KALYANI GOVERNMENT ENGINEERING COLLEGE

DEPARTMENT OF INFORMATION TECHNOLOGY

Assignment on

Various types of turning machines

Submitted by

Name : Abhoy Biswas
Roll.No.-10200321037

Registration .No-211020100310009

Dept: Information Technology
Subject Code : PCC-CS-403
Subject: Formal language And Autamata
Sem: 4th sem
Year : 2nd Year

Contents

Topics

Various types of Turning Machine

Multi-tape turing machine

- Multi-head turing machines
- Multi-track turing machines
- Semi-infinite turing machines
- Universal Turing Machine
- Alternating Turing machine
- Non-Deterministic Turing Machine
- Unambiguous Turing Machine
- Quantum Turing Machine
- Read-Only Turing Machine
- Probabilistic Turing Machine

Various types of turning machines

.

Abhoy Biwas,10200321037

Kalyani Government Engineering College, Department of Information Technology, 2nd Year 4th Semester, 2023

Abstract

The Turning Machine is a theoretical model of computation that has played a central role in the development of computer science and the foundations of mathematics. In this project, we explore the Turning Machine in depth, analyzing its properties, limitations, and practical applications. We begin by introducing the basic concept of the Turning Machine and its relevance to the study of algorithms and complexity. We then examine its formal definition and explore some of its key features, such as the ability to simulate any computer program and the concept of universality. We also investigate some of the theoretical limitations of the Turning Machine, such as the Halting Problem and the incompleteness of formal systems. Finally, we consider some practical applications of the Turning Machine in areas such as artificial intelligence and cryptography, and explore how its insights can be applied to real-world problems. Overall, this project provides a comprehensive overview of the Turning Machine and its significance for the study of computation.

Introduction

The Turning Machine is a theoretical model of computation that plays a fundamental role in the study of computer science and the foundations of mathematics. It was first introduced by the mathematician Alan Turing in 1936 as a way to investigate the limits of mechanical computation. Despite its simplicity, the Turning Machine[1] has proven to be a powerful tool for exploring the boundaries of what is computable, and has been used to prove fundamental results such as the undecidability of the Halting Problem. Today, the Turning Machine remains an important concept in the study of algorithms and complexity, and continues to inspire new developments in the field of theoretical computer science. In this project, we will explore the Turning Machine in depth, examining its properties, its limitations, and its significance for the foundations of computation. We will also look at some of the practical applications of the Turning Machine in areas such as artificial intelligence[2] and cryptography, and consider how its insights can be applied to real-world problems.

Discussion

<u>Various types of turning machines</u>

The following are the different types of turing machines:

- Multi-tape turing machine
- Multi-head turing machines
- Multi-track turing machines
- Semi-infinite turing machines
- Universal Turing Machine
- Alternating Turing machine
- Non-Deterministic Turing Machine
- Unambiguous Turing Machine
- Quantum Turing Machine
- Read-Only Turing Machine
- Probabilistic Turing Machine

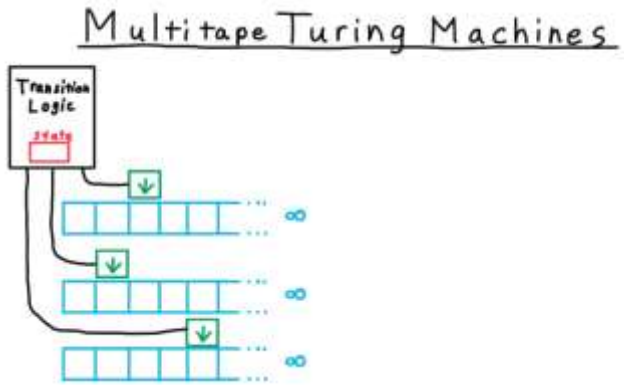
Multi-tape turing machine

This is just like any other ordinary turing machine except it has multiple tapes where each tape has its head for reading and writing[1].

In the beginning, the input is only on the first tape while the rest of the tapes remain blank. In the next step, the machine reads consecutive symbols under its head and prints a symbol on each tape then shifts its head.

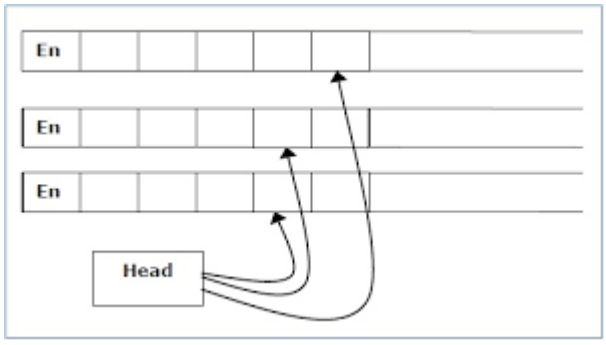
This may seem better than a single tape but regardless of the number of tapes, it can be simulated by a single tape turing machine.[2]

To conclude, multi-tape TMs have multiple tapes and each tape is accessed by a different head that moves independently of other heads.



Multi-head turing machines

These have a single tape with *n* number of heads each reading symbols on the same tape. In a single step, all heads determine the scanned symbols and write then shift left, right, or stay. These actions are independent of any other head.[1]

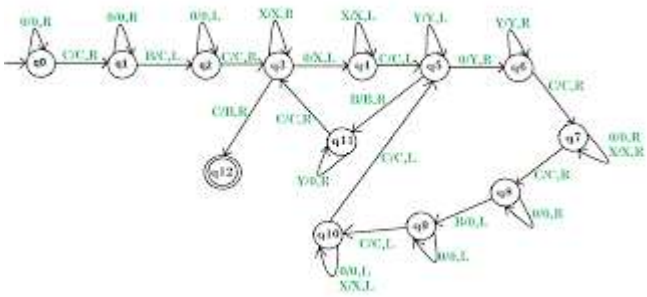


Multi-track turing machines

This turing machine has multiple tracks but a single tape head that reads and writes on all tracks.

These machines accept recursively enumerable languages just like a normal single-track single-tape turing machine.

Also, for every single-track turing machine, there exists an equivalent multi-track turing machine.



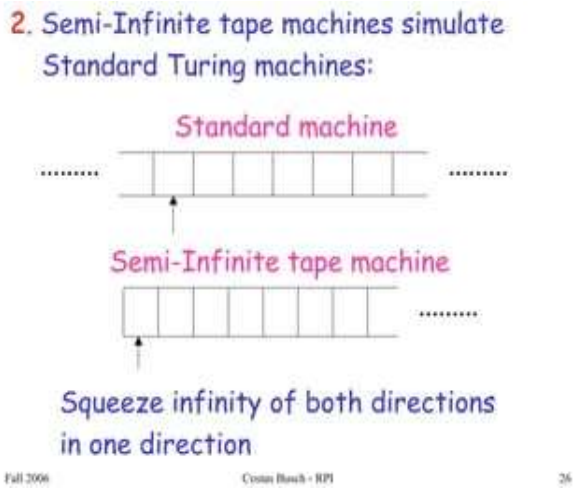
Semi-infinite turing machines

Such machines have left ends limited by an end marker and an infinite right end.

They also have two tracks, that is, the *upper track* that represents cells to the right of the initial head position and a *lower track* that represents cells to the left of the initial head position in reverse order.[6]

The machine starts from a state *q0* and scans from the left end marker. At each step it reads the symbol on the tape under its head then writes a new symbol on that tape and shifts the head to the left or right of one tape cell.

A *transition function*[2] determines the actions to be taken.

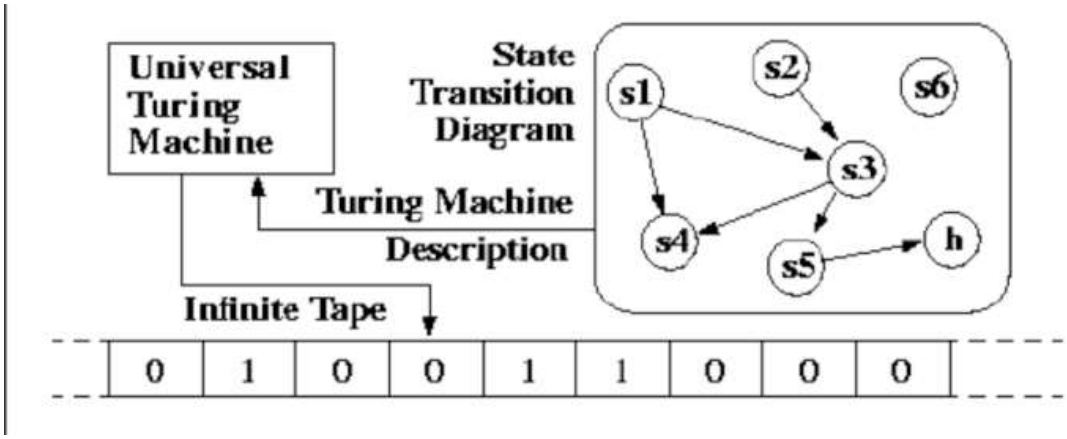


Semi-infinite machines have two special states, the *accept* and *reject* states. The former means that the input is accepted and the latter means that input is rejected when the machine enters that state. In other cases, the machine will continue infinitely without any rejections or acceptances for certain inputs.

Universal Turing Machine

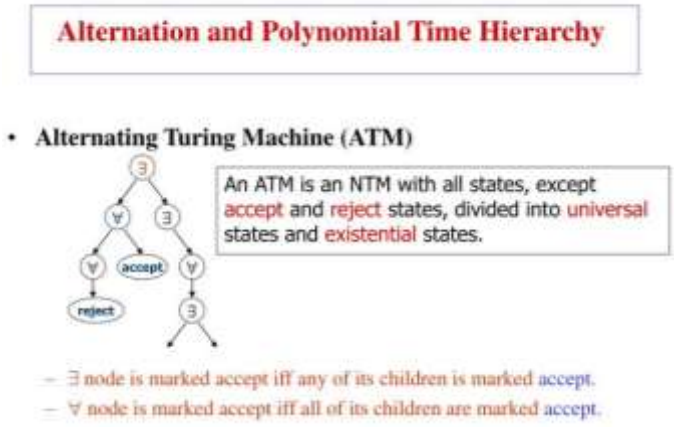
This is a type of turing machine that simulates an arbitrary turing machine on an arbitrary input. It does this by having three pieces of information for the machine it is simulating, the first is the basic description of the machine to be simulated, the second is the input to that machine's tape, and finally the internal state of the machine.

It controls the machine by changing its state based on the input.



Alternating Turing machine

This is a *Non-Deterministic* turing machine that accepts computations that generalize rules used in the definition of complexity classes of NP and *Co-NP*.[4]



Its states are divided into two sets namely *existential* and *universal* states. The former is accepting if a transition leads to an accepting state while the latter is accepting if every transition leads to an accepting state.[6]

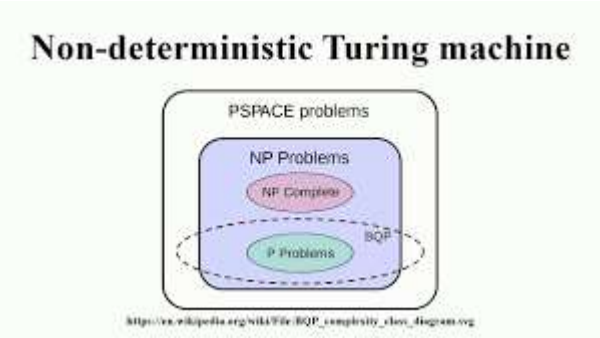
These machines are commonly used in complexity theory since they yield elegant characterizations of complexity classes.

Non-Deterministic Turing Machine

In such turing machines, for every state and symbol, there exists a group of actions the machine can have. Computations of such machines form a tree of configurations that are

reachable from the start configuration and therefore input is accepted if there is at least one node in the tree that is an accept configuration otherwise the input is not accepted.

Compared to a deterministic model of computing where a command leads to a single action, *NDTs* given specific commands allow for a range of actions. For example, an input *X* would lead to a variety of actions *Y(array)*.[3]



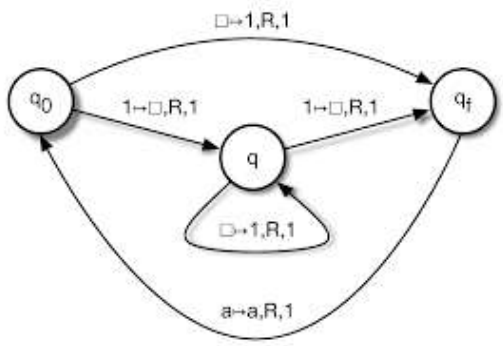
Unambiguous Turing Machine

This is a special type of a non-deterministic turing machine where for each input, there is exactly a single possible computation. In many ways, it is similar to a deterministic turing machine.

Quantum Turing Machine

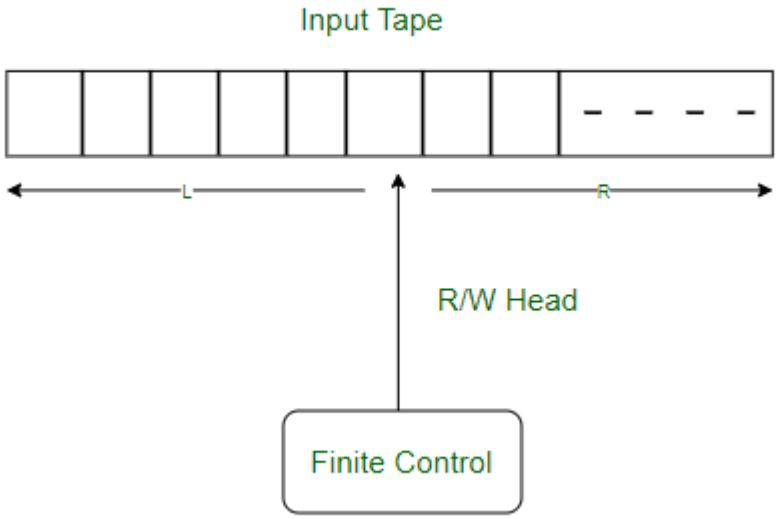
Also referred to as a *universal quantum computer*. It is an abstract machine that models the effects of a quantum computer by providing a simple model that covers the power of quantum computation.[5]

This means that any quantum algorithm is can be expressed formally as a quantum turing machine.



Read-Only Turing Machine

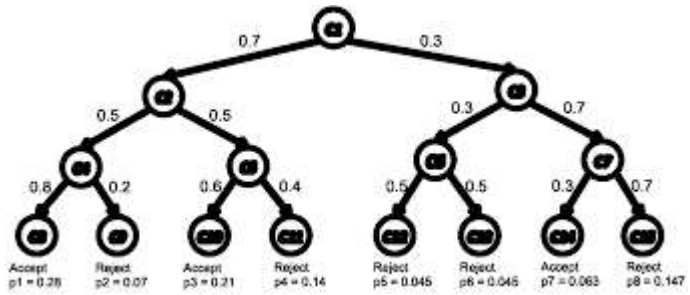
Also referred to as a *two-way deterministic finite-state automaton*.[4] It is a class of models of computability that behave like a standard turing machine and move in both directions across a given input except it cannot write to the input tape, only read it.
In its bare form, it is similar to a deterministic finite-state machine in computational power and therefore only parses regular languages.



Probabilistic Turing Machine

This is a turing machine that is modified to execute randomized computations. These machines make decisions based on the outcomes of unbiased coin tosses.
Every non-deterministic turing machine can be simulated by a probabilistic turing machine with a small error of probability.[5]

Also unlike deterministic turing machines, they have stochastic results whereby on a given input and instruction state machine, it can have different run times or may not halt at all, furthermore, it may accept input in one execution and reject the same input in another execution.



Common applications of such machines are modeling a robot that handles accidents in a nuclear plant or modeling space missions that have strong radiations.

Conclusion

In conclusion, the Turing machine is a theoretical device that has played a pivotal role in the development of modern computer science. Its simplicity and universality have allowed researchers to prove fundamental results about computation and the limits of what can be computed. Through this project, we have explored the workings of the Turing machine and its various components, including the tape, head, and state transition function. We have also examined its ability to perform computations by using it to simulate the behavior of other machines, such as a binary counter and a simple programming language. Additionally, we have discussed the limitations of the Turing machine and its relevance to modern computing. While the Turing machine is a powerful tool for theoretical analysis, it does not reflect the complexities and practicalities of real-world computing. Nevertheless, the concepts and principles behind the Turing machine continue to be fundamental to the study of computer science and the development of new technologies. Overall, this project has deepened our understanding of the Turing machine and its importance in the history and ongoing evolution of computing.

Reference

1. Fast algorithms
   A multitape Turing machine implementation

2. Combinators
   A Centennial View
   Stephen Wolfram
3. Fundamental Concept of Turing Machine
   Raghvendra Kumar, Prasant Kumar Pattnaik
   Alan Mathison Turing

4. Machines take me by surprise with great frequency
   Fast algorithms: A multitape Turing machine implementation January 1, 1994

5. **Introduction to Algorithms,** Writers: Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein.

6. **Data Structures and Algorithms Made Easy,** Writer: **Narsimha Karumanchi.**