Monash University
Faculty of Information Technology

Lecture 15
Universal Turing Machines

Slides by David Albrecht (2011), modified by Graham Farr (2013).

FIT2014 Theory of Computation

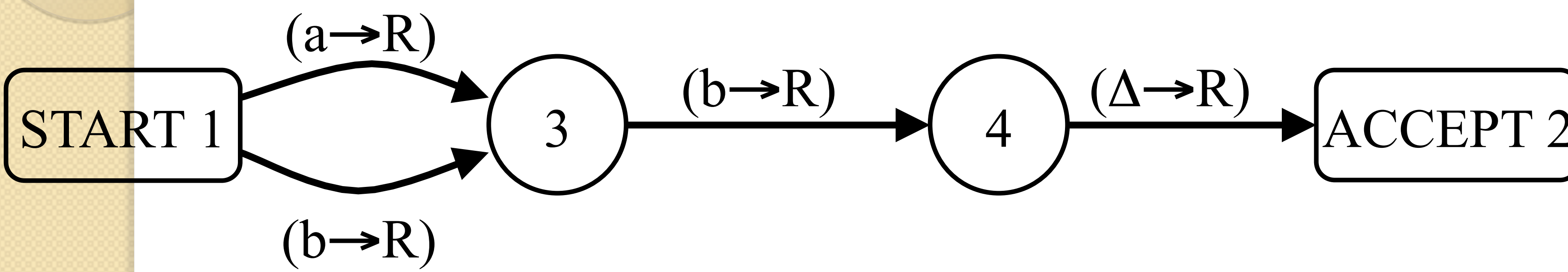
Overview

- Tables for Turing Machines
- Encoding
- Decoding
- Definition of a Universal Turing Machine
- Algorithm for a Universal Turing Machine

Assumptions

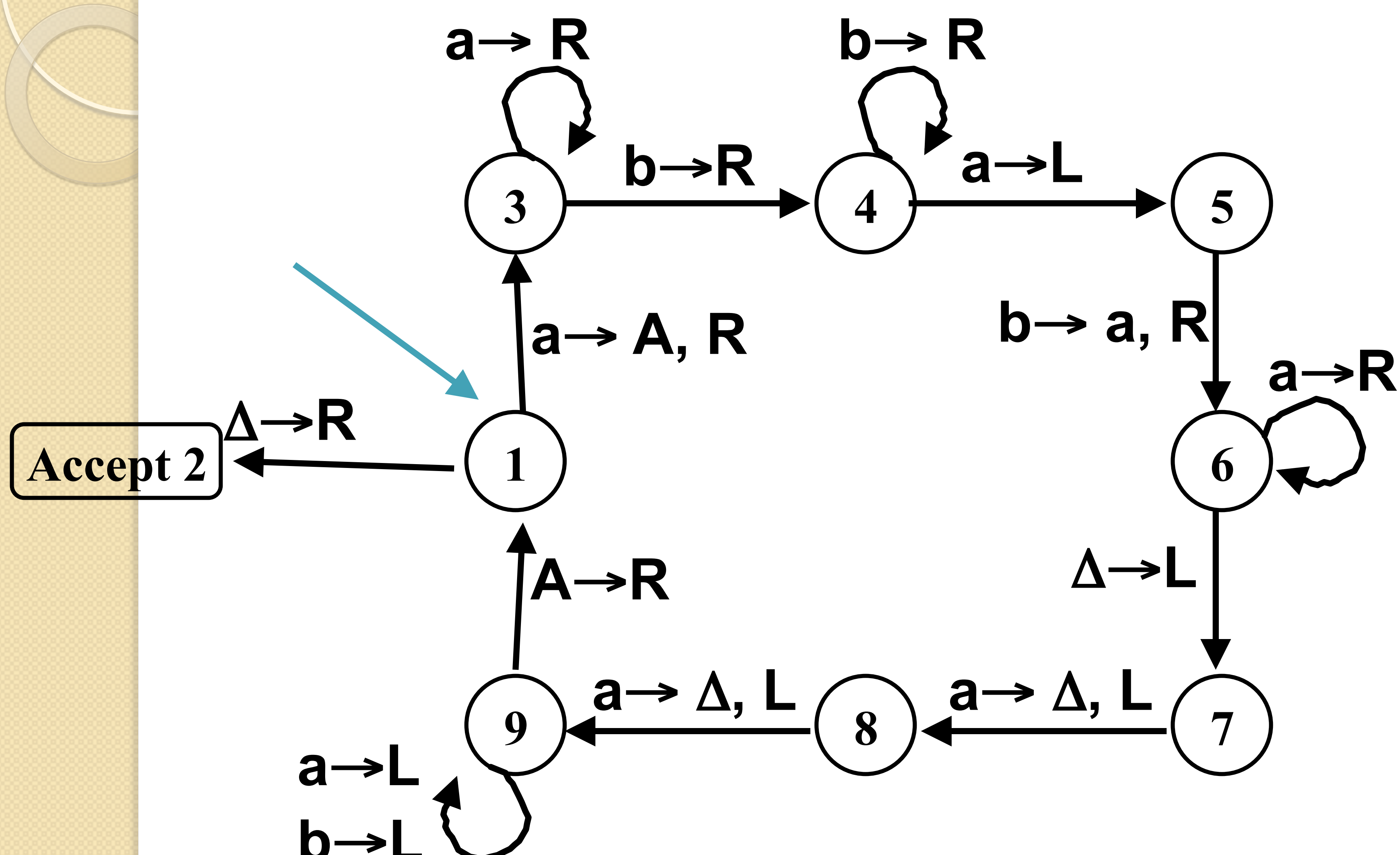
- Input Alphabet
  a b
- Tape Alphabet
  a b #
- Start State
  - Numbered 1.
- Accept State
  - Numbered 2

Example



From	To	Read	Write	Move
1	3	a	a	R
1	3	b	b	R
3	4	b	b	R
4	2	Δ	Δ	R

TM for $a^nb^na^n$



Table

From	To	Read	Write	Move
1	3	a	#	R
3	3	a	a	R
3	4	b	b	R
4	4	b	b	R
4	5	a	a	L
5	6	b	a	R
6	6	a	a	R
6	7	Δ	Δ	L
7	8	a	Δ	L
8	9	a	Δ	L
9	9	a	a	L
9	9	b	b	L
9	1	#	#	R
1	2	Δ	Δ	R

Conditions to Check

- Check whether there is a row with a 1 in the From column.
- Check that there is no row with a 2 in the From column.
- Check there are no two rows with the same numbers in the From and the same letter in the Read column.

Coding

- Integer *n*
  - Code as: $\mathbf{a}^{n}\mathbf{b}$

Letter	Code
a	**aa**
b	**ab**
Δ	**ba**
#	**bb**
L	**a**
R	**b**

Coding the Table

From	To	Read	Write	Move	Code
1	3	a	#	R	abaaabaabbb
3	3	a	a	R	aaabaaabaaaab
3	4	b	b	R	aaabaaaabababb
4	4	b	b	R	aaaabaaaabababb
4	5	a	a	L	aaaabaaaaabaaaaa
5	6	b	a	R	aaaaabaaaaaababaab
6	6	a	a	R	aaaaaabaaaaaabaaaab
6	7	Δ	Δ	L	aaaaaabaaaaaaabbabaa
7	8	a	Δ	L	aaaaaaabaaaaaaaabaabaa
8	9	a	Δ	L	aaaaaaaabaaaaaaaaabaabaa
9	9	a	a	L	aaaaaaaaabaaaaaaaaabaaaaa
9	9	b	b	L	aaaaaaaaabaaaaaaaaabababa
9	1	#	#	R	aaaaaaaaababbbbbb
1	2	Δ	Δ	R	abaabbabab

Encoding of the TM

abaaabaabbbaaabaaabaaaabaaabaaaabababba
aaabaaaabababbaaaabaaaaabaaaaaaaaaabaa
aaaababaabaaaaaabaaaaaabaaaabaaaaaabaa
aaaaabbabaaaaaaaaabaaaaaaaabaabaaaaaaa
aabaaaaaaaaabaabaaaaaaaaaaabaaaaaaaaaba
aaaaaaaaaaaaabaaaaaaaaababaaaaaaaaaab
abbbbbbabaabbabab

Code Word Language (CWL)

- **CWL** is the regular language

$$(a^+ba^+b\,(a \cup b)^5)^*$$

- Words which encode a TM belong to **CWL**.
- Note:

Not all words in CWL encode a TM

Decode

abaaabaaaababaaabababbaaabaaaabababbaaaabaabbabab

From	To	Read	Write	Move
1	3	a	a	R
1	3	b	b	R
3	4	b	b	R
4	2	Δ	Δ	R

Algorithm

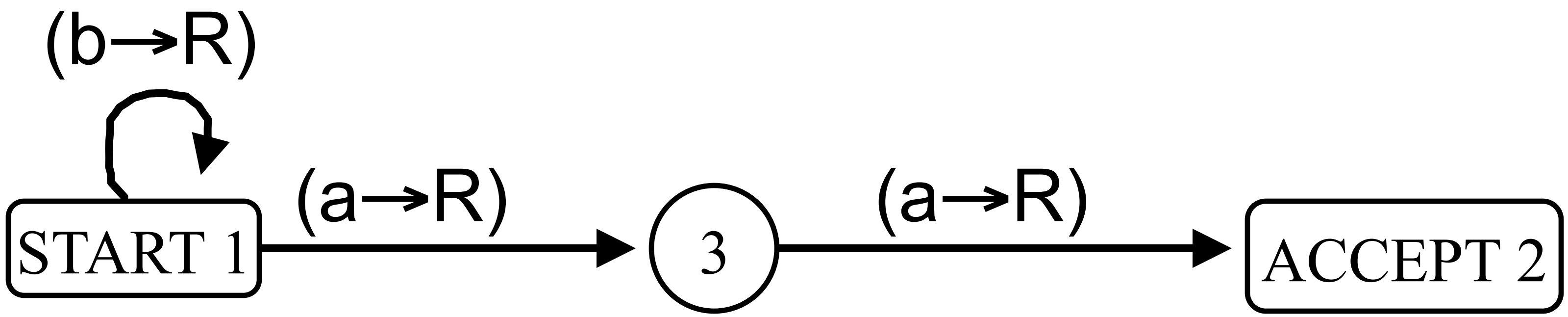
1. Count the initial clump of **a**’s.
2. Count the next clump of **a**’s.
3. Read the next two letters.
4. Read the next two letters.
5. Read the next letter.
6. Repeat until there are no more letters.

Definition

An **Universal Turing Machine (UTM)** is

- A Turing Machine
- Can run any TM on any input data.

Example



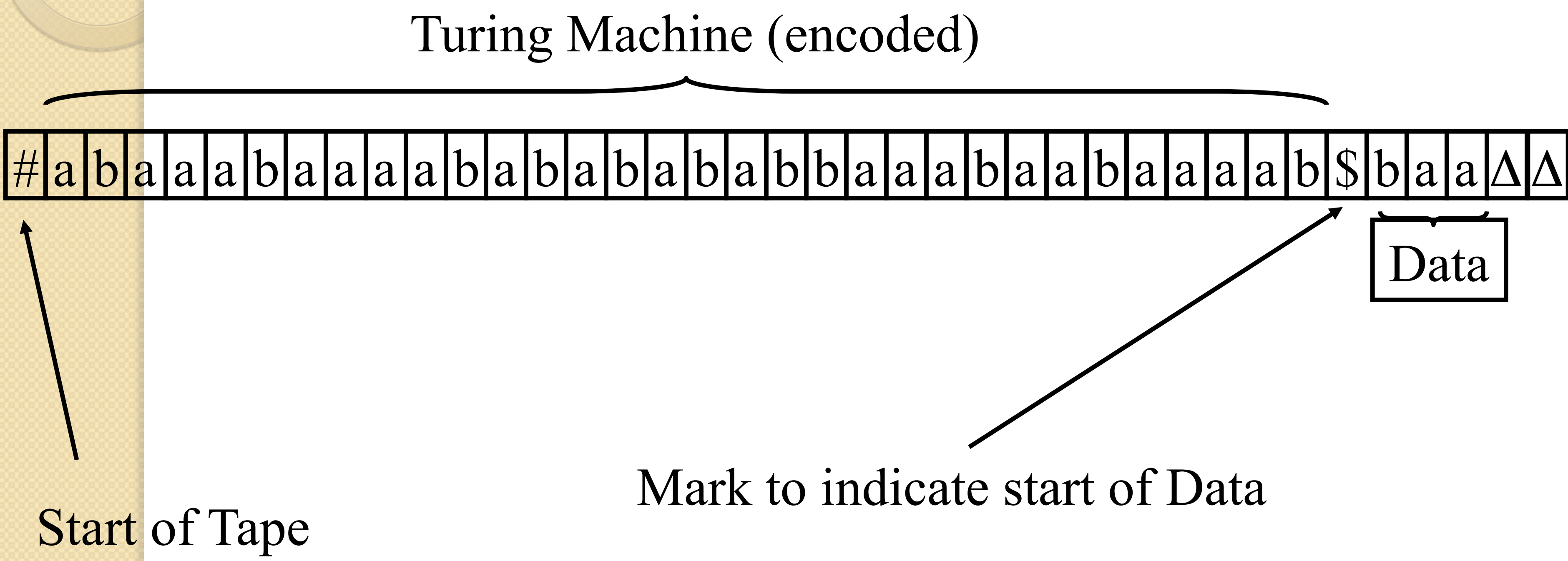
- Turing Machine

  abaaabaaaabababababbaaabaabaaaab

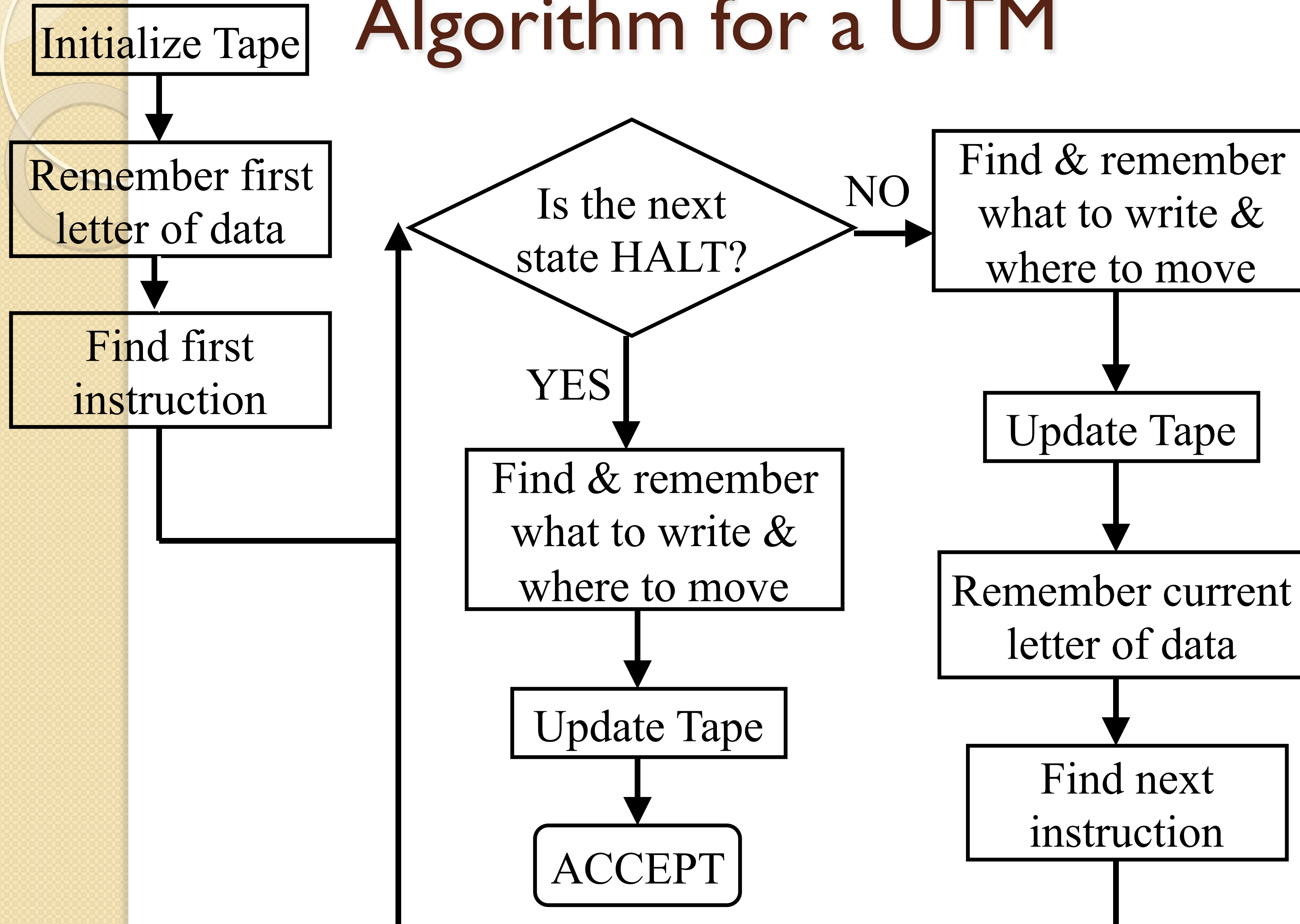
- Data

  baa

Input for UTM



Algorithm for a UTM



Exercise

Suppose:

- $\boldsymbol{U}$ is a UTM,
- $\boldsymbol{T}$ is a TM
- $\boldsymbol{x}$ is an input string for $\boldsymbol{T}$, with length($\boldsymbol{x}$) = $\boldsymbol{n}$.
- When $\boldsymbol{T}$ is run on input $\boldsymbol{x}$, it takes time $\boldsymbol{t}$ and visits at most $\boldsymbol{s}$ tape cells.

Using the algorithm outline of the previous slide, and the encoding scheme for TMs given in this lecture:

Determine an upper bound for the time taken by $\boldsymbol{U}$ to simulate the running of $\boldsymbol{T}$ on input $\boldsymbol{x}$.

Give the bound in terms of $\boldsymbol{t}$, $\boldsymbol{s}$ and $\boldsymbol{n}$.

Importance of UTMs

- theoretical model of one computer simulating another

- Stored-program computer
- von Neumann architecture

- Enables us to ask whether various problems about computers can be solved *algorithmically*.

- For example, are there *algorithmic* solutions for:
  - Given a program, with some input, does it eventually stop?
  - Given two programs, do they always behave the same?
  - Does a given program meet its specifications?

Revision

- Know how to encode a Turing Machine.
- Know how to decode Turing Machine representation.
- Know what a Universal Turing Machine is, and what it does.
- Understand why UTMs exist.