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>

CO - 4

TUTORIAL SESSION 20:

Variants of TM: Multi-Tape and NDTM

Concept Building

Multi-Tape Turing Machine (MTM)

A Multi-Tape Turing Machine (MTM) is an extension of the classical single-tape Turing machine. It consists of multiple tapes, each with its own head for reading and writing. Here are some key features:

Multiple Tapes: Instead of one tape, an MTM has several tapes, typically kkk tapes where kkk is a positive integer.

Independent Heads: Each tape has its own head that can move independently of the others, either left, right, or stay in place.

Transition Function: The transition function is more complex and takes into account the symbols under all kkk heads and dictates the new states, symbols to write, and movements of all heads.

Enhanced Computational Power: While MTMs do not provide more computational power in terms of solvable problems (they are equivalent to single-tape Turing machines in this respect), they can be more efficient, potentially solving problems faster by simulating certain computations more effectively.

Non-Deterministic Turing Machine (NDTM)

A Non-Deterministic Turing Machine (NDTM) is a theoretical model of computation that extends the classical deterministic Turing machine with the ability to explore multiple computation paths simultaneously. Here are the key features:

5. **Non-Deterministic Transitions:** At any given state and tape configuration, an NDTM can have multiple possible transitions.
6. **Parallel Computation Paths:** The machine can be thought of as exploring all possible transitions in parallel. If any computation path leads to an accepting state, the NDTM accepts the input.
7. **Acceptance Criteria:** An input is accepted if there exists at least one sequence of transitions leading to an accepting state.
8. **Theoretical Power:** NDTMs are used primarily in theoretical computer science to study problems in the complexity class NP (nondeterministic polynomial time). They are not realizable as physical machines but provide a useful abstraction for understanding computational complexity.

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

1. Differentiate deterministic and non-deterministic Turing Machine.

Solution:.

Deterministic Turing Machine (DTM):

- Has a unique transition for each state and input.
- Only one computation path is followed.
- Simpler, with predictable behavior.

Non-Deterministic Turing Machine (NDTM):

- Can have multiple transitions for a state and input.
- Multiple computation paths can be explored simultaneously.
- Useful in theoretical models for complex problems.

2. Can a non-deterministic Turing machine be used to solve Halting Problem? Explain.

Solution:

No, an NDTM cannot solve the Halting Problem.

Reasons:

- The Halting Problem is undecidable.
- Non-determinism doesn't solve undecidability.
- No machine can decide halting for all inputs.

3. Write the transition function for a k-tape Turing machine.

Solution:

$\delta(q, s_1, s_2, \ldots, s_k) = (q', r_1, r_2, \ldots, r_k, d_1, d_2, \ldots, d_k)$

where:

- **q**: current state.
- **$s_1, s_2, \ldots, s_k$**: symbols read on **k** tapes.
- **q'**: new state.
- **$r_1, r_2, \ldots, r_k$**: symbols to write.
- **$d_1, d_2, \ldots, d_k$**: move directions (L, R, S).

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

1. What is the primary advantage of a multi-tape Turing machine over a single-tape Turing machine?

Solution:

- **Efficiency**: Simultaneous read/write on multiple tapes.
- **Reduced scanning**: Avoids repetitive tape movements.
- **Faster computation**: Complex tasks are completed more quickly.

2. Explain how multiplication of two integers can be done using multi-tape Turing Machine.

Solution:

1. **Tape 1**: First integer.
2. **Tape 2**: Second integer.
3. **Tape 3**: Result (initialized to 0).
4. **Process**: Multiply bit by bit, shifting and adding partial sums efficiently using multiple tapes.

3. What is an example of a problem that is easier to solve with an NDTM?

Solution:

An example is the **context-free language (CFL) recognition** problem.

- **NDTM**: Can guess the correct derivation path and check efficiently.
- **DTM**: Must exhaustively explore all possibilities, which is slower.

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

1. How is the time complexity of an NDTM measured?

Solution:

Time complexity of an NDTM is measured by the longest path it explores, assuming all paths are checked in parallel.

2. Can the behavior of an NDTM be simulated by a DTM? Explain with an example.

Solution:

Yes, an NDTM can be simulated by a DTM.

Example: For **3-SAT**:

- **NDTM**: Simultaneously tests all variable assignments.
- **DTM**: Checks each assignment one by one, leading to exponential time.

3. What is the relationship between the time complexity of a non-deterministic Turing machine and a deterministic Turing machine?

Solution:

- **NDTM**: Can solve certain problems in **polynomial time**.
- **DTM**: May take **exponential time** to solve the same problems.
- **Relationship**: **Time complexity of NDTM** is generally **better** (polynomial vs. exponential) than **DTM** for NP problems.

Viva – Questions

1. Explain the concept of a multi-tape Turing machine and its advantages over a single-tape Turing machine.

Solution:

A **multi-tape Turing machine** has multiple tapes for simultaneous read/write.

Advantages:

- **Faster** computation.
- **Increased efficiency** for complex tasks.

2. Can a Turing machine with a finite number of states recognize all recursively enumerable languages? Explain.

Solution:

No, a finite-state Turing machine can't recognize all recursively enumerable languages due to limited memory.

(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: