Pokhara University
Faculty of Science and Technology

Course No.: CMP264	Full marks: 100
Course title: **Theory of Computation (3-1-0)**	Pass marks: 45
Nature of the course: Theory	Total Lectures: 45 hrs
Level: Bachelor	Program: BE (Computer)

1. Course Description

This course is designed to provide basic knowledge of the theory of automata, formal languages and computational complexity.

2. General Objectives

- To acquaint the students with the basic knowledge of automata and formal languages.
- To develop the skills in students to design various types of automata and analyze them.
- To acquaint the students with the concepts of computability, computational bounds and computational complexity.

3. Methods of Instruction

Lecture, Discussion, Readings, Tutorials.

4. Contents in Detail.

Specific Objectives	**Contents**
- Understand the concept of alphabet and language	**Unit 1: Introduction (4 hrs.)** 1.1 Review of set, relation and function 1.2 Proof techniques– proof by contradiction, pigeon hole principle, induction and diagonalization. 1.3 Alphabets and language 1.4 Chomsky's hierarchy?

- Design and implement the deterministic and non-deterministic finite automata. - Develop the equivalence of regular languages and finite automata	**Unit 2: Finite Automata and Regular Language (10 hrs)** 2.1 Deterministic Finite Automata, Non-Deterministic Finite Automata 2.2 Regular expressions and regular language, equivalence of regular language and finite automata 2.3 Properties of regular language 2.4 Pumping lemma for regular sets 2.5 Closure properties of regular sets 2.6 Decision algorithms for regular sets
- Explain the theory and design of context-free grammar and pushdown automata and their equivalence. - Explore the derivation trees, simplification and formal forms of context-free grammar.	**Unit 3: Context-Free Language and Pushdown Automata (13 hrs)** 3.1 Context-free grammar 3.2 Derivative trees and simplification of context-free grammar 3.3 Normal forms (CNF, GNF) 3.4 Pushdown automata (formal description and final state PDA design) 3.5 Equivalence of pushdown automata and context-free grammar 3.6 Properties of context-free languages (CFL) 3.7 Pumping lemma for CFL's 3.8 Closure properties of CFL's 3.9 Decision algorithms for CFL's
- Explain the theory and significance of Turing machines - Explain computing mechanism and extensions of Turing machines - Understand the computable languages, functions and unrestricted grammar	**Unit 4: Turing Machines (10 hrs)** 4.1 Introduction to Turing machine 4.2 Computing with Turing machine 4.3 Extensions of Turing machine 4.4 Unrestricted grammar 4.6 Recursively enumerable languages
- Use the idea of undecidability introducing Church-Turing thesis and the halting problem - Understand the universal Turing machines and undecidable problems of Turing machines	**Unit 5: Undecidability (4 hrs)** 5.1 The Church-Turing thesis 5.2 Halting Problem 5.3 Universal Turing machines 5.4 Undecidable problems about Turing machines 5.5 Properties of Recursive and Recursively enumerable languages.

- Understand the concept of computational complexity and different classes of problems	**Unit 6: Computational Complexity (4 hrs)** 6.1 Introduction to Complexity theory, tractable and intractable problems. 6.3 Class P and Class NP problems 6.4 NP-complete problems.

5. List of Tutorials:

The various tutorial activities that suit this course should cover all the content of this course to give students a space to engage more actively with the course content in the presence of the instructor. Students should submit tutorials as assignments or class-works to the instructor for evaluation. The following tutorial activities of 15 hours per group of maximum 24 students should be conducted to cover the content of this course:

A. Discussion-based Tutorials: (2 hrs)

1. Review discussion on set theory concepts like functions, relations, etc.
2. Discussion on the theoretical meaning and significance of computation and computers
3. A study and presentation can be done on the history of computation and Alan Turing as the father of computation (Oral Presentation).

B. Problem solving-based Tutorials: (8 hrs)

1. Solve problems to design deterministic and non-deterministic finite automata (DFA and NDFA) to recognize/generate given regular languages.
2. To develop regular expressions to recognize/generate given regular languages.
3. To convert a given NDFA to DFA.
4. To design Context-Free Grammar (CFG) to recognize/generate given Context-Free Language (CFL)
5. To simply a given CFG and convert it into CNF and GNF
6. To design Pushdown Automata (PDA) to recognize/generate given Context-Free Language (CFL)
7. To design Turing machines to recognize/generate given languages

C. Review and Question/Answer-based Tutorials: (5 hrs)

1. Case study of examples of DFA, PDA and Turing Machines for some practical tasks followed by Oral Presentation/demonstration in class.
2. Case study of Unsolvable problems such as the Tiling problem followed by Oral Presentation in class.
3. Case study of Class NP and NP-complete problems (eg, Travelling Salesman Problem) followed by Oral Presentation in class.

7. **Evaluation system and Students' Responsibilities**

Internal Evaluation

The internal evaluation of a student may consist of assignments, attendance, internal assessment, etc. The internal evaluation scheme for this course is as follows:

Internal Evaluation	**Weight**	**Marks**	**External Evaluation**	**Marks**
Theory		50	Semester-End examination	50
Attendance & Class Participation	10%			
Assignments	20%			
Presentations/Quizzes	10%			
Internal Assessment	60%			
Total Internal		50		
Full Marks: 50 + 50 = 100				

Student Responsibilities:

Each student must secure at least 45% marks separately in internal assessment and practical evaluation with 80% attendance in the class in order to appear in the Semester End Examination. Failing to get such a score will be given NOT QUALIFIED (NQ) to appear for the Semester-End Examinations. Students are advised to attend all the classes, formal exam, test, etc. and complete all the assignments within the specified time period. Students are required to complete all the requirements defined for the completion of the course.

8. Prescribed Books and References

Text Books:

1. Lewis, H. R., & Papadimitriou, C. H., *Elements of theory of computation*, Pearson Education.
2. R. McNaughton, *Elementary Computability, Formal Languages, and Automata*, Prentice Hall of India.
3. Hopcroft, J. E., Motwani, R., & Ullman, J. D. (2008). *Introduction to automata theory, languages, and computation*. Pearson.

References:

1. M. Sipser, *Introduction to the Theory of Computation*, Thomson Course Technology
2. E. Engeler, *Introduction to the Theory of Computation*, Academic Press.