COURSE UNIT DESCRIPTION

Course unit title	Course unit code
Algorithm theory	

Lecturer(s)	Department where the course unit is delivered
Coordinator: Adomas Birštunas **Other lecturers:**	Department of Informatics Faculty of Mathematics and Informatics Vilnius University

Cycle	Type of the course unit
1st (BA)	Compulsory

Mode of delivery	Semester or period when the course unit is delivered	Language of instruction
Face-to-face	2 semester	Lithuanian

Prerequisites
Prerequisites: Discrete Mathematics.

Number of credits allocated	Student‘s workload	Contact hours	Individual work
5 ETCS	132	71	61

Purpose of the course unit: programme competences to be developed
Purpose of the course unit – to introduce computer science fundamentals, to teach to understand and analyze complex problems, to teach students to apply nontrivial theoretical knowledge, to teach to learn. ***Generic competences:*** • Life-long learning *(GK2)*. ***Specific competences:*** • Knowledge and skills of underlying conceptual basis *(SK4).*

Learning outcomes of the course unit: students will be able to	Teaching and learning methods	Assessment methods
demonstrate knowledge of theoretical computer science fundamentals: to identify the main approaches of algorithm formalizations and their impact on computer science, to identify how complexity of the problems are calculated, to discuss about tractable and intractable problems.	problematical teaching, explanation, analysis of complex samples, group discussion	Examination in a written form, theorem proving, solving problems applying theory, mini tests
understand formal information: to read and use information that is provided by means of formal methods and various notations.		
analyze and assess problems and their solutions: to determine problem decidability, to differ complexity of the problem and complexity of the problem solving algorithm, to calculate and compare time and space complexity of the problems and their solutions.		
apply algorithm theory knowledge: to apply theorems in problem solving, to conclude and argue using theoretical knowledge in the particular situations.	problematical teaching, explanation, analysis of complex samples, problem solving, wrong solution analysis, group discussion	
explain proofs of the theorems.		

solve various problems of the algorithm theory: Turing machines, finite-state machines, lambda calculus, recursive functions.	explanation, problem solving, wrong solution analysis	Solving standard problems, solving problems applying theory, mini tests
apply different calculus for propositional logic.		
understand and apply various calculi for both mathematical logic and other disciplines.		

Course content: breakdown of the topics	**Contact hours**							**Individual work: time and assignments**	
	Lectures	Tutorials	Seminars	Practice	Laboratory work (LW)	*Tutorial during Practice*	**Contact hours**	**Individual work**	**Assignments**
1. Algorithm formalization; the Church‘s thesis	2						**2**		
2. Hilbert-style calculus for propositional logic	1			2		*3*	**3**	**2**	Problem solving (Hilbert calculus without Deduction theorem) (2)
3. Deduction theorem	2			2			**4**	**3**	Problem solving (Hilbert calculus with Deduction theorem) (1), analysis of the theorem proofs (2)
4. Sequent calculi	1			4			**5**	**3**	Problem solving (sequent calculi) (2), scientific literature reading (introduction to different sequent calculi) (1)
5. Resolution method	3			4			**7**	**4**	Problem solving (resolution method) (2), analysis of the theorem proofs (2)
6. Turing machines and their variants	2			2		*1*	**4**	**3**	Problem solving (one-tape and multi-tape Turing machines) (2), analysis of the theory application for problem solving (Turing machines) (1)
7. Finite automata	1			2			**3**	**3**	Problem solving (finite automata) (2), analysis of the theorem proofs (1)
8. Pairing function	1						**1**	**1**	Problem solving (Cantor tuple functions) (1)
9. The halting problem	2						**2**	**3**	Analysis of the theorem proofs (2), scientific literature reading (outcomes of the halting problem) (1)
10. Algorithms complexity	1			2			**3**	**1**	Problem solving (time and space complexity of the Turing machines) (1)
11. The lambda calculus	2			4		*3*	**6**	**3**	Problem solving (beta reduction, lambda calculus) (2), analysis of the theory application for problem solving (lambda calculus) (1)
12. Primitive recursive functions	2			4			**6**	**4**	Problem solving (primitive recursive functions) (3), analysis of the theorem proofs (1)
13. Partial recursive functions	2			2			**4**	**3**	Problem solving (partial recursive functions) (3)
14. Recursive and recursively enumerable sets	3			2			**5**	**3**	Analysis of the theorem proofs (3), analysis of the theory application for problem solving (recursive and recursively enumerable sets) (1)
15. Ackermann functions	2					*1*	**2**	**2**	Analysis of the theorem proofs (2)
16. Universal functions	4			2			**6**	**5**	Analysis of the theorem proofs (3), analysis of the theory application for problem solving (universal functions, function graph, function extension) (2)

17. Post canonical system	1						**1**		
18. Colloquium and preparation		2					**2**	**6**	Content revision, problem solving.
19. Examination and preparation		5					**5**	**12**	Content revision, analysis of the theorem proofs, analysis of the theory application for problem solving, problem solving.
Total	**32**	**7**		**32**		*8*	**71**	**61**	

Assessment strategy	**Weight %**	**Deadline**	**Assessment criteria**
Problem solving during practice	4,4	During year of study	During practice it is possible to collect up to 0,5 accumulative grade-points. Grade-points are given for correct and fast problem solving during practice.
Mini tests during practice	7,9	During year of study	Mini test consist of 1-5 questions with given possible answers. Mini test duration about 5 minutes. Mini test is passed if more than 50 % answers are correct. Student get 0,1 grade point for every passed mini test. 9 mini tests should be organized during year of study.
Colloquium	26,3	April - May	During colloquium it is possible to collect up to 3 accumulative grade-points. Colloquium consists of 3 - 4 tasks (standard problems). Every task is assessed from 0,5 to 1,2 grade-point.
Examination in written form (if necessary, additional questions in oral form)	61,4	June	Student can take part in the examination only if at least one of the following conditions are satisfied: - student passed at least 6 mini tests, - student’s colloquium work was assessed at least by 1 grade-point. During examination it is possible to collect up to 7 accumulative grade-points. Examination consists of 3 - 6 tasks: - problem with theory application (required, 2,5 - 4,5 grade-points), - full or partial theorem proof (required, 1 - 3 grade-points), - definitions and theorems (optional, 0 - 1,2 grade-points), - standard problems (optional, 0 - 1,2 grade-points). The tasks are prepared so that: - 2 grade-points are given for the practical part (part of the problem with theory application and standard problems), - 5 grade-points are given for the theory part (full or partial theorem proofs, definitions, theorems and part of the problem with theory application).

Author	**Publishing year**	**Title**	**Number or volume**	**Publisher or URL**
Required reading				
S. Norgėla	2007	Logika ir dirbtinis intelektas		Vilnius, TEV
Recommended reading				
Adomas Birštunas	2012	Lecture Notes of the Algorithm Theory		http://www.mif.vu.lt/~adomas/konspektai/AlgorithmTheory.pdf
R. Lassaigne, M. de Rougemont	1999	Logika ir algoritmų sudėtingumas		Vilnius, Žodynas
S. Norgėla	2004	Matematinė logika		Vilnius, TEV
Lawrence C. Paulson	2007	Logic and Proof		Computer Laboratory, University of Cambridge
Achim Jung	2007	The Halting Problem (adopted by Volker Sorge, Steve Vickers in 2012)		The University of Birmingham

Lance Fortnow	2009	The status of the P versus NP problem	No 9, pp. 78-86	Communications of the ACM 52,
Peter Selinger	2007	Lecture Notes on the Lambda Calculus		Department of Mathematics and Statistics, Dalhousie University, Halifax, Canada