Models of Computation

Course Description

Basic Information

Field of Study : Analytical Computer Science

Path : -

Organizational Unit : Faculty of Mathematics and Computer Science

Education Level : first-cycle studies

Form of Studies : full-time studies

Study Profile : general academic

Obligatory Status : mandatory

Education Cycle : 2022/23

Course Code : UJ.WMIIANS.180.03346.22

Languages of Instruction : Polish

Course Related to Scientific Research : Yes

Disciplines : Computer Science

ISCED Classification : 0613 Software and applications development and analysis

USOS Code : WMI.TCS.MO.OL

Course Coordinator

Michał Wrona

Course Instructors

Michał Wrona, Maciej Ślusarek

Period Semester 4	Form of verification of learning outcomes exam Form of instruction and hours lecture: 30 tutorials: 30	Number of ECTS credits 6.0

Learning Outcomes for the Course

Code	Effects in terms of	Field-specific learning outcomes	Verification methods
Knowledge – The student knows and understands:			
W1	what a formal language is and the basic facts about formal languages,	IAN_K1_W02	written exam, graded credit
W2	the basic tools: minimization of finite automata, mutual simulations of equivalent models, pumping lemmas, diagonal method	IAN_K1_W11	written exam, graded credit
W3	the concept of undecidability and basic computational complexity classes	IAN_K1_W02, IAN_K1_W11	written exam, graded credit
Skills – The student can:			
U1	define a model describing a formal language and place the class of languages described by the defined model in the language hierarchy	IAN_K1_U01, IAN_K1_U04, IAN_K1_U11, IAN_K1_U17, IAN_K1_U21	written exam, graded credit
U2	select an appropriate model for the problem being solved	IAN_K1_U01, IAN_K1_U04, IAN_K1_U11, IAN_K1_U17, IAN_K1_U21	written exam, graded credit
U3	construct finite automata, context-free grammars, and Turing machines	IAN_K1_U09	written exam, graded credit
Social competences – The student is ready to:			

Code	Effects in terms of	Field-specific learning outcomes	Verification methods
K1	prove theorems in computability theory; understand the profound implications of computability theory for broadly defined science and philosophy, e.g., knows and understands Church's thesis	IAN_K1_K01	written exam, graded credit

ECTS Credit Balance

Student activity form	Average number of hours* dedicated to completed activity types
lecture	30
tutorials	30
preparation for tutorials	90
exam preparation	28
participation in exam	2

Total student workload	Number of hours 180	ECTS credits 6.0

* hour (lesson) means 45 minutes

Program Content

No.	Program Content	Learning outcomes for the course
1.	1. Formal languages and their properties. 2. Finite automata and regular expressions. 3. Pumping lemma and Myhill-Nerode theorem. 4. Minimization of finite automata. 5. Properties of regular languages; problems and algorithms. 6. Grammars and context-free languages; stack automata. 7. Pumping lemma for context-free languages and properties of context-free languages. 8. Deterministic stack automata. 9. Turing machines; recursive and recursively enumerable languages. 10. Universal Turing machine; halting problem and undecidable problems, Rice's theorem. 11. Basics of computational complexity: P, NP, coNP, PSPACE.	W1, W2, W3, U1, U2, U3, K1

Extended Information

Teaching Methods:

conventional lecture, problem solving, subject tutorials

Type of classes	Forms of credit	Course completion conditions
lecture	written exam	positive grade on the exam, preceded by admission to it based on a positive grade from tutorials
tutorials	graded credit	solving problems at the board, two tests

Prerequisites and Additional Requirements

Formal Methods in Computer Science

Literature

Required

1. J. Hopcroft, J. Ullman, "Introduction to Automata Theory, Languages, and Computation" (1st ed.), Addison-Wesley, 1979