MODULE 1: AUTOMATA, COMPUTABILITY, AND COMPLEXITY
WEEK 1

Learning Outcomes:

After completing this course you are expected to demonstrate the following:

1. Understand the three tradinonally central areas of the theory of computation: automata, computability and complexity.

A. Engage

Trivia

Two neurophysiologist, were the first to present a description of finite automata in 1943. Their paper entitled, "A Logical Calculus Immanent in Nervous Acitivity", made significant contributions to sthe study of neural network theory, theory of automata, the theory of computation and cybernetics.

Warren McCulloch and Walter Pitts

Figure 1.1

B. Explore

Video Title: **What is Automata Theory**
YouTube Link: **https://www.youtube.com/watch?v=nO1z6F_lwnM&t=30s**
Module Video Filename: **Week 1** - **What is Automata Theory**

C. Explain

Introduction

Automata Theory is an exciting, theoretical branch of computer science. It established its roots during the 20th Century, as mathematicians began developing - both theoretically and literally - machines which imitated certain features of man, completing calculations more quickly and reliably. The word **automaton** itself, closely related to the word "automation", denotes automatic processes carrying out the production of specific processes. Simply stated, automata theory deals with the logic of computation with respect to simple machines, referred to as **automata**. Through automata, computer scientists are able to understand how machines compute functions and solve problems and more importantly, what it means for a function to be defined as ***computable*** or for a question to be described as ***decidable***.

D. Elaborate

Basics of Complexity Theory

Computer problems come in different varieties; some are easy, and some are hard. For example, the sorting problem is an easy one. Say that you are need to arrange a list of numbers in ascending order. Even a small computer can sort a million number rather quickly. Compare that to a scheduling problem. Say that you must find a schedule of classes for the entire university to satisy some reasonable constraints, such as that no two classes take place in the same room at the same time. The scheduling probem seems to be much harder than the sorting problem. If you have just a thousand classes, fiinding the best schedule may require centuries, even with a supercomputer.

Basics of Computability Theory

During the first half of the twentieth century, mathematical such as Kurt Godel, Alan Turing, and Alonzo Church discovered that certain basic problems cannot be solved by computers. One example of this phenomenon is the problem of determining whether a mathematical statement is true or false. This task is the bread and butter of mathematicians. It seems like a natural for solution by computer because it lies strickly within the realm of mathematics. But no computer algorithm can perform this task.

Basics of Automata Theory

Automatons are abstract models of machines that perform computations on an input by moving through a series of states or configurations. At each state of the computation, a transition function determines the next configuration on the basis of a finite portion of the present configuration. As a result, once the computation reaches an accepting configuration, it accepts that input. The most general and powerful automata is the **Turing machine**.

The **major objective** of automata theory is to develop methods by which computer scientists can describe and analyze the dynamic behaviour of discrete systems, in which signals are sampled periodically. The behaviour of these discrete systems is determined by the way that the system is constructed from storage and combinational elements. Characteristics of such machines include:

- **Inputs:** assumed to be sequences of symbols selected from a finite set *I* of input signals. Namely, set *I* is the set $\{x_1, x_{,2}, x_3... x_k\}$ where *k* is the number of inputs.

- **Outputs:** sequences of symbols selected from a finite set Z. Namely, set *Z* is the set $\{y_1, y_2, y_3 ... y_m\}$ where *m* is the number of outputs.

- **States:** finite set *Q*, whose definition depends on the type of automaton.

The four major families of automaton:

1. Finite-State Machine
2. Pushdown Automata
3. Linear-Bounded Automata
4. Turing Machine

The families of automata above can be interpreted in a hierarchal form, where the finite-state machine is the simplest automata and the Turing machine is the most complex. The focus of this project is on the finite-state machine and the Turing machine. A Turing machine is a finite-state machine yet the inverse is not true.

E. Evaluate

ASSESSMENT:
Instructions: You may write your answer on the Answer Sheet (AS) provided in this module.

CONTENT FOR ASSESSMENT:
Identification. For 2-points each.

1. The two person who presented the first finite automata paper entiled “A Logical Calculus Immanent in Nervous Acitivy”.
2. It is an abstract models of machines that perform computation on an input by moving through a series of states.
3. The four major families of automaton.
4. The most general and powerful automata is the?
5. They are sequences of symbols selected from a finite set.

References

1. Introduction to the Theory of Computation , Second Edition by MICHAEL SIPSER
2. https://cs.stanford.edu/people/eroberts/courses/soco/projects/2004-05/automata-theory/refs.html#basics

Facilitated By:		
Name	:	
MS Teams Account (email)	:	
Smart Phone Number	:	