Course: Automata Theory

Lecture 1: Introduction to Automata Theory and Sets

Lecturer: Martha Gichuki

Course description

- The course begins with an introduction to logic and formal grammar where learners will do a recap on sets, logic and truth tables.
- A coverage of finite state machines, Push Down automata and Turing Machines (The Church's thesis) will culminate the study of various models of computation.
- Formal language and grammar will then follow to enable learners differentiate regular and context free languages.
- An evaluation of the computability and complexity of practical computational problems which are the foundations of automata theory will then be done and the outcome will be problem description.

Course objectives

Upon successful completion of the Automata Theory course, the learners should have an understanding of finite state and pushdown automata, regular languages and context free languages.

The learning outcomes of the course will be to enable learners:-

(i)Describe various models of automata,

(ii) Identify applications of automata theory

(ii)Simulate machines that illustrate automata theory concepts.

Learning outcomes lecture 1: Introduction to Automata Theory and Sets

At the end of the lecture, you will be able to:

(i) Define automata theory

(ii) Explain the importance of automata theory to computing.

(iii) Define set theory terms as used in automata theory

(iv) Solve problems involving set theory

Introduction to automata theory

•Automata theory is the study of abstract machines and problems they can solve. It introduces the theory of computation and complexity.

•Computability theory presents an analysis of solvable and unsolvable problems. Complexity theory answers the question *"What makes problems computationally hard?"* e.g. class timetabling problems have been found to be complex and computationally hard.

Applications of Automata Theory

- Hardware verification
  - Coin machines e.g. telephone booths, candy machines, parking ticket machines etc.
  - Automatic doors (elevators/lifts)
  - Automatic toy cars.
  - Household appliances e.g. Microwaves, Washing machines etc.
- Compiler design techniques

Prerequisite courses

- We introduce the course by reminding ourselves the basic mathematical foundations covered in the pre-requisites of the class i.e. Discrete Mathematics and Fundamentals of Computer Systems.

- Automata theory is closely related to **formal language theory** because automatons are often classified by the class of formal languages they can recognize.

Definition of key terms in automata theory

•An automaton is always anchored on the basic concepts of **symbols, words alphabets and strings.** These are defined briefly as follows: -

•**Symbol –** A character or letter (An arbitrary datum) that has some meaning to or effect on the machine. Symbols are sometimes called letters.

•**Word –** A finite string formed by the concatenation of several symbols.

•**Alphabet –** A finite set of symbols. It is frequently denoted by Σ, which is the set of letters in an alphabet.

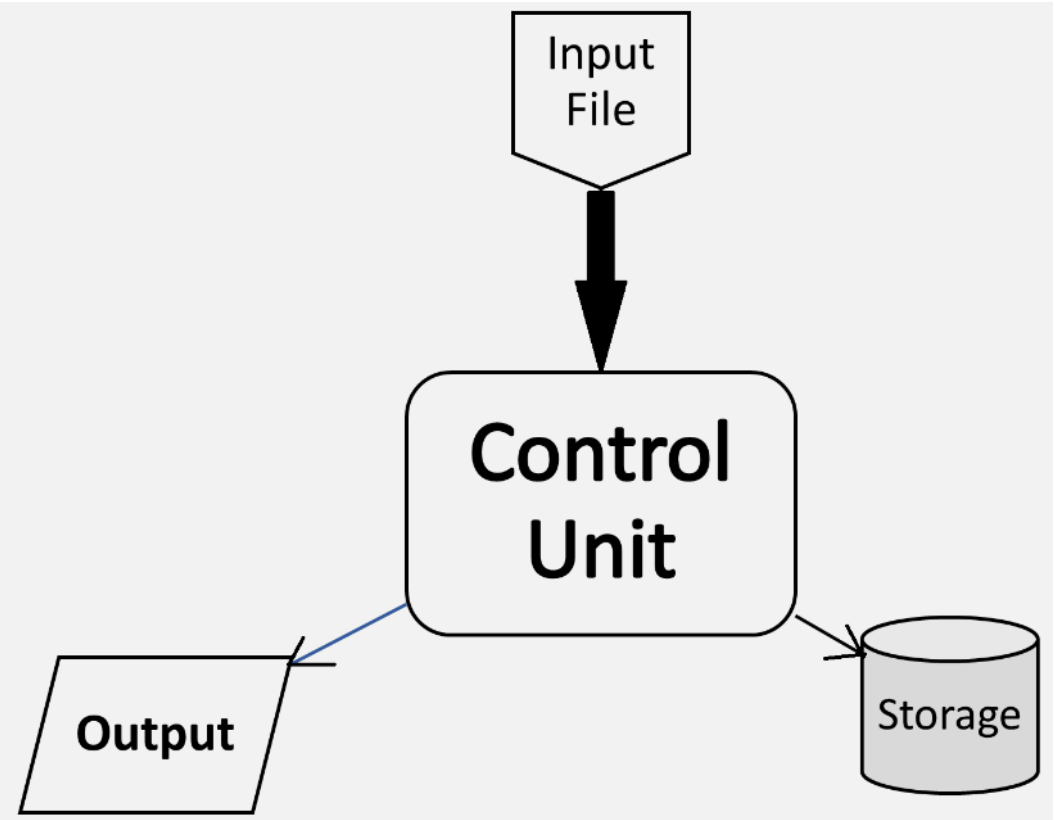
Definitions continued…

Language - refers to a set of words formed by symbols in each alphabet.

Kleene Closure – A language may be thought of as a sub-set of possible words. The set of all possible words may in turn be thought of as the set of all possible concatenations of strings. Formally, this set of all possible strings is called a free monoid. It is denoted as * and the super script $^{(*)}$ is called the **Kleene Star**.

How an Automata Works:

An automaton has a mechanism to **read input** which is a **string over a given alphabet**. This input is written on an "**input file**", which can be read, by the automaton but cannot be changed. An input file is divided into cells, each of which can hold one symbol. The automaton has a **temporary storage device** which has unlimited number of cells and the contents of which, can be altered by the automaton. An automaton has a **control uni**t which is said to be in one of a finite number of internal states. The automaton can change states in a defined way.



Definition of terms

Set: It is a well-defined collection of objects or elements e.g. set $A = \{a, e, i, o, u\}$ is the set of the English alphabets.

A set is a non-repeating, unordered collection of objects (elements, members). E.g. the collection of four letters; a, b, c d is a set which is written as: L = {a, b, c, d}.

Definition continued…

Elements: The objects comprising a set are called its **elements or members**.

Singleton: It is a set having only one element e.g. set A = { 3} is a singleton.

Empty set: A set E with no elements is called an **empty set** and it is denoted by ∅ or { }

Determining members of a set

There are two ways of determining the members of a set:

i). **Listing** all the set elements e.g. {a, e, i, o, u}

ii). Providing an **algorithm or a rule such as grammar** or predicate e.g.

Set $X = \{x | 1 < x \leq 10, x \in \mathbb{N}\}$

Set Notation

The notations being used to denote sets are:

- To indicate that **x is a member of the set S**, we write **x ∈ S**.
- If every element of set A is also an element of set B, we say that **A is a subset of B** and is denoted as **A ⊆ B**
- If every element of Set A is also an element of set B, but B also has some elements not contained in A, then we say that **A is a proper subset of B** and is written as **A ⊂ B**.

Union of sets

The union of two sets is the set that has objects that are elements of at least one of the two given sets, and possibly both i.e. the union of sets A & B written as A∪B is a set that contains **everything in A or in B or in both**. E.g. if

$$A \cup B = \{a: a \in A \cup a \in B\}$$

Example if $A = \{1,2,3\}; B = \{4,5,6\} => A \cup B = \{1,2,3,4,5,6\}$

Intersection of sets

The intersection of sets A and B written as $A \cap B$ is a set that contains exactly those elements that are in A & B.

Example if

$A = \{1,2,3\}; B = \{3,4\}$ and $C = \{a, e\} \Rightarrow A \cap B = \{3\}$ while $A \cap C = \{\}$or $A \cap C = \emptyset$

Set Difference

- The set difference of a set A and set B written as A-B is a set that contains everything in A but not in B.

Example if A = {1, 2, 3}, B = {3, 4, 9}

A-B = {1, 2} i.e. A-B = {x: X∈A & X∉B }

- **Example** if A= {C, C++, Lisp, Java, FORTRAN}, B= {C, Lisp}

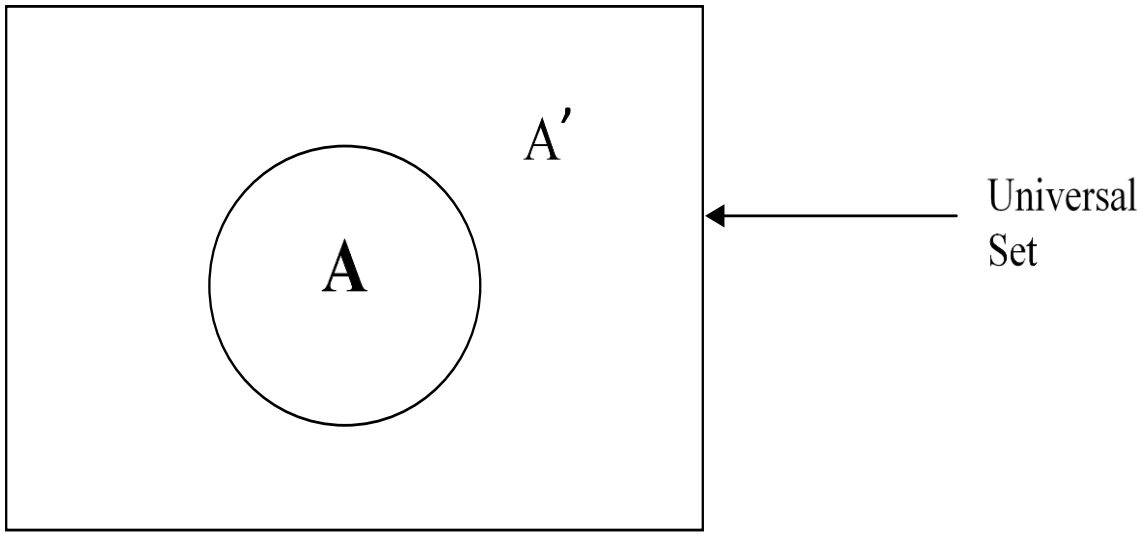
A-B = { C++, Java, FORTRAN} {x: X∈A & X∉B }

B-A = { } (empty set) {x: X∈B & X∉A }

Complement of a set

- The complement of a set A, written as $\overline{A}$ or ~A or $A^{/}$ or ¬A is the set containing everything that is not in A or simply Not A e.g.

everything that is not in A or simply Not A e.g.



Set Cardinality

- This refers to the **size of set subsets or the number of elements in set A**.
- It is denoted by |A| or n(A). For example the cardinality of set $X = \{1,2,3,4, t\}$ is $|X| = 5$

Power Set

- This refers to a **set of all subsets**.

Example if set A = {1, 2, 3},

P (A) also written as 2^A={∅, {1}, {2}, {3}, {1, 2}, {1, 3}, {2, 3}, {1, 2, 3}}

$$|P(A)| = 2^{|A|} = 2^3 = 8$$

$$\therefore |P(A)| = 2^{|A|}$$

Properties of set operations

The following laws hold for the three given sets A, B, & C.

- Independent Law

$A \cup A = A$; $A \cap A = A$

Example if $A = \{1, 2, 3\}$,

$A \cup A = \{1, 2, 3\}$

$A \cap A = \{1, 2, 3\}$

Commutative Law

- $A \cup B = B \cup A$;
- $A \cap B = B \cap A$

Example if $A = \{1, 2, 3\}$, $B = \{3, 4, 9\}$

- $A \cup B = B \cup A = \{1, 2, 3, 4, 9\}$
- $A \cap B = B \cap A = \{3\}$

Associative law

- $(A \cup B) \cup C = A \cup (B \cup C)$
- $(A \cap B) \cap C = A \cap (B \cap C)$

Distributive law

- $(A \cup B) \cap C = (A \cap C) \cup (B \cap C)$
- $(A \cap B) \cup C = (A \cup C) \cap (B \cup C)$

Absorption Law

- $(A \cup B) \cap A = A$
- $(A \cap B) \cup A = A$

De Morgan's Laws

- De Morgan's laws states that union and intersection interchange under complementation.
- $\sim(A \cup B) \equiv \sim A \cap \sim B$
- $\sim(A \cap B) \equiv \sim A \cup \sim B$

Exercises Continued…

1. For any alphabet Σ, the set of all strings over Σ is denoted by Σ^*, suppose Σ is in the two symbol alphabet {a, b}, what are the possible subsets of Σ^*
2. Let A be the set {x, y, z}, and B be the set {x, y}:
   a) Is **A** a subset of B?
   b) Is **B** a subset of A?
   c) What is $A \cup B$?
   d) What is $A \cap B$?
   e) What is A X B?
   f) What is the power set of B?

Exercises Continued…

1. Let the sets: A={a, b}, B={c, d} and C={e, f}; use the three sets to define and proof the following set laws -
   a) Commutative law
   b) Associative law
   c) Distributive law
   d) Demorgan law
2. If set **A** has ***a*** elements and set **B** has ***b*** elements, how many elements are in ***A X B***? Explain your answer.
3. If **C** is a set with ***c*** elements, how many elements are in the ***power set of C***? Explain your answer**.**

Exercises Continued…

1. Using examples, describe two differences between a set and an ordered pair
2. Describe the following sets by regular expressions:

a) {01, 10}

b) {101}

c) {λ, 1,1,1,1,1,1,…}

d) {1,1,1,1,1,1,1,…}

Exercises Continued…

Let set A = {1, 2, 3}; Set B = {4, 5, 6} and Set C = {7, 8, 9} .Proof the following properties of sets (laws) using the set membership.

a) $A \cup B = \{1, 2, 3, 4, 5, 6\}$
b) $B \cup C = \{4, 5, 6, 7, 8, 9\}$
c) $(A \cup B) \cap (B \cup C) = \{4, 5, 6\}$
d) $A - B = \{\}$
e) $B - C = \{\}$
f) $A' = (B \cup C)$

Course references

1. Rowan G. & John T., (2009), *Discrete Mathematics: Proofs, Structures and Applications*, CRC Press, ISBN: 9781439812808.
2. W. D. Wallis (2003), *A Beginners Guide to Discrete Mathematics*, Springer Science & Business Media, ISBN: 978-0817642693.
3. Introduction to the theory of computation (3rd ed.), Michael, S. Boston, Cengage Learning. ISBN-13: 978-1133187790, (2012).
4. Introduction to languages and the theory of computation (3rd ed.), Martin, J., New York: McGraw-Hill. ISBN-13: 978-0072322002, (2002)
5. Introduction to the theory of neural computation, John, A. H., Anders, S. K., & Richard, G. P., Boulder: Westview Press. ISBN-10: 0201515601, (1991).