Automata Formal Languages & Logic

Preet Kanwal

Department of Computer Science & Engineering

Automata Formal Languages & Logic

Unit 4 - Turing Machines

Preet Kanwal

Department of Computer Science & Engineering

$a^n b^n c^n$?
ww ?
Context-Free Languages
$a^n b^n$
ww^R
Regular Languages
$a*$
$a*b*$

Languages accepted by
Turing Machines
$a^n b^n c^n$
ww
Context-Free Languages
$a^n b^n$
ww^R
Regular Languages
$a*$
$a*b*$

A Standard Turing Machine has the following components:

1. Infinite Input tape - unbounded in both directions
2. Control Unit
3. R/W head - changes position in each move

- By convention, input string is preloaded on to the tape.
- Special markers can be used to indicate start and end of the input string.

Turing Machine M, is a 7-tuple :

$$\Sigma \subseteq \Gamma$$

$$\square \in \Gamma$$

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

Note:
Textbook specifies the transition as:
a,a,R

Other accepted notations:
a,a|R
or
a|a|R
or
a|a,R

Read
Write
Move Left
$a \rightarrow b, L$
q_1
q_2

Time 1
..... ◊ ◊ a b a c ◊ ◊ ◊ .....
q_1

Time 2
..... ◊ ◊ a b b c ◊ ◊ ◊ .....
q_2

Move Right
q_1
$a \rightarrow b, R$
q_2
Time 1
..... ◊ ◊ a b a c ◊ ◊
q_1
Time 2
..... ◊ ◊ a b b c ◊ ◊
q_2

Allowed:

No transition

for input symbol c

The machine **halts** in a state if there is no transition to follow.

For example :

or

- Accepting states have no outgoing transitions.
- The machine halts and accepts.

Allowed

Not Allowed

Accept Input string →	If machine halts in an accept state
Reject Input string →	If machine halts in a non-accept state or If machine enters an infinite loop

Unit 4 - Infinite Loop Example

Time 0

Time 1

Time 2

Because of the infinite loop:

- **The accepting state cannot be reached**
- **The machine never halts**
- **The input string is rejected**

Given a Turing Machine M and a string w, when we run w on M, there are three outputs possible:

Given a Turing Machine M and a string w, when we run w on M, there are three outputs possible:

Language recognized by a Turing Machine is called :

Turing Recognizable Language

or

Turing Acceptable Language

or

Recursively enumerable Language

Turing Machine Decider:

Turing Machines that halt on all inputs.

Language recognized by a Turing Machine Decider is called Recursive Language.

Recursive languages are closed under Complement.

Computation :

- It is a series of moves a Turing Machine makes given an input until it halts in some configuration.

Computable Function :

- A function f is computable if we can construct a Turing Machine Decider for that function.

For example: Following functions are computable:

1. f(x) = 2x
2. f(x,y) = x + y
3. $$f(x,y) = \begin{cases} 1 & \text{if } x > y \\ 0 & \text{if } x \leq y \end{cases}$$

Current Configuration

Instantaneous description: $ca\ q_1\ ba$

Initial configuration: $q_0\ w$

Input string

Final Configuration : w q_f
where q_f is the final state

Unit 4 - Examples

Construct Turing Machine for the following examples:

1. $a^n b^n$, n>=1. Trace the input string : aabb
2. $0^n 1^n 2^n$, n>=1
3. Given input : $a^n b^m c^k$ such that k=n-m. Write the difference(in terms of no. of c's) on the tape. If the difference is negative append a -ve sign. For example if k=3 write ccc on the tape. If k=-2, write -cc on the tape. (Language of subtraction). Input and output must be separated by a □.
4. Given input : $0^m 1 0^n$, write 0^{m*n} as output on the tape (Language of Unary multiplication). Input and output must be separated by a □.
5. $0^n 1^{n^2}$, n>=1
6. 0^{2^n}, n>=0
7. ww, w ∈ $\{a,b\}^+$
8. 0^{n^2}, n>=1

Automata Formal Languages and Logic

Unit 4 - Homework

Construct Turing Machine to recognize the following language:

1. $a^n b^m c^n d^m$, $n>=1$
2. Given input : $0^m 1 0^n$, write 0^{m+n} as output on the tape (Language of addition), $m,n>=1$
3. Given a string $w \in \{a,b\}^+$, write w as output on the tape separated by a blank. Final contents on the tape must look like : w□w (Copy operation)
4. Given a string $w \in \{a,b\}^+$, sort the symbols in the input string. For example if the input is babaa, output should be : aaabb. Final contents on the tape must look like: babaa□aaabb
5. Given a string $w \in \{a,b\}^+$, write w^R as the output on the tape (Reversing a string).Final contents on the tape must look like : w□w^R

Example:

$$f(x,y) = \begin{cases} x+y & \text{if } x > y \\ 0 & \text{if } x \le y \end{cases}$$

By combining Turing Machines that perform simple tasks, complex algorithms can be implemented.

The Church-Turing Thesis claims that every effective method of computation is either equivalent to or weaker than a Turing machine.

Alan turing came up with the notion that, what can be computed using Turing machine is known as computable.

All variations of the Turing Machine are equivalent in computing capability.

Algorithmically computable means computable by Turing Machine.

Warning :

Completely different term.

Used in Artificial Intelligence.

It is a test to determine whether a computer or program has same kind of intelligence as that of a human.

Alan turing designed both Turing Machine and Turing Test. They are completely different.

THANK YOU

Preet Kanwal

Department of Computer Science & Engineering

preetkanwal@pes.edu

+91 80 6666 3333 Extn 724

1) $L = \{a^n b^n, n \geq 1\}$

Logic

- Match a 'b' for every 'a'
- Mark the matched symbols

Encode
matched 'a' as 'X'
matched 'b' as 'Y'

Turing Machine

Initial input tape Conf:-

final input tape Conf:-

Trace aabb

$q_0\, aabb$
$\vdash X q_1 abb$
$\vdash X a q_1 bb$
$\vdash X q_2 aYb$
$\vdash q_2 XaYb$
$\vdash X q_0 aYb$
$\vdash XX q_1 Yb$
$\vdash XX Y q_1 b$
$\vdash XX q_2 YY$
$\vdash X q_2 XYY$
$\vdash X X q_0 YY$
$\vdash XXY q_3 Y$
$\vdash XXYY q_3 \square\square$
$\vdash XXYY\square F\square$

2) $L = \{a^n b^n c^n, n \geq 1\}$

<u>Logic</u>

Match a 'b' and a 'c' for every a.

Encode all the marked symbols as:-

a – 'X'

b – 'Y'

c – 'Z'

<u>Input tape</u>

Initial conf:

a	a	b	b	c	c		

final conf:

X	X	Y	Y	Z	Z		

<u>Turing machine</u>

3) Construct a Turing Machine for the language of multiplication

Input tape

Initial conf:

0	0	1	0	0	0	

final conf:

0	0	1	0	0	0		0	0	0	0	0	0

Given input:

$0^m 1 0^n$

produce output:

0^{m*n}

Logic :-

Repeat n (m times)

3) Continued ...

Input tape

Time																
0		0	0	1	0	0	0									→ Initial conf.
1		X	0	1	0	0	0									
2		X	0	1	Y	0	0		0							
3		X	0	1	Y	Y	0		0	0						
4		X	0	1	Y	Y	Y		0	0	0					
5		X	0	1	0	0	0		0	0	0					
6		X	X	1	0	0	0		0	0	0					
7		X	X	1	Y	0	0		0	0	0	0				
8		X	X	1	Y	Y	0		0	0	0	0	0			
9		X	X	1	Y	Y	Y		0	0	0	0	0	0		
10		X	X	1	0	0	0		0	0	0	0	0	0		
11		0	0	1	0	0	0		0	0	0	0	0	0		→ final conf.

3) Continued . . . .

0|0,R
0|0,R
0|0,L
0|x,R
1|1,R
0|Y,R
□|□,R
□|0,L
□|□,L
1|1,L
x|x,R
□|□,L
Y|Y,R
0|0,L
Y|0,L
x|0,L
x|0,L
Y|0,L
1|1,L
□|0,R

4) $L = \{ww, w \in \{a,b\}^+\}$

ww is an even length string.

Two main steps :-

1) Find the mid point :- Break the even length string into 2 equal parts.

for example :- if the input is aabaab

output should be aab□aab

2) Match w□w

4) Continued....

a	a	b	a	a	b			
X	a	b	a	a		b		
X	X	b	a		a	b		
X	X	Y		a	a	b		
a	a	b		a	a	b		
X	a	b		X	a	b		
X	X	b		X	X	b		
X	X	Y		X	X	Y		

Divides even length string into 2 equal halves

Matches w□w

4) Continued..

b|b,R
a|a,R
b|□,R
□|b,L
b|Y,R
a|X,R
□|□,L
a|□,R
□|a,L
□|□,L
a|a,L
b|b,L
X|X,R
Y|Y,R
□|□,L
X|a,L
Y|b,L
□|□,R
a|X,R
b|b,R
a|a,R
□|□,R
Y|Y,R
X|X,R
a|X,L
X|X,L
Y|Y,L
□|□,R
b|Y,R
a|a,R
b|b,R
□|□,R
Y|Y,R
X|X,R
b|Y,L
□|□,L
X|X,R
Y|Y,R
a|a,L
b|b,L

5) $L = \{0^n 1^{n^2}, n \geq 1\}$

<u>Logic :-</u>

↳ We mark the first 0 as Z and check for 'n' copies of '1'.

↳ Second 0 is then marked as Z and 'n' copies of '1' are checked.

↳ We repeat the above process until n times n copies of '1' are checked.

5) Continued....

O	O	O	I	I	I	I	I	I	I	I	I
Z	O	O	I	I	I	I	I	I	I	I	I
W	O	O	Y	I	I	I	I	I	I	I	I
W	X	O	Y	Y	I	I	I	I	I	I	I
W	X	X	Y	Y	Y	I	I	I	I	I	I
Z	O	O	Y	Y	Y	I	I	I	I	I	I
Z	Z	O	Y	Y	Y	I	I	I	I	I	I
W	Z	O	Y	Y	Y	Y	I	I	I	I	I
W	W	O	Y	Y	Y	Y	Y	I	I	I	I
W	W	X	Y	Y	Y	Y	Y	Y	I	I	I
Z	Z	O	Y	Y	Y	Y	Y	Y	I	I	I
Z	Z	Z	Y	Y	Y	Y	Y	Y	I	I	I
W	Z	Z	Y	Y	Y	Y	Y	Y	Y	I	I
W	W	Z	Y	Y	Y	Y	Y	Y	Y	Y	I
W	W	W	Y	Y	Y	Y	Y	Y	Y	Y	Y
Z	Z	Z	Y	Y	Y	Y	Y	Y	Y	Y	Y

5) Continued...

z|z,R
0|z,L
z|z,L
□|□,R
z|w,R
0|x,R
z|z,R
y|y,R
0|0,R
1|y,L
0|0,L
y|y,L
z|z,L
y|y,R
□|□,R
y|y,L
w|w,R
x|x,R
y|y,R
x|□,L
w|z,L
□|□,R

6) Language of Subtraction $= \{a^n b^m c^k,\ k = n-m,\ n, m \geq 1\}$

Input :- $a^n b^m c^k$

output :- $a^n b^m c^k \square c^k$ if k is positive

$a^n b^m c^k \square -c^k$ if k is negative

Logic :-

↳ Match a's & b's

↳ if a's are more it should match with # c's.

- A copy of each c should then be made after the □.

↳ if #b's > #a's, leftover b's must be equal to # c's

- A copy of #c's, preceded with a '-' sign should be made after the blank.

6) Continued

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
a|X,R
b|Y,L
c|Z,R
x|x,R
y|y,R
a|a,R
y|y,L
a|a,L
c|c,R
□|□,R
□|c,L
x|x,R
□|□,L
c|c,L
y|y,L
a|a,L
z|z,L
a|a,R
y|y,R
a/X,R
y|y,R
z|z,R
z|z,R
□|□,R
y|y,R
□|□,R
b|Y,R
b|b,R
z|z,R
c|z,R
c|c,R
□|□,R
□|-,R
-|-,R
c|c,R
□|c,L
□|□,L
-|-,L
c|c,L
z|z,L
b|b,L
y|y,R
b|Y,R
z|z,R
z|z,R
□|□,R

6) Continued.. $L = \{a^n b^m c^k \ , \ k = n-m \ , \ n, m \geq 1\}$

Logic

- Match a's & b's. 3 cases are possible :-

  1) #a's > #b's

  2) #a's = #b's

  3) #b's > #a's

- In case ① and ③ we must match leftover a's and b's respectively with the #c's in the string.
- In case #b's > #a's we must write a '$\sqcup$' sign before copying the c's.

7) $L = \{0^{2^n}, n \geq 0\}$

$L = \{0, 00, 0000, 00000000, \ldots\}$

<u>Logic</u>

Recursively divide the #0's by 2 until only 1 zero is left.

<u>Example</u>

Mark alternate 0's as X

	#0's
0 0 0 0 0 0 0 0	8
0 X 0 X 0 X 0 X	4 (8/2)
0 X X X 0 X X X	2 (4/2)
0 X X X X X X X	1 (2/2) ⇒ accept

7) Continued ...

Acknowledgement - The Notes are prepared by Prof. Preeti Kanwal.

8) $L = \{0^{n^2}, n \geq 1\}$ n is a perfect square

$L = \{0, 0000, 000000000, 0000000000000000, \ldots\}$

$n=1$ $n=2$ $n=3$ $n=4$

<u>Logic</u> :-

- We will have to guess the value of n.
- In order to do that we begin with $n=1$ by placing an X at the beginning separated by a # from the input.
- In case the value of $n>1$ we add one more X and make the check.
- Above process is repeated until the machine halts.

8) Continued...

0|0,L
□|#,L
□|x,R
#|#,R
0|0,R
□|0,R
0|0,L
□|x,L
#|#,L, 0|0,L, x|x,L, z|x,L
0|0,L
□|0,R
z|z,L
x|z,L
x|z,L
0|0,R
#|#,L
□|□,R
#|#,R
□|□, R
y|x,L
w|z,L
z|z,R
M|0,R
z|w,R
x|y,R
x|x,R
z|z,R
w|w,R
y|y,R
#|#,R
M|M,R
0|M,L
#|#,L
M|M,L
x|x,L
z|z,L

0^4

				0	**0**	**0**	**0**		
			#	**0**	**0**	**0**	**0**		
		X	**#**	**0**	**0**	**0**	**0**		
	X	**X**	**#**	**0**	**0**	**0**	**0**		
	Z	**X**	**#**	**0**	**0**	**0**	**0**		
	W	**X**	**#**	**M**	**0**	**0**	**0**		
	W	**Y**	**#**	**M**	**M**	**0**	**0**		
	Z	**X**	**#**	**M**	**M**	**0**	**0**		
	Z	**Z**	**#**	**M**	**M**	**0**	**0**		
	W	**Z**	**#**	**M**	**M**	**M**	**0**		
	W	**W**	**#**	**M**	**M**	**M**	**M**		
	Z	**Z**	**#**	**0**	**0**	**0**	**0**	**ACCEPT**	

0^6

					0	0	0	0	0	0	
				#	0	0	0	0	0	0	
			X	#	0	0	0	0	0	0	
		X	X	#	0	0	0	0	0	0	
		Z	X	#	0	0	0	0	0	0	
		W	X	#	M	0	0	0	0	0	
		W	Y	#	M	M	0	0	0	0	
		Z	X	#	M	M	0	0	0	0	
		Z	Z	#	M	M	0	0	0	0	
		W	Z	#	M	M	M	0	0	0	
		W	W	#	M	M	M	M	0	0	
		Z	Z	#	0	0	0	0	0	0	
		X	X	#	0	0	0	0	0	0	
	X	X	X	#	0	0	0	0	0	0	
	Z	X	X	#	0	0	0	0	0	0	
	W	X	X	#	M	0	0	0	0	0	
	W	Y	X	#	M	M	0	0	0	0	
	W	Y	Y	#	M	M	M	0	0	0	
	Z	X	X	#	M	M	M	0	0	0	
	Z	Z	X	#	M	M	M	0	0	0	
	W	Z	X	#	M	M	M	M	0	0	
	W	W	X	#	M	M	M	M	M	0	
	W	W	Y	#	M	M	M	M	M	M	
	Z	Z	X	#	M	M	M	M	M	M	
	Z	Z	Z	#	M	M	M	M	M	M	
	W	Z	Z	#	M	M	M	M	M	M	No corresponding 0 to match, hence rejected

0^9

					0	0	0	0	0	0	0	0	0
				#	0	0	0	0	0	0	0	0	0
			X	#	0	0	0	0	0	0	0	0	0
		X	X	#	0	0	0	0	0	0	0	0	0
		Z	X	#	0	0	0	0	0	0	0	0	0
		W	X	#	M	0	0	0	0	0	0	0	0
		W	Y	#	M	M	0	0	0	0	0	0	0
		Z	X	#	M	M	0	0	0	0	0	0	0
		Z	Z	#	M	M	0	0	0	0	0	0	0
		W	Z	#	M	M	M	0	0	0	0	0	0
		W	W	#	M	M	M	M	0	0	0	0	0
		Z	Z	#	0	0	0	0	0	0	0	0	0
		X	X	#	0	0	0	0	0	0	0	0	0
	X	X	X	#	0	0	0	0	0	0	0	0	0
	Z	X	X	#	0	0	0	0	0	0	0	0	0
	W	X	X	#	M	0	0	0	0	0	0	0	0
	W	Y	X	#	M	M	0	0	0	0	0	0	0
	W	Y	Y	#	M	M	M	0	0	0	0	0	0
	Z	X	X	#	M	M	M	0	0	0	0	0	0
	Z	Z	X	#	M	M	M	0	0	0	0	0	0
	W	Z	X	#	M	M	M	M	0	0	0	0	0
	W	W	X	#	M	M	M	M	M	0	0	0	0
	W	W	Y	#	M	M	M	M	M	M	0	0	0
	Z	Z	X	#	M	M	M	M	M	M	0	0	0
	Z	Z	Z	#	M	M	M	M	M	M	0	0	0
	W	Z	Z	#	M	M	M	M	M	M	M	0	0
	W	W	Z	#	M	M	M	M	M	M	M	M	0
	W	W	W	#	M	M	M	M	M	M	M	M	M
	Z	Z	Z	#	M	M	M	M	M	M	M	M	M
	Z	Z	Z	#	0	0	0	0	0	0	0	0	0