Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

TUTORIAL SESSION 17:

Recursive and Recursively Enumerable Language

Concept Building

Recursive Languages

1. **Definition**:
   - A language is called **recursive** (or **decidable**) if there exists a Turing machine that will halt and accept for every string in the language, and halt and reject for every string not in the language.
2. **Characteristics**:
   - The Turing machine must always halt, regardless of whether the input string is in the language or not.
   - There exists an algorithm that can decide membership in the language.
3. **Examples**:
   - The set of all valid arithmetic expressions.
   - The set of all syntactically correct programs in a given programming language.

Recursively Enumerable Languages

1. **Definition**:
   - A language is called **recursively enumerable** (RE) if there exists a Turing machine that will halt and accept for every string in the language. However, for strings not in the language, the Turing machine may either halt and reject or run forever without halting.
2. **Characteristics**:
   - If a string is in the language, the Turing machine will eventually halt and accept it.
   - If a string is not in the language, the Turing machine may never halt.
   - The language can be "enumerated" by a Turing machine: there is a Turing machine that generates all strings in the language in some order, possibly with repetition.
3. **Examples**:
   - The set of all valid proofs in a formal system (like Peano arithmetic).
   - The set of all programs that halt for a given input (Halting problem instances).

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Pre-Tutorial (To be completed by student before attending tutorial session)

1. **Write the difference between recursive and recursive enumerable language**

Solution:

- **Recursive**: The Turing machine always halts, and you can always determine if a string is part of the language.
- **Recursively Enumerable**: The Turing machine may halt for strings in the language but might run forever for strings outside the language.

2. **State the Decidability and Undecidability**

Solution:

Decidability and Undecidability:

- **Decidable**: A problem is decidable if a computer can always solve it in a finite time with a correct answer.
- **Undecidable**: A problem is undecidable if no algorithm can always solve it in a finite time, or it may run forever.

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

3. List out the properties of Enumerable Languages

Properties of Enumerable Languages:

1. **Recognizable**: A Turing machine accepts strings in the language but may not halt for others.
2. **Semi-decidable**: Can decide membership for strings in the language, but not outside.
3. **Closure**:
    - Union, intersection, and concatenation of enumerable languages are enumerable.
4. **Not Closed Under Complement**: The complement may not be enumerable.
5. **Subset of Recursive**: All recursive languages are enumerable, but not all enumerable languages are recursive.

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

IN-TUTORIAL (To be carried out in presence of faculty in classroom)

1. Write the characteristics of recursive and recursive enumerable language.

Solution:

Characteristics of Recursive Languages:

1. The Turing machine **always halts** for any input.
2. It **decides** whether a string belongs to the language.
3. Every recursive language is also **recursively enumerable**.
4. **Decidable** in finite time.

Characteristics of Recursively Enumerable Languages:

1. The Turing machine **may not halt** for strings not in the language.
2. It **recognizes** strings in the language.
3. The machine **accepts** strings in the language but might run indefinitely for others.
4. **Not closed** under complement.

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

2. Give an example of a recursive language and a recursively enumerable language?

Solution:

Example of a Recursive Language:

- **Even number of 1's**: The language $L = \{w \mid w \text{ contains an even number of 1's}\}$.

  A Turing machine can check the number of 1's and always halt with a decision (even or odd).

Example of a Recursively Enumerable Language:

- **Halting Problem**: The language $L = \{\langle M, w\rangle \mid M \text{ halts on input } w\}$.

  A Turing machine can accept the pair $\langle M, w\rangle$ if M halts on w, but it may not halt if M doesn't halt.

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

3. Briefly discuss about Properties of recursive and recursively enumerable languages.

Properties of Recursive Languages:

1. **Decidable**: Always halts and gives the correct answer.
2. **Closed**: Closed under union, intersection, complement.
3. **Subset**: All recursive languages are recursively enumerable.

Properties of Recursively Enumerable Languages:

1. **Semi-decidable**: May halt for strings in the language, but not for others.
2. **Closed**: Closed under union, intersection, but not complement.
3. **Subset**: Recursive languages are recursively enumerable.

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Post-Tutorial (To be carried out by student after attending tutorial session)

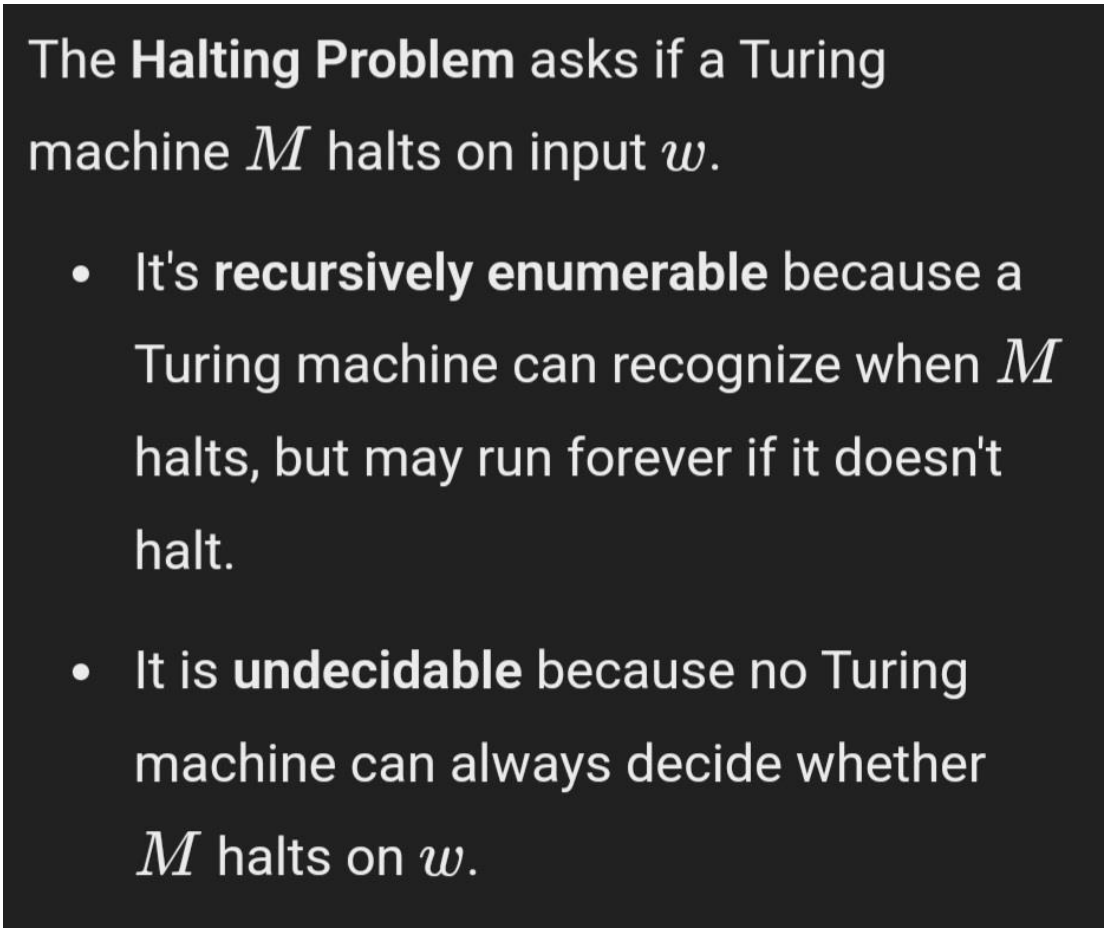
1. .How are recursive and recursively enumerable languages related?

Solution:

1. **Subset**: All **recursive** languages are **recursively enumerable**.
2. **Decidability**: Recursive languages are decidable, while recursively enumerable languages may not be.
3. **Complement**: The complement of a recursive language is recursive, but not necessarily for recursively enumerable languages.

2. explain the Halting problem in the context of recursively enumerable languages?

Solution:



Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

3. Why is a Turing machine called Linear Bounded Automata? Discuss the advantages of Linear Bounded Automata.

Solution:

A **Turing machine** is called a **Linear Bounded Automaton (LBA)** when its tape usage is restricted to a linear function of the input size.

Advantages of LBA:

1. **Space Efficiency**: Uses less memory compared to general Turing machines.
2. **Decidability**: Problems solvable by LBAs are decidable.
3. **Time Complexity**: Can often solve problems in polynomial time.

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Viva – Questions

1. Give an example of a recursively enumerable language that is not recursive.

Solution:

The Halting Problem: $L = \{\langle M, w\rangle \mid M \text{ halts on } w\}$.

2. How are recursive languages related to the complexity class P?

Solution:

Recursive languages are decidable, and the complexity class P consists of problems solvable in polynomial time.

(For Evaluator's use only)

Comment of the Evaluator (if Any)	Evaluator's Observation Marks Secured: out of **50** Full Name of the Evaluator: Signature of the Evaluator Date of Evaluation: