Monash University
Faculty of Information Technology

FIT2014 Theory of Computation

Lecture 19

Universal Turing Machines

slides by Graham Farr
based in part on previous slides by David Albrecht

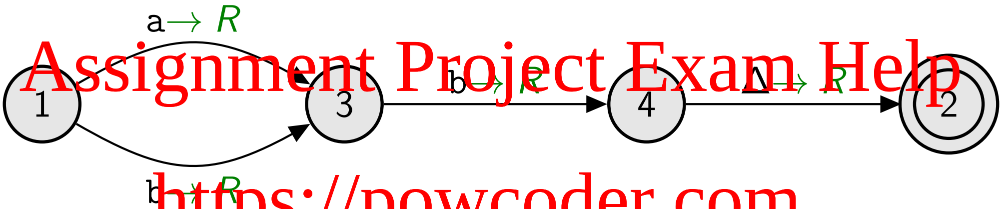
Overview

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

Assumptions

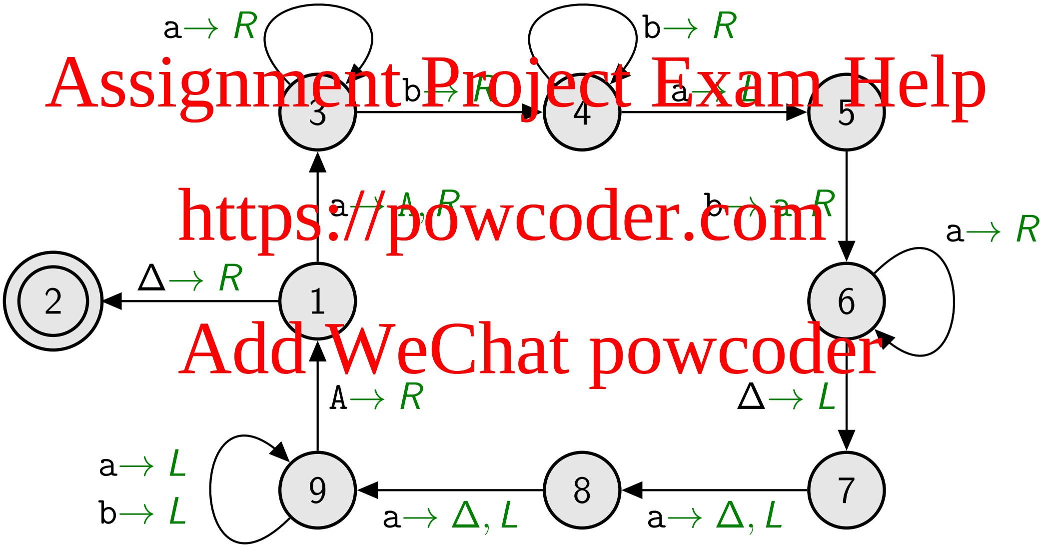
- Input Alphabet: $\{a,b\}$
- Tape Alphabet: $\{a,b,\#\}$
- Start State: numbered 1
- Accept State: numbered 2

Tables for TMs



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 $\{a^n b^n a^n : n \geq 0\}$



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

State number	Code
n	$a^n b$

Letter	Code
a	aa
b	ab
Δ	ba
#	bb

Direction	Code
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	aaabaaaabababb
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

abaaabaabbbaaabaaabaaaabaaabaaaabababbaaabaaaabababbaaaabaaaaabaaaaa
aaaaabaaaaaababaabaaaaaabaaaaaabaaaabaaaaaabaaaaaaabbabaa
aaaaaaabaaaaaaaabaabaaaaaaaaabaaaaaaaaaaababaaaaaaaaaaabaaaaaaaaabaaaaa
aaaaaaaaabaaaaaaaaababab aaaaaaaaaababbbbbbabaabbabab

. . . as one long string, without breaks or spaces.

Code Word Language (CWL)

The **Code-Word Language (CWL)** is the regular language defined by the regular expression

$$(aa^*baa^*b(a \cup b)^*)^*$$

Words which encode a TM belong to CWL.

BUT: Not all words in CWL encode a TM.

Quantifier practice:

$$
\begin{array}{llcl}
\neg\forall w \in \mathrm{CWL} & \exists M & : & w \text{ encodes } M \\
\exists w \in \mathrm{CWL} & \neg\exists M & : & w \text{ encodes } M \\
\exists w \in \mathrm{CWL} & \forall M & : & \neg(w \text{ encodes } M) \\
\exists w \in \mathrm{CWL} & \forall M & : & w \text{ does not encode } M
\end{array}
$$

Decode

abaaabaaaababbaaabababbaaabaaaabababbaaaabaabbabab

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

While there are unread letters

1. Read and count the next clump of a's, then read the `b` after it.
   - Interpret clump of a's as the state number, in unary, that this transition goes from.
2. Read and count the next clump of a's, then read the `b` after it.
   - Interpret clump of a's as the state number, in unary, that this transition goes to.
3. Read the next two letters.
   - Interpret it as the letter to be read for this transition.
4. Read the next two letters.
   - Interpret it as the letter to be written for this transition.
5. Read the next letter.
   - Interpret it as the direction for this transition.

Universal Turing Machine (UTM)

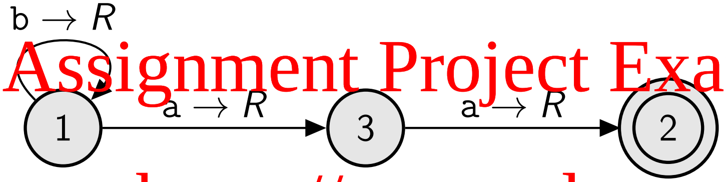
Definition

A **Universal Turing Machine (UTM)** is a Turing Machine that takes, as input,

- an encoding of some Turing Machine M, together with
- a string x, to be used as input to M

and simulates the execution of M on x.

Example

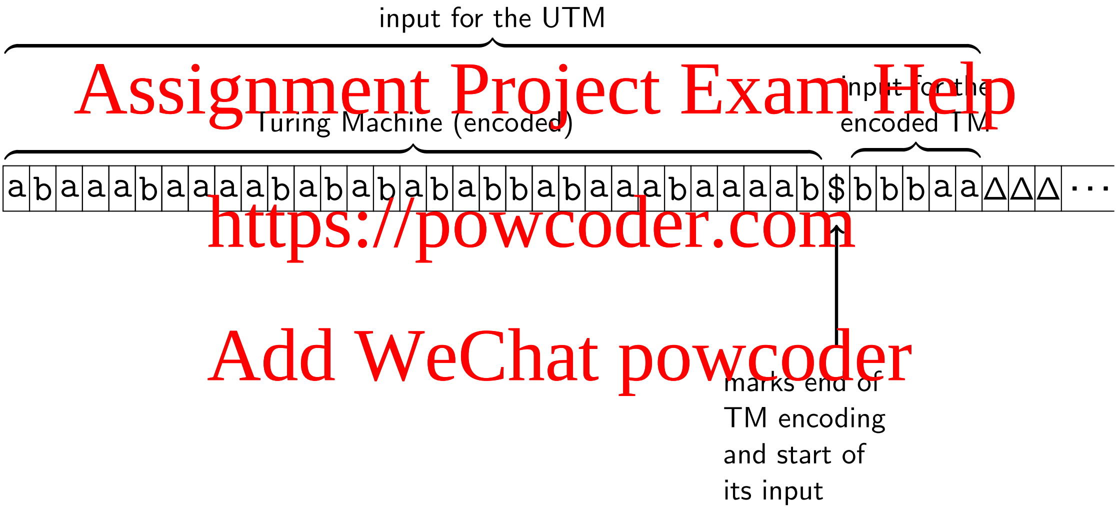


Suppose we want a UTM to simulate the execution of this TM on the string `bbbaa`.

Input to the UTM:

Turing Machine:	`abaaabaaaababababbaaabaabaaaab`
Data:	`bbbaa`

Input for UTM



Algorithm for a UTM

1. Move rightwards to first letter of the encoded TM's input.
   Read it. Mark it, so we can come back to it (e.g., a $\mapsto$ `A`, b $\mapsto$ `B`).
   - Remember it (in choice of state).
2. Move leftwards to first instruction.
3. If the next state in current instruction in encoded TM is the Accept state
   - Find (from current instruction) what to write and direction of next move.
   - Remember it (in choice of state).
   - Move rightwards back to current position in encoded TM's input.
   - Write the required letter, move in the required direction, and Accept.

   else
   - Find (from current instruction) what to write and direction of next move.
   - Remember it (in choice of state).
   - Move rightwards back to current position in encoded TM's input.
   - Write the required letter, move in the required direction.
   - Read current letter in encoded TM's input. Mark it, so we can come back to it.
   - Remember it (in choice of state).
   - Move leftwards to find the next instruction (using remembered letter).

Exercise

Suppose:

- U is a UTM;
- T is a TM
- x is an input string for T, with $|x| = n$.
- When T is run on input x, it takes time t and visits at most 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 U to simulate the running of T on input x.
- Give the bound in terms of t, s and n.

Importance of UTMs

- theoretical model of one computer simulating another
- Stored-program computer
- von Neumann architecture

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.