ADDIS ABABA SCIENCE AND TECHNOLOGY UNIVERSITY
UNIVERSITY FOR INDUSTRY

SWEG3101

Formal Language and Automata Theory

College of Engineering

Department of Software Engineering

Course Outline

2nd Semester 2023/2024

Table of Contents

1. Course Information

Course Code	SWEG3101
Course Name	Formal Language and Automata Theory
Year/Semester	III / II
Academic Year	2023/2024
Pre-requisite(s)	SWEG2105 & SWEG3103
Co-requisite(s)	-
Credit Hour / ECTS	3
Course schedule	Lecture [Mon | 7:30p - 10:30p] - [B59 4-001] – Sec1 Lecture [Tue | 7:30p-10:30p] - [B59 4-002]- Sec2 Lecture [Fri | 02:30a - 06:20a] - [B59 4-003] – Sec3 Lecture [Fri | 7:30p - 10:30p] - [B59 4-004] – Sec4

2. Instructor Information

Course Coordinator

Name	Sebahadin N.
E-mail	Sebahadin@gmail.com
Telephone number <Optional>	---
Consultation hour	Wednesday 2:00 -4:00 am
Office (Block/Room)	68 / 407

Course Instructor/s

Name	Sebahadin N.
E-mail	Sebahadin@gmail.com
Telephone number <Optional>	---
Consultation hour	Wednesday 2:00 -4:00 am
Office (Block/Room)	68 / 407

3. Course Overview

Formal languages and automata theory is based on mathematical computations. These computations are used to represent various mathematical models. Automata theory is a theory of models. The working of every process can be represented by means of models. The model can be a theoretical or mathematical model. The model helps in representing the concept of every activity. In this Basics of all fundamental concepts of automata theory and those are strings, languages, operations on the languages will be discussed. Formal languages including grammar, finite automaton, regular expression, pushdown automaton, Context-free Grammar, Context-free Languages, Backus- Naur, Chomsky Normal Form, and the Turing machine also will be discussed. Not only do they form basic models of computation, but they are also the foundation of many branches of computer science, e.g. compilers, software engineering, and concurrent systems.

4. Course Objectives

- To discuss concepts in automata theory and theory of computation
- To describe the central concepts of automata theory
- To acquire insights into the relationship among formal languages, formal grammars, and automata.
- To design grammars and recognizer for different formal languages
- To explain models of computation, resources (time and space), algorithms, computability, and complexity.

5. Course Learning Outcomes

CLO1 Acquire a fundamental understanding of the core concepts in automata theory and formal languages

CLO2 An ability to design grammars and automata (recognizers) for different language classes. And an ability to identify formal language classes and prove language membership properties

CLO3 An ability to prove and disprove theorems establishing key properties of formal languages and automata

CLO4 Acquire a fundamental understanding of core concepts relating to the theory of computation and computational models including (but not limited to) decidability and intractability.

Up on successful completion of this course, students should be able to:		Program learning outcomes
CLO 1	Acquire a fundamental understanding of the core concepts in automata theory and formal languages	[PLO1]
CLO 2	An ability to design grammars and automata (recognizers) for different language classes. And an ability to identify formal language classes and prove language membership properties	[PLO3]
CLO 3	An ability to prove and disprove theorems establishing key properties of formal languages and automata	[PLO2]
CLO 4	Acquire a fundamental understanding of core concepts relating to the theory of computation and computational models including (but not limited to) decidability and intractability.	[PLO2]

Graduate Attributes

PLO1	Engineering Knowledge		PLO7	Environment and sustainability
PLO2	Problem Analysis		PLO8	Ethics
PLO3	Design/development of solutions		PLO9	Individual and teamwork
PLO4	Investigation		PLO10	Communication
PLO5	Modern tool usage		PLO11	Project management and finance
PLO6	The engineer and society		PLO12	Lifelong learning

6. Course Schedule and Instructional Method

The course lecture slots will be used for presenting and discussing every topic Principles of Compiler Design.

Week	Topic [Module]	Learning and Teaching Activity	CLO
Week 1	**Chapter 1**: **Mathematical Preliminaries and Notation** 1.1 Set 1.2 Function and Relations 1.3 Graphs and Trees 1.4 Proof Techniques 1.5 Basic Concepts Automata • Automata Machine • Symbols. String and Language • Introduction to Grammars	Lecture and Discussion	1

Week 2 **Week 3**	**Chapter 2: Finite Automata** 2.1 Deterministic Finite Automata (DFA) 2.2 Non-deterministic Finite Automata (NFA) 2.3 Equivalence of NFA and DFA 2.4 Conversion NFA-DFA 2.5 Reduction of the Number of States of Finite Automata 2.6 Two way Finite Automata	Lecture and Discussion	*2,3*
Week 4 **Week 5**	**Chapter 3: Regular Expressions, Regular Languages andRegular Grammars** 3.1 Formal Definition of a Regular Expression 3.2 Languages Associated with Regular Expressions 3.3 Regular Expressions to Finite Automata, 3.4 Finite Automata to Regular Expressions 3.5 Applications of Regular Expressions 3.6 Right- and Left-Linear Grammars • Right-Linear Grammars Generate Regular Languages • Left-Linear Grammars for Regular Languages • Conversion of Left-Linear Grammars into Right-Linear Grammars 3.7 Right-Linear Grammars and NFAs 3.8 Regular Language Equivalence and DFA Minimization	Lecture and Discussion	3
Week 6 **Week 7** **Week 8**	**Chapter 4**: **Context free Languages and Context freeGrammars** 4.1 Context free languages 4.2 Context free Grammars 4.3 Derivation tree or Parse tree • Definition, Relationship between parse treesand derivations. • Sentential forms • Partial Derivation Tree • Left-Linear| Right-Linear| Mixed Derivation • Parsing and ambiguity 4.4 Context-Free Grammars and Programming Language 4.5 Simplification of context free grammar • Methods for Transforming Grammars • Chomsky's hierarchy of grammars • Chomsky Normal Form	Lecture and Discussion	3
Week 9	**Chapter 5**: **Pushdown automata**	Lecture and	3

Week 10 **Week 11**	5.1 Non deterministic Pushdown Automata 5.2 Pushdown Automata for Context Free Languages 5.3 Context-Free Languages for Pushdown Automata	Discussion	4
Week 12 **Week 13**	**Chapter 6: Turing Machine** 6.1 The standard Turing machine 6.2 Turing machine as machine accepters 6.3 Turing machine as Transducers 6.4 Universal Turing Machines	Lecture and Discussion	3,4
Week 14	**Chapter 7: Complexity Theory** 7.1 Introduction 7.2 Polynomial- Time Algorithm 7.3 Non- Deterministic Polynomial Time Algorithm 7.4 NP Problems	Lecture and Discussion	5

7. Assessment Tasks

No	Assessment Method	Weight	Week Due	CLO Assessed
1	Tests	20 %	7	1,2,3,5
2	Assignment	10 %	12	1, 2, 3
3	Quiz	20 %	5	*3*
6	Final Exam	50%	16	1, 2, 3, 4,5,6

Submission of Assessment Tasks

- Assessment tasks, i.e. Assignment must be submitted electronically unless you are informed explicitly by the instructor.
- Each assessment task will have a specified deadline, clearly communicated during the class time.
- It is better to submit your work before the designated deadline to avoid any late submission penalties.
- Follow the instructions provided by the instructor or the proper documentation report/Assignment format for each assessment task

Late submission penalty (without prior permission)

- If you anticipate difficulties in meeting the deadline due to extenuating circumstances, communicate with the instructor as soon as possible to discuss possible accommodations.
- The penalty for late submission will be a deduction of 15% from the original score of the assessment task per day.
- If the assignment or the project is not submitted in four consecutive days from the due date the result will be zero

Feedback on Assessments

- Feedback and grades for assessment tasks will be provided within 2 weeks from the submission date.
- Feedback would be given in the instructor office mentioned above
- If you have questions or require clarification regarding the feedback received, do not hesitate to reach out to the instructor.
- Review the feedback provided to understand your strengths and areas for improvement.

8. Learning Resources

Textbooks

1. Introduction To Formal Languages And Automata, 6Th Edn by Peter Linz, Jones & Bartlett, 2016, Paperback, 9789384323219
2. Introduction to Automata Theory, Languages, and Computation: Global Edition, 3rd edition John E. Hopcroft , Rajeev Motwani, Jeffrey D. Ullman

9. Grading Policy

Grading of courses will be according to the university's rules and regulation. Examinations are graded and the letter grading system with corresponding points shown below.

Marks	Grade	Grade Points
[90,100]	**A+**	4.00
[85,90)	**A**	4.00
[80,85)	**A-**	3.75
[75,80)	**B+**	3.50
[70,75)	**B**	3.00
[65,70)	**B-**	2.75
[60,65)	**C+**	2.50
[50,60)	**C**	2.00
[50	**F**	0.00

Pass requirements

To pass this course a student must:

- attempt all assessments.
- achieve a minimum of 40% in the final exam.

10. Attendance

As per university guideline, a minimum 80% during lecture and 100% during practical work sessions except for some unprecedented mishaps. Failure to fulfill this requirement result in barring.

11. Academic Integrity, Referencing and Plagiarism

Academic Integrity

AASTU values academic integrity. Therefore, all students must understand the meaning and consequences of cheating, plagiarism and other academic offenses under the Code of Student Conduct and Disciplinary Procedures. Work submitted must be your own, and others' ideas should be appropriately acknowledged. If you don't follow these rules, plagiarism may be detected in your work using plagiarism detection tool 'Turnitin'.

Referencing

Referencing is a way of acknowledging the sources of information that you use to research your assignments. You need to provide a reference whenever you draw on someone else's words, ideas or research. Not referencing other people's work can constitute plagiarism.

Referencing Style

The University advises students to use the ["IEEE Referencing Style"] for written work and oral presentations.