CS 4410

Automata, Computability, and Formal Language

Dr. Xuejun Liang

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
  - Given two classes of automata C_1 and C_2, if for every automaton in C_1 there is an equivalent automaton in C_2, the class C_2 is at least as powerful as C_1
  - If the class C_1 is at least as powerful as C_2, and the converse also holds, then the classes C_1 and C_2 are equivalent
- Equivalence can be established either through a constructive proof or by simulation
  - Use one machine to **simulate** another machine

Turing Machines with a Stay-Option

In a Turing Machine with a Stay-Option, the read-write head has the option to stay in place after rewriting the cell content

Transition function: δ: $Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, S\}$

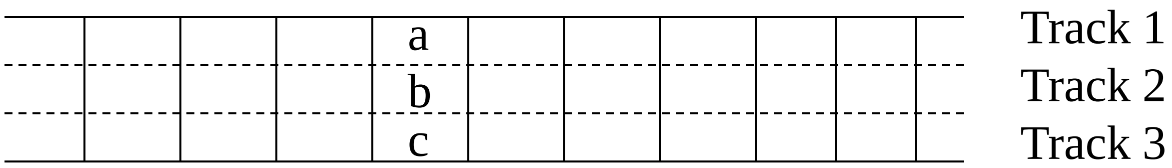
Theorem 10.1: The class of Turing machines with stay-option is equivalent to the class of standard Turing machine

To show equivalence, we argue that any machine with a stay-option can be simulated by a standard Turing machine, since the stay-option can be accomplished by

- A rule that rewrites the symbol and moves right, and
- A rule that leaves the tape unchanged and moves left

Turing Machines with Semi-infinite Tape

Turing machines with multiple tracks



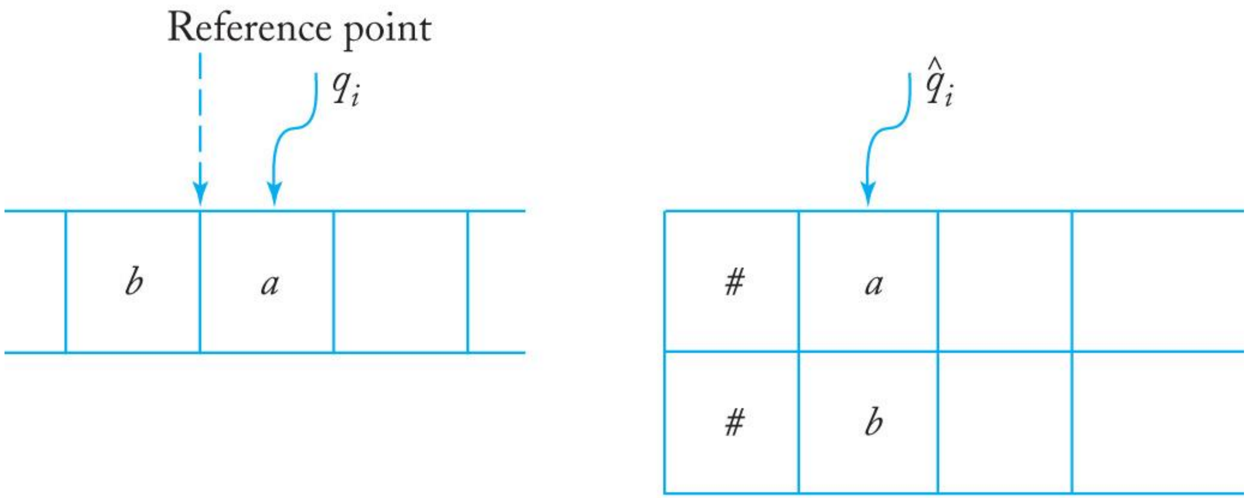
Turing machines with semi-infinite tape

The tape has a left boundary
No left move at the left boundary

A Turing machine with semi-infinite tape is otherwise identical to the standard model, except that no left move is possible when the read-write head is at the tape boundary

Equivalence of Standard Turing Machines and Semi-Infinite Tape Machines

The classes are equivalent because, as shown below, any standard Turing machine can be simulated by a machine with a semi-infinite tape



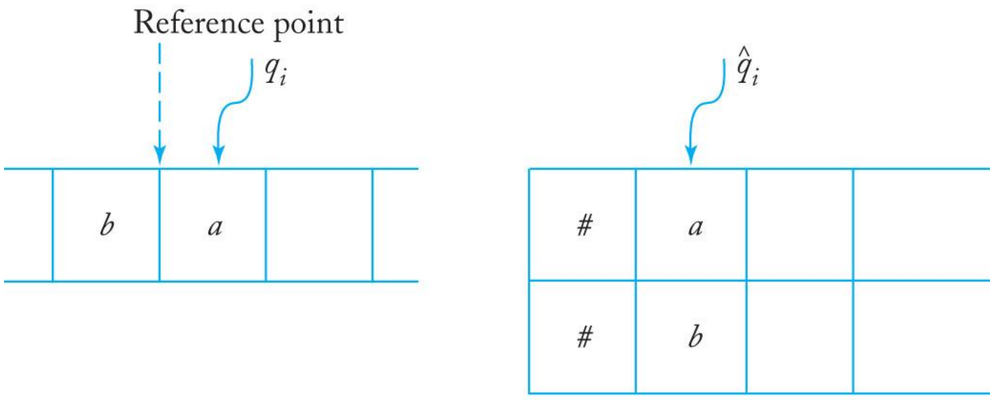
(a) (b)

(a) Machine to be simulated

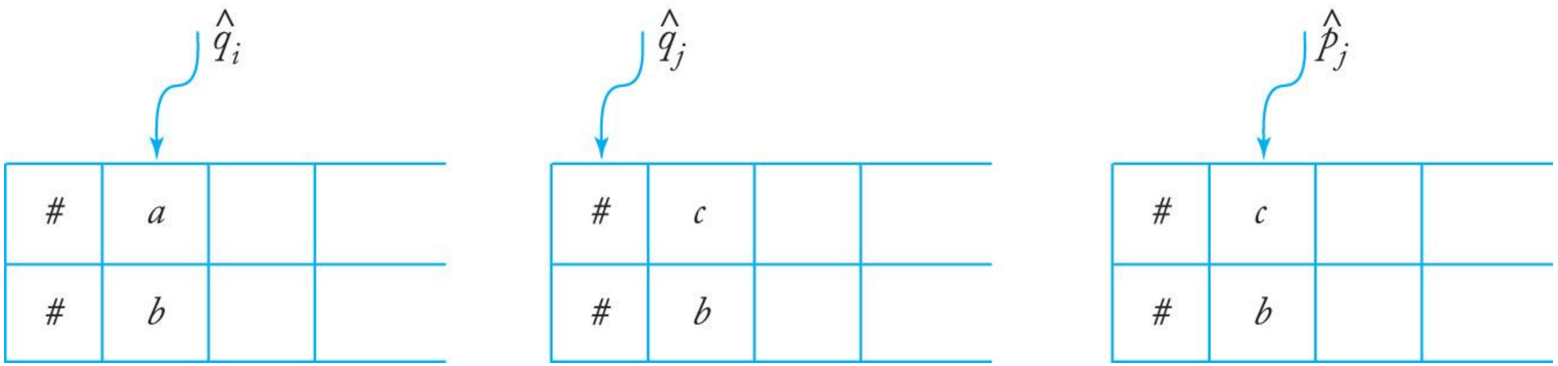
(b) Simulating machine.

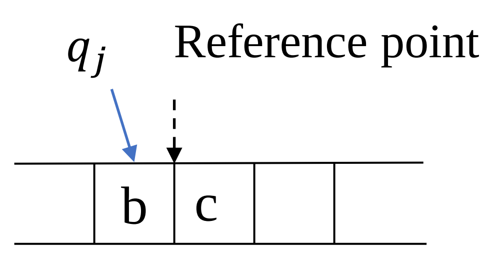
Track 1 for right part of standard tape

Track 2 for left part of standard tape

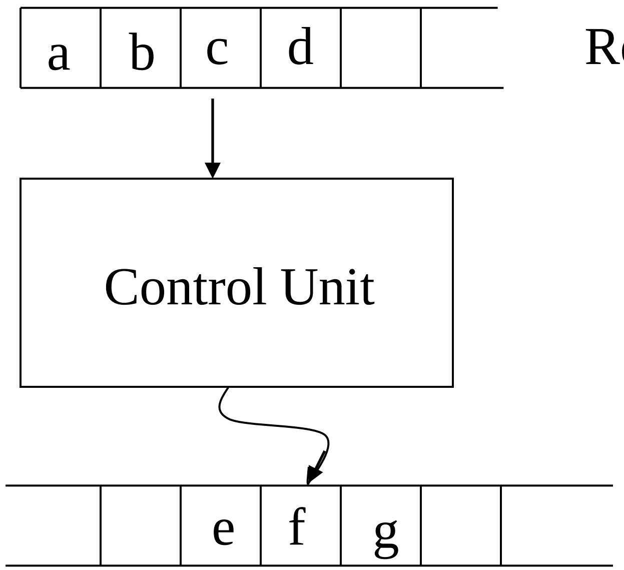

Reference point
q_i
b
a
$\hat{q}_i$
#
a
#
b

(a) (b)


$\hat{q}_i$
#
a
#
b
$\hat{q}_j$
#
c
#
b
$\hat{p}_j$
#
c
#
b


q_j
Reference point
b
c

The Off-Line Turing Machine



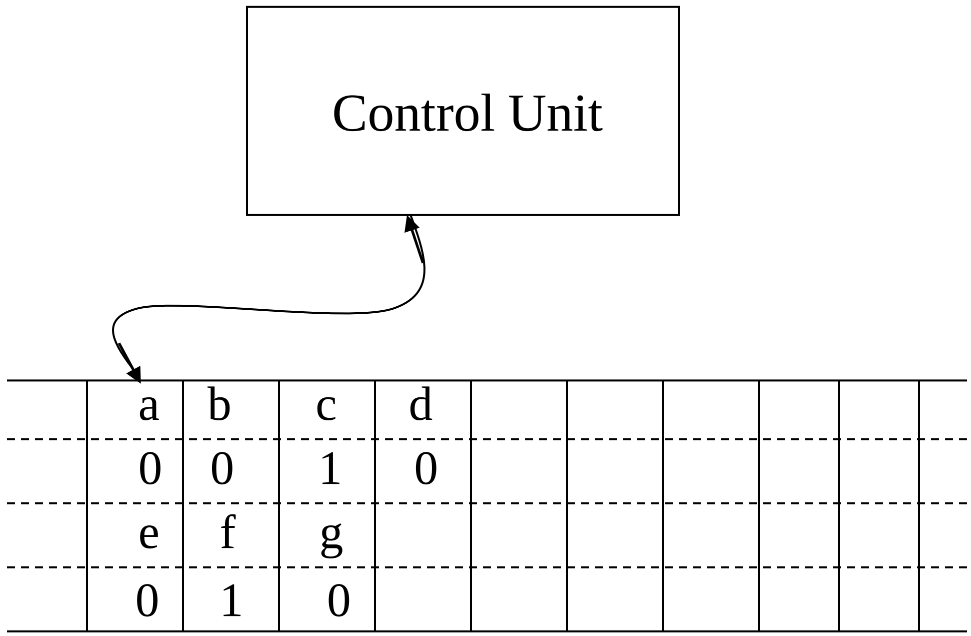
Read-only input file

As shown on the left, an off-line Turing machine has a read-only input file in addition to the read-write tape

Transitions are determined by both the current input symbol and the current tape symbol

Equivalence of Standard Turing Machines and Off-Line Turing Machines

A standard Turing machine with four tracks can simulate the computation of an off-line machine

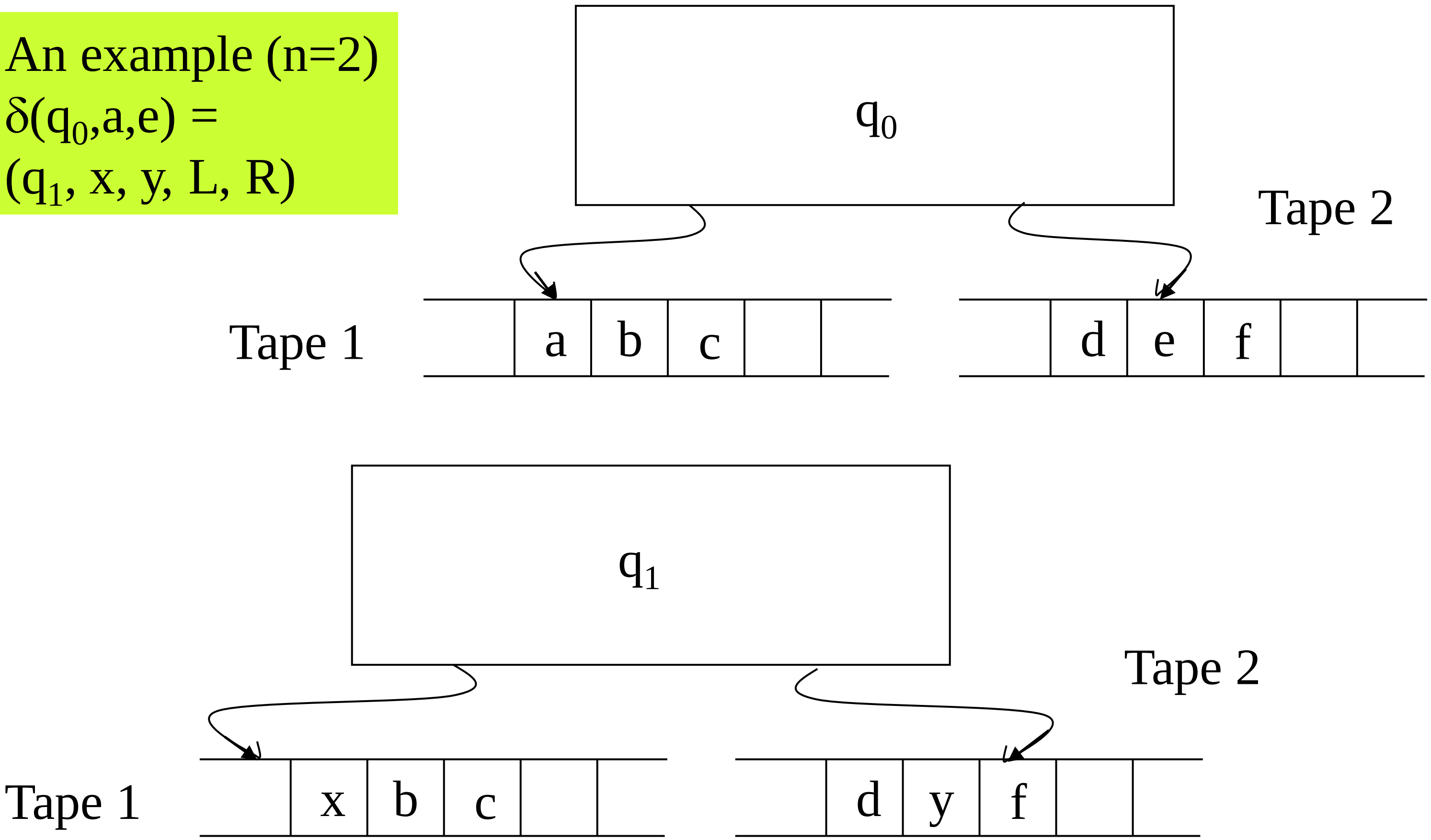


- Two tracks are used to store the input file contents and current position,
- The other two tracks store the contents and current position of the read-write tape

Multitape Turing Machines

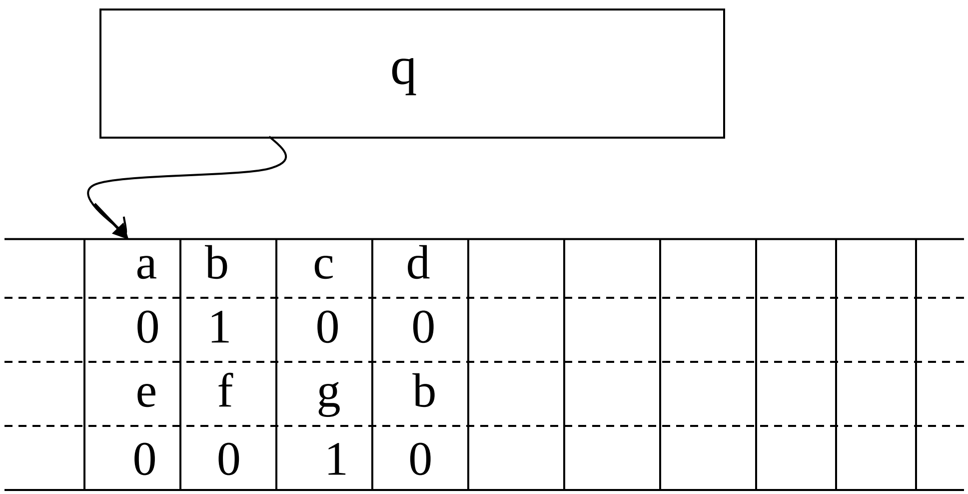
Transition function $\delta: Q \times \Gamma^n \rightarrow Q \times \Gamma^n \times \{L, R\}^n$

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



Equivalence of Standard Turing Machines and Multitape Turing Machines

A standard Turing machine with four tracks can simulate the computation of a two-tape machine



- Two tracks are used to store the contents and current position of tape 1
- The other two tracks store the contents and current position of tape 2

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

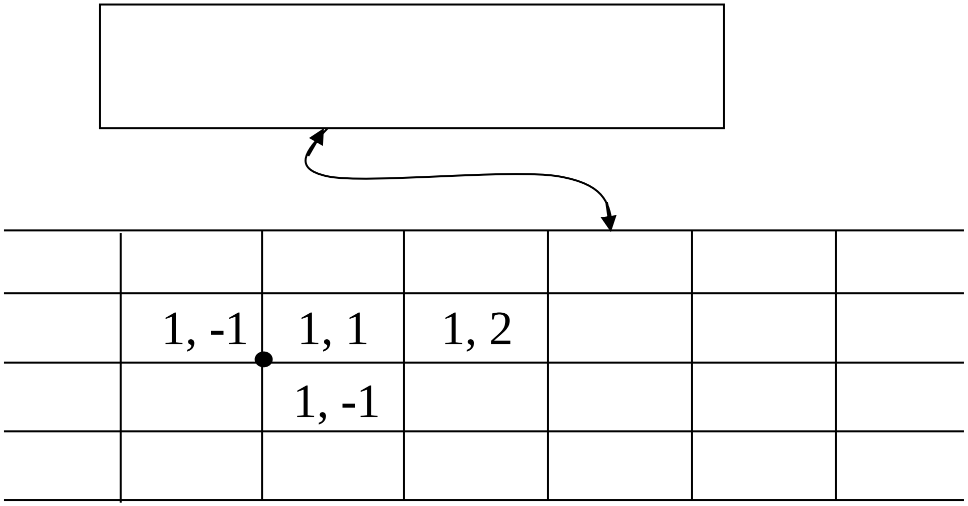
Multidimensional Turing Machine

A multidimensional Turing machine has a tape that can extend infinitely in more than one dimension

In the case of a two-dimensional machine, the transition function must specify movement along both dimensions

Transition function of a two-dimensional Turing machine

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



two-dimensional address scheme

Equivalence of Standard Turing Machines and Multidimensional Turing Machines

A standard Turing machine with two tracks can simulate the computation of a two-dimensional machine

Simulate two-dimensional Turing machine

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

In the simulating machine, one track is used to store the cell contents and the other one to keep the associated address

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

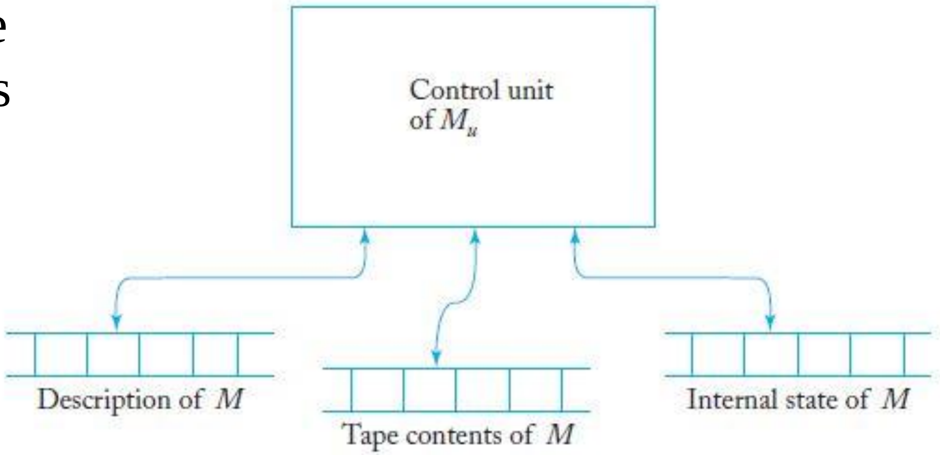


FIGURE 10.16

Encoding of a Turing machine

$$\delta(q_1, a_2) = (q_2, a_3, L)$$

$$\downarrow$$

$$\cdots 1\,0\,1\,1\,0\,1\,1\,0\,1\,1\,1\,0\,1\,0 \cdots$$

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
- Input can be considered as bracketed by two special symbols or markers which can be neither overwritten nor skipped by the read-write head
- Linear bounded automata are assumed to be nondeterministic and accept languages in the same manner as other Turing machine accepters

Languages Accepted by Linear Bounded Automata

- 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
  - It is difficult to come up with a concrete and explicitly defined language to use as such an example