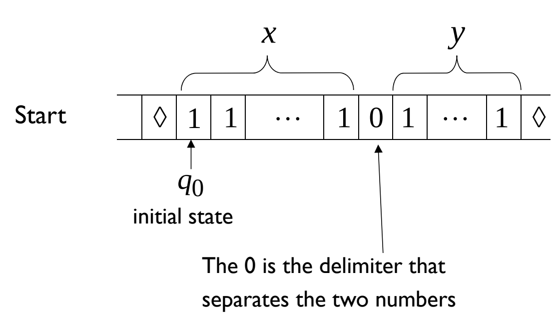
x, y are integers

Turing Machine:

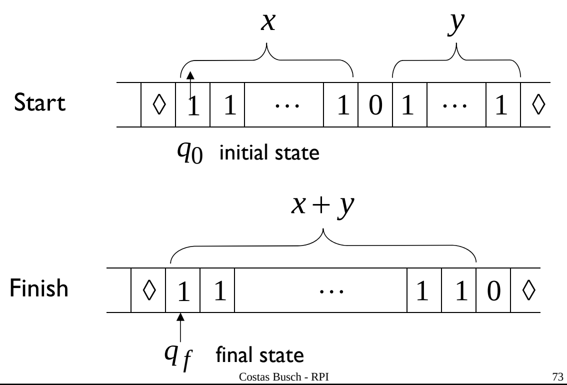
Input string: $x0y$ unary

Output string: $xy0$ unary

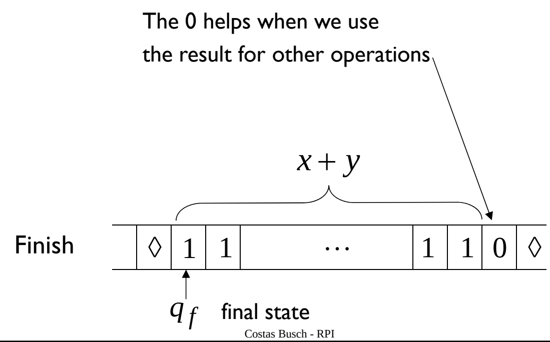
Computing Functions with Turing Machines



Computing Functions with Turing Machines

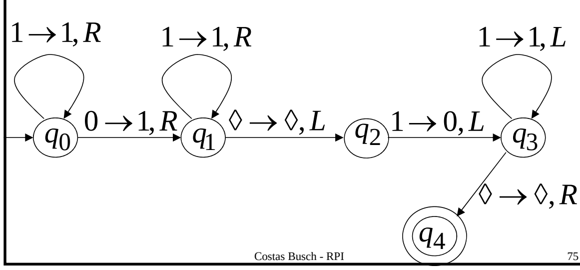


Computing Functions with Turing Machines



Computing Functions with Turing Machines

Turing machine for function $f(x, y) = x + y$

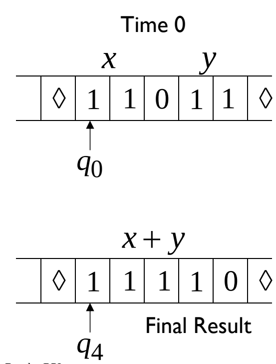


Computing Functions with Turing Machines

Execution Example:

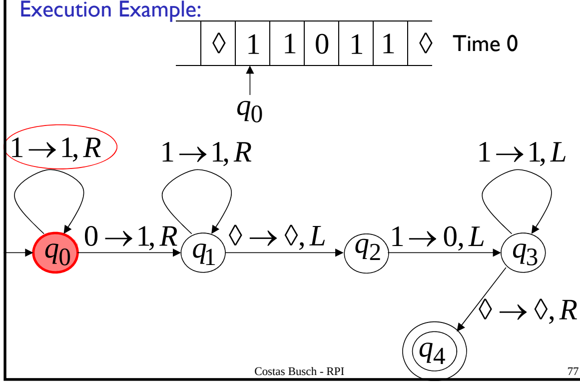
$x = 11$ (2)

$y = 11$ (2)



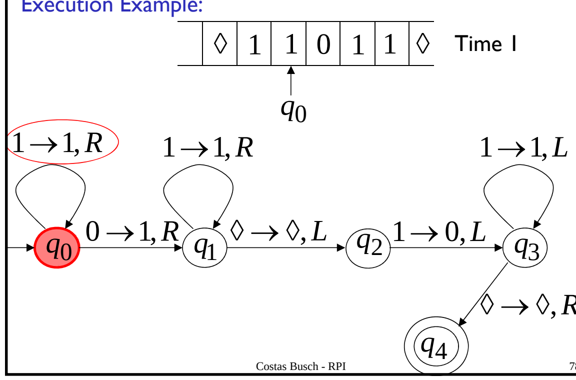
Computing Functions with Turing Machines

Execution Example:



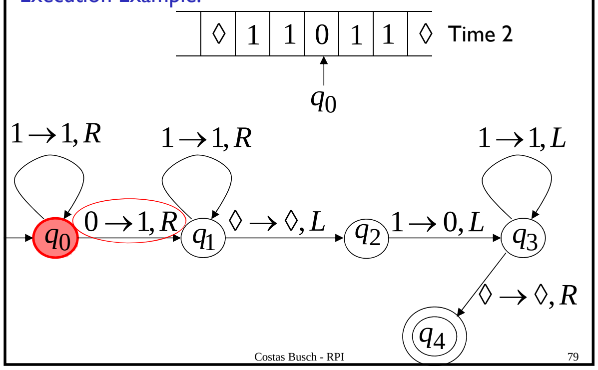
Computing Functions with Turing Machines

Execution Example:



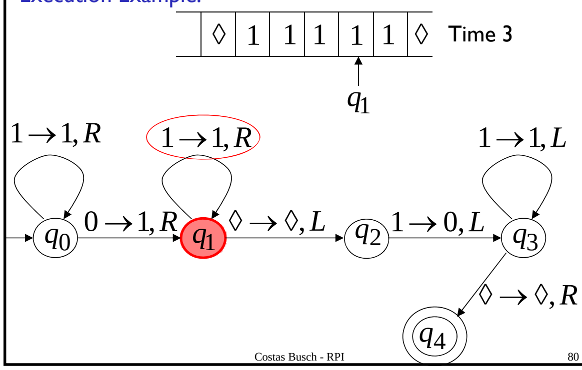
Computing Functions with Turing Machines

Execution Example:



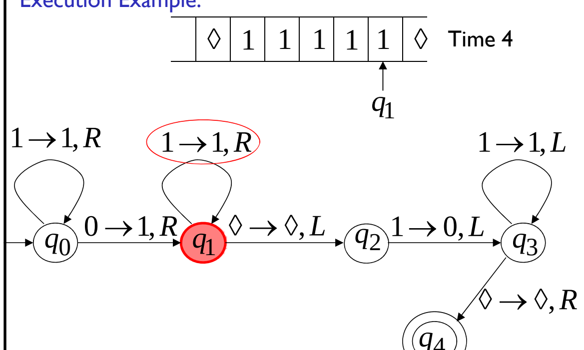
Computing Functions with Turing Machines

Execution Example:



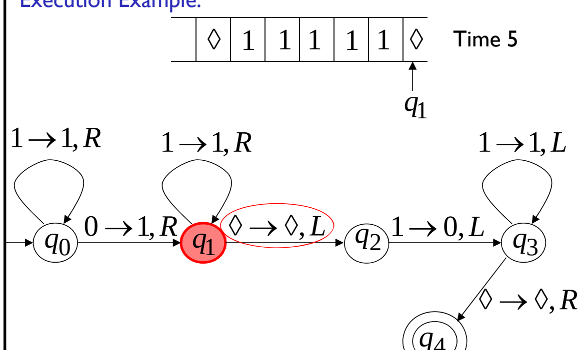
Computing Functions with Turing Machines

Execution Example:



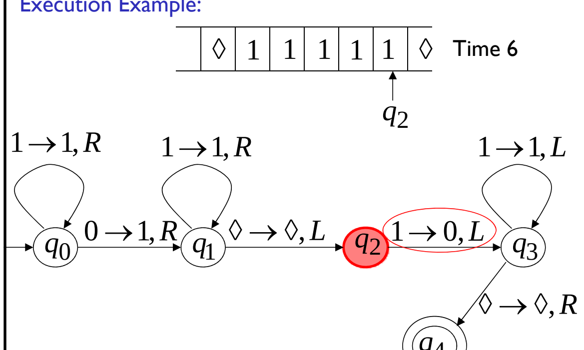
Computing Functions with Turing Machines

Execution Example:



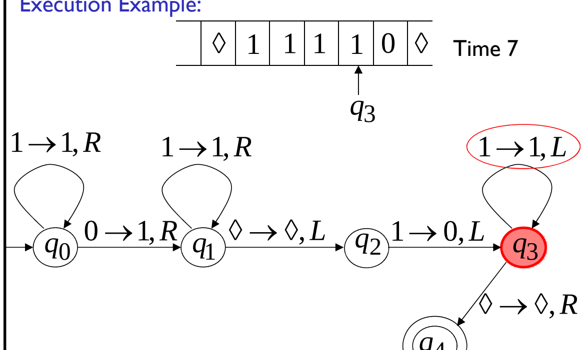
Computing Functions with Turing Machines

Execution Example:



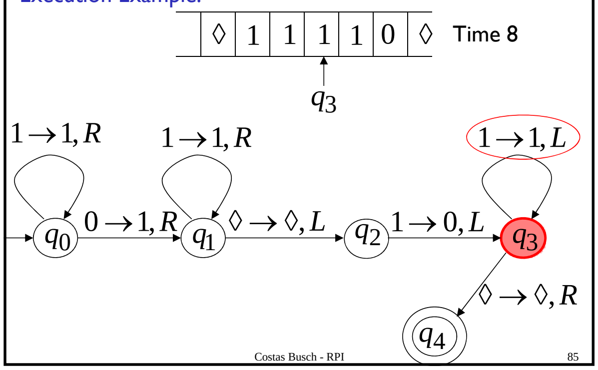
Computing Functions with Turing Machines

Execution Example:



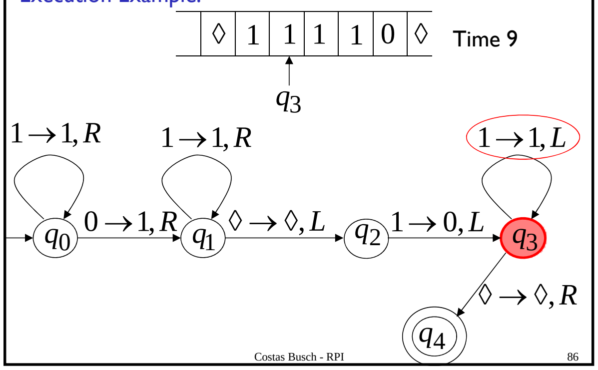
Computing Functions with Turing Machines

Execution Example:



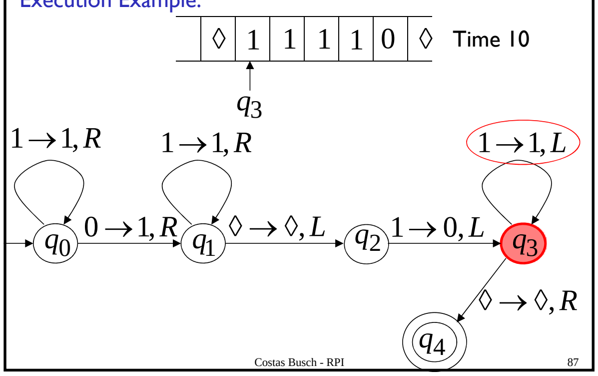
Computing Functions with Turing Machines

Execution Example:



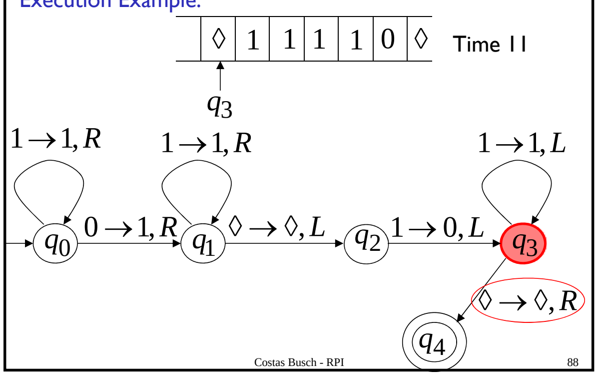
Computing Functions with Turing Machines

Execution Example:



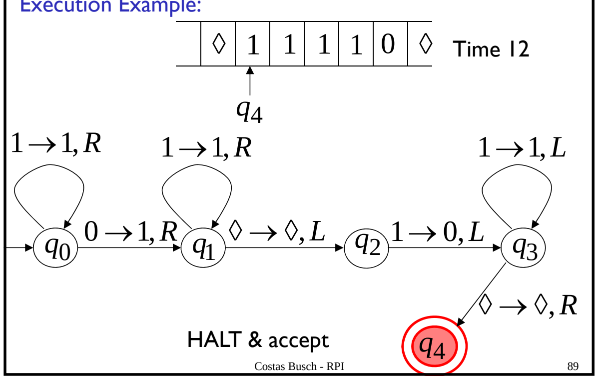
Computing Functions with Turing Machines

Execution Example:



Computing Functions with Turing Machines

Execution Example:



Computing Functions with Turing Machines

The function $f(x) = 2x$ is computable

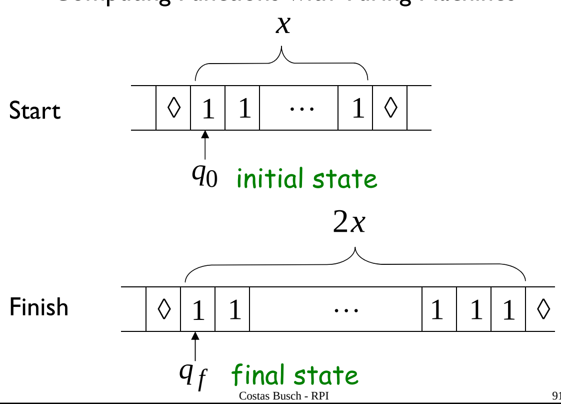
x is integer

Turing Machine:

Input string: x unary

Output string: xx unary

Computing Functions with Turing Machines



Computing Functions with Turing Machines

Turing Machine Pseudocode for $f(x) = 2x$

- Replace every 1 with $

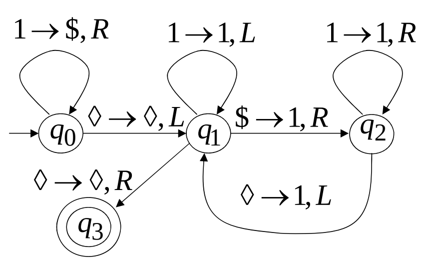
Repeat:

- Find rightmost $, replace it with 1
- Go to right end, insert 1

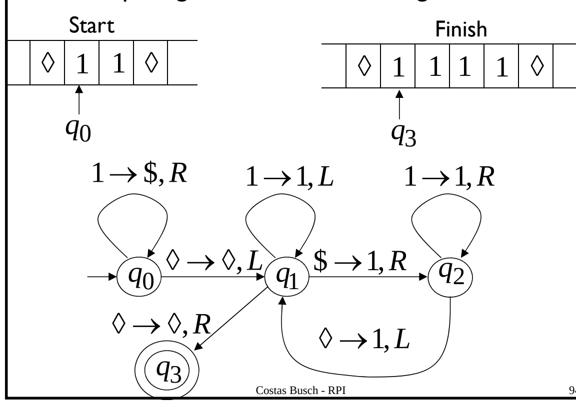
Until no more $ remain

Computing Functions with Turing Machines

Turing Machine for $f(x) = 2x$



Computing Functions with Turing Machines



Computing Functions with Turing Machines

The function is computable

$$f(x,y) = \begin{cases} 1 & \text{if } x > y \\ 0 & \text{if } x \le y \end{cases}$$

Computing Functions with Turing Machines

Turing Machine for

$$f(x,y) = \begin{cases} 1 & \text{if } x > y \\ 0 & \text{if } x \le y \end{cases}$$

Input: $x0y$

Output: 1 or 0

Computing Functions with Turing Machines

Turing Machine Pseudocode:

Repeat

Match a 1 from x with a 1 from y

Until all of x or y is matched

If a 1 from x is not matched

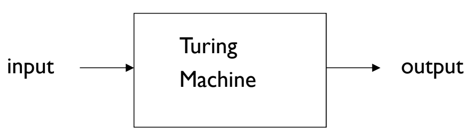
erase tape, write 1 $(x > y)$

else

erase tape, write 0 $(x \leq y)$

Combining Turing Machines

Block Diagram



Computing Functions with Turing Machines

Example:

$$f(x,y) = \begin{cases} x+y & \text{if } x > y \\ 0 & \text{if } x \leq y \end{cases}$$