Computer Discrete Mathematics

Syllabus

Requisites of the Course	
Cycle of Higher Education	*First cycle of higher education (Bachelor's degree)*
Field of Study	*12 Information Technologies*
Speciality	*121 Software engineering*
Education Program	*Computer Systems Software Engineering*
Type of Course	*Normative*
Mode of Studies	*full-time*
Year of studies, semester	*1 year (1 semester)*
ECTS workload	*5 credits (ECTS). Time allotment -150 hours, including 72 hours of classroom work, and 78 hours of self-study.*
Testing and assessment	*1 semester – Exam*
Course Schedule	*2 classes per week by the timetable http://rozklad.kpi.ua/*
Language of Instruction	*English*
Course Instructors	Lecturer: *D.Sc., Seniour Scientist, Anatoliy Sergiyenko, mobile +380688123376, email anat.srg@gmail.com, personal web page http://kanyevsky.kpi.ua* Teacher of practical work: *M.Sc., Anastasiia Molchanova, email an.ser.313kpi@gmail.com*
Access to the course	*https://bbb.comsys.kpi.ua/b/ana-3nr-kpr*

Outline of the Course

1. Course description, goals, objectives, and learning outcomes

The teaching of computer discrete mathematics at the university has the following aspects:

— Academic aspect. The discrete mathematics is an exact science that gives fundamental knowledge. The aim is to familiarize the students with the basic concepts and methods of discrete mathematics.

— Professional aspect. Knowledge of the discrete mathematics is basically the algorithms for solving most typical problems, the basis for their program implementation. The purpose of this course is to form the skills in the use of discrete mathematical methods.

— Intellectual and educational aspects. Studying the discrete mathematics contributes to the development of cognitive skills, which is an essential intellectual factor in the process of creating the new algorithms and programs.

As a result of studying this discipline, the following learning outcomes are achieved.

Competencies:

• ZK01 ability to abstract thinking, analysis and synthesis;

• ZK02 Ability to apply knowledge in practical situations

• ZK05 Ability to learn and master modern knowledge

• FK08 Ability to apply fundamental and interdisciplinary knowledge for successful solution of software engineering tasks

• FK14 Ability to algorithmic and logical thinking

Knowledge:

• PRN05 Know and apply relevant mathematical concepts, domain methods, system and object-oriented analysis and mathematical modeling for software development

Skills:

• solve problems of analysis and synthesis of models and algorithms based on the discrete mathematics concepts;

• think systematically and apply creative abilities to form new ideas.

2. Prerequisites and post-requisites of the course (the place of the course in the scheme of studies in accordance with curriculum)

Prerequisites, ie., disciplines, the study of which must precede the study of this discipline:

• basic level of English language proficiency not lower than B1.

Postrequisites, ie., disciplines, the study of which must be preceded by the study of this discipline:

• computer architecture;
• data bases;
• program engineering;
• system software;
• numeric methods;
• computer networks;
• artificial intelligence.

3. Content of the course

Theme 1: Introduction.

Subject of the discrete mathematics. Short history of mathematics.

Basics of discrete mathematics.

Objects. Sets. Mathematical structures. Relations and their properties. Operations. Algebraic structures. Lattices. Graphs. Algorithms.

Theme 2: Theory of sets.

Sets and operations with them. Multisets. Partitition and covering of sets. Power set. Algebra of sets. Problems with sets. Representation of sets in programs.

Relations.

Relation definitions. Direct product of sets. Binary relations. Operations with relations. Relation properties. Shuffle relations. Representation of relationships in programs. Closure of relations.

Functions.

Function concept. Injection, surjection, and bijection. Superposition of functions. Presentation of functions in the programs.

Lattices.

Order relation. Hasse diagram. Lattice definition. Operations with the lattice elements. Lexicographic order.

Theme 3. Logical functions.

Definition of a Boolean algebra. Boolean function definition and representation. Boolean functions of one and two arguments. Boolean function superposition. Boolean algebra of switching functions. Logical reduction functions. Implementation of the Boolean functions in computers.

Analytical representation of Boolean functions .

Canonical forms of Boolean function. Channon expansion. Zhegalkin algebra. Functionally complete systems of boolean functions. Boolean function duality principle.

Boolean function minimization

Disjunctive normal form. Conjunctive normal form. Quine method. Carnaugh maps. Partially defined boolean function minimization. Absolutely minimum normal form.

Combinatorial network synthesis

Combinatorial network synthesis process. Boolean function minimization during the network synthesis. Adder. Subtractor. Decoder. Multiplexor. Boolean functions in ROM.

Theme 4. Abstract finite state machines

Basic concepts of the discrete automaton. Miley, Moore, C-automata. Finite state machine (FSM) representation. Bubble diagram. FSM with the Boolean signals. Block diagram of the algorithm. FSMs in programming and computer engineering.

Linear FSMs

Concept of the linear group code. Cyclic codes. Counters, accumulators, multipliers with accumulators. Linear FSMs for operations with the cyclic codes. Hamming codes. Cyclic redundancy check codes. Quantum computer as linear FSM.

4. Coursebooks and teaching resources

Base text books

1. Graham, Ronald L.; Knuth, Donald E.; Patashnik, Oren. Concrete Mathematics - A foundation for computer science (2nd ed.). Reading, MA, USA: Addison-Wesley Professional. 1994. pp. xiv+657. ISBN 0-201-55802-5.

2. Haggard G., Schlipf J., Whitesides S. Discrete Mathematics for Computer Science. Brooks Cole, 2005. - 718 P. ISBN 0-534-49501-X.

3. Sergiyenko A., Molchanova A. Discrete Mathematics. Lesson book. https://kanyevsky.kpi.ua/en/for-students/student-books/

Additional text books

1. Gallier J. Discrete Mathematics for Computer Science, Some Notes. 2008. DOI: 10.1007/978-1-4419-8047-2. https://www.researchgate.net/publication/1922282_Discrete_Mathematics_for_Computer_Science_Some_Notes

2. Doerr A., Levasseur Ken. Applied Discrete Stuctures. University of Massachusetts Lowell. 2020. https://faculty.uml.edu/klevasseur/ads2/

3. Pass R., Tseng W-L. D. A Course in Discrete Structures. Cornell University. 2011. 153 p. https://www.cs.cornell.edu/~rafael/discmath.pdf

4. O'Donnel J. T., Hall C. V., Page R. L. Discrete mathematics using a computer (2.ed.). Springer, 2006. 441 p. https://www.researchgate.net/publication/220695320_Discrete_mathematics_using_a_computer_2_ed

Lesson slides are stored in https://kanyevsky.kpi.ua/en/for-students/

5. Methodology

The educational content of the discipline consists of lessons and practicum.

Lessons:

Theme 1: Introduction.

Lesson 1. Introduction.

Goal. Education materials. Rating system. Place of the discrete mathematics. Discrete mathematics directions. Short history of mathematics. Mathematical theory structure.

Lesson 2. Basics of discrete mathematics.

Objects. Complex constructive objects. Sets. Mathematical structures. Relations and their properties. Operations. Algebraic structures. Monoid, group, ring, field. Graphs.

Lesson 3. Algorithms.

History of algorithms. Turing machine. Solvability problem. Algorithm definitions. Markov's definition. Kolmogorov definition. Post-Turing definition. Graph algorithm model. Petri net. Algorithm block diagram. Knuth's model. Schönhage model. Aho-Ullman-Hopcroft model. Gandi model. Abstract state machine. Discrete mathematics structure.

Theme 2: Theory of sets.

Lesson 4. Sets.

Set presentation methods. Russel's paradox. Multisets. Set comparizon. Finite and infinite sets. Operations with sets. Partitition and covering of sets. Power set. Algebra of sets. Set operation properties, theorems.

Lesson 5. Problems with sets.

To simplify the expression. Equation solving. Representation of sets in programs. Bit scale. Sets in arrays. Sets in hash tables.

Lesson 6. Relations.

Ordered sets. Direct product of sets. Binary relations. *m*-ary relation. Relation kinds. Relation operations. Relation composition. Relation power. Relation properties.

Lesson 7. Relations (cont).

Representation of relations in programs. Relation kernel. Closure of relations. Relations and intelligence.

Lesson 8. Functions.

Function definition. Function concept. Total and partial functions. Function of *n* arguments. Injection, surjection, and bijection. Superposition of functions. Presentation of functions in the programs.

Lesson 9. Functions. Lattices.

Order relation. Hasse diagram. Lattice definition. Lattice algebra. Operations with the lattice elements. Lexicographic order.

Theme 3. Logical functions.

Lesson 10. Boolean algebra and logical functions.

Definition of a Boolean algebra. Boolean function definition. Axioms and theorems of the Boolean algebra. Boolean function superposition. Algebra of switching functions. The truth table. Geometric representation. Analytical representation. Fully and incompletely defined function. Boolean functions of one and two arguments. Superposition of Boolean functions. Boolean algebra of switching functions.

Lesson 11. Boolean algebra and logical functions (cont.).

Boolean algebra of switching functions. Formula simplifying. Logical functions of reduction. Boolean function implementation in computers. Boolean function graphic images. Canonical forms of the Boolean function.

Lesson 12. Boolean algebra and logical functions (end).

Shannon's expansion. Zhegalkin's algebra. Functionally complete systems of Boolean functions. Duality principle of the Boolean functions.

Lesson 13. Boolean function minimization.

Properties of DNF. Quine’s method. Karnaugh–Veitch method. Absolute minimal form of representation of Boolean functions. Library of elementary components.

Lesson 14. Combinational network synthesis.

Mapping Boolean equations in combinational networks. Combinational network in a given element basis. Adder and subtractor. Decoder and multiplexer.

Theme 4. Abstract finite state machines

Lesson 15. Abstract digital automata

Discrete automaton. Finite state machine (FSM). Miley automaton. Moore automaton. Miley automaton representations. FSM.

Lesson 16. Abstract digital automata

FSM with the Boolean signals. Automata in the computers. Stack machine. Concept of the operational and control automata. Method of the canonical synthesis of digital automata. Microprogramming method. FSM design using VHDL language. Functional network of the abstract computer.

Lesson 17. Linear automata

Linear automaton definition. A group of integers in the computers. Integer automata. Accumulator. Multiplier with accumulator. Linear sequential automata. The pseudo-random sequence generator.

Lesson 18. Linear automata (cont.)

The Galois field. The cyclic code. Cyclic redundancy check (CRC) method. Using Galois fields. Linear automata implementation in the quantum computers.

Practicum

Lesson 1.

Investigation of the Euclid's algorithm. Diophantine equations. Solving the problem of the Pythagorean triples.

Lesson 2.

Problems with the constructive objects. Sets. Problems with the primitive relations. Examples of monoid, ring, field. Examples of graphs.

Lesson 3.

Investigation of the Geron algorithm. Graph algorithm model. Examples of the algorithm representation. Modeling the Petri net.

Lesson 4.

Set presentation methods. Finite and infinite sets. Operations with sets. Partitition and covering of sets. Power set. Problems with sets. To simplify the expression.

Lesson 5.

Problems with sets. Equation solving. Bit scale.

Lesson 6.

Ordered sets. Direct product of sets. Binary relations. Relation operations. Relation composition.

Lesson 7.

Relation power. Relation properties.

Lesson 8.

Relation kernel. Closure of relations.

Lesson 9.

Problems with the function properties. Injection, surjection, and bijection. Superposition of functions.

Lesson 10.

Problems with the order relation, Hasse diagram, lattices, lattice algebra.

Lesson 11.

Problems with the axioms and theorems of the Boolean algebra, Boolean function superposition. Examples of the truth table, geometric representation, analytical representation. Problems with fully and incompletely defined function.

Lesson 12.

Problems with the Boolean algebra of switching functions. Formula simplifying. Logical functions of reduction. Boolean function graphic images. Canonical forms of the Boolean function.

Lesson 13.

Problems with the Shannon's expansion. Zhegalkin's algebra. Mapping the Boolean functions into different functionally complete systems of Boolean functions. Problems with the duality principle of the Boolean functions.

Lesson 14.

Properties of DNF, KNF. Function minimization by the Quine’s method, Karnaugh–Veitch method.

Lesson 15.

Absolute minimal form of representation of Boolean functions. Combinational network synthesis. Mapping Boolean equations in combinational networks. Combinational network in a given element basis.

Lesson 16.

Examples of the Miley, Moore automaton representations.

Lesson 17.

Problems with the canonical synthesis of the digital automata.

Lesson 18.

Module control work.

6. Self-study

The self-study includes the independent work of students and is as follows:

• preparation for lectures by studying the previous lecture material as well as literary sources on which the material of previous lectures is based (the list of sources and the list of sections is provided together with the lecture material);

• preparation for the practicum by the study of theoretical material needed to answer test questions and be ready to solve the problems;

• performing the tasks being given at the previous practicum lessons for the home work;

• performing the module calculating work;

• performing two home control works;

• writing the essay for the given thema devoted to some particular discrete mathematic problem solving.

The module calculating work is usually performed for two or three weeks; the home control works are performed in a week; the essay is written for a month.

Policy and Assessment

7. Course policy

The system of requirements for students:

• the student is obliged to attend lectures and practicum classes and actively work on mastering the material taught at them;

• at the lecture the lecturer uses his own presentation material;

• at the practicum classes the students solve the problems at the black board, defend their essays taking for that the score notes;

• modular test is written in a lecture using all available materials;

The starting rating (during the semester) consists of points that the student receives for:

- work in practical classes (18 classes);
- performance of two regular control works;
- performance of the modular control work.

Scoring criteria are:

Work on practical classes:

- active creative work - 2 points;
- fruitful work or completed homework - 1 point;
- passive work -0 points.

Execution of control work:

- the work is done flawlessly - 6 points;
- the work was performed with minor shortcomings - 5 points;
- the work is done with certain errors - 4-3 points;
- the work was performed with the number of errors approaching half of their maximum possible number - 2 points;
- the work is not credited (it is not completed or it has an excessive number of errors) - 0 points.

Execution of the modular control work:

- creative work - 12-11 points;
- the work is performed with minor shortcomings - 10-9 points;

- the work is done with certain errors - 8-7 points:

- the work is not credited (the task is not completed or there are huge errors) - 0 points.

Incentive -1-2 points are awarded for timely submission of settlement work for inspection.

The condition of the first attestations to receive at least 10 points (at the time of attestation). The condition of the second attestation is to receive at least 25 points (at the time of attestation).

The student can receive up to 6 incentive points for performing creative works from the credit module (compiling essays, participating in competitions, in research, etc.).

8. Monitoring and grading policy

At the first class the students are acquainted with the grading policy which is based on Regulations on the system of assessment of learning outcomes https://document.kpi.ua/files/2020_1-273.pdf. The student's rating in the course consists of points that he/she receives for participation in 18 practical classes (R1), incentive points (R2) and an exam (R3).

Rs=R1+R2+R3=100 points

As a result, the maximum average weight score is equal to:

16 classes x 2 points = 32 points
2 regular control works = 12 points
modular control work = 12 points
Incentive points = 6 points
Exam = 40 points

According to the university regulations on the monitoring of the student's academic progress (https://kpi.ua/document_control) there are two assessment weeks (attestation), usually during 7th/8th and 14th/15th week of the semester, when students take the Progress and Module tests respectively, to check their progress against the criteria of the course assessment policy.

Students whose final performance score is more than 30 are required to pass the exam. Students whose score is below 30 are not allowed to pass the exam.

At the exam, the students perform a written test. Each task contains two theoretical questions (tasks) and one practical one. The list of questions is given in the Recommendations for mastering the credit module. Each theoretical question (task) is evaluated with 15 points according to the following criteria:

- "excellent", full answer, not less than 90% of the required information), performed in accordance with the requirements for the level of "skills" - 15-14 points;

- "good", a sufficiently complete answer, not less than 75% of the required information, performed in accordance with the requirements for the level of skills or minor inaccuracies - 13-11 points;

- "satisfactory", incomplete answer, not less than 60% of the required information, performed in accordance with the requirements for the "stereotypical" level and some errors - 10-8 points;

- "unsatisfactory", the answer does not meet the conditions for "satisfactory" - 0 points.

The practical task is evaluated at 10 points according to the following criteria:

- "excellent", complete, error-free problem solving - 10 points;

- "good", complete problem solving with minor inaccuracies or shortcomings - 9-8 points;

- "satisfactory", the task is performed with errors - 7-5 points;
- "unsatisfactory", the answer does not meet the conditions for "satisfactory" - 0 points.

The final performance score R_S is adopted by university grading system as follows:

Score	*Grade*
100-95	Excellent
94-85	Very good
84-75	Good
74-65	Satisfactory
64-60	Sufficient
Below 60	Fail
Course requirements are not met	Not Graded

9. Additional information about the course

• Questions in the exam
repeat the themes of the lessons given in the chapter 5 of this syllabus.
• Questions in the modular control work are the test questions as follows:
1. Reasoning in mathematics is correct if it is
a.logically perfect,
b. coincide with practical results,
c.agree with the proved theorems.
2. There are sets {0,1,2} and {2,3}. Their Cartesian product is equal to
a. {(0,2), (0,3),(1,2),(1,3),(2,2),(2,3)}
b. {0, 0, 2, 3, 4, 6}
c. {{0,2}, {0,3},{1,2},{1,3},{2,2},{2,3}}
3. The relation R is symmetric if
a. xR y = true, then y Rx = true for arbitrary x,y
b. xR x = true for arbitrary x,
c. xR y = true, then y Rx = true, but xR x = false for arbitrary x,y.
4. 85.   The symmetric difference of the sets {1, 2, 6, 7} and {2, 3, 6} is
a. {1, 3, 7}
b. {1, 7}
c. { 1, 2, 6, 7}
5. 173. The Boolean function f (a, b, c) is dual if
a. f(a,b,c) = ~f (~a,~b,~c)
b. f(a,b,c) = f(c,b,a)
c. f(a,b,c) = f (~a,~b,~c).

Syllabus of the course

Is designed by teacher D.Sc., Senior Scientist, Anatolii Sergiyenko

Adopted by Department of Computing Engineering (protocol № 10 , 25.05.2022)

Approved by the Faculty Board of Methodology (protocol № 10 , 13.06. 2022)