SASTRA
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 OF THE UGC ACT, 1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

CSE211-Formal Languages and Automata Theory

U4L3_Diagonalization and A Language which is not Recursively Enumerable

Dr. P. Saravanan
School of Computing
SASTRA Deemed University

Agenda

- Recap of previous session
- Recursively enumerable and recursive-def
- A Language which is not Recursively Enumerable-proof
- Diagonalization
- Diagonalization language

Let L be a recursively enumerable language

and M the Turing Machine that accepts it

For string : w

if $w \in L$ then M halts in a final state

if $w \notin L$ then M halts in a non-final state

or loops forever

Let L be a recursive language

and M the Turing Machine that accepts it

For string w:

if $w \in L$ then M halts in a final state

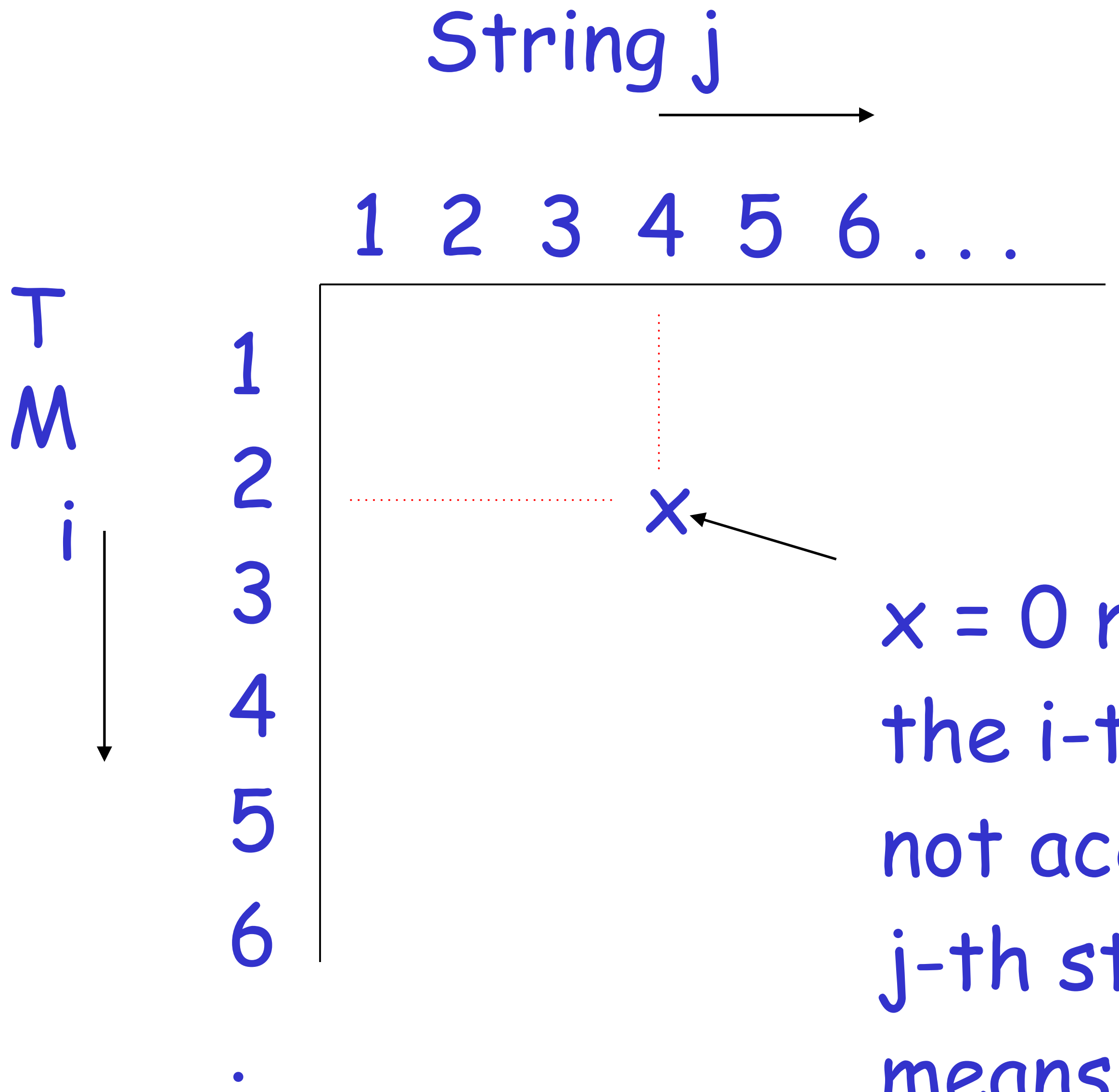
if $w \notin L$ then M halts in a non-final state

A Language which is not Recursively Enumerable

We want to find a language that is not Recursively Enumerable

This language is not accepted by any Turing Machine

Table of Acceptance



x = 0 means the i-th TM does not accept the j-th string; 1 means it does.

Diagonalization Again

- Whenever we have a table like the one on the previous slide, we can *diagonalize* it.
  - That is, construct a sequence D by complementing each bit along the major diagonal.
- Formally, D = $a_1 a_2 \ldots$, where $a_i = 0$ if the (i, i) table entry is 1, and vice-versa.

Diagonalization - (2)

Consider the diagonalization language

L_d = {w | w is the i-th string, and the i-th TM does not accept w}.

Consider alphabet $\{a\}$

Strings: $a, \; aa, \; aaa, \; aaaa, \; \sqcup$

$a^1 \; a^2 \; a^3 \; a^4$ ' '

Consider Turing Machines
that accept languages over alphabet $\{a\}$

They are countable:

$$M_1, \ M_2, \ M_3, \ M_4, \ \ldots$$

Example language accepted by M_i

$$L(M_i) = \{aa, aaaa, aaaaaa\}$$

$$L(M_i) = \{a^2, a^4, a^6\}$$

Alternative representation

	a^1	a^2	a^3	a^4	a^5	a^6	a^7	$\cdots$
$L(M_i)$	0	1	0	1	0	1	0	$\cdots$

	a^1	a^2	a^3	a^4	$\cdots$
$L(M_1)$	0	1	0	1	$\cdots$
$L(M_2)$	1	0	0	1	$\cdots$
$L(M_3)$	0	1	1	1	$\cdots$
$L(M_4)$	0	0	0	1	$\cdots$

Consider the language

$$L = \{a^i : a^i \in L(M_i)\}$$

L consists from the 1's in the diagonal

	a^1	a^2	a^3	a^4	$\cdots$
$L(M_1)$	0	1	0	1	$\cdots$
$L(M_2)$	1	0	0	1	$\cdots$
$L(M_3)$	0	1	$\textcircled{1}$	1	$\cdots$
$L(M_4)$	0	0	0	$\textcircled{1}$	$\cdots$

$$L = \{a^3, a^4, \ldots\}$$

Consider the language $\overline{L}$

$$L = \{a^i : a^i \in L(M_i)\}$$

$$\overline{L} = \{a^i : a^i \notin L(M_i)\}$$

$\overline{L}$ consists of the 0's in the diagonal

$$
\begin{array}{c|ccccc}
 & a^1 & a^2 & a^3 & a^4 & \cdots \\
\hline
L(M_1) & \textcircled{0} & 1 & 0 & 1 & \cdots \\
L(M_2) & 1 & \textcircled{0} & 0 & 1 & \cdots \\
L(M_3) & 0 & 1 & 1 & 1 & \cdots \\
L(M_4) & 0 & 0 & 0 & 1 & \cdots
\end{array}
$$

$$\overline{L} = \{a^1, a^2, \square\}$$

Theorem:

Language $\overline{L}$ is not recursively enumerable

Proof:

Assume for contradiction that

$\overline{L}$ is recursively enumerable

There must exist some machine M_k that accepts $\overline{L}$

$$L(M_k) = \overline{L}$$

	a^1	a^2	a^3	a^4	' '
$L(M_1)$	$\textcircled{0}$	1	0	1	' '
$L(M_2)$	1	$\textcircled{0}$	0	1	' '
$L(M_3)$	0	1	1	1	' '
$L(M_4)$	0	0	0	1	' '

Question: $M_k = M_1$?

$$\begin{array}{c|ccccc}
 & a^1 & a^2 & a^3 & a^4 & \cdots \\
\hline
L(M_1) & \textcircled{0} & 1 & 0 & 1 & \cdots \\
L(M_2) & 1 & \textcircled{0} & 0 & 1 & \cdots \\
L(M_3) & 0 & 1 & 1 & 1 & \cdots \\
L(M_4) & 0 & 0 & 0 & 1 & \cdots
\end{array}$$

$a^1 \in L(M_k)$

Answer: $M_k \neq M_1$

$a^1 \notin L(M_1)$

$$
\begin{array}{c|ccccc}
 & a^1 & a^2 & a^3 & a^4 & \cdots \\
\hline
L(M_1) & \boxed{0} & 1 & 0 & 1 & \cdots \\
L(M_2) & 1 & \boxed{0} & 0 & 1 & \cdots \\
L(M_3) & 0 & 1 & 1 & 1 & \cdots \\
L(M_4) & 0 & 0 & 0 & 1 & \cdots
\end{array}
$$

Question: $M_k = M_2$?

	a^1	a^2	a^3	a^4	$\cdots$
$L(M_1)$	0	1	0	1	$\cdots$
$L(M_2)$	1	0	0	1	$\cdots$
$L(M_3)$	0	1	1	1	$\cdots$
$L(M_4)$	0	0	0	1	$\cdots$

$$a^2 \in L(M_k)$$

Answer: $M_k \neq M_2$

$$a^2 \notin L(M_2)$$

$$\begin{array}{c|ccccc}
 & a^1 & a^2 & a^3 & a^4 & \cdots \\
\hline
L(M_1) & \boxed{0} & 1 & 0 & 1 & \cdots \\
L(M_2) & 1 & \boxed{0} & 0 & 1 & \cdots \\
L(M_3) & 0 & 1 & 1 & 1 & \cdots \\
L(M_4) & 0 & 0 & 0 & 1 & \cdots
\end{array}$$

Question: $M_k = M_3$?

$$\begin{array}{c|ccccc}
 & a^1 & a^2 & a^3 & a^4 & \cdots \\
\hline
L(M_1) & \boxed{0} & 1 & 0 & 1 & \cdots \\
L(M_2) & 1 & \boxed{0} & 0 & 1 & \cdots \\
L(M_3) & 0 & 1 & 1 & 1 & \cdots \\
L(M_4) & 0 & 0 & 0 & 1 & \cdots
\end{array}$$

$$a^3 \notin L(M_k)$$

Answer: $M_k \neq M_3$

$$a^3 \in L(M_3)$$

Similarly: $M_k \neq M_i$ for any i

Because either:

$a^i \in L(M_k)$	or	$a^i \notin L(M_k)$

$a^i \in L(M_k)$

$a^i \notin L(M_i)$

or

$a^i \notin L(M_k)$

$a^i \in L(M_i)$

Therefore, the machine M_k cannot exist

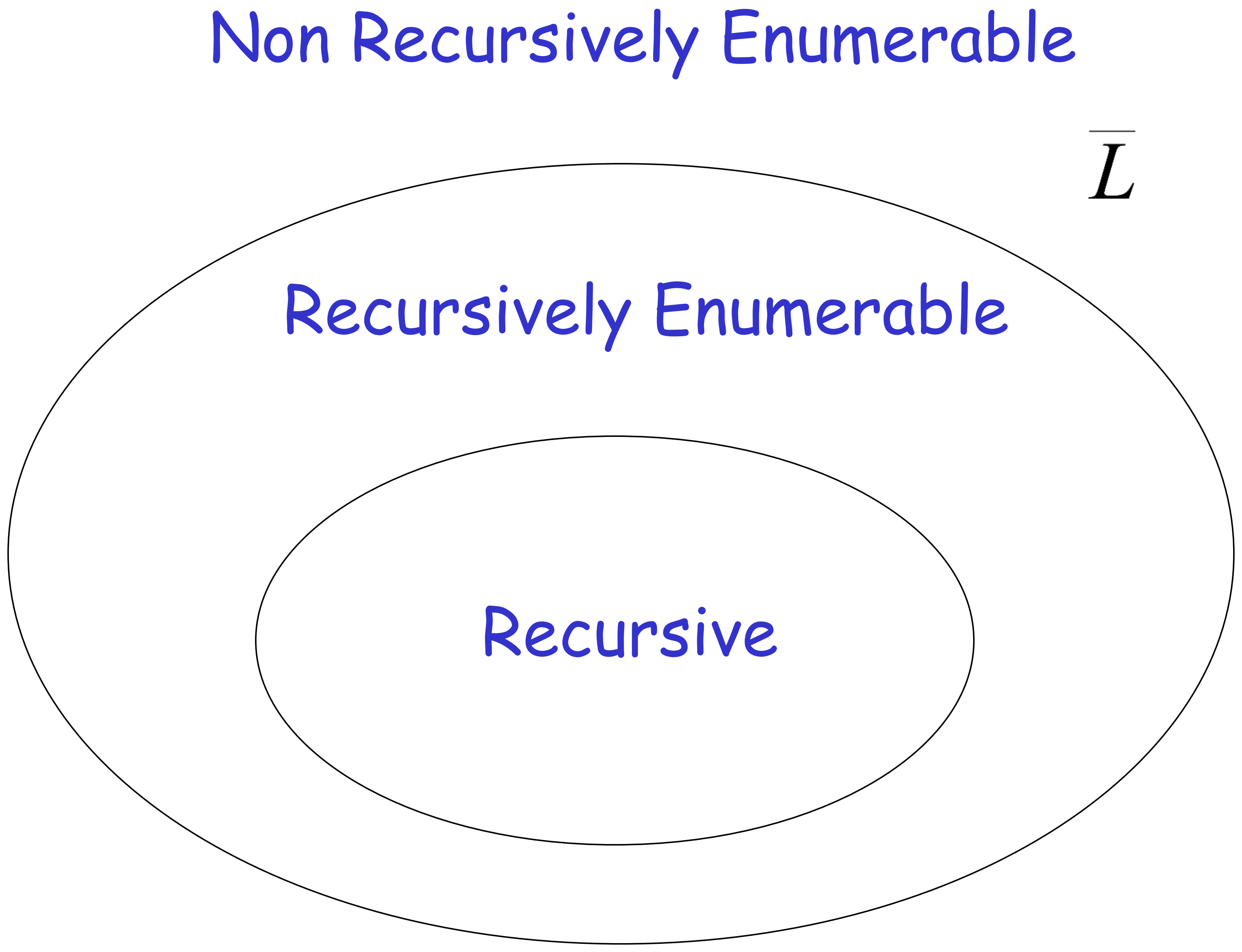
Therefore, the language $\overline{L}$ is not recursively enumerable

End of Proof

Observation:

There is no algorithm that describes $\overline{L}$

(otherwise $\overline{L}$ would be accepted by some Turing Machine)


Non Recursively Enumerable
$\overline{L}$
Recursively Enumerable
Recursive

References

John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman, *Introduction to Automata Theory*, Languages, and Computation, Pearson, 3rd Edition, 2011.

Peter Linz, *An* Introduction to Formal Languages and *Automata*, Jones and Bartle Learning International, United Kingdom, 6th Edition, 2016.

Next Class: Unit IV

Universal Language

Thank you.