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

Variations of the Turing Machine

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

Church Turing's Thesis

The Church-Turing Thesis asserts that every solvable decision problem can be transformed into an equivalent Turing machine problem.

In other word, the Church-Turing thesis asserts that

- a decision problem *P has a solution if, and only if, there* exists a TM that determines answer for every $p \in P$
- if no such TM exists, the problem is said to be *undecidable*

Church Turing's Thesis

The Church-Turing Thesis for Decision Problems:

A decision problem

- consists of a set of questions whose answers are either yes or no
- is undecidable if no algorithm that can solve the problem; otherwise, it is decidable

- An unsolvable problem is a problem such that there does not exist any TM that can solve the problem
- A solution to a decision problem is a equivalent to the question of membership in a *recursive language.*

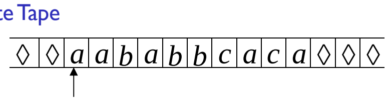
Church Turing's Thesis

The Church-Turing thesis for Recognition Problems:

- A decision problem P is *partially solvable if, and only if, there is* a TM that accepts precisely the instances of P whose answer is "yes".
- A partial solution to a decision problem is equivalent to the question of membership in a *recursively enumerable language*

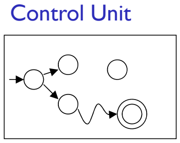
The Standard Model

Infinite Tape



Read-Write Head (Left or Right)

Control Unit



Deterministic

Variations of the Standard Model

Turing machines with:

- Stay-Option
- Semi-Infinite Tape
- Off-Line
- Multitape
- Multidimensional
- Nondeterministic

Variations of the Standard Model

- The variations form different Turing Machine Classes
- Each Class has the same power with the Standard Model

Variations of the Standard Model

All Variants have the same power as the Standard TM

Same power of two classes means that both classes of Turing machines accept the same languages

Variations of the Standard Model

Same Power of two classes means:

– For any machine M_1 of first class there is a machine M_2 of second class such that:

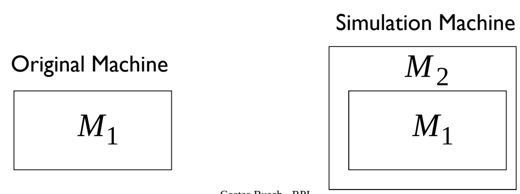
$$L(M_1) = L(M_2)$$

and vice-versa

Variations of the Standard Model

Simulation: A technique to prove same power

– Simulate the machine of one class with a machine of the other class



Variations of the Standard Model

- Configurations in the Original Machine correspond to configurations in the Simulation Machine

Original Machine: $d_0 \succ d_1 \succ \cdots \succ d_n$

$\updownarrow \quad \updownarrow \quad \updownarrow$

Simulation Machine: $d'_0 \overset{*}{\succ} d'_1 \overset{*}{\succ} \cdots \overset{*}{\succ} d'_n$

Variations of the Standard Model

Final Configuration:

Original Machine: d_f

$\updownarrow$

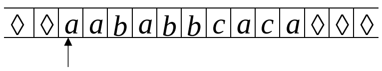
Simulation Machine: d'_f

- The Simulation Machine and the Original Machine accept the same language

Variations of the Standard Model

Turing Machines with Stay-Option:

The head can stay in the same position



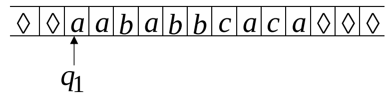
Left, Right, Stay

L,R,S: moves

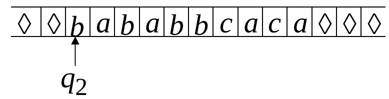
Variations of the Standard Model

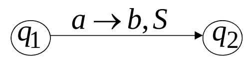
Example:

Time 1



Time 2





Variations of the Standard Model

Stay-Option Machines:

Theorem: Stay-Option Machines have the same power with Standard Turing machines

Variations of the Standard Model

Proof:

Part 1: Stay-Option Machines are at least as powerful as Standard machines

Proof:

- A Standard machine is also a Stay-Option machine (that never uses the S move)

Variations of the Standard Model

Proof:

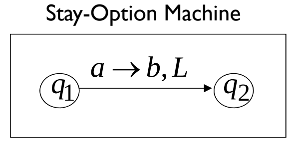
Part II: Standard machines are at least as powerful as Stay-Option Machines

Proof:

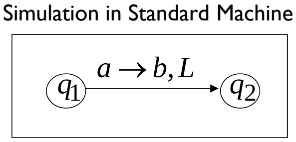
- A Standard machine can simulate a Stay-Option machine

Variations of the Standard Model

Stay-Option Machine



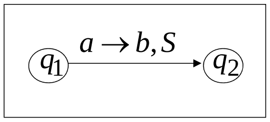
Simulation in Standard Machine



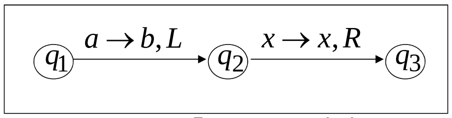
Similar for Right moves

Variations of the Standard Model

Stay-Option Machine



Simulation in Standard Machine

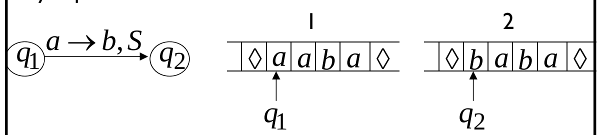


For every symbol x

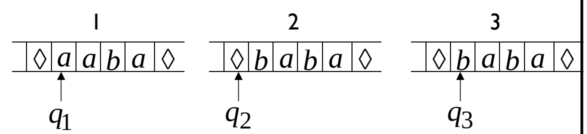
Variations of the Standard Model

Example:

Stay-Option Machine:

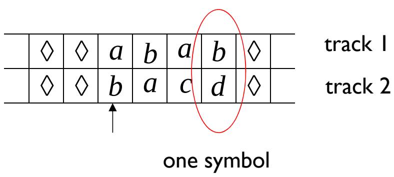


Simulation in Standard Machine:

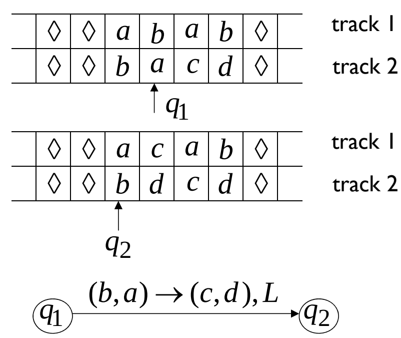


Variations of the Standard Model

Standard Machine--Multiple Track Tape

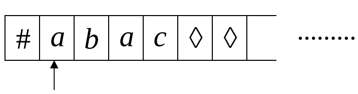


Variations of the Standard Model



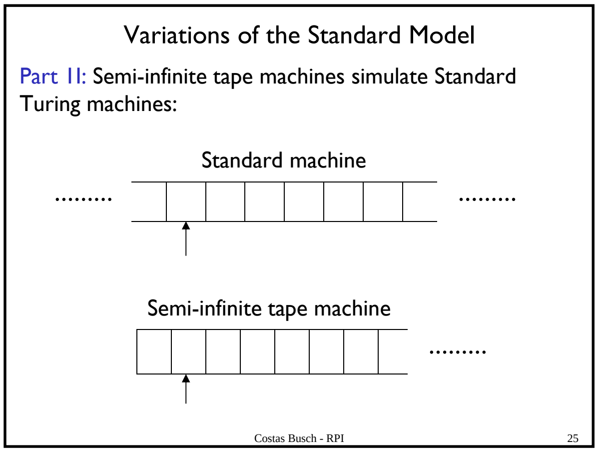
Variations of the Standard Model

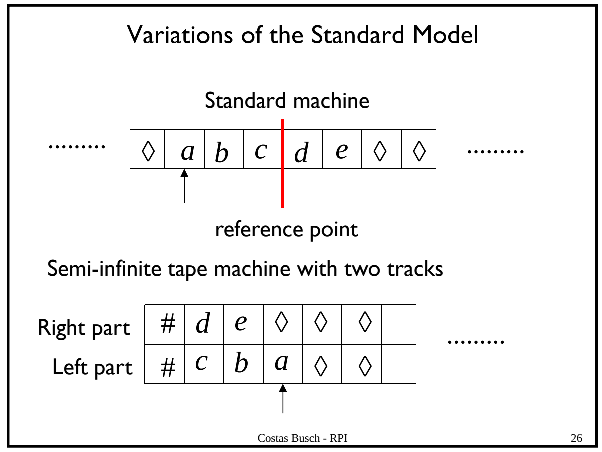
Semi-Infinite Tape:

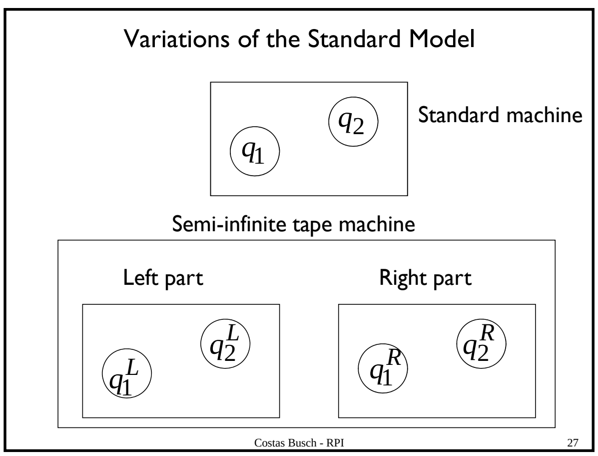


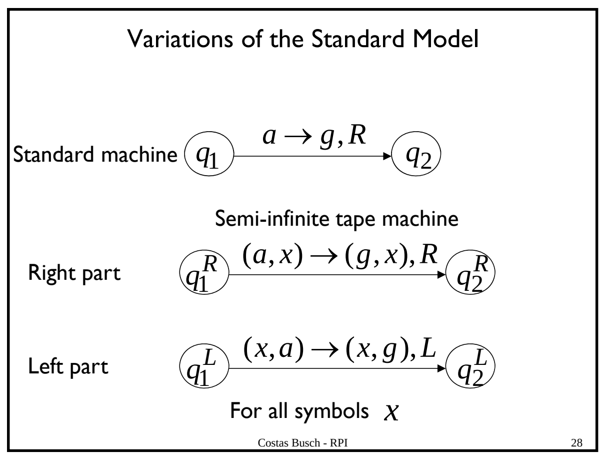
Variations of the Standard Model

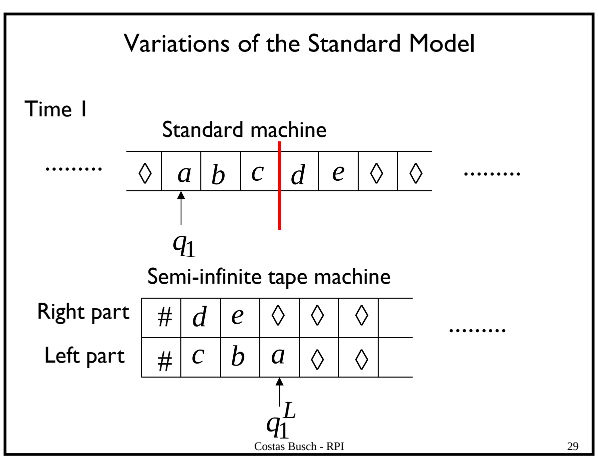
Part 1: Standard Turing machines simulate Semi-infinite tape machines

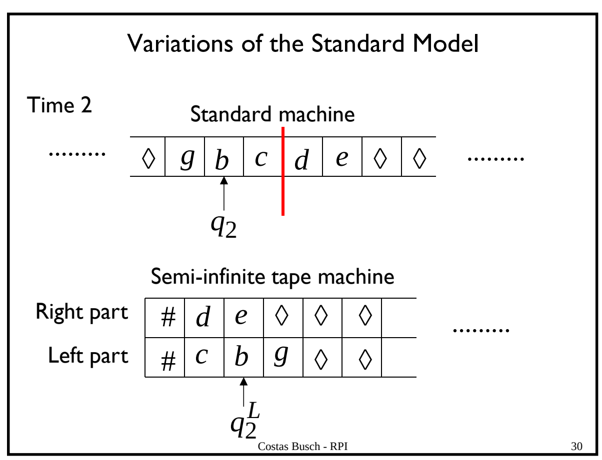

Variations of the Standard Model
Part II: Semi-infinite tape machines simulate Standard Turing machines:
Standard machine
....
....
Semi-infinite tape machine
....
Costas Busch - RPI
25


Variations of the Standard Model
Standard machine
....
◊ a b c d e ◊ ◊
....
reference point
Semi-infinite tape machine with two tracks
Right part # d e ◊ ◊ ◊
Left part # c b a ◊ ◊
....
Costas Busch - RPI
26


Variations of the Standard Model
q_1 q_2
Standard machine
Semi-infinite tape machine
Left part
Right part
q_1^L q_2^L
q_1^R q_2^R
Costas Busch - RPI
27

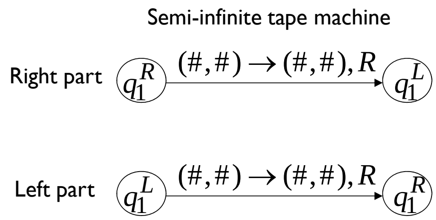

Variations of the Standard Model
Standard machine q_1 $a \to g, R$ q_2
Semi-infinite tape machine
Right part q_1^R $(a, x) \to (g, x), R$ q_2^R
Left part q_1^L $(x, a) \to (x, g), L$ q_2^L
For all symbols x
Costas Busch - RPI
28


Variations of the Standard Model
Time 1
Standard machine
....
◊ a b c d e ◊ ◊
....
q_1
Semi-infinite tape machine
Right part # d e ◊ ◊ ◊
Left part # c b a ◊ ◊
....
q_1^L
Costas Busch - RPI
29


Variations of the Standard Model
Time 2
Standard machine
....
◊ g b c d e ◊ ◊
....
q_2
Semi-infinite tape machine
Right part # d e ◊ ◊ ◊
Left part # c b g ◊ ◊
....
q_2^L
Costas Busch - RPI
30

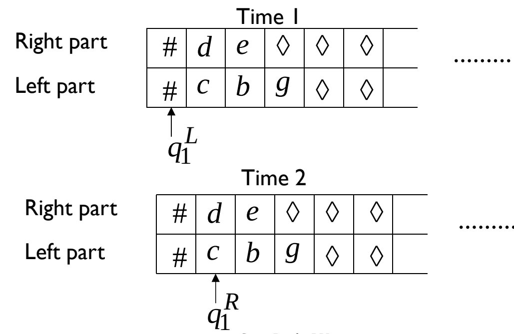
Variations of the Standard Model

At the border:



Variations of the Standard Model

Semi-infinite tape machine:

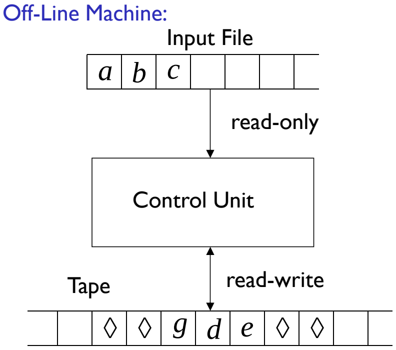


Variations of the Standard Model

Theorem: Semi-infinite tape machines have the same power with Standard Turing machines

Variations of the Standard Model

The Off-Line Machine:



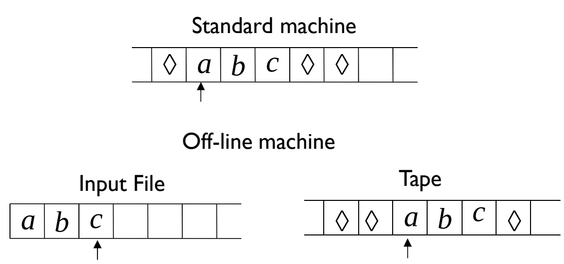
Variations of the Standard Model

Part I: Off-line machines simulate Standard Turing Machines

Off-line machine:

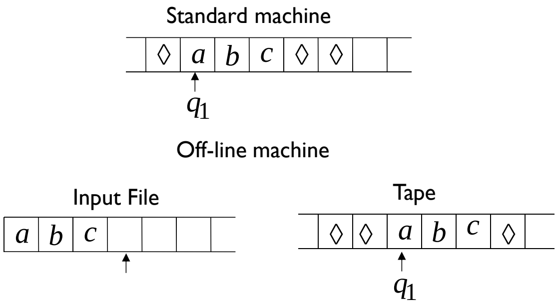
1. Copy input file to tape
2. Continue computation as in Standard Turing machine

Variations of the Standard Model



1. Copy input file to tape

Variations of the Standard Model

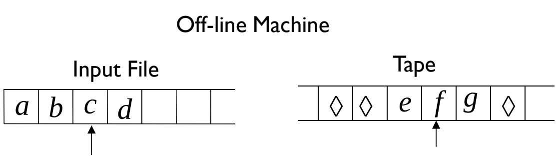


2. Do computations as in Turing machine

Variations of the Standard Model

- Part II: Standard Turing machines simulate Off-line machines:
  – Use a Standard machine with four track tape to keep track of the Off-line input file and tape contents

Variations of the Standard Model



Four track tape - Standard Machine

#	a	b	c	d		Input File
#	0	0	1	0		head position
	e	f	g			Tape
	0	1	0			head position

Reference point

#	a	b	c	d		Input File
#	0	0	1	0		head position
	e	f	g			Tape
	0	1	0			head position

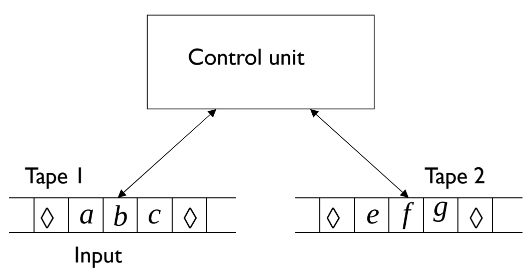
Repeat for each state transition:
- Return to reference point
- Find current input file symbol
- Find current tape symbol
- Make transition

Variations of the Standard Model

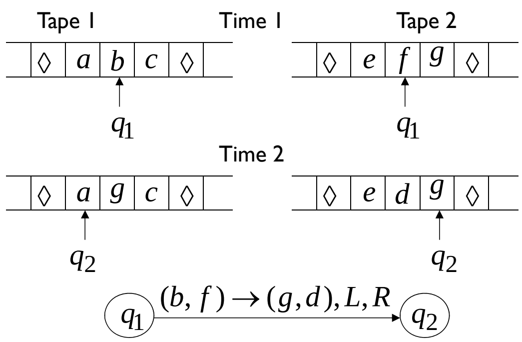
Theorem: Off-line machines have the same power with Standard machines

Variations of the Standard Model

Multitape Turing Machines:



Variations of the Standard Model



Variations of the Standard Model

Part I: Multitape machines simulate Standard Machines:

– Use just one tape

Variations of the Standard Model

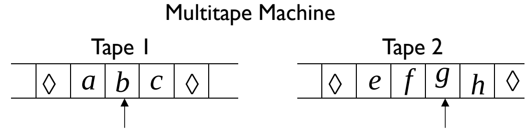
Part II: Standard machines simulate Multitape machines:

Standard machine:

- Use a multi-track tape
- A tape of the Multiple tape machine corresponds to a pair of tracks

Variations of the Standard Model

Multitape Machine



Standard machine with four track tape

		a	b	c				Tape 1
		0	1	0				head position
		e	f	g	h			Tape 2
		0	0	1	0			head position

Reference point

	#	a	b	c			Tape 1
	#	0	1	0			head position
	#	e	f	g	h		Tape 2
	#	0	0	1	0		head position

Repeat for each state transition:

- Return to reference point
- Find current symbol in Tape 1
- Find current symbol in Tape 2
- Make transition

Variations of the Standard Model

Theorem: Multi-tape machines have the same power with Standard Turing Machines

Variations of the Standard Model

Same power doesn't imply same speed:

Language $L = \{a^n b^n\}$

	Acceptance Time
Standard machine	n^2
Two-tape machine	n

Variations of the Standard Model

Standard machine: $L = \{a^n b^n\}$

Go back and forth n^2 times

Two-tape machine:

Copy b^n to tape 2 (n steps)

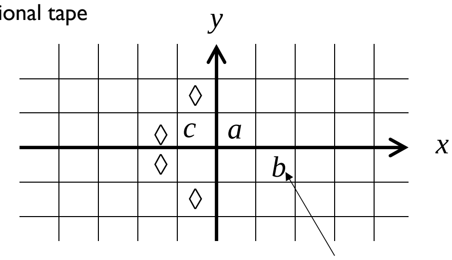
Leave a^n on tape 1 (n steps)

Compare tape 1 and tape 2 (n steps)

Variations of the Standard Model

MultiDimensional Turing Machines:

Two-dimensional tape



MOVES: L,R,U,D

U: up D: down

HEAD

Position: +2, -1

Variations of the Standard Model

Part I: Multidimensional machines simulate Standard machines:

– Use one dimension

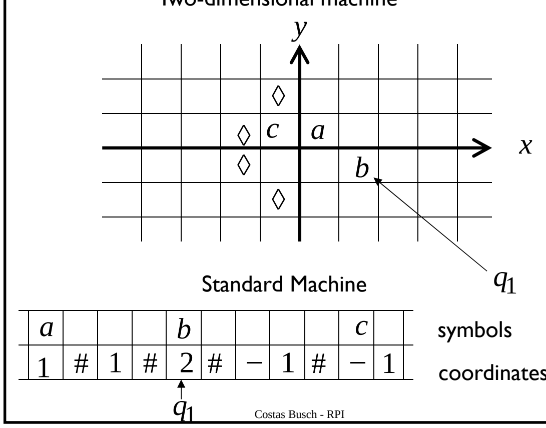
Variations of the Standard Model

Part II: Standard machines simulate Multidimensional machines:

Standard machine:

– Use a two track tape
– Store symbols in track 1
– Store coordinates in track 2

Two-dimensional machine



Variations of the Standard Model

Standard machine:

- Repeat for each transition
  - Update current symbol
  - Compute coordinates of next position
  - Go to new position

Variations of the Standard Model

Theorem: Multi-Dimensional Machines have the same power with Standard Turing Machines

Turing Machines and Computers

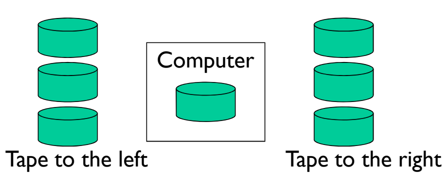
- A computer can simulate a Turing Machine
- A Turing Machine can simulate a computer

Computer Simulation of a TM

- No computer with a finite amount of memory is able to simulate a Turing Machine
  - Consider that the hard drive is the tape – you can buy very large ones, but they are finite
  - A TM can solve problems that require more memory than the memory available to the computer
    - Consider a string so awfully large that is larger than the computer's memory

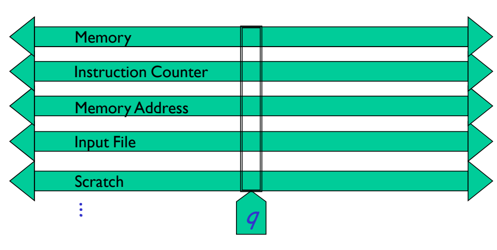
Computer Simulation of a TM

- Assume
  - Unbounded amounts of removable media
  - Each disk represents a segment of the tape
  - The computer can have one segment at any given time



TM Simulation of a Computer

- A computer can be simulated by a multi-tape Turing Machine

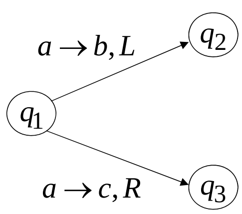


TM Simulation of a Computer

- The good news …
  - The simulation only takes polynomial time!

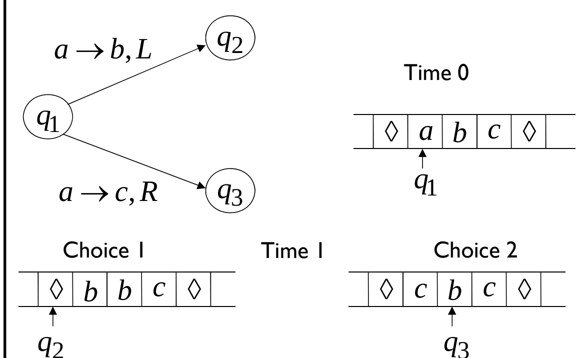
NonDeterministic Turing Machines

NonDeterministic Turing Machines



Non Deterministic Choice

NonDeterministic Turing Machines



NonDeterministic Turing Machines

Input string w is accepted if this a possible computation

$$q_0 w \overset{*}{\succ} x\, q_f\, y$$

Initial configuration

Final Configuration

Final state

NonDeterministic Turing Machines

- NonDeterministic Machines simulate Standard (deterministic) Machines:
  - Every deterministic machine is also a nondeterministic machine

NonDeterministic Turing Machines

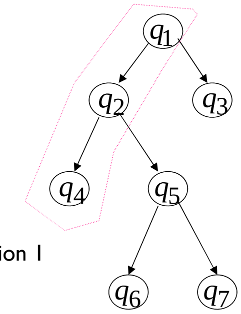
- Deterministic machines simulate NonDeterministic Machines:

Deterministic Machine:

- Keeps track of all possible computations
- Stores computations in a two-dimensional tape

NonDeterministic Turing Machines

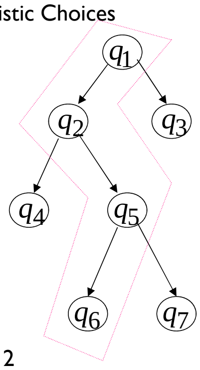
Non-Deterministic Choices



Computation 1

NonDeterministic Turing Machines

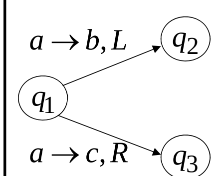
Non-Deterministic Choices



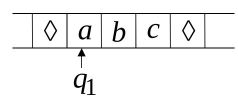
Computation 2

NonDeterministic Turing Machines

NonDeterministic machine:



Time 0

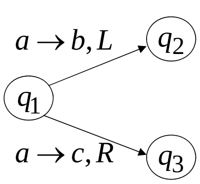


Deterministic machine

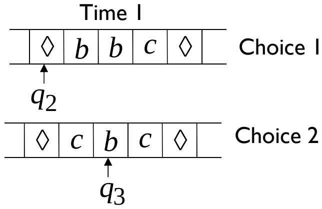
Computation 1

#	#	#	#	#	#
#	a	b	c	#	
#	q_1			#	
#	#	#	#	#	

NonDeterministic machine



Time 1



Deterministic machine

	#	#	#	#	#	#	
#		b	b	c	#		Computation 1
#	q_2				#		
#		c	b	c	#		Computation 2
#			q_3		#		

NonDeterministic Turing Machines

Repeat

Execute a step in each computation:

If there are two or more choices in current computation:

1. Replicate configuration
2. Change the state in the replica

NonDeterministic Turing Machines

Theorem: NonDeterministic Machines have the same power with Deterministic machines

Remark:

- The simulation in the Deterministic machine takes time exponential time compared to the NonDeterministic machine