CSC 413 – AUTOMATA THEORY AND COMPUTABILITY (2 UNITS) LECTURE 1

BALOGUN JEREMIAH ADEMOLA

ASSISTANT LECTURER,

DEPARTMENT OF COMPUTER SCIENCE AND MATHEMATICS

MOUNTAIN TOP UNIVERSITY, OGUN STATE, NIGERIA

COURSE OUTLINE

- Automata Theory
  - Roles of Models in Computation
  - Finite State Automata (FSA)
  - Push-Down Automata (PDA)
  - Formal Grammars
  - Parsing
  - Relative Powers of Formal Models

- Computability
  - Finite State Machines
  - Turing Machines
  - Universal Turing Machines
  - Churih's Thesis
  - Solvability
  - Decidability

ROLES OF MODELS IN COMPUTATION

- In Computer Science, a model of computation is a model which describes how an output of a mathematical function is computed from an input.
- A model describes how units of computations, memories, and communications are organized.
- Using a model allows studying the performance of algorithms independently of the variations that are specific to particular implementations and specific technology.
- Models of computations can be classified as follows:
  - Sequential models – Finite State Machines, Pushdown Automata, Random Access Machines, Turing Machines etc.
  - Functional models – Lambda Calculus, General recursive functions, Combinatory Logic, Abstract Rewriting Systems etc.
  - Concurrent models – Cellular Automaton, Digital Circuits, Kahn Process Networks, Petri Nets, Synchronous Data Flow, Interaction Nets, Actor Model etc.

ROLES OF MODELS IN COMPUTATION…

- Models differ in their expressive power; for example, each function that can be computed by a Finite State Machine can also be computed by a Turing Machine, but not vice versa.

- A nondeterministic model of computation is associated with some of these models of computation
  - Nondeterministic models are not useful for practical computation; they are used in the study of computational complexity of algorithms.

- Models of can be broadly classified into the following broad categories:
  - Abstract Machines and models equivalent to it – used in proofs of computability and upper bounds on computational complexity of algorithms
  - Decision Tree Models – used in proofs of lower bounds on computational complexity of algorithmic problems.

ROLES OF MODELS IN COMPUTATION…

- Automata theory is the study of abstract machines and automata, as well as the computational problems that can be solved using them.

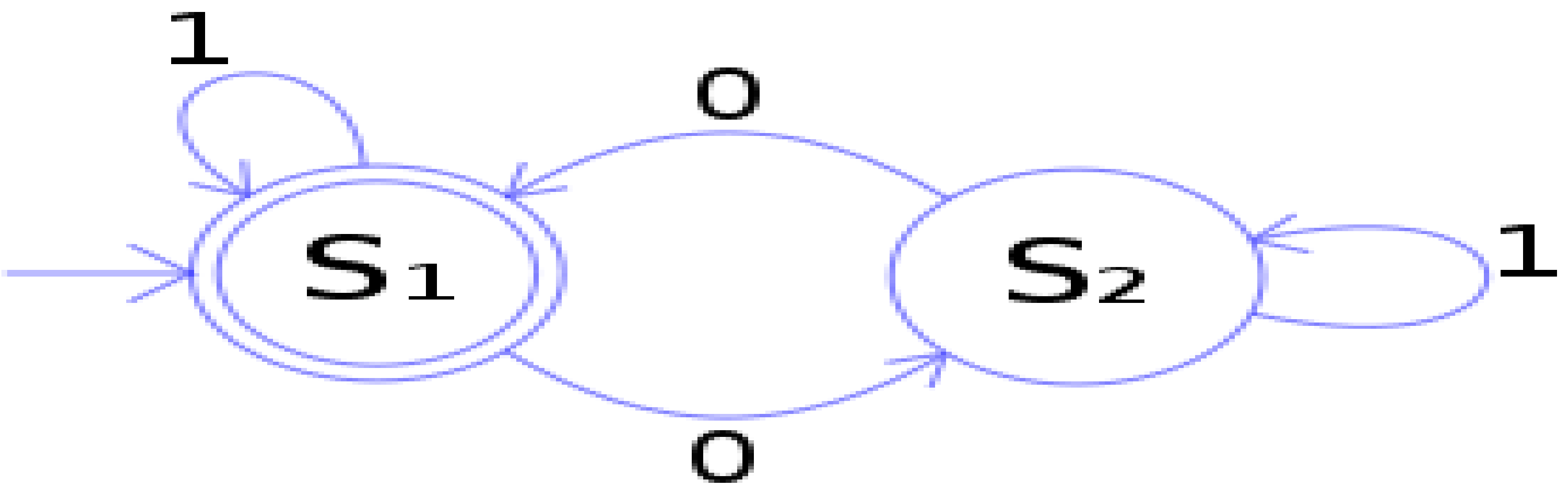
- The word automata (the plural of automaton) comes from the Greek word αὐτόματα, which means "self-making".

- Automata theory is closely related to formal language theory.
  - An automaton is a finite representation of a formal language that may be an infinite set.
  - Automata are often classified by the class of formal languages they can recognize, typically illustrated by the Chomsky hierarchy, which describes the relations between various languages and kinds of formalized logics.

- Automata play a major role in theory of computation, compiler construction, artificial intelligence, parsing and formal verification.

ROLES OF MODELS IN COMPUTATION…

- The figure below illustrates a finite-state machine, which belongs to a well-known type of automaton.
  - This automaton consists of states (represented in the figure by circles) and transitions (represented by arrows).
  - As the automaton sees a symbol of input, it makes a transition (or jump) to another state, according to its transition function, which takes the current state and the recent symbol as its inputs.



ROLES OF MODELS IN COMPUTATION…

- 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
  - what it means for a function to be defined as *computable* or for a question to be described as *decidable.*

- 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.

ROLES OF MODELS IN COMPUTATION…

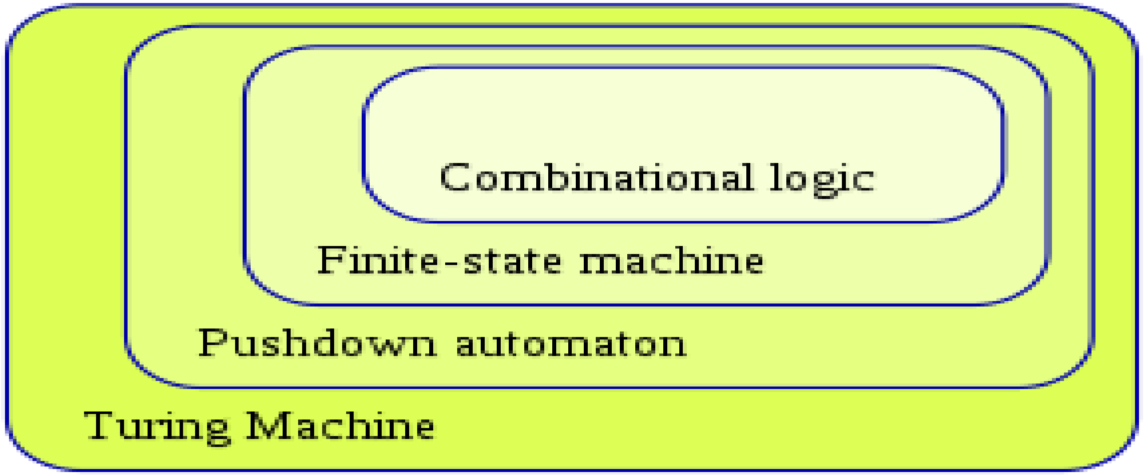
- The major objective of automata theory is to develop methods by which computer scientists can describe and analyze the dynamic behavior of discrete systems, in which signals are sampled periodically.
  - The behavior of these discrete systems is determined by the way that the system is constructed 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.
  - Outputs – assumed to be sequences of symbols selected from a finite set Z.
  - States – assumed to be a finite set Q, whose definition depends on the type of automaton.

- There are four major families of automaton :
  - Finite-state machine
  - Pushdown automata
  - Linear-bounded automata
  - Turing machine

FINITE STATE AUTOMATA (FSA)

- A Finite State Automata (FSA) or Finite State Machine (FSM) or a State Machine is a mathematical model of computation.
  - It is an abstract machine that can be in exactly one of a finite number of states at any given time.
  - The FSA can change from one state to another in response to some *inputs*; the change from one state to another is called a *transition*.
  - Thus, an FSA is defined by a list of its states, its initial state, and the inputs that trigger each transition.
- They are of two (2) types, namely: Deterministic and Nondeterministic FSA.



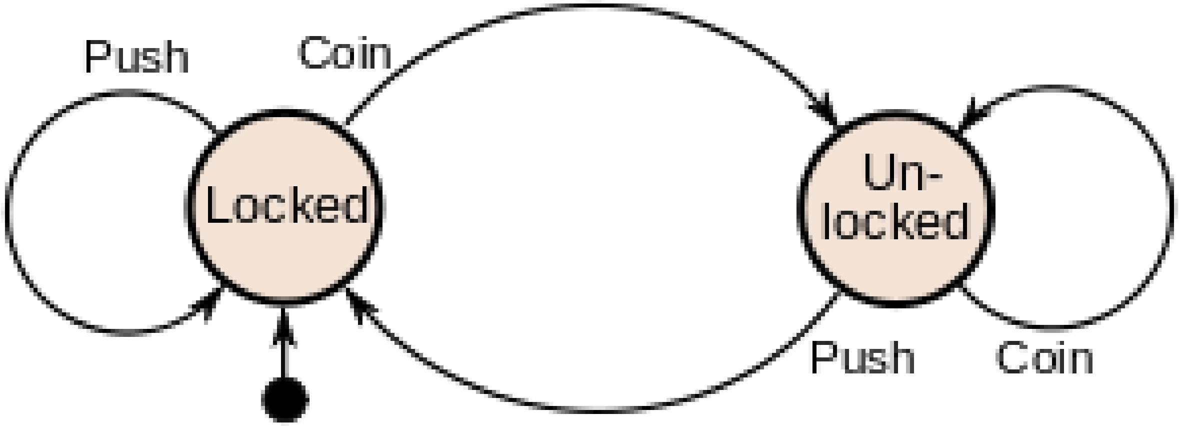
FINITE STATE AUTOMATA (FSA)…

- An example of a simple mechanism that can be modeled by a state machine is a turnstile.
- A turnstile, used to control access to subways and amusement park rides, is a gate with three rotating arms at waist height, one across the entryway.
- Initially the arms are locked, blocking the entry, preventing patrons from passing through.
- Depositing a coin or token in a slot on the turnstile unlocks the arms, allowing a single customer to push through.
- After the customer passes through, the arms are locked again until another coin is inserted.

FINITE STATE AUTOMATA (FSA)…

- Considered as a state machine, the turnstile has two possible states: ***Locked*** and ***Unlocked***.
- There are two possible inputs that affect its state: putting a coin in the slot (***coin***) and pushing the arm (***push***).
- In the locked state, pushing on the arm has no effect; no matter how many times the input ***push*** is given, it stays in the locked state.
- Putting a coin in – that is, giving the machine a ***coin*** input – shifts the state from ***Locked*** to ***Unlocked***.
- In the unlocked state, putting additional coins in has no effect; that is, giving additional ***coin*** inputs does not change the state.
- However, a customer pushing through the arms, giving a ***push*** input, shifts the state back to ***Locked***.

FINITE STATE AUTOMATA (FSA)…



Current State	Input	Next State	Output
Locked	coin	Unlocked	Unlocks the turnstile so that the customer can push through.
	push	Locked	None
Unlocked	coin	Unlocked	None
	push	Locked	When the customer has pushed through, locks the turnstile.

CLASSIFICATION OF FINITE STATE AUTOMATA (FSA)…

ACCEPTORS

- **Acceptors** (also called **detectors** or **recognizers**) produce binary output, indicating whether or not the received input is accepted.
  - Each state of an acceptor is either *accepting* or *non accepting*.

- Once all input has been received, if the current state is an accepting state, the input is accepted; otherwise it is rejected.
  - As a rule, input is a sequence of symbols (characters); actions are not used.

- The start state can also be an accepting state, in which case the acceptor accepts the empty string.

CLASSIFICATION OF FINITE STATE AUTOMATA (FSA)…

TRANSDUCERS

- **Transducers** produce output based on a given input and/or a state using actions.
  - They are used for control applications and in the field of computational linguistics.
- In control applications, two types are distinguished:
  - Moore Machine - The FSA uses only entry actions, i.e., output depends only on state. The advantage of the Moore model is a simplification of the behaviour.
  - Mealy Machine - The FSA also uses input actions, i.e., output depends on input and state. The use of a Mealy FSM leads often to a reduction of the number of states.

MATHEMATICAL MODELING OF FSA

- A deterministic finite-state machine or deterministic finite-state acceptor is a quintuple defined as:

$$(\boldsymbol{\Sigma}, \boldsymbol{S}, \boldsymbol{s_0}, \boldsymbol{\delta}, \boldsymbol{F})$$

Where:

- $\boldsymbol{\Sigma}$ is the input alphabet (a finite non-empty set of symbols)
- $\boldsymbol{S}$ is a finite non-empty set of states
- $\boldsymbol{s_0}$ is an initial state, an element of **S**
- $\boldsymbol{\delta}$ is the state-transition function: $\delta: S \times \Sigma \longrightarrow S$ (in a non-deterministic FSA, it would be $\delta: S \times \Sigma \longrightarrow \mathcal{P}(S)$ i.e. δ would return a set of states)
- $\boldsymbol{F}$ is the set of final states, a (possibly empty) subset of $\boldsymbol{S}$

MATHEMATICAL MODELING OF FSA

- A deterministic finite-state machine or deterministic finite-state transducer is a sextuple defined as:

$$(\Sigma, \Gamma, S, s_0, \delta, \omega)$$

Where:

- Σ is the input alphabet (a finite non-empty set of symbols)
- Γ is the output alphabet (a finite non-empty set of symbols)
- S is a finite non-empty set of states
- s_0 is an initial state, an element of S
- δ is the state-transition function: $\delta: S \times \Sigma \longrightarrow S$
- ω is the output function
  - $\omega : S \times \Sigma \longrightarrow \Gamma$ (for Mealy Machine) or $\omega : S \longrightarrow \Gamma$ (for Moore Machine)

BIBLIOGRAPHY

Brutscheck, M., Berger, S., Franke, M., Schwarzbacher, A., Becker, S.: *Structural Division Procedure for Efficient IC Analysis*. IET Irish Signals and Systems Conference, (ISSC 2008), pp.18–23. Galway, Ireland, 18–19 June 2008.

Fernández, M. (2009). Models of Computation: An Introduction to Computability Theory. *Undergraduate Topics in Computer Science*. Springer.

Ferraguti, A., Micheli, G. & Schnyder, R. (2018), Irreducible compositions of degree two polynomials over finite fields have regular structure, *The Quarterly Journal of Mathematics,* ***69***, Oxford University Press, pp. 1089–1099,

Hamon, G. (2005). *A Denotational Semantics for State flow*. International Conference on Embedded Software. Jersey City, NJ: ACM. pp. 164–172

Koshy, T. (2004). *Discrete Mathematics With Applications*. Academic Press. p. 762. ISBN 978-0-12-421180-3.

Pouly, M. & Kohlas, J. (2011). *Generic Inference: A Unifying Theory for Automated Reasoning*. John Wiley & Sons. Chapter 6. Valuation Algebras for Path Problems, p. 223

Savage, J.E. (1998). *Models Of Computation: Exploring the Power of Computing*. Addison-Wesley.

Wang, J. (2019). *Formal Methods in Computer Science*. CRC Press. p. 34. ISBN 978-1-4987-7532-8

Yan, Song Y. (1998). *An Introduction to Formal Languages and Machine Computation*. Singapore: World Scientific Publishing Co. Pte. Ltd. pp. 155–156.78-0201895391.