Module Specification

Discrete Mathematics [TSI]

Version: 2023-24, v2.0, 09 Aug 2023

Contents

Part 1: Information

Module title: Discrete Mathematics [TSI]

Module code: UFCFRW-12-1

Level: Level 4

For implementation from: 2023-24

UWE credit rating: 12

ECTS credit rating: 6

College: College of Arts, Technology and Environment

School: CATE School of Computing and Creative Technologies

Partner institutions: Transport and Telecommunication Institute

Field:

Module type: Module

Pre-requisites: None

Excluded combinations: None

Co-requisites: None

Continuing professional development: No

Professional, statutory or regulatory body requirements: None

Part 2: Description

Overview: Not applicable

Features: Not applicable

Educational aims: The aim of this module is to instruct students in methods and models of computer mathematics: logical functions, logical networks, finite automaton, graph theory, flows in networks, algorithm theory, formal systems (theories). The module adopts the use of modern software such as R, Python, Mathcad to complete the practical assignments. The module provides a strong practical element giving ample opportunity to learn and practise new skills.

Outline syllabus: Introduction to graph theory;
Non-oriented graph main characteristics. Eulerian graphs;
Trees and oriented trees. Spanning trees of the non-oriented graphs. The Kruskal's algorithm of minimal spanning tree searching;
Tree of shortest paths and its searching algorithm;
Networks and flows in the networks. Algorithm of the maximal flow and the minimal cost flow searching;
Transport problem;
Definition and representation ways of logical functions (LF). Elementary (LF) and their properties. LF superposition;
Disjunctive and conjunctive normal forms of LF. Completeness of LF family. Post's theorem;
A problem of LF minimization. Quine - Mc Klosky algorithm for the enumeration of LF simple implicative functions. LF minimal form searching. Covering problem;
Finite state automatons (machines);
A minimization of the numbers of states for the finite automaton;
Intuitive requirements to algorithm. Various models of the algorithms. Church thesis;
A definition and representation ways of the Turing machine, Universal Turing;
machine. The way of its construction;
Stopping and self-applicability problem;
A proof of its algorithmically non-resolvability. General algorithm theory;
Introduction to the logic. Predicates and propositions, Truth formulas;
Principles of formal systems (theories) construction, Propositional calculus;
Logic of the propositional calculus;
Predicate calculus. Logic of predicate calculus;
The formal arithmetic;
Meta-theory of the formal systems;

Part 3: Teaching and learning methods

Teaching and learning methods: This module will adopt a mix of lectures and practical classes. During lectures, theoretical aspects of the course will be provided to students by the teaching staff. Lectures will be supported by presentation published and available to the students on e.tsi.lv under the module section. Also, additional materials, like publications on the internet, videos etc will be presented in e.tsi.lv.

During practical classes each student receives an individual task to perform. Each practical task should be completed and uploaded to e.tsi.lv (under specific practical task element), it will be checked by the teaching staff and feedback will be provided. Defence of the performed task is not required. The teaching staff note the task is passed or failed. Modern software such as R, Python or Mathcad will be used in practical classes (students' choice). As a demonstration of the idea of the algorithm studied in the current practical task, the code on the Mathcad is provided as the most formalised and similar to pseudocode.

Module Learning outcomes: On successful completion of this module students will achieve the following learning outcomes.

MO1 Apply computational mathematical models and methods for solving practical problems

MO2 Specify graphs in different ways

MO3 Solve various problems in graph theory, for example, the problem of finding the shortest path, the problem of finding the maximum flow.

MO4 Solve various problems in the theory of logical functions

MO5 Build a finite state machine and implement any algorithm using a Turing machine

MO6 Solve various problems in the theory of algorithms and formal systems

MO7 Use different modern software such as Mathcad, R or Python to solve simple problems.

Hours to be allocated: 120

Contact hours:

Independent study/self-guided study = 96 hours

Face-to-face learning = 64 hours

Total = 160

Reading list: The reading list for this module can be accessed at readinglists.uwe.ac.uk via the following link https://rl.talis.com/3/uwe/lists/3E565164-0661-0B67-D58F-7F3A4F8C1019.html?lang=en-gb&login=1

Part 4: Assessment

Assessment strategy: To assess the learning outcomes of this course, several types of activities are provided, which
include
1) two midterm tests (summative assessment)
2) laboratory works (summative assessment)
3) Laboratory works
4) examination (summative assessment).

Midterm tests are used as a formative type of knowledge assessment and are designed for assessment of the knowledge acquired by the student. This will allow students to pay attention to material that they have not mastered enough.

The course ends with an exam, which is aimed at assessing the theoretical knowledge and practical skills acquired by the student in the process of studying the course.

The students will be required to resit failed tasks during the resit period.

Assessment tasks:

Portfolio (First Sit)
Description: A portfolio of four practical assignments, which provides results of assignments execution.

Weighting: 40 %

Final assessment: No

Group work: No

Learning outcomes tested: MO1, MO2, MO3, MO4, MO5, MO7

Examination (First Sit)

Description: Final 2-hour written closed-book
examination which consists of two
parts: theoretical and practical.

Weighting: 60 %

Final assessment: Yes

Group work: No

Learning outcomes tested: MO1, MO2, MO3, MO4, MO5, MO6

Portfolio (Resit)

Description: A portfolio of practical assignments, which provides results of assignments execution.

Weighting: 40 %

Final assessment: No

Group work: No

Learning outcomes tested: MO1, MO2, MO3, MO4, MO5, MO7

Examination (Resit)

Description: Examination (2 hours)
written closed-book examination which consists of two parts: theoretical and practical.

Weighting: 60 %

Final assessment: Yes

Group work: No

Learning outcomes tested: MO1, MO2, MO3, MO4, MO5, MO6

Part 5: Contributes towards

This module contributes towards the following programmes of study:

Computer Science and Software Development {Double Degree} {Foundation} [TSI] BSc (Hons) 2022-23

Computer Science and Software Development {Double Degree} {Foundation} [TSI] BSc (Hons) 2022-23