CS 4410

Automata, Computability, and Formal Language

Dr. Xuejun Liang

Spring 2019

Chapter 10

Other Models of Turing Machines

1. Minor Variations on the Turing Machine Theme
   - Equivalence of Classes of Automata
   - Turing Machine with a Stay-Option
   - Turing Machine with Semi-Infinite Tape
   - The Off-Line Turing Machine
2. Turing Machines with More Complex Storage
   - Multitape Turing Machines
   - Multidimensional Turing Machine
3. Nondeterministic Turing Machines
4. A Universal Turing Machine
5. Linear Bounded Automata

Learning Objectives

At the conclusion of the chapter, the student will be able to:

- Explain the concept of equivalence between classes of automata
- Describe how a Turing machine with a stay-option can be simulated by a standard Turing machine
- Describe how a standard Turing machine can be simulated by a machine with a semi-infinite tape
- Describe how off-line and multidimensional Turing machines can be simulated by standard Turing machines
- Construct two-tape Turing machines to accept simple languages
- Describe the operation of nondeterministic Turing machines and their relationship to deterministic Turing machines
- Describe the components of a universal Turing machine
- Describe the operation of linear bounded automata and their relationship to standard Turing machines

Equivalence of Classes of Automata

Definition 10.1: Two automata are equivalent if they accept the same language. Consider two classes of automata C_1 and C_2. If for every automaton M_1 in C_1. There is an automaton M_2 in C_2 such that

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

we say that C_2 is at least as powerful as C_1. If the converse also holds and for every M_2 in C_2 there is an M_1 in C_1 such that $L(M_1)=L(M_2)$, we say that C_1 and C_2 are equivalent.

Turing machines with stay-option: $\delta: Q\times\Gamma \rightarrow Q\times\Gamma\times\{L, R, S\}$

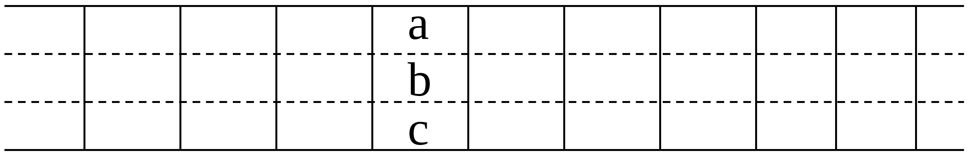
Theorem 10.1: The class of Turing machines with stay-option is equivalent to the class of standard Turing machine

Idea of the equivalence proof:

Use one machine to **simulate** another machine

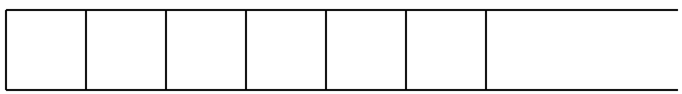
Turing Machines with Semi-infinite Tape

Turing machines with multiple tracks



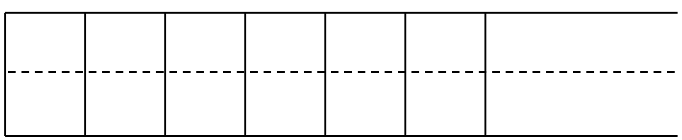
Track 1
Track 2
Track 3

Turing machines with semi-infinite tape



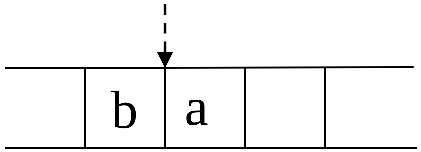
Have a left boundary
No left move at the left boundary

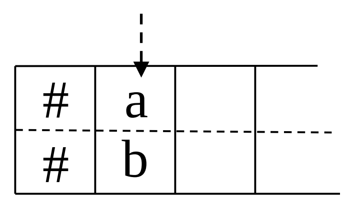
Simulate standard Turing machines



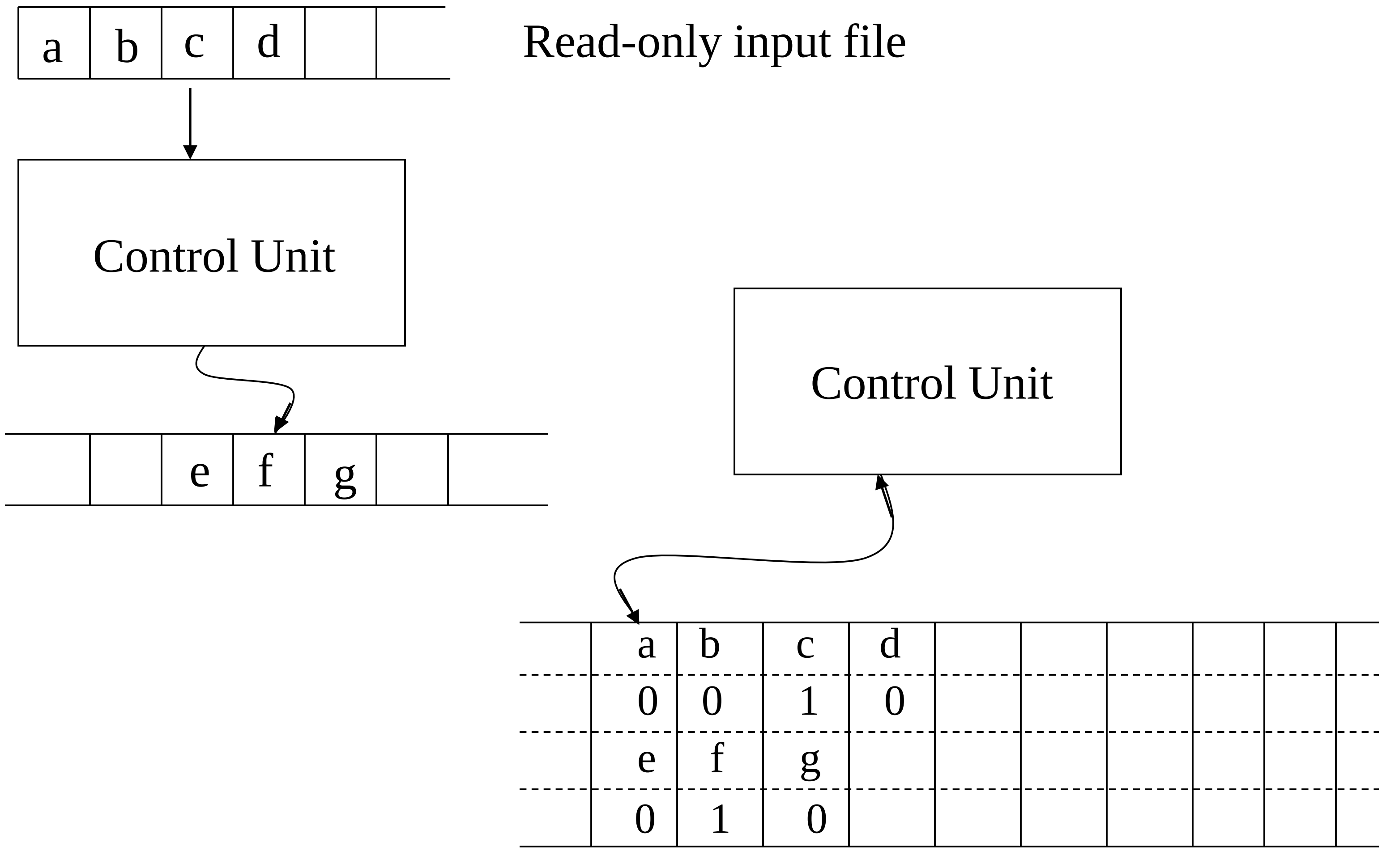
Track 1 for right part of standard tape
Track 2 for left part of standard tape

Reference point





The Off-Line Turing Machine

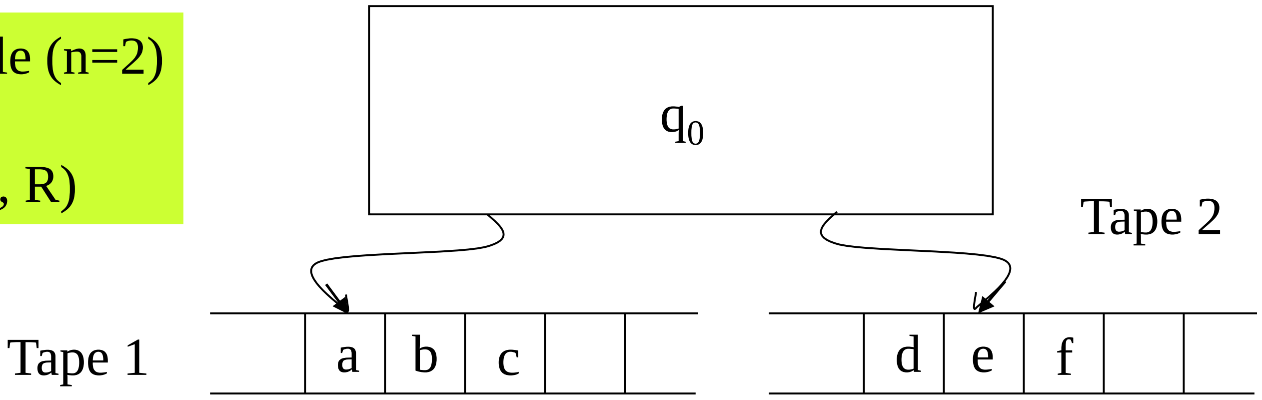


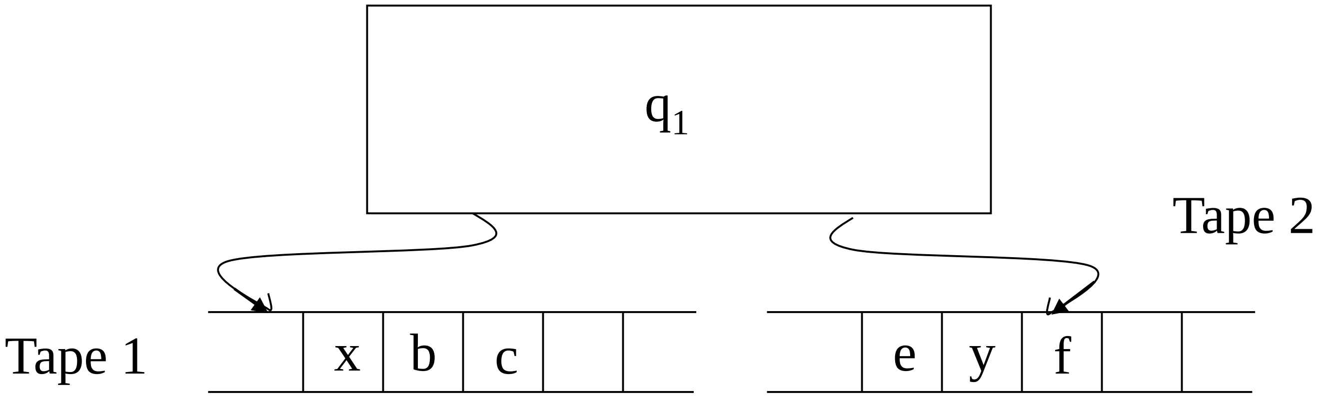
Multitape Turing Machines

Transition function

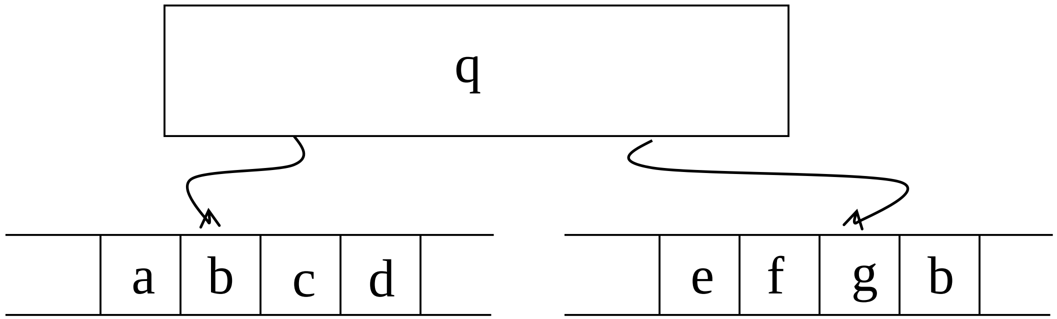
$\delta: Q \times \Gamma^n \rightarrow Q \times \Gamma^n \times \{L, R\}^n$

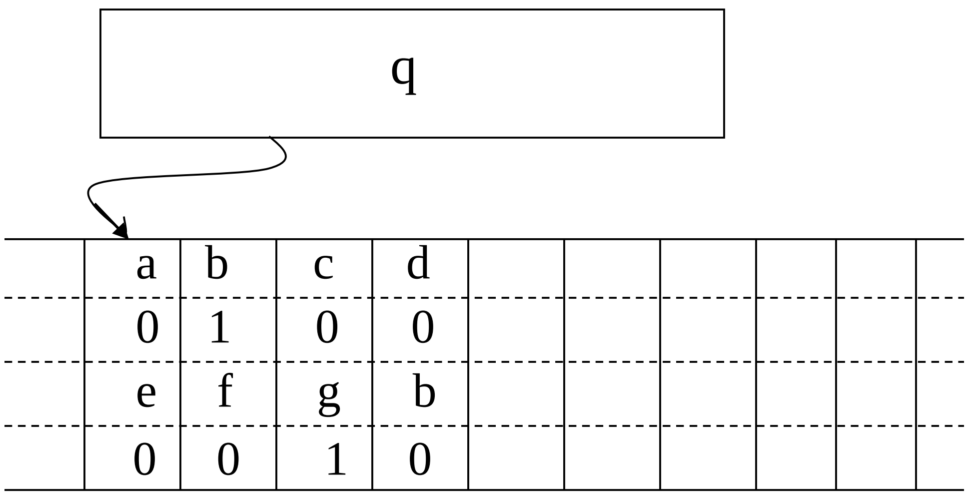
An example (n=2)
$\delta(q_0,a,e) =$
(q_1, x, y, L, R)





Simulate a Two-Tape Machine



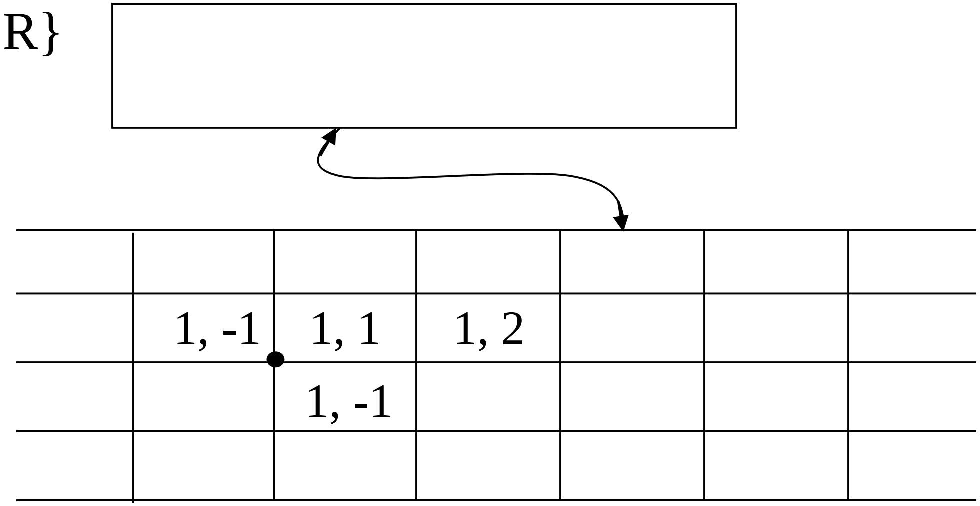


Example 10.1: Two-tape machine that accepts the language $\{a^n b^n : n>0\}$

Multidimensional Turing Machine

Transition function of a two-dimensional Turing machine

$\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, U, R\}$



Simulate two-dimensional Turing machine

	a				b						
	1	#	2	#	1	0	#	-	3	#	

Nondeterministic Turing Machines

Definition 10.2: A nondeterministic Turing machine is an automaton as Given by Definition 9.1, except that δ is now a function

$$\delta : Q \times \Gamma \rightarrow 2^{Q \times \Gamma \times \{L,R\}}$$

Example 10.2: If a Turing machine has transitions specified by

$$\delta(q_0,a)=\{(q_1, b, R), (q_2, c, L)\},$$

it is nondeterministic.

Theorem 10.2: The class of deterministic Turing machines and the class of nondeterministic Turing machine are equivalent

Simulation of a nondeterministic move

	#	#	#	#	#
	#	a	a	a	#
	#	q_0			#
	#	#	#	#	#

#	#	#	#	#	#
#		b	a	a	#
#			q_1		#
#		c	a	a	#
#	q_2				#
#	#	#	#	#	#

A Universal Turing Machine

- A *universal Turing machine* is a reprogrammable Turing machine which, given as input the description of a Turing machine M and a string w, can simulate the computation of M on w
- A universal Turing machine has the structure of a multitape machine, as shown in Figure 10.16

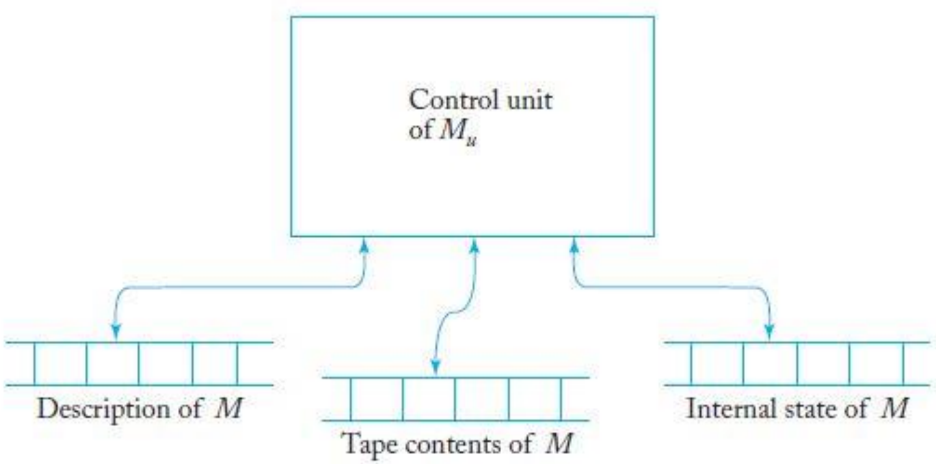


FIGURE 10.16

A Universal Turing Machine (Cont.)

Encoding of a Turing machine

Countable and Uncountable Infinite.

Example: {p/q : p, q are possible integer}

Definition 10.4: Let S be a set of strings on some alphabet Σ. Then an enumeration procedure for S is a Turing machine that can carry out the sequence of steps

$$q_0\Box \overset{*}{\vdash} q_s x_1 \# s_1 \overset{*}{\vdash} q_s x_2 \# s_2$$

with $x_i \in \Gamma^* - \{\#\}$, $s_i \in S$, in such a way that any s in S is produced in a finite number steps. The state q_s is a state signifying membership in S; that is, whenever q_s is entered, the string following # must be in S.

Example 10.3: Let $\Sigma=\{a, b, c\}$. Then $S= \Sigma^+$ is countable.

Theorem 10.3: The set of all Turing machines, though infinite, is countable.

Linear Bounded Automata

- The power of a standard Turing machine can be restricted by limiting the area of the tape that can be used
- A *linear bounded automaton* is a Turing machine that restricts the usable part of the tape to exactly the cells used by the input
- Linear bounded automata are assumed to be nondeterministic and accept languages in the same manner as other Turing machine accepters
- It can be shown that any context-free language can be accepted by a linear bounded automaton
- In addition, linear bounded automata can be designed to accept languages which are not context-free, such as $L = \{ a^n b^n c^n : n \geq 1 \}$
- Finally, linear bounded automata are not as powerful as standard Turing machines

Linear Bounded Automata (Cont.)

Definition 10.5: A linear bounded automaton is a Turing machine M= (Q, Σ, Γ, δ, q_0, □, F), as in Definition 10.2, subject to the restriction that Σ must contain two special symbols [and], such that δ(qi, [) can contain only elements of the form (q_j, [, R), and δ(qi,]) can contain only elements of the form (q_j,], L).

Definition 10.6: A string is accepted by a linear bounded automaton if there is a possible sequence of moves

$$q_0[w] \vdash^* [x_1 q_f x_2]$$

for some $q_f \in F$, $x_1, x_2 \in \Gamma^*$. The language accepted by the lba is the set of all such accepted strings.

Example 10.4: The language $L=\{a^n b^n c^n : n \geq 1\}$ is accepted by some linear bounded automaton.

Example 10.5: Find a linear bounded automaton that accepts the language $L=\{a^{n!} : n \geq 0\}$.