SYLLABUS

Computational Logic

University year 2025-2026

1. Information regarding the programme

1.1. Higher education institution	**Babeş-Bolyai University, Cluj-Napoca**
1.2. Faculty	**Faculty of Mathematics and Computer Science**
1.3. Department	**Department of Computer Science**
1.4. Field of study	**Computer Science**
1.5. Study cycle	**Bachelor**
1.6. Study programme/Qualification	**Computer Science in English**
1.7. Form of education	**Full time**

2. Information regarding the discipline

2.1. Name of the discipline	**Computational Logic**					Discipline code	**MLE5055**
2.2. Course coordinator					**Lecturer Ph.D. Lupea Mihaiela**		
2.3. Seminar coordinator					**Lecturer Ph.D. Lupea Mihaiela**		
2.4. Year of study	1	2.5. Semester	1	2.6. Type of evaluation	E	2.7. Discipline regime	Compulsory

3. Total estimated time (hours/semester of didactic activities)

3.1. Hours per week	**4**	of which: 3.2 course	**2**	3.3 seminar/laboratory/project	**2**
3.4. Total hours in the curriculum	**56**	of which: 3.5 course	**28**	3.6 seminar/laboratory/project	**28**
Time allotment for individual study (ID) and self-study activities (SA)					**hours**
Learning using manual, course support, bibliography, course notes (SA)					25
Additional documentation (in libraries, on electronic platforms, field documentation)					10
Preparation for seminars/labs, homework, papers, portfolios and essays					30
Tutorship					9
Evaluations					20
Other activities:					

3.7. Total individual study hours	**94**
3.8. Total hours per semester	**150**
3.9. Number of ECTS credits	**6**

4. Prerequisites (if necessary)

4.1. curriculum	-
4.2. competencies	-

5. Conditions (if necessary)

5.1. for the course	-
5.2. for the seminar /lab activities	-

6.1. Specific competencies acquired [1]

Professional/essential competencies	• Use of theoretical foundations of computer science as well as of formal models. • Use of artificial intelligence concepts and techniques to solve real-world problems.
Transversal competencies	• Application of organized and efficient work rules, of responsible attitudes towards the didactic-scientific field, to bring creative value to own potential, with respect for professional ethics principles and norms. • Use of efficient methods and techniques to learn, inform, research and develop the abilities to bring value to knowledge, to adapt at the requirements of a dynamical society and to communicate efficiently in Romanian language and in an international language.

6.2. Learning outcomes

Knowledge	• The graduate is familiar with international academic research databases and digital libraries (Web of Science, ACM Digital Library, IEEE Xplore, Springer, Elsevier, CiteSeerX, etc.). • The graduate has the necessary knowledge for literature review.
Skills	• The graduate is able to define/identify/understand research problems in computer science. • The graduate is able to present and explain methods, algorithms, paradigms and techniques used in various branches of computer science. • The graduate has the necessary skills to use research support tools.
Responsibility and autonomy:	• The graduate has the ability to understand and communicate information effectively. • The graduate has the knowledge to select and use appropriate instructional procedures to facilitate the process of knowledge assimilation.

7. Objectives of the discipline (outcome of the acquired competencies)

7.1 General objective of the discipline	• To introduce the **logical foundations of computer science**: propositional calculus and predicate calculus, theorem proving methods, Boolean algebras and Boolean functions. The connection with logic circuits is presented. • To introduce internal **representations** of integer and real numbers.

[1] One can choose either competences or learning outcomes, or both. If only one option is chosen, the row related to the other option will be deleted, and the kept one will be numbered 6.

7.2 Specific objective of the discipline	• Understand how integer and real numbers are represented and manipulated internally by a computer. • Understand the functionality of some simple logic circuits from the hard component of a computer. • Identify and apply appropriate logical (propositional/predicate) models and proof methods to solve real problems in the domain of human and mathematical reasoning.

8. Content

8.1 Course	Teaching methods	Remarks
Course 1. Numeration systems 1. Definitions, representation of numbers in a base **b.** 2. Conversions between bases using the *substitution method* and the *method of successive divisions/multiplications* for integer and rational numbers. 3. Rapid conversions (bases 2,4,8,16).	Exposure: description, explanation, examples, discussion of case studies	
Course 2. Internal representations of numbers 1. Representation of unsigned integers, operations. 2. Representation of signed integers: direct code, inverse code, complementary code, operations. 3. Fixed-point and floating-point representations of real numbers.	Exposure: description, explanation, examples, discussion of case studies	
Course 3. **Propositional logic – syntax and semantics** 1. Syntax: connectives, formulas. 2. Semantics: interpretation, model, consistent formula, inconsistent formula, tautology, logical consequence, truth table for a formula. 3. Laws (logical equivalences): DeMorgan, absorption, commutativity, associativity, distributivity, idempotency. 4. Clauses and normal forms: conjunctive normal form (CNF) and disjunctive normal form (DNF), algorithm for transformation of a formula into DNF and CNF.	Exposure: description, explanation, examples, discussion of case studies, debate, dialog	
Course 4. Propositional logic – formal system 1. Formal (axiomatic) system associated to propositional logic, deduction, theorem. 2. Theorem of deduction and its consequences. 3. Properties of propositional logic.	Exposure: description, explanation, examples, discussion of case studies, proofs, dialog	
Course 5. Predicate (first-order) logic 1. Syntax: connectives, quantifiers, terms, atoms, formula, clause, literal, closed formula, free formula, the formal (axiomatic) system. 2. Semantics of predicate logic: interpretation, model, valid formula, consistent formula, inconsistent formula, logical consequence. 3. Properties of predicate logic.	Exposure: description, explanation, examples, discussion of case studies, dialog	

Course 6. Semantic tableaux method – a refutation proof method for propositional/predicate logic. 1. Classes of formulas, decomposition rules, branch (open, closed), construction of a semantic tableau. 2. Build the models and anti-models of a propositional/predicate formula from its semantic tableau.	Exposure: description, explanation, examples, discussion of case studies, proofs	
Course 7. Resolution in propositional logic 1. Resolution as a formal system. 2. Strategies of resolution: level saturation, set-of–support. 3. Refinements of resolution: lock resolution, linear resolution.	Exposure: description, explanation, examples, discussion of case studies	
Course 8. Resolution in predicate logic 1. Normal forms: prenex, Skolem and clausal forms. 2. Substitutions and unifications. 3. Predicate resolution – formal system. 4. Refinements of predicate resolution.	Exposure: description, explanation, examples, discussion of case studies, proofs	
Course 9. Modeling the common-sense human reasoning and mathematical reasoning using propositional and predicate logics.	Exposure: explanation, examples, discussion of case studies	
Course 10. Boolean algebras and Boolean functions **1.** Boolean algebras: definitions, properties, principle of duality, examples; **2.** Boolean functions: definitions, maxterms, minterms, the canonical disjunctive/conjunctive forms, transformation. **3.** Maximal and central monoms, factorization.	Exposure: description, explanation, examples, discussion of case studies, dialog, debate	
Course 11. Simplification of Boolean functions (I) 1. Veitch-Karnaugh diagrams method for functions of 2-3-4 variables. **2.** The dual simplification algorithm for the canonical conjunctive form.	Exposure: description, explanation, examples	
Course 12. Simplification of Boolean functions (II) 1. Quine's analytical method. 2. Moisil's algebraic method.	Exposure: description, explanation, examples, discussion of case studies	
Course 13. Logic circuits 1. Definitions, representations for basic gates and derived gates. 2. Logic circuit analysis and synthesis. 3. Comparator, adder, subtractor, encoder, decoder.	Exposure: description, explanation, examples, discussion of case studies	
Course 14. Combinational logic circuits - examples	Exposure: description, explanation, examples	

Bibliography

1. M. Ben-Ari: Mathematical Logic for Computer Science, Ed. Springer, 2001.
2. F. Boian, Bazele Matematice ale Calculatoarelor, Editura Presa Universitara Clujeana, 2002 – library.
3. M. Cocan, B. Pop: Bazele matematice ale sistemelor de calcul, Editura Albastra, Cluj-Napoca, 2001 – UBB library.
4. M.Fitting: First-order logic and Automated Theorem Proving, Ed.Springer Verlag, 1996.
5. M.Huth, M. Ryan: Logic in Computer Science - Modelling and Reasoning about Systems, 2nd edition, Cambridge, 2004.
6. D.W. Loveland: Automated Theorem Proving: a Logical Basis, North Holland, 2014.
7. M. Lupea, A. Mihis: Logici clasice şi circuite logice. Teorie şi exemple, ediţia 3, Editura Albastra, Cluj-Napoca, 2011.

8. M. Lupea, A. Mihis: A Computational Approach to Classical Logics and Circuits, Editura Presa Universitară Clujeană, Cluj-Napoca, 2016.
9. Mihaela Malita, Mircea Malita, Bazele Inteligentei Artificiale, Vol. I, Logici propozitionale, Ed. Tehnica, Bucuresti, 1987.
10. L.C. Paulson: Logic and Proof, Univ. Cambridge, 2000, on-line course.
11. M. Possega: Deduction Systems, Inst. of Informatics, 2002, on-line course.
12. A.Wasilewska: Logics for Computer Science: Classical and Non-Classical, Springer, 2021, e-book.

8.2 Seminar / laboratory	Teaching methods	Remarks
Seminar 1. Exercises Operations (addition, subtraction, division, multiplication) in different numeration bases.	Dialogue, case studies, examples	Attendance to seminars is mandatory for at least 75%.
Seminar 2: Exercises 1. Conversions between bases for integer and rational numbers using the methods: substitution, successive divisions/multiplications. 2. Rapid conversions between bases: 2,4,8,16.	Dialogue, case studies, examples	
Seminar 3. Exercises 1. Representation of signed integers: direct code, inverse code, complementary code, operations. 2. Representations of real numbers: fixed-point and floating-point representations.	Dialogue, case studies, examples	
Seminar 4. Exercises: 1. Using the truth table, decide whether a formula is consistent/ tautology/inconsistent or not,. Write all the models/anti-models of a propositional formula. 2. Transform a formula into their normal equivalent forms (DNF, CNF) and using these forms decide the validity or inconsistency of a formula.	Dialogue, debate, case studies, examples, students presentations	
Seminar 5. Exercises 1. Apply the theorem of deduction to prove the syllogism rule, separations of premises rule, reunion of premises rule. 2. Using the axiomatic system prove that a propositional formula is a theorem.	Dialogue, debate, case studies, examples, proofs, students presentations	
Seminar 6. **One hour – midterm exam: written paper with subjects from courses 1-2 and seminars 1-3.** **Exercises - predicate logic** 1. Transform natural language sentences into predicate formulas. 2. Build models and anti-models for a predicate formula.	Dialogue, debate, case studies, examples, students presentations	Attendance to the midterm exam is mandatory
Seminar 7. Exercises – semantic tableaux method (I) 1. Build the semantic tableau of a propositional formula, write all its models and anti-models. 2. Solve the decision problems in propositional logic.	Dialogue, debate, case studies, examples, students presentations	
Seminar 8. Exercises – semantic tableaux method (II) 1. Using the semantic tableaux method solve the decision problems in predicate logic. 2. From a semantic tableau of a predicate formula build the models of that formula.	Dialogue, debate, case studies, examples, students presentations	

Seminar 9. Exercises – propositional resolution (I) 1. Using resolution check the inconsistency of a set of propositional clauses. 2. Check whether a propositional formula is a theorem/ deductible from a set of formulas using resolution or one of its strategies.	Dialogue, debate, case studies, examples, students presentations	
Seminar 10. Exercises – propositional resolution (II) 1. Apply the refinements of resolution and combinations of strategies and refinements to solve the decisions problems in propositional logic. 2. Details regarding the implementation of lock resolution and linear resolution.	Dialogue, debate, case studies, examples, students presentations	
Seminar 11. Exercises – predicate resolution 1. Build the prenex, Skolem and clausal normal forms of a predicate formula. 2. Compute the most general unifier of two or more atoms. 3. Apply resolution and its refinements to solve the decision problems in predicate logic.	Dialogue, debate, case studies, examples, students presentations	
Seminar 12. Exercises – Boolean functions (I) 1. Build the canonical forms for a Boolean function. 2. Apply Veitch-Karnaugh diagrams method to simplify functions with 2, 3 and 4 variables.	Dialogue, debate, case studies, examples, students presentations	
Seminar 13. Exercises - Boolean functions (II) 1. Apply Quine's method and Moisil's method to simplify Boolean functions. 2. Implementation of the corresponding logic circuit.	Dialogue, debate, case studies, examples, students presentations	
Seminar 14. Exercises – logic circuits 1. Implement a simplified combinational circuit for the 7-segments electronic display. 2. Implement the simplified combinational circuits for the conversion between BCD and Gray codes.	Dialogue, debate, case studies, examples, students presentations	

Bibliography

1. W.Bibel: Automated theorem proving, View Verlag, 1987.
2. Cl.BENZAKEN: Systeme formels. Introduction a la logique, ed.Masson, 1991.
3. J. Harrison: Handbook of Practical Logic and Automated Reasoning, Cambridge University Press, 2009.
4. D.Tatar: Inteligenta artificiala: demonstrare automata de teoreme si NLP, Ed. Microinformatica, 2001.
5. (ed) A.Thayse: From standard logic to Logic Programming, Ed. J.Wiley, vol1(1989), vol2,vol3(1990).
6. A.B Marcovitz: Introduction to Logic Design, Mc.Graw-Hill Higher Education, 2005.

9. Corroborating the content of the discipline with the expectations of the epistemic community, professional associations and representative employers within the field of the program

- The course respects the IEEE and ACM Curricula Recommendations for Computer Science studies;
- The course exists in the studying program of all major universities in Romania and abroad;
- The content of the course offers a theoretical base for the applicative direction of building automated proof systems useful in mathematics, software engineering, intelligent agents, robotics, natural languages and computer vision.

10. Evaluation

Activity type	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Percentage of final grade
10.4 Course	- Know the basic principles of the domain;	Written paper (regular session) with subjects from courses 3-13.	60%

	- Apply the course concepts, methods and algorithms in problem solving.		
	- Know to perform operations and conversions in different numeration bases; - Know to represent integer and real numbers.	Midterm exam - written paper (seminar 6 - one hour) with subjects from courses 1-2 and seminars 1-3.	20%
10.5 Seminar/laboratory	Solve at home and present during the seminars exercises from an existing benchmark of problems	Seminars' activity: responses and individual presentations of solved exercises.	20%
	Exercises: modelling reasoning using propositional/predicate logic.	Optional homework (can increase the final grade).	10%
10.6 Minimum standard of performance			
• At least grade 5 (from a scale of 1 to 10) at written papers and seminar activity.			

11. Labels ODD (Sustainable Development Goals)[2]

Not applicable.

Date:	Signature of course coordinator	Signature of seminar coordinator
15.04.2025	Lect. PhD. Mihaiela LUPEA	Lect. PhD. Mihaiela LUPEA

Date of approval:	Signature of the head of department
...	Assoc. Prof. PhD. Adrian STERCA

[2] Keep only the labels that, according to the *Procedure for applying ODD labels in the academic process*, suit the discipline and delete the others, including the general one for *Sustainable Development* – if not applicable. If no label describes the discipline, delete them all and write „*Not applicable*.".