

Master of Computer Science

Programme coordinator: prof. RNDr. Roman Barták, Ph.D.

Study started in 2015 and later

1 General Information

The study programme Master of Computer Science consists of the following study branches and their specializations:

- Discrete Models and Algorithms
 - Discrete mathematics and algorithms
 - Geometry and mathematical structures of computer science
 - Optimization
- Theoretical Computer Science
- Software and Data Engineering
 - Software engineering
 - Software development
 - Web engineering
 - Database systems
 - Big data processing
- Software Systems
 - System programming
 - Dependable systems
 - High performance computing
- Computational Linguistics
 - Computational and formal linguistics
 - Statistical methods and machine learning in computational linguistics
- Artificial Intelligence
 - Intelligent agents
 - Machine learning
 - Robotics
- Computer Graphics and Game Development
 - Computer graphics
 - Computer game development

While your study branch has been specified already in your application, the choice of your specialization (when applicable) is up to your later decision (when enrolling for the state final exam).

Assumed knowledge

It is assumed that an incoming student has a sufficient knowledge of mathematics, theoretical computer science, and programming. In particular, students are

expected to have a good knowledge of mathematics at the level of our bachelor courses NMAI054 Mathematical Analysis I, NMAI055 Mathematical Analysis II, NMAI059 Probability and Statistics, NMAI062 Algebra I, and NOPT048 Optimization Methods.

Students are also expected to have knowledge equivalent to the courses NDMI002 Discrete Mathematics, NTIN060 Algorithms and Data Structures I, NTIN061 Algorithms and Data Structures II, NTIN071 Automata and Grammars, and NAIL062 Propositional and Predicate Logic. Knowledge from these courses is also expected for the state final exam.

We also expect students to have good knowledge of programming at least at the level of our bachelor courses NPRG030 Programming I, NPRG031 Programming II.

Students missing knowledge in some of the above-mentioned areas are advised to consider taking the relevant bachelor courses in the first year of their Master's studies. Please do not hesitate to contact the programme coordinator in case of doubt.

Software Project

One of the obligatory courses for students of study branches Software and Data Engineering, Software Systems, and Computer Graphics and Game Development is NPRG023 Software Project, a team software project. For students of study branches Computational Linguistics, Artificial Intelligence this course is elective and highly recommended. The goal of this subject is to practice team work in large software projects lasting typically around 9 months. The work on the project is finished by public presentation.

Students can enroll in the courses NPRG023 Software Project and NPRG027 Course Credit for Project anytime, not just in the usual enrollment period of a given academic year. Nevertheless, each student can try to complete these courses at most twice during the studies.

State Final Exam

The state final exam consists of two parts: a defense of the Master's (diploma) thesis and an oral examination. The student can enroll for each part separately. To finish the studies, both parts of the state final exam must be completed successfully.

Requirements to enroll for the state final exam

- obtaining at least 120 credits,
- passing all obligatory courses of a given branch and specialization,
- obtaining a given number of credits from the elective courses of a given branch and specialization,
- submitting the Master's thesis by the specified deadline (for defence of the Master's thesis).

Master's (diploma) thesis

Students are advised to select the topic of their Master's (diploma) thesis at the end of the first semester. The faculty departments offer many topics for Master's theses each year and students may also suggest their own topics. We recommend selecting the topic of your thesis primarily from those offered by the department that coordinates your study branch. If you prefer a topic offered by another department or your own topic, please consult the coordinator of your study branch.

After the topic is assigned, the student enrolls in the following obligatory courses:

Code	Subject	Credits	Winter	Summer
NSZZ023	Diploma Thesis I	6	—	0/4 C
NSZZ024	Diploma Thesis II	9	0/6 C	—
NSZZ025	Diploma Thesis III	15	—	0/10 C

Course credits for these courses are appointed by the thesis advisor on the basis of student's work on the thesis. We suggest the students to discuss with their advisor the expected amount of work and the milestones for each of these courses. All three courses can be enrolled in both winter and summer semesters.

Oral examination

The oral part of the state final exam has the same structure for all study branches. The student is examined from two obligatory areas covering the theoretical foundations of computer science (complexity, computability, and data structures) and from three areas specific to a given study branch and selected specialization. The student will select these three examination areas when registering for the final exam. Please note that some study branches have additional obligatory examination areas and some other restrictions might apply.

Note that not all the courses are available in English every year. We recommend students to contact the study branch coordinator for the selected study branch and discuss individual study plans prior the beginning of each semester.

The obligatory examination areas:

1. Foundations of complexity and computability

Computational models (Turing machines, RAM). Recursive and recursively enumerable functions. Algorithmically undecidable problems (Halting Problem). Non-deterministic computational model. Basic complexity classes and their relationship. Hierarchy theorems. NP-complete problems, Cook-Levin Theorem. Pseudo-polynomial algorithms, strong NP-completeness. Approximation algorithms and schemes.

Recommended courses

Code	Subject	Credits	Winter	Summer
NTIN090	Introduction to Complexity and Computability	5	2/1 C+Ex	—

2. Data structures

Search trees ((a,b)-trees, Splay trees). Heaps (regular, binomial). Hashing, handling collisions, universal hashing, selecting a hash function. Analysis of worst-case, amortized-case and expected complexity of data structures. Behavior and analysis of data structures on systems with memory hierarchy.

Recommended courses

Code	Subject	Credits	Winter	Summer
NTIN066	Data Structures I	5	2/1 C+Ex	—

2 Degree Plans - Discrete Models and Algorithms

Coordinated by: Department of Applied Mathematics

Study branch coordinator: Doc. RNDr. Martin Klazar, Dr.

Specializations:

- Discrete mathematics and algorithms
- Geometry and mathematical structures of computer science
- Optimization

The study branch Discrete Models and Algorithms provides education in the area of discrete (meaning non-continuous) mathematical structures used in Computer Science. It deals also with combinatorial (and other) algorithms and with modeling phenomena and processes by means of such structures and algorithms. In the specialization Optimization it puts emphasis on solid grasp of various kinds of optimization. The study branch enables to its graduates to be in contact with current scientific results and ideally it prepares them for independent research activity.

The graduate knows in depth discrete mathematics and discrete structures used in computer science and can model, using algorithms, various phenomena and processes. According to chosen specialization the graduate has advanced knowledge in one or more of the areas: combinatorics and graph theory, random techniques and methods in discrete mathematics and algorithms, algebraic and topological methods, and finally optimization of various kinds. The graduate can use this knowledge in research when solving difficult theoretical and practical questions in the area of applied mathematics and computer science, in technical and economical practice, and in interdisciplinary research. The graduate can work in research and development in either academia or industry in any position requiring logical reasoning, analytical capabilities, an algorithmic approach, and the exploitation of modern methods of computer science.

2.1 Obligatory courses

Code	Subject	Credits	Winter	Summer
NTIN090	Introduction to Complexity and Computability	5	2/1 C+Ex	—
NTIN066	Data Structures I	5	2/1 C+Ex	—
NMAI064	Mathematical Structures	6	—	2/2 C+Ex
NSZZ023	Diploma Thesis I	6	0/4 C	0/4 C
NSZZ024	Diploma Thesis II	9	0/6 C	0/6 C
NSZZ025	Diploma Thesis III	15	0/10 C	0/10 C

2.2 Elective courses - Set 1

The student needs to obtain at least 45 credits for the courses from the following set:

Code	Subject	Credits	Winter	Summer
NAIL076	Logic Programming I	3	2/0 Ex	—
NDMI009	Combinatorial and Computational Geometry I	6	2/2 C+Ex	—

NDMI010	Graph Algorithms	3	2/0 Ex	—
NDMI013	Combinatorial and Computational Geometry II	6	—	2/2 C+Ex
NDMI015	Combinatorial Counting	3	—	2/0 Ex
NDMI018	Approximation and Online Algorithms	6	—	2/2 C+Ex
NDMI025	Randomized Algorithms	6	—	2/2 C+Ex
NDMI028	Linear Algebra Applications in Combinatorics	6	2/2 C+Ex	—
NDMI036	Combinatorial Structures	3	—	2/0 Ex
NDMI037	Geometric Representations of Graphs I	3	—	2/0 Ex
NDMI045	Analytic and Combinatorial Number Theory	3	—	2/0 Ex
NDMI055	Selected Chapters on Combinatorics I	3	2/0 Ex	—
NDMI056	Selected Chapters on Combinatorics II	3	—	2/0 Ex
NDMI059	Graph Minors and Tree Decompositions	3	2/0 Ex	—
NDMI060	Coloring of Graphs and Other Combinatorial Structures	3	2/0 Ex	—
NDMI064	Applied Discrete Mathematics	3	2/0 Ex	—
NDMI065	Matroid Theory	6	2/2 C+Ex	—
NDMI066	Algebraic Number Theory	3	2/0 Ex	—
NDMI067	Flows, Paths and Cuts	3	2/0 Ex	—
NDMI073	Combinatorics and Graph Theory III	6	2/2 C+Ex	—
NDMI074	Algorithms and Their Implementation	6	—	2/2 C+Ex
NDMI088	Graph Algorithms II	3	—	2/0 Ex
NMAG337	Introduction to Group Theory	5	2/2 C+Ex	—
NMAI040	Introduction to Number Theory	3	2/0 Ex	—
NMAI065	Fundamentals of Category Theory for Computer Scientists	3	2/0 Ex	—
NMAI066	Topological and Algebraic Methods	3	—	2/0 Ex
NMAI067	Logic in Computer Science	3	2/0 Ex	—
NMAI071	Math++	6	2/2 C+Ex	—
NMMA901	Introduction to Complex Analysis (O)	5	2/2 C+Ex	—
NMMA903	Measure and Integration Theory (O)	8	4/2 C+Ex	—
NMMA931	Introduction to Functional Analysis (O)	8	4/2 C+Ex	—
NOPT008	Nonlinear Optimisation Algorithms	6	—	2/2 C+Ex
NOPT016	Integer Programming	6	—	2/2 C+Ex

NOPT017	Multiobjective Optimisation	3	—	2/0 Ex
NOPT018	Fundamentals of Nonlinear Optimization	6	2/2 C+Ex	—
NOPT034	Mathematical Programming and Polyhedral Combinatorics	5	2/1 C+Ex	—
NOPT042	Constraint Programming	6	2/2 C+Ex	—
NOPT051	Interval Methods	6	2/2 C+Ex	—
NTIN017	Parallel Algorithms	3	—	2/0 Ex
NTIN022	Probabilistic Techniques	6	2/2 C+Ex	—
NTIN063	Complexity	5	—	2/1 C+Ex
NTIN064	Computability	3	—	2/0 Ex
NTIN067	Data Structures II	3	—	2/0 Ex

2.3 Elective courses - Set 2

The student needs to obtain at least 6 credits for the courses from the following set¹:

Code	Subject	Credits	Winter	Summer
NDMI073	Combinatorics and Graph Theory III	6	2/2 C+Ex	—
NOPT018	Fundamentals of Nonlinear Optimization	6	2/2 C+Ex	—

¹For specializations Discrete mathematics and algorithms, Geometry and mathematical structures of computer science we recommend the course NDMI073; for the specialization Optimization we recommend the course NOPT018. After completing one course from Set 2, the credits are counted for that set and the minimal credit requirement for Set 2 is satisfied. If the student completes both courses from Set 2, the credits for the second course are counted for Set 1.

2.4 State Final Exam

In addition to the two examination areas that are obligatory for all study branches, the student will select three examination areas from the following lists. At least two examination areas must be selected from a chosen specialization, one area may be selected from another specialization. In total, each student will get five questions.

a) Specialization *Discrete mathematics and algorithms*

Examination areas

1. Combinatorics and graph theory
2. Probabilistic methods and combinatorial enumeration
3. Combinatorial optimization

Knowledge requirements

1. *Combinatorics and graph theory*

Graph colorings (and variants - choosability), graph minors, tree width and its relation to complexity, geometric representations of graphs (characterization theorems, recognizing algorithms), algebraic properties of graphs, matching theory, Ramsey theory and Szemerédi's regularity lemma, set systems (Steiner triple systems, finite geometries).

Recommended courses

Code	Subject	Credits	Winter	Summer
NDMI037	Geometric Representations of Graphs I	3	—	2/0 Ex
NDMI059	Graph Minors and Tree Decompositions	3	2/0 Ex	—
NDMI060	Coloring of Graphs and Other Combinatorial Structures	3	2/0 Ex	—
NDMI073	Combinatorics and Graph Theory III	6	2/2 C+Ex	—

2. *Probabilistic methods and combinatorial enumeration*

Combinatorial counting, generating functions, recurrences, basic probabilistic models, linearity of expectation, variance and its uses, Markov's inequality and its applications to particular examples, Chernov's inequality, Lovasz local lemma, asymptotic estimates of functions, probabilistic constructions and algorithms.

Recommended courses

Code	Subject	Credits	Winter	Summer
NDMI015	Combinatorial Counting	3	—	2/0 Ex
NDMI025	Randomized Algorithms	6	—	2/2 C+Ex
NTIN022	Probabilistic Techniques	6	2/2 C+Ex	—

3. *Combinatorial optimization*

Graph algorithms, theory of polyhedra, travelling salesman problem, classes of special matrices, integrality, matchings and flows in networks, matroid theory, ellipsoid method.

Recommended courses

Code	Subject	Credits	Winter	Summer
NTIN090	Introduction to Complexity and Computability	5	2/1 C+Ex	—
NDMI010	Graph Algorithms	3	2/0 Ex	—
NDMI065	Matroid Theory	6	2/2 C+Ex	—
NOPT034	Mathematical Programming and Polyhedral Combinatorics	5	2/1 C+Ex	—
NDMI088	Graph Algorithms II	3	—	2/0 Ex

b) Specialization ***Geometry and mathematical structures of computer science*****Examination areas**

1. Combinatorial and computational geometry
2. Algebraic and topological structures in Computer Science
3. Category theory in Computer Science
4. Number theory in Computer Science

Knowledge requirements

1. Combinatorial and computational geometry

Geometric problems in finite-dimensional spaces, combinatorial properties of geometric configurations, algorithmic applications, design of geometric algorithms, geometric representations of graphs.

Recommended courses

Code	Subject	Credits	Winter	Summer
NDMI009	Combinatorial and Computational Geometry I	6	2/2 C+Ex	—
NDMI013	Combinatorial and Computational Geometry II	6	—	2/2 C+Ex

2. Algebraic and topological structures in Computer Science

Posets (partially ordered sets), suprema and infima, semilattices and lattices. Fix point theorems. Special ordered structures in Computer Science (DCPO, domains). Fundamentals of general topology, topological constructions. Topological approaches in Computer Science (Scott's topology, continuous lattices). Categories of topological spaces and certain kinds of posets used in Computer Science.

Recommended courses

Code	Subject	Credits	Winter	Summer
NMAI064	Mathematical Structures	6	—	2/2 C+Ex
NMAI066	Topological and Algebraic Methods	3	—	2/0 Ex

3. Category theory in Computer Science

Categories, functors, transformations, examples. Limits and colimits, special constructions. Adjunction, relation to categorical constructions. Reflections and coreflections. Examples of adjoint situations. Cartesian closed categories. Categories and structures, especially structures used in Computer Science. Monadic algebras.

Recommended courses

Code	Subject	Credits	Winter	Summer
NMAI065	Fundamentals of Category Theory for Computer Scientists	3	2/0 Ex	—

4. Number theory in Computer Science

Diophantine approximation (Dirichlet's theorem, Farey fractions transcendental numbers). Diophantine equations (Pell's equation, Thue equation, four squares theorem, Hilbert's tenth problem). Prime numbers (bounds on the prime number function, Dirichlet's theorem). Geometry of numbers (lattices, Minkowski's theorem). Congruences (quadratic residues). Integer partitions (identities, e.g., the pentagonal identity).

Recommended courses

Code	Subject	Credits	Winter	Summer
NMAI040	Introduction to Number Theory	3	2/0 Ex	—

c) Specialization **Optimisation****Examination areas**

1. Nonlinear programming
2. Discrete optimisation processes
3. Multiobjective and integer programming
4. Parametric programming and interval methods

Knowledge requirements*1. Nonlinear programming*

Properties of convex sets and convex functions. Generalizations of convex functions. Necessary and sufficient optimality conditions for free and constrained extrema in problems of nonlinear programming. Quadratic programming. Semidefinite programming. Duality in nonlinear programming. Methods for solving problems on free and constrained extrema, including penalization and barrier methods. One-dimensional optimization.

Recommended courses

Code	Subject	Credits	Winter	Summer
NOPT018	Fundamentals of Nonlinear Optimization	6	2/2 C+Ex	—

2. Discrete optimisation processes

Algorithmic game theory, election mechanisms, electronic auctions, application of submodular functions in economy. Optimization based on enumeration, generating functions of edge cuts and perfect matchings, enumerative dualities, the maximum cut problem for graphs on surfaces.

Recommended courses

Code	Subject	Credits	Winter	Summer
NDMI064	Applied Discrete Mathematics	3	2/0 Ex	—
NOPT018	Fundamentals of Nonlinear Optimization	6	2/2 C+Ex	—

3. Multiobjective and integer programming

Various approaches to solving problems with several criteria. Functional associated to a problem of vector programming. Pareto optimal solution. Problems of linear and nonlinear vector optimization. Methods for obtaining Pareto optimal solutions. Problems of linear programming with integrality conditions or with binary variables. Nonlinear optimization problems with integrality conditions.

Recommended courses

Code	Subject	Credits	Winter	Summer
NOPT016	Integer Programming	6	—	2/2 C+Ex
NOPT017	Multiobjective Optimisation	3	—	2/0 Ex
NOPT018	Fundamentals of Nonlinear Optimization	6	2/2 C+Ex	—

4. Parametric programming and interval methods

Domains of stability of solutions. Domains of solvability. Solvability function for one-parametric and multi-parametric programming. Interval linear algebra (systems of linear equations, regularity, eigenvalues). Linear programming with imprecise data. Deterministic global optimization, lower and upper bounds on objective function and optimum value.

Recommended courses

Code	Subject	Credits	Winter	Summer
NOPT017	Multiobjective Optimisation	3	—	2/0 Ex
NOPT018	Fundamentals of Nonlinear Optimization	6	2/2 C+Ex	—
NOPT051	Interval Methods	6	2/2 C+Ex	—

2 Degree Plans - Theoretical Computer Science

Coordinated by: Department of Theoretical Computer Science and Mathematical Logic; Computer Science Institute of Charles University

Study branch coordinator: Doc. Mgr. Michal Koucký, Ph.D.

This study branch has no specializations.

Theoretical Computer Science provides comprehensive education in theoretical aspects of computational models, algorithm and data structure design, and structural properties of Boolean functions. Students gain understanding of the state-of-the-art techniques in the design of efficient algorithms and data structures, and also learn the limits and possibilities for solving algorithmic problems. In addition to that students acquire mathematical tools necessary to analyze and model algorithmic processes. Students can utilize gained knowledge in practical setting or they can continue by a doctoral study in theoretical computer science or related areas.

The graduate thoroughly understands the limits and possibilities of computational systems, has a broad overview of algorithmic techniques, and is able to apply these techniques to new problems. He also has skills necessary to convey abstract ideas with precision and clarity. The graduate can apply his skills in the design and analysis of complex systems and in the development of innovative solutions and transformative technologies. The graduate is also well prepared for doctoral studies in theoretical computer science and related areas.

3.1 Obligatory courses

Code	Subject	Credits	Winter	Summer
NTIN090	Introduction to Complexity and Computability	5	2/1 C+Ex	—
NTIN066	Data Structures I	5	2/1 C+Ex	—
NTIN022	Probabilistic Techniques	6	2/2 C+Ex	—
NTIN063	Complexity	5	—	2/1 C+Ex
NTIN100	Introduction to Information Transmission and Processing	5	—	2/1 C+Ex
NSZZ023	Diploma Thesis I	6	0/4 C	0/4 C

NSZZ024	Diploma Thesis II	9	0/6 C	0/6 C
NSZZ025	Diploma Thesis III	15	0/10 C	0/10 C

3.2 Elective courses

The student needs to obtain at least 45 credits for the courses from the following set:

Code	Subject	Credits	Winter	Summer
NAIL021	Boolean Functions and Their Applications	3	2/0 Ex	—
NAIL031	Representations of Boolean Functions	3	—	2/0 Ex
NAIL094	Decision Procedures and Verification	6	2/2 C+Ex	—
NALG117	Introduction to Complexity of CSP	3	2/0 Ex	—
NDMI010	Graph Algorithms	3	2/0 Ex	—
NDMI013	Combinatorial and Computational Geometry II	6	—	2/2 C+Ex
NDMI018	Approximation and Online Algorithms	6	—	2/2 C+Ex
NDMI025	Randomized Algorithms	6	—	2/2 C+Ex
NDMI067	Flows, Paths and Cuts	3	2/0 Ex	—
NDMI074	Algorithms and Their Implementation	6	—	2/2 C+Ex
NDMI077	Algorithms for Specific Graph Classes	3	—	2/0 Ex
NDMI088	Graph Algorithms II	3	—	2/0 Ex
NMAG446	Logic and Complexity	3	—	2/0 Ex
NMAG536	Proof Complexity and the P vs. NP Problem	3	—	2/0 Ex
NMAI067	Logic in Computer Science	3	2/0 Ex	—
NOPT034	Mathematical Programming and Polyhedral Combinatorics	5	2/1 C+Ex	—
NSWI072	Data Compression Algorithms	3	2/0 Ex	—
NTIN006	Algebraic Algorithms	3	—	2/0 Ex
NTIN017	Parallel Algorithms	3	—	2/0 Ex
NTIN018	Probabilistic Analysis of Algorithms	3	2/0 Ex	—
NTIN033	Experimental Analysis of Algorithms	6	—	2/2 C+Ex
NTIN064	Computability	3	—	2/0 Ex
NTIN067	Data Structures II	3	—	2/0 Ex
NTIN073	Recursion	3	2/0 Ex	—
NTIN081	Structural Complexity	3	—	2/0 Ex
NTIN082	Computational Complexity	3	2/0 Ex	—
NTIN084	Bioinformatics Algorithms	6	2/2 C+Ex	—

NTIN085	Selected Topics in Computational Complexity I	5	2/1 C+Ex	—
NTIN086	Selected Topics in Computational Complexity II	5	—	2/1 C+Ex
NTIN087	String Algorithms	3	2/0 Ex	—
NTIN088	Algorithmic Randomness	3	—	2/0 Ex
NTIN096	Pseudo-Boolean Optimization	3	—	2/0 Ex
NTIN097	Hypercube Problems	3	2/0 Ex	—
NTIN098	Advanced Data Structures	3	2/0 Ex	—
NTIN099	Algorithmic Aspects of Boolean Functions and Parameterized Complexity	3	—	2/0 Ex

3.3 Other recommended courses

Code	Subject	Credits	Winter	Summer
NOPT016	Integer Programming	6	—	2/2 C+Ex
NOPT042	Constraint Programming	6	2/2 C+Ex	—
NTIN023	Dynamic Graph Data Structures	3	2/0 Ex	—

3.4 State Final Exam

In addition to the two examination areas that are obligatory for all study branches, the student will select three other examination areas. Two of them must be from the following list of examination areas; the last one can be either from the following list as well, or it can be any area from the study branch Discrete Models and Algorithms, any area from the specialization Intelligent agents or the specialization Machine learning of the study branch Artificial Intelligence, or any area from the specialization Computer graphics of the study branch Computer Graphics and Game Development. In total, each student will get five questions from the five examination areas.

Examination areas

1. Complexity and computability
2. Boolean functions
3. Algorithms
4. Advanced data structures

Knowledge requirements

1. *Complexity and computability*

Oracle computation and relativized complexity classes. Polynomial hierarchy. Non-uniform models of computation. Probabilistic complexity classes. Interactive protocols, PCP Theorem. One-way functions and pseudo-random generators. Communication complexity. Proof complexity. Relationships and separations among complexity classes. Recursion theorems and their application. Effectively inseparable sets. Relativized computability and the jump operation. Arithmetic hierarchy.

Recommended courses

Code	Subject	Credits	Winter	Summer
NTIN063	Complexity	5	—	2/1 C+Ex

NTIN081	Structural Complexity	3	—	2/0 Ex
NTIN082	Computational Complexity	3	2/0 Ex	—
NMAG536	Proof Complexity and the P vs. NP Problem	3	—	2/0 Ex
NTIN064	Computability	3	—	2/0 Ex
NTIN100	Introduction to Information Transmission and Processing	5	—	2/1 C+Ex

2. Boolean functions

Resolution and its completeness. Classes of Boolean functions with special properties. Algorithms for SAT and MAXSAT. Representing Boolean functions using BDD's and OBDD's. SAT solvers and their use for the SMT Problem. Parameterized complexity. Hypercube graphs.

Recommended courses

Code	Subject	Credits	Winter	Summer
NTIN099	Algorithmic Aspects of Boolean Functions and Parameterized Complexity	3	—	2/0 Ex
NAIL094	Decision Procedures and Verification	6	2/2 C+Ex	—
NTIN097	Hypercube Problems	3	2/0 Ex	—
NAIL021	Boolean Functions and Their Applications	3	2/0 Ex	—
NAIL031	Representations of Boolean Functions	3	—	2/0 Ex

3. Algorithms

Advanced graph algorithms, flows in graphs, algorithms for planar graphs. Linear and semidefinite programming, polynomial algorithms, applications in graph and approximation algorithms. Combinatorial approximation algorithms and schemes. Probabilistic algorithms, approximate counting, hashing and its applications. Interactive protocols and verification, PCP Theorem and its applications. Parallel models of computation and complexity classes, techniques and examples of parallel algorithms. String algorithms, sequences, sub-sequences, regular expressions and search.

Recommended courses

Code	Subject	Credits	Winter	Summer
NDMI010	Graph Algorithms	3	2/0 Ex	—
NDMI018	Approximation and Online Algorithms	6	—	2/2 C+Ex
NDMI025	Randomized Algorithms	6	—	2/2 C+Ex
NTIN017	Parallel Algorithms	3	—	2/0 Ex
NTIN087	String Algorithms	3	2/0 Ex	—

4. Advanced data structures

Entropy and information. Error-correcting codes. Data compression. Data structures for string processing. Dynamic data structures for graphs. Dictionary data struc-

tures. Probabilistic search data structures. Advanced heaps. Data structures for storing integers. Persistent data structures. Self-adjusting data structures. Cache oblivious analysis and optimal algorithms. Data-streaming algorithms.

Recommended courses

Code	Subject	Credits	Winter	Summer
NTIN100	Introduction to Information Transmission and Processing	5	—	2/1 C+Ex
NTIN067	Data Structures II	3	—	2/0 Ex
NTIN087	String Algorithms	3	2/0 Ex	—
NDMI010	Graph Algorithms	3	2/0 Ex	—
NTIN098	Advanced Data Structures	3	2/0 Ex	—
NSWI072	Data Compression Algorithms	3	2/0 Ex	—

3 Degree Plans - Software and Data Engineering

Coordinated by: Department of Software Engineering

Study branch coordinator: Doc. RNDr. Tomáš Skopal, Ph.D.

Specializations:

- Software engineering
- Software development
- Web engineering
- Database systems
- Big data processing

The study branch Software and data engineering aims at expertise in analysis, design and development of complex software solutions, and systems focused on big data processing. The portfolio of courses provided in the study covers a number of technological platforms, from classic, web-based, to modern cloud and distributed solutions. A required part of the study is a work on large software project where students apply not only the theoretical knowledge and technological skills but also team work abilities.

The graduate gains a deep knowledge of software and data engineering based on her/his specialization. With the specialization Software engineering the graduate is able to analyse requirements for software solutions, to design architectures, and to lead the development process. The specialization Software development prepares the graduate for leading a team of SW developers. The development of internet applications is covered by the specialization Web engineering. The graduate of Database systems is able to design schemas of databases and to implement complex database applications. With the Big data processing specialization the graduate is prepared for the role of data scientist with abilities in data mining and related data analytics knowledge.

4.1 Obligatory courses

Code	Subject	Credits	Winter	Summer
NTIN090	Introduction to Complexity and Computability	5	2/1 C+Ex	—
NTIN066	Data Structures I	5	2/1 C+Ex	—

NSWI004	Operating Systems	5	2/1 C+Ex	—
NPRG027	Credit for Project	6	0/4 C	0/4 C
NPRG023	Software Project	9	0/6 C	0/6 C
NSZZ023	Diploma Thesis I	6	0/4 C	0/4 C
NSZZ024	Diploma Thesis II	9	0/6 C	0/6 C
NSZZ025	Diploma Thesis III	15	0/10 C	0/10 C

4.2 Elective courses

The student needs to obtain at least 48 credits for the courses from the following set:

Code	Subject	Credits	Winter	Summer
NDBI001	Query Languages I	5	2/2 C+Ex	—
NDBI006	Query Languages II	5	—	2/2 C+Ex
NDBI016	Transactions	3	—	2/0 Ex
NDBI019	Stochastic Methods in Databases	3	—	2/0 Ex
NDBI021	Querying with Preferences	5	—	2/2 C+Ex
NDBI023	Data Mining	9	—	4/2 C+Ex
NDBI034	Retrieval of Multimedia Content on the Web	4	2/1 C+Ex	—
NDBI040	Big Data Management and NoSQL Databases	4	2/1 C+Ex	—
NDBI042	Data Visualization Techniques	3	—	2/1 C+Ex
NDBI043	Text Search Techniques	3	—	2/0 Ex
NMAI060	Probabilistic Methods	3	2/0 Ex	—
NPFL054	Introduction to Machine Learning	6	2/2 C+Ex	—
NPRG014	Concepts of Modern Programming Languages	2	0/2 C	—
NPRG024	Design Patterns	3	—	0/2 MC
NPRG039	Advanced aspects and new trends in XML	5	2/1 C+Ex	—
NPRG042	Programming in Parallel Environment	6	—	2/2 C+Ex
NPRG043	Recommended Programming Practices	6	—	2/2 C+Ex
NPRG054	High Performance Software Development	5	—	2/2 C+Ex
NPRG059	Advanced Programming Praxis	2	0/1 C	—
NSWI021	Computer Networks II	3	—	2/0 Ex
NSWI035	Principles of Distributed Systems	3	2/0 Ex	—
NSWI041	Introduction to Software Engineering	5	—	2/2 C+Ex
NSWI045	TCP/IP Protocol Suite	3	—	2/0 Ex
NSWI068	Object and Component Systems	5	2/2 C+Ex	—
NSWI072	Data Compression Algorithms	3	2/0 Ex	—
NSWI073	Modern Networking Solutions	3	0/2 C	—

NSWI080	Middleware	5	—	2/1 C+Ex
NSWI101	System Behaviour Models and Verification	6	2/2 C+Ex	—
NSWI108	Web Semantization	5	2/2 C+Ex	—
NSWI126	Advanced Tools for Software Development and Monitoring	3	—	0/2 C
NSWI130	Software System Architectures	5	2/2 C+Ex	—
NSWI131	Performance Evaluation of Computer Systems	5	—	2/1 C+Ex
NSWI144	Data Integration and Quality	4	2/1 C+Ex	—
NSWI145	Web Services	5	—	2/2 C+Ex
NSWI149	Software Engineering in Practice	3	—	2/0 C
NSWI150	Virtualization and Cloud Computing	3	2/0 Ex	—
NSWI152	Cloud Application Development	3	—	0/2 C
NSWI153	Advanced Technologies for Web Applications	3	—	2/0 C
NTIN043	Formal Foundations of Software Engineering	5	2/2 C+Ex	—
NTIN067	Data Structures II	3	—	2/0 Ex

4.3 State Final Exam

In addition to the two examination areas that are obligatory for all study branches, there are two additional obligatory areas based on the selected specialization as indicated below. The student will select the last examination area from the following list. In total, each student will get five questions.

Examination areas

1. Software analysis and architectures (obligatory for the specialization Software engineering)
2. Advanced programming (obligatory for the specializations Software engineering and Software development)
3. Software technologies (obligatory for the specialization Software development)
4. Web technologies (obligatory for the specialization Web engineering)
5. Databases - formal foundations and query languages (obligatory for the specializations Web engineering and Database systems)
6. Databases - implementation and administration (obligatory for the specialization Database systems)
7. Big and unstructured data processing (obligatory for the specialization Big data processing)
8. Data mining (obligatory for the specialization Big data processing)

Knowledge requirements

1. *Software analysis and architectures*

SW development processes, development phases. Business processes and their modeling using BPMN. UML and its use for analysis and design of structure and behavior of SW. Design patterns. SW testing, impact and change analysis. SW project

planning, cost estimation, levels of project management. Legal aspects of SW, principal legal environment for IT projects. Types of SW architecture. Modeling and documentation of SW architecture. Classification of SW architecture quality attributes, their description using scenarios and tactics. Service oriented architectures. Algebraic methods, many sorted algebras, initial models. Formal principles of the UML language. OCL as a specification language, formal base of specification. Formal principles of RDF and OWL, the description logic.

Recommended courses

Code	Subject	Credits	Winter	Summer
NSWI130	Software System Architectures	5	2/2 C+Ex	—
NSWI041	Introduction to Software Engineering	5	—	2/2 C+Ex
NSWI094	Database Application Development	5	2/1 C+Ex	—
NTIN043	Formal Foundations of Software Engineering	5	2/2 C+Ex	—

2. *Advanced programming*

Object concepts of modern languages. Concepts of classless languages. Generic programming and metaprogramming, generics and templates, policies, traits, type inference. Modern constructs of programming languages. Object references and lifetime. Advanced aspects of imperative languages. Exceptions, exception-safe programming. Implementation of object properties, runtime support, calling conventions, garbage collection. Modern language constructs and code performance. Design patterns. Scripting languages, prototype-based languages. Domain-specific languages. Functional programming. Principles of code quality, best practices. Refactoring. Testing, debugging, monitoring.

Recommended courses

Code	Subject	Credits	Winter	Summer
NPRG059	Advanced Programming Praxis	2	0/1 C	—
NPRG014	Concepts of Modern Programming Languages	2	0/2 C	—
NPRG024	Design Patterns	3	—	0/2 MC
NPRG043	Recommended Programming Practices	6	—	2/2 C+Ex
NSWI068	Object and Component Systems	5	2/2 C+Ex	—

3. *Software technologies*

Operating system architectures, process management, memory management, communication and synchronization, parallelism, virtualization, paging. File systems, access rights and security. Portability, multiplatform applications. Testing and monitoring of performance and functionality. Architectures of web applications, server-side and client-side scripting, cooperation with database systems. Architecture of data servers, transactions, performance optimization. Cluster, Grid, and Cloud. IaaS, PaaS, and SaaS. MapReduce. Load balancing, high availability.

Recommended courses

Code	Subject	Credits	Winter	Summer
NSWI126	Advanced Tools for Software Development and Monitoring	3	—	0/2 C
NSWI153	Advanced Technologies for Web Applications	3	—	2/0 C
NDBI036	A Practical Approach to Database Systems	4	2/1 C+Ex	—
NSWI150	Virtualization and Cloud Computing	3	2/0 Ex	—

4. Web technologies

Overview of basic web technologies. Network services for web technologies. Web services. Architecture of client-server applications, server-side and client-side scripting, web frameworks. Database systems in web applications, NoSQL databases, multimedia databases. Indexing and document searching, principles of web search engines. Linked Data, integration of semantic data to web pages. Security of information systems in the Internet environment, authentication, authorization, security models, cryptography basics, data security.

Recommended courses

Code	Subject	Credits	Winter	Summer
NSWI130	Software System Architectures	5	2/2 C+Ex	—
NSWI153	Advanced Technologies for Web Applications	3	—	2/0 C
NSWI145	Web Services	5	—	2/2 C+Ex
NDBI034	Retrieval of Multimedia Content on the Web	4	2/1 C+Ex	—
NPRG043	Recommended Programming Practices	6	—	2/2 C+Ex

5. Databases - formal foundations and query languages

Relational calculi, relational algebras. Relational completeness. Safe expressions, equivalences of relational query languages. Transitive closure of relation. Semantics of SQL. SQL standards. Object extension of relational data model. SQL/XML. Text databases – Boolean and vector model. Searching in text collections: Boolean and vector indexes, signatures, query result ranking. Querying XML data, similarity of XML documents. Datalog. Deductive databases. Recursion in SQL. Tableau queries. Preference modeling and querying, top-k algorithms, Fuzzy Datalog, recommending systems. RDF data model, SPARQL query language, similarity search in multimedia databases, metric indexes for similarity search.

Recommended courses

Code	Subject	Credits	Winter	Summer
NDBI001	Query Languages I	5	2/2 C+Ex	—
NDBI006	Query Languages II	5	—	2/2 C+Ex
NDBI021	Querying with Preferences	5	—	2/2 C+Ex

NDBI040	Big Data Management and NoSQL Databases	4	2/1 C+Ex	—
NDBI043	Text Search Techniques	3	—	2/0 Ex

6. Databases - implementation and administration

Architectures of database systems. Models and properties of transactions: locking protocols, time stamps. Transaction isolation, resource allocation. Distributed transactions. Error/failure recovery, journals. Distribution with horizontal fragmentation, implementation of NoSQL databases, CAP theorem. Indexing relational data. Spatial access methods. Algorithms for implementation of relational operations and aggregation functions. Query evaluation and optimization. Indexing XML data. Query evaluation in XML query languages. Data compression: text models, coding, Huffman coding, arithmetic coding, LZ algorithms, bitmap compression, sparse matrices compression, Burrows-Wheeler transformation.

Recommended courses

Code	Subject	Credits	Winter	Summer
NDBI016	Transactions	3	—	2/0 Ex
NPRG039	Advanced aspects and new trends in XML	5	2/1 C+Ex	—
NSWI072	Data Compression Algorithms	3	2/0 Ex	—
NSWI144	Data Integration and Quality	4	2/1 C+Ex	—
NDBI040	Big Data Management and NoSQL Databases	4	2/1 C+Ex	—
NTIN066	Data Structures I	5	2/1 C+Ex	—

7. Big and unstructured data processing

Distribution with horizontal fragmentation, implementation of NoSQL databases, CAP theorem. Big Data management - distribution, scalability, replication, transactions. MapReduce. Key-value storages. Column storages. Document storages. Models for fulltext querying - vector, Boolean, probabilistic models. Data compression: text models, coding. Similarity search in multimedia databases, metric indexes for similarity search. Data visualization techniques. RDF(S) models, description and dynamic logic, web query languages, model of web semantization.

Recommended coursesy

Code	Subject	Credits	Winter	Summer
NDBI043	Text Search Techniques	3	—	2/0 Ex
NDBI040	Big Data Management and NoSQL Databases	4	2/1 C+Ex	—
NDBI034	Retrieval of Multimedia Content on the Web	4	2/1 C+Ex	—
NDBI042	Data Visualization Techniques	3	—	2/1 C+Ex
NSWI108	Web Semantization	5	2/2 C+Ex	—

8. Data mining

Basic principles of database systems, data warehouses and OLAP technology. Data mining – data preprocessing, concept description techniques, methods for mining asso-

ciation rules, methods for data classification and prediction, cluster analysis methods, data mining in database systems. Statistical methods for data mining. Discovery of different types of dependencies. Bayesian analysis, bayesian networks. Probabilistic models of information retrieval. Methods of learning for classification and regression. Support Vector Machines and kernel functions. Experiment evaluation. Data visualization techniques.

Recommended courses

Code	Subject	Credits	Winter	Summer
NDBI023	Data Mining	9	—	4/2 C+Ex
NDBI019	Stochastic Methods in Databases	3	—	2/0 Ex
NAIL029	Machine Learning	3	—	2/0 Ex
NDBI042	Data Visualization Techniques	3	—	2/1 C+Ex

4 Degree Plans - Software Systems

Coordinated by: Department of Distributed and Dependable Systems

Study branch coordinator: Doc. Ing. Petr Tůma, Dr.

Specializations:

- System programming
- Dependable systems
- High performance computing

The study branch puts emphasis on system-oriented programming in one of three focus domains. The specialization System programming focuses on coding the basic layers of a computer system (middleware, operating system). In the specialization Dependable systems, the curriculum deals with systematic construction of systems with high reliability, such as embedded and real-time systems. The specialization High performance computing introduces techniques for software development on high performance computing systems (highly parallel systems, distributed systems, clouds). All focus domains pay attention to both the programming tools and methods and the associated architectural knowledge.

The graduate possesses robust programming skills in the given focus domain: System programming for modern operating systems and system-related technologies (middleware, virtual machines), Dependable systems for dealing with the systematic construction of systems with guaranteed reliability, and High performance computing for software development on modern parallel and distributed systems. The graduate has absorbed both the necessary theoretical foundations and the skills required for solving practical programming tasks. He can use modern programming languages and tools. He can adapt to the fast-moving technologies of today and use these technologies in team software projects. He can solve problems individually and systematically, and apply deep system knowledge in delivering outside-the-box solutions.

5.1 Obligatory courses

Code	Subject	Credits	Winter	Summer
NTIN090	Introduction to Complexity and Computability	5	2/1 C+Ex	—

NTIN066	Data Structures I	5	2/1 C+Ex	—
NMAI060	Probabilistic Methods	3	2/0 Ex	—
NSWI004	Operating Systems	5	2/1 C+Ex	—
NPRG043	Recommended Programming Practices	6	—	2/2 C+Ex
NPRG059	Advanced Programming Praxis	2	0/1 C	—
NSWI126	Advanced Tools for Software Development and Monitoring	3	—	0/2 C
NSWI150	Virtualization and Cloud Computing	3	2/0 Ex	—
NPRG027	Credit for Project	6	0/4 C	0/4 C
NPRG023	Software Project	9	0/6 C	0/6 C
NSZZ023	Diploma Thesis I	6	0/4 C	0/4 C
NSZZ024	Diploma Thesis II	9	0/6 C	0/6 C
NSZZ025	Diploma Thesis III	15	0/10 C	0/10 C

5.2 Elective courses

The student needs to obtain at least 30 credits for the courses from the following set:

Code	Subject	Credits	Winter	Summer
NPRG042	Programming in Parallel Environment	6	—	2/2 C+Ex
NPRG054	High Performance Software Development	5	—	2/2 C+Ex
NPRG056	Developing Applications for Mobile Devices	3	0/2 C	—
NSWE001	Embedded and Real Time Systems	6	—	2/2 C+Ex
NSWI035	Principles of Distributed Systems	3	2/0 Ex	—
NSWI041	Introduction to Software Engineering	5	—	2/2 C+Ex
NSWI068	Object and Component Systems	5	2/2 C+Ex	—
NSWI080	Middleware	5	—	2/1 C+Ex
NSWI101	System Behaviour Models and Verification	6	2/2 C+Ex	—
NSWI109	Compiler Design	4	—	2/1 C+Ex
NSWI131	Performance Evaluation of Computer Systems	5	—	2/1 C+Ex
NSWI132	Program Analysis and Code Verification	6	—	2/2 C+Ex
NSWI029	Modern Trends in Computer Science	2	0/2 C	0/2 C
NSWI054	Software Engineering for Dependable Systems	3	—	0/2 MC

5.3 State Final Exam

In addition to the two examination areas that are obligatory for all study branches, the student will get three questions from the following areas based on the chosen specialization. In total, each student will get five questions.

a) Specialization **System programming**

Examination areas

1. System aspects of computers (SP)
2. Parallel and distributed systems (SP)
3. Modern programming concepts (SP)

Knowledge requirements

1. *System aspects of computers (SP)*

Process and thread management, communication (IPC), synchronization, memory models. Memory management (paging, heap structure, garbage collection). Runtime support, calling conventions (ABI), virtualization. Device management. File systems. All topics considered in the context of contemporary parallel architectures.

Recommended courses

Code	Subject	Credits	Winter	Summer
NSWI004	Operating Systems	5	2/1 C+Ex	—

2. *Parallel and distributed systems (SP)*

Modern memory architectures (multiprocessors, GPU, NUMA), relationship between data location and access efficiency. Communication and coordination in distributed environment, algorithms and technologies (specific technologies for RPC, DSM, messaging based on current development).

Recommended courses

Code	Subject	Credits	Winter	Summer
NPRG042	Programming in Parallel Environment	6	—	2/2 C+Ex
NPRG054	High Performance Software Development	5	—	2/2 C+Ex
NSWI035	Principles of Distributed Systems	3	2/0 Ex	—
NSWI080	Middleware	5	—	2/1 C+Ex

3. *Modern programming concepts (SP)*

Modern object-oriented concepts, metaprogramming, reflection, aspects. Features of modern programming languages (for example annotations, iterators, generics, lambda functions). Code quality metrics, documentation, refactoring, testing and debugging. Design patterns. All topics considered in the context of contemporary programming languages.

Recommended courses

Code	Subject	Credits	Winter	Summer
NPRG043	Recommended Programming Practices	6	—	2/2 C+Ex

NSWI068	Object and Component Systems	5	2/2 C+Ex	—
---------	------------------------------	---	----------	---

b) Specialization *Dependable systems***Examination areas**

1. System aspects of computers (DS)
2. Parallel and distributed systems (DS)
3. Formal methods (DS)

Knowledge requirements*1. System aspects of computers (DS)*

Process and thread management, communication (IPC), synchronization, memory models. Real-time and embedded systems (RTES), real-time scheduling, design and modeling of RTES, operating systems for RTES. Mobile devices, design and deployment of mobile applications, resource management.

Recommended courses

Code	Subject	Credits	Winter	Summer
NSWI004	Operating Systems	5	2/1 C+Ex	—
NSWE001	Embedded and Real Time Systems	6	—	2/2 C+Ex
NPRG056	Developing Applications for Mobile Devices	3	0/2 C	—

2. Parallel and distributed systems (DS)

Synchronization in concurrent systems. Causality and consensus in distributed environments. Communication, coordination, replication and mobility in distributed environment, algorithms and technologies (specific technologies for RPC, DSM, messaging based on current development).

Recommended courses

Code	Subject	Credits	Winter	Summer
NSWI035	Principles of Distributed Systems	3	2/0 Ex	—
NSWI080	Middleware	5	—	2/1 C+Ex

3. Formal methods (DS)

Program behavior modeling, formal tools (timed automata, LTS, Kripke Structure, lattice, LTL, CTL, bisimulation). Contracts and code verification. Static analysis (principles, basic algorithms). Model checking, state explosion, infinite state model checking, compositional reasoning, bounded model checking, CEGAR, symbolic execution, runtime verification, abstract interpretation).

Recommended courses

Code	Subject	Credits	Winter	Summer
NSWI101	System Behaviour Models and Verification	6	2/2 C+Ex	—
NSWI132	Program Analysis and Code Verification	6	—	2/2 C+Ex

c) Specialization **High performance computing****Examination areas**

1. Distributed systems (HPC)
2. Parallel programming (HPC)
3. System aspects of computers (HPC)

Knowledge requirements*1. Distributed systems (HPC)*

Virtualization technologies, clusters, clouds. Frameworks for cloud application development and cloud computing, deployment. Causality and consensus in distributed environment. Communication, coordination, replication and mobility in distributed environment, algorithms and technologies (specific technologies for RPC, DSM, messaging based on current development).

Recommended courses

Code	Subject	Credits	Winter	Summer
NSWI150	Virtualization and Cloud Computing	3	2/0 Ex	—
NSWI035	Principles of Distributed Systems	3	2/0 Ex	—
NSWI080	Middleware	5	—	2/1 C+Ex

2. Parallel programming (HPC)

Modern memory architectures (multiprocessor, GPU, NUMA), relationship between data location and access efficiency. Virtualization support. Converting computations for data parallelism, parallel solutions of non-homogeneous tasks. Efficient parallel implementation of basic algorithms. Synchronization, load balancing.

Recommended courses

Code	Subject	Credits	Winter	Summer
NPRG042	Programming in Parallel Environment	6	—	2/2 C+Ex
NPRG054	High Performance Software Development	5	—	2/2 C+Ex

3. System aspects of computers (HPC)

Process and thread management, communication (IPC), synchronization, memory models. Memory management (paging, heap structure, garbage collection). Runtime support, calling conventions (ABI), virtualization. Device management. File systems. Performance metrics, methods for performance evaluation, simulation and modeling. All topics considered in the context of contemporary parallel architectures.

Recommended courses

Code	Subject	Credits	Winter	Summer
NSWI004	Operating Systems	5	2/1 C+Ex	—
NSWI131	Performance Evaluation of Computer Systems	5	—	2/1 C+Ex

5 Degree Plans - Computational Linguistics

Coordinated by: Institute of Formal and Applied Linguistics

Study branch coordinator: Doc. RNDr. Markéta Lopatková, Ph.D.

Specializations:

- Computational and formal linguistics
- Statistical methods and machine learning in computational linguistics

The aim of the study branch Computational Linguistics is to get the students ready for research in the area of natural language processing and development of applications dealing with both written and spoken language. Examples of such applications are systems of information retrieval, machine translation, grammar checking, text summarization and information extraction, automatic speech recognition, voice control, spoken dialogue systems, and speech synthesis. The emphasis is put on deep understanding of formal foundations and their practical applicability. The study branch Computational Linguistics can be studied in two specializations: (i) computational and formal linguistics, and (ii) statistical methods and machine learning in computational linguistics.

The graduate is familiar with the theoretical foundations of the formal description of natural languages, the mathematical and algorithmic foundations of automatic natural language processing, and state-of-the-art machine learning techniques. Graduates have the ability to apply the knowledge acquired during their studies in the design and development of systems automatically processing natural language and large quantities of both structured and unstructured data, such as information retrieval, question answering, summarization and information extraction, machine translation and speech processing. They are equipped with reasonable knowledge, skills, and experience in software development and teamwork applicable in all areas involving the development of applications aiding human-computer interaction and/or machine learning.

6.1 Obligatory courses

Code	Subject	Credits	Winter	Summer
NTIN090	Introduction to Complexity and Computability	5	2/1 C+Ex	—
NTIN066	Data Structures I	5	2/1 C+Ex	—
NPFL063	Introduction to General Linguistics	5	2/1 C+Ex	—
NPFL067	Statistical Methods in Natural Language Processing I	6	2/2 C+Ex	—
NPFL092	NLP Technology	5	1/2 MC	—
NSZZ023	Diploma Thesis I	6	0/4 C	0/4 C
NSZZ024	Diploma Thesis II	9	0/6 C	0/6 C
NSZZ025	Diploma Thesis III	15	0/10 C	0/10 C

6.2 Elective courses

The student needs to obtain at least 42 credits for the courses from the following set:

Code	Subject	Credits	Winter	Summer
NPFL006	Introduction to Formal Linguistics	3	2/0 Ex	—
NPFL038	Fundamentals of Speech Recognition and Generation	6	2/2 C+Ex	—
NPFL068	Statistical Methods in Natural Language Processing II	6	—	2/2 C+Ex
NPFL070	Language Data Resources	5	—	1/2 MC
NPFL075	Prague Dependency Treebank	6	—	2/2 C+Ex
NPFL079	Algorithms in Speech Recognition	6	—	2/2 C+Ex
NPFL082	Information Structure of Sentences and Discourse Structure	3	—	1/1 Ex
NPFL083	Linguistic Theory and Grammar Formalisms	6	—	2/2 C+Ex
NPFL087	Statistical Machine Translation	6	—	2/2 C+Ex
NPFL093	NLP Applications	5	—	2/1 MC
NPFL094	Morphological and Syntactic Analysis I	3	2/0 MC	—
NPFL095	Modern Methods in Computational Linguistics	3	0/2 C	—
NPFL096	Computational Morphology	4	—	2/1 Ex
NPFL099	Statistical Dialogue Systems	5	—	2/1 C+Ex
NPFL103	Information Retrieval	6	2/2 C+Ex	—
NPFL104	Machine Learning Methods	5	—	1/2 C+Ex
NPRG027	Credit for Project	6	0/4 C	0/4 C
NPRG023	Software Project	9	0/6 C	0/6 C

6.3 State Final Exam

In addition to the two examination areas that are obligatory for all study branches, there is one obligatory area for this study branch, one obligatory area dependent on the specialization. As the last examination area, the student can select either the obligatory area of the other specialization of the study branch Computational Linguistics, or any area from the specialization Intelligent agents or the specialization Machine learning of the study branch Artificial Intelligence, or any area from the specialization Computer graphics of the study branch Computer Graphics and Game Development. In total, each student will get five questions from the five examination areas.

Examination areas

1. Fundamentals of natural language processing (obligatory for both specializations)
2. Linguistic theories and formalisms (obligatory for the specialization Computational and formal linguistics)
3. Statistical methods and machine learning in computational linguistics (obligatory for the specialization Statistical methods and machine learning in computational linguistics)
4. Multimodal technologies and data
5. Applications in natural language processing

Knowledge requirements*1. Fundamentals of natural language processing*

Fundamentals of general linguistics. System of layers in language description. Dependency syntax, formal definition of dependency trees and their characteristics. The Chomsky hierarchy of languages, context free languages, phrase grammars, unification-based grammars and categorial grammars for a natural language. Design and evaluation of linguistic experiments, evaluation metrics. Basic stochastic methods. Language modeling, basic methods for training stochastic models. Basic algorithms.

Recommended courses

Code	Subject	Credits	Winter	Summer
NPFL067	Statistical Methods in Natural Language Processing I	6	2/2 C+Ex	—
NPFL063	Introduction to General Linguistics	5	2/1 C+Ex	—

2. Linguistic theories and formalisms

Functional Generative Description. Prague Dependency Treebank. Other basic grammar formalisms (Government and Binding, unification-based grammars, feature structures, HPSG, LFG, categorial grammars, (L)TAG). Phonetics, phonology. Computational Morphology. Syntax. Computational lexicography. Topic-focus articulation; information structure, discourse. Coreference. Linguistic typology. Formal grammars and their application in rule-based morphology and parsing.

Recommended courses

Code	Subject	Credits	Winter	Summer
NPFL063	Introduction to General Linguistics	5	2/1 C+Ex	—
NPFL083	Linguistic Theory and Grammar Formalisms	6	—	2/2 C+Ex
NPFL075	Prague Dependency Treebank	6	—	2/2 C+Ex
NPFL094	Morphological and Syntactic Analysis I	3	2/0 MC	—
NPFL006	Introduction to Formal Linguistics	3	2/0 Ex	—

3. Statistical methods and machine learning in computational linguistics

Generative and discriminative models. Supervised machine learning for classification and regression (linear models, other methods: Naive Bayes, decision trees, example-based learning). Support Vector Machines and Kernel functions. Logistic regression. Unsupervised machine learning methods. Bayesian Networks. Bias-variance tradeoff. Language models and noisy channel models. Smoothing, model combination. HMM, trellis, Viterbi, Baum-Welch. Algorithms for statistical tagging. Algorithms for phrase-based and dependency-based statistical parsing.

Recommended courses

Code	Subject	Credits	Winter	Summer
NPFL067	Statistical Methods in Natural Language Processing I	6	2/2 C+Ex	—
NPFL068	Statistical Methods in Natural Language Processing II	6	—	2/2 C+Ex

NPFL104	Machine Learning Methods	5	—	1/2 C+Ex
NPFL087	Statistical Machine Translation	6	—	2/2 C+Ex

4. *Multimodal technologies and data*

Fundamentals of speech production and perception. Methods of speech signal processing. HMM acoustic modeling of phonemes. The implementation of the Baum-Welch and Viterbi algorithms in speech recognition systems. Continuous speech recognition using large dictionaries. Adaptation techniques. Speech summarization. Topic and keyword spotting in speech corpora. Speaker recognition. Methods of speech synthesis. Text processing for speech synthesis. Prosody modeling. Basic components of a dialog system. Spoken language understanding. Dialog control – MDP and POMDP systems. Reinforcement learning. Dialogue state tracking in MDP and POMDP systems. User simulation. Speech generation. Dialog systems quality evaluation. Search and indexing in audio-visual archives.

Recommended courses

Code	Subject	Credits	Winter	Summer
NPFL038	Fundamentals of Speech Recognition and Generation	6	2/2 C+Ex	—
NPFL079	Algorithms in Speech Recognition	6	—	2/2 C+Ex
NPFL099	Statistical Dialogue Systems	5	—	2/1 C+Ex

5. *Applications in natural language processing*

Spell-checking and grammar-checking. Input methods. Machine translation. Machine-aided translation. Statistical methods in machine translation. Quality evaluation of machine translation. Information retrieval, models for information retrieval. Query expansion and relevance feedback. Document clustering. Web search. Duplicate detection and plagiarism detection. Information retrieval evaluation. Sentiment analysis, social network analysis. Search systems (Lucene, SOLR, Terrier). NLP toolkits (GATE, NLTK, NLPTools).

Recommended courses

Code	Subject	Credits	Winter	Summer
NPFL087	Statistical Machine Translation	6	—	2/2 C+Ex
NPFL103	Information Retrieval	6	2/2 C+Ex	—
NPFL093	NLP Applications	5	—	2/1 MC

6 Degree Plans - Artificial Intelligence

Coordinated by: Department of Theoretical Computer Science and Mathematical Logic

Study branch coordinator: Prof. RNDr. Roman Barták, Ph.D.

Specializations:

- Intelligent agents
- Machine learning
- Robotics

The study branch Artificial Intelligence provides education in the area of theoretical and applied knowledge for design of intelligent systems in various areas including data analysis, automated problem solving, and robotic applications. The emphasis is put on deep understanding of formal theoretical foundations and their practical applicability. Students will gain knowledge about design of efficient data structures, about formal modeling of problems and knowledge by using techniques of mathematical logic and probability theory, about algorithms (classical and nature-inspired) for problem solving, for control of autonomous agents, for machine learning, and for data mining, and about complexity analysis of computational methods. The students will learn how to apply these techniques and how to extend them both for abstract (data) and physical (robotic) worlds in single-agent and multi-agent environments. The study branch Artificial Intelligence can be studied in three specializations: Intelligent agents, Machine learning, and Robotics.

Graduates can apply and further extend techniques for the design of intelligent systems, including knowledge modeling and formal modeling of complex systems by means of mathematical logic and probability theory, automated problem solving, planning and scheduling, control of autonomous agents (both virtual and physical), machine learning, and data mining. They are also able to analyse and formally model a complex decision problem, propose an appropriate solving technique, and implement it. Graduates can work in research and development in either academia or industry in any position requiring logical reasoning, analytical capabilities, an algorithmic approach, and the exploitation of modern methods of computer science (declarative and nature-inspired programming).

7.1 Obligatory courses

Code	Subject	Credits	Winter	Summer
NTIN090	Introduction to Complexity and Computability	5	2/1 C+Ex	—
NTIN066	Data Structures I	5	2/1 C+Ex	—
NMAI060	Probabilistic Methods	3	2/0 Ex	—
NAIL070	Artificial Intelligence II	3	—	2/0 Ex
NSZZ023	Diploma Thesis I	6	0/4 C	0/4 C
NSZZ024	Diploma Thesis II	9	0/6 C	0/6 C
NSZZ025	Diploma Thesis III	15	0/10 C	0/10 C

7.2 Elective courses

The student needs to obtain at least 60 credits for the courses from the following set:

Code	Subject	Credits	Winter	Summer
NAIL002	Neural Networks	9	4/2 C+Ex	—
NAIL013	Applications of Neural Networks Theory	3	—	2/0 Ex
NAIL060	Neural Networks Implementation I	6	2/2 C+Ex	—
NAIL021	Boolean Functions and Their Applications	3	2/0 Ex	—

Master of Computer Science

NAIL022	Methods of Logic Programming	3	2/0 Ex	—
NAIL025	Evolutionary Algorithms I	6	2/2 C+Ex	—
NAIL086	Evolutionary Algorithms II	6	—	2/2 C+Ex
NAIL028	Introduction to Robotics	6	2/2 C+Ex	—
NAIL029	Machine Learning	3	—	2/0 Ex
NAIL004	Seminar on Artificial Intelligence I	3	0/2 C	—
NAIL052	Seminar on Artificial Intelligence II	3	—	0/2 C
NAIL061	Seminar on Mobile Robotics	3	—	0/2 C
NAIL065	Evolutionary Robotics	5	—	2/1 C+Ex
NAIL068	Human-like Artificial Agents	6	—	2/2 C+Ex
NAIL069	Artificial Intelligence I	5	2/1 C+Ex	—
NAIL071	Planning and Scheduling	3	—	2/0 Ex
NAIL073	Robot I	3	0/2 C	—
NAIL074	Robot II	3	—	0/2 C
NAIL076	Logic Programming I	3	2/0 Ex	—
NAIL077	Logic Programming II	3	—	2/0 Ex
NAIL078	Lambda Calculus and Functional Programming I	5	2/1 C+Ex	—
NAIL087	Computers and Cognitive Sciences I	6	3/1 C+Ex	—
NAIL094	Decision Procedures and Verification	6	2/2 C+Ex	—
NAIL101	Probabilistic Robotics	6	—	2/2 C+Ex
NAIL104	Probabilistic graphical models	3	2/0 Ex	—
NAIL105	Internet and Classification Methods	3	—	1/1 C+Ex
NAIL106	Multiagent Systems	6	—	2/2 C+Ex
NAIL107	Machine Learning in Bioinformatics	6	—	2/2 C+Ex
NAIL108	Mobile Robotics	3	—	1/1 MC
NDBI023	Data Mining	9	—	4/2 C+Ex
NDBI031	Statistical Methods in Data Mining Systems	3	1/1 C+Ex	—
NMAI061	Methods of Mathematical Statistics	5	—	2/1 C+Ex
NMAI067	Logic in Computer Science	3	2/0 Ex	—
NOPT021	Game Theory	3	2/0 Ex	—
NOPT042	Constraint Programming	6	2/2 C+Ex	—
NPFL067	Statistical Methods in Natural Language Processing I	6	2/2 C+Ex	—
NPFL068	Statistical Methods in Natural Language Processing II	6	—	2/2 C+Ex
NPFL097	Selected Problems in Machine Learning	3	0/2 C	—
NPFL104	Machine Learning Methods	5	—	1/2 C+Ex
NPGR001	Computer vision and robotics	3	2/0 Ex	—
NPRG023	Software Project	9	0/6 C	0/6 C
NPRG027	Credit for Project	6	0/4 C	0/4 C
NPRG037	Microcontroller Programming	6	2/2 C+Ex	—
NSWE001	Embedded and Real Time Systems	6	—	2/2 C+Ex

NSWI054	Software Engineering for Dependable Systems	3	—	0/2 MC
---------	---	---	---	--------

7.3 Other recommended courses

Code	Subject	Credits	Winter	Summer
NAIL015	Neural Networks Implementation II	6	—	2/2 C+Ex
NAIL031	Representations of Boolean Functions	3	—	2/0 Ex
NAIL079	Lambda Calculus and Functional Programming II	5	—	2/1 C+Ex
NAIL110	Practical Course in Robotics	6	—	1/3 MC
NPFL038	Fundamentals of Speech Recognition and Generation	6	2/2 C+Ex	—
NPFL054	Introduction to Machine Learning	6	2/2 C+Ex	—
NPFL079	Algorithms in Speech Recognition	6	—	2/2 C+Ex
NPFL087	Statistical Machine Translation	6	—	2/2 C+Ex
NPFL103	Information Retrieval	6	2/2 C+Ex	—
NPGR002	Digital Image Processing	5	3/0 Ex	—
NTIN084	Bioinformatics Algorithms	6	2/2 C+Ex	—
NSWI103	Project Management – System Dynamics I	3	0/2 C	—
NSWI104	Company Management – System Dynamics II	3	—	0/2 C

7.4 State Final Exam

In addition to the two examination areas that are obligatory for all study branches, the student will select three examination areas from the following lists. At least two examination areas must be selected from a chosen specialization (one area is obligatory for the specialization Intelligent agents), one area may be selected from another specialization. In total, each student will get five questions.

a) Specialization *Intelligent agents*

Examination areas

1. Knowledge representation (obligatory for the specialization Intelligent agents)
2. Problem solving and planning
3. Nonprocedural programming
4. Multi-agent systems
5. Nature inspired computing

Knowledge requirements

1. Knowledge representation

Formal systems, first-order logic, syntax, axioms, inference rules. Propositional logic, its semantics, tautology and satisfiability, provability. Normal forms of formulae. Automated theorem proving, model checking (DPLL), forward and backward chaining, resolution and unification.

Conditional independence, Bayesian networks, evaluation in Bayesian networks, naive Bayes classifier, decision graphs, MDP, POMDP, reinforcement learning, conditional random fields.

Recommended courses

Code	Subject	Credits	Winter	Summer
NAIL062	Propositional and Predicate Logic	6	2/2 C+Ex	—
NMAI060	Probabilistic Methods	3	2/0 Ex	—
NAIL069	Artificial Intelligence I	5	2/1 C+Ex	—
NAIL070	Artificial Intelligence II	3	—	2/0 Ex
NAIL104	Probabilistic graphical models	3	2/0 Ex	—

2. *Problem solving and planning*

Knowledge representation: state space, production systems, logical representation. Search algorithms: tree, graph, and local search, heuristics. SAT solving and constraint satisfaction. Problem modeling. Automated planning: planning domain and problem, planning operators. Planning techniques and algorithms, heuristics, and extensions of planning techniques. Planning and scheduling.

Recommended courses

Code	Subject	Credits	Winter	Summer
NAIL069	Artificial Intelligence I	5	2/1 C+Ex	—
NAIL071	Planning and Scheduling	3	—	2/0 Ex
NOPT042	Constraint Programming	6	2/2 C+Ex	—
NAIL094	Decision Procedures and Verification	6	2/2 C+Ex	—

3. *Nonprocedural programming*

Differences between procedural and non-procedural styles of programming. Principles of functional and logic programming. Lambda calculus, its syntax, and reduction principles. Church and Rosser property and consistency of calculus. Fixed point theorems. Normal form of terms. Typed lambda calculus.

Substitution and unification. Horn clauses, SLD resolution and logic programs. Pure Prolog, negation as failure, general logic programs. Sufficient conditions of program termination. Implementation of Prolog. Constraint logic programming.

Recommended courses

Code	Subject	Credits	Winter	Summer
NAIL076	Logic Programming I	3	2/0 Ex	—
NAIL077	Logic Programming II	3	—	2/0 Ex
NAIL078	Lambda Calculus and Functional Programming I	5	2/1 C+Ex	—
NAIL022	Methods of Logic Programming	3	2/0 Ex	—
NOPT042	Constraint Programming	6	2/2 C+Ex	—

4. *Multi-agent systems*

Autonomous agent architectures; agent perception, agent action selection mechanism, agent memory. Psychological inspiration. Methods for agent control; symbolic

and connectionist reactive planning, hybrid approaches. Path search problem, steering rules, terrain representation. Communication and knowledge in multiagent systems, ontologies, speech acts, FIPA-ACL, protocols. Distributed problem solving, cooperation, Nash equilibria, Pareto efficiency, source allocation, auctions. Agent design methodologies, agent languages and environments. Ethological inspiration, models of population dynamics. Methods for agent learning; reinforcement learning, basic forms of animal learning.

Recommended courses

Code	Subject	Credits	Winter	Summer
NAIL106	Multiagent Systems	6	—	2/2 C+Ex
NAIL068	Human-like Artificial Agents	6	—	2/2 C+Ex

5. *Nature inspired computing*

Genetic algorithms, genetic and evolutionary programming. Schemata theory, probabilistic models of simple genetic algorithm. Evolutionary strategies, differential evolution, coevolution, open ended evolution. Swarm optimization algorithms. Memetic algorithms, hill climbing, simulated annealing. Application of evolutionary algorithms (expert systems evolution, neuroevolution, combinatorial optimization, multi-objective optimization).

Recommended courses

Code	Subject	Credits	Winter	Summer
NAIL025	Evolutionary Algorithms I	6	2/2 C+Ex	—
NAIL086	Evolutionary Algorithms II	6	—	2/2 C+Ex
NAIL065	Evolutionary Robotics	5	—	2/1 C+Ex

b) Specialization *Machine learning*

Examination areas

1. Machine learning and its applications
2. Neural networks
3. Data mining

Knowledge requirements

1. *Machine learning and its applications*

Machine learning; version space search, supervised learning and self-organization, probabilistic approaches, theoretical aspects of machine learning. Evolutionary algorithms; fundamental concepts and theoretical knowledge, the building block hypothesis, coevolution, applications of evolutionary algorithms. Machine learning in computational linguistics and statistical parsing algorithms. Probabilistic algorithms for biological sequence analysis; search for motifs in DNA sequences, strategies for gene detection and protein structure prediction.

Recommended courses

Code	Subject	Credits	Winter	Summer
NAIL029	Machine Learning	3	—	2/0 Ex

NPFL067	Statistical Methods in Natural Language Processing I	6	2/2 C+Ex	—
NAIL025	Evolutionary Algorithms I	6	2/2 C+Ex	—
NAIL107	Machine Learning in Bioinformatics	6	—	2/2 C+Ex

2. *Neural networks*

Neurophysiological minimum. Models for supervised learning, the back-propagation algorithm, strategies to speed up the training process, regularization techniques and generalization. Associative memories; Hebbian learning and the search for suboptimal solutions, stochastic models. Artificial neural networks based on the principle of unsupervised learning. Modular, hierarchical and hybrid models of neural networks. Genetic algorithms and their application for training of artificial neural networks.

Recommended courses

Code	Subject	Credits	Winter	Summer
NAIL002	Neural Networks	9	4/2 C+Ex	—
NAIL060	Neural Networks Implementation I	6	2/2 C+Ex	—
NAIL013	Applications of Neural Networks Theory	3	—	2/0 Ex
NAIL065	Evolutionary Robotics	5	—	2/1 C+Ex

3. *Data mining*

Basic paradigms of data mining. Data preparation, attribute selection and methods for relevance analysis of attributes. Data mining methods; association rules, approaches based on supervised learning and cluster analysis. Methods for the extraction of characteristic and discriminant rules and measures of their interestingness. Representation, evaluation and visualization of the extracted knowledge. Models and evaluation of information retrieval. Practical applications of data mining techniques and methods for information retrieval.

Recommended courses

Code	Subject	Credits	Winter	Summer
NDBI023	Data Mining	9	—	4/2 C+Ex
NPFL103	Information Retrieval	6	2/2 C+Ex	—
NAIL105	Internet and Classification Methods	3	—	1/1 C+Ex
NAIL099	Seminar of machine learning and modelling I	2	0/1 C	—

c) Specialization **Robotics**

Examination areas

1. Localization and mapping
2. Control systems
3. Robotic systems
4. Planning and navigation

Knowledge requirements*1. Localization and mapping*

Basic localization methods. Probabilistic localization, particle filters, Monte-Carlo methods. Environment representation, map formats, correspondence problem, mapping in dynamic environment. Localization and mapping relation, SLAM.

Recommended courses

Code	Subject	Credits	Winter	Summer
NAIL028	Introduction to Robotics	6	2/2 C+Ex	—
NAIL101	Probabilistic Robotics	6	—	2/2 C+Ex
NAIL108	Mobile Robotics	3	—	1/1 MC

2. Control systems

Robot control systems. Signal processing, object recognition, feature matching and tracking. Modeling systems, virtual robotics, simulators. Distributed algorithms, multirobot control systems, communication, synchronization, coordination. Software implementation, programming for specific runtime environment, debugging tools and techniques.

Recommended courses

Code	Subject	Credits	Winter	Summer
NAIL028	Introduction to Robotics	6	2/2 C+Ex	—
NPGR001	Computer vision and robotics	3	2/0 Ex	—
NPGR002	Digital Image Processing	5	3/0 Ex	—
NSWI035	Principles of Distributed Systems	3	2/0 Ex	—

3. Robotic systems

Basic kinematic and dynamic model, inverse kinematics and dynamics. Low-level hardware and software, embedded systems. Sensor and actuator types, principles and typical usage. High-level robot systems and their control: manipulators, mobile robotics, autonomous robotics.

Recommended courses

Code	Subject	Credits	Winter	Summer
NAIL028	Introduction to Robotics	6	2/2 C+Ex	—
NAIL108	Mobile Robotics	3	—	1/1 MC
NSWE001	Embedded and Real Time Systems	6	—	2/2 C+Ex

4. Planning and navigation

Basic navigation techniques: dead-reckoning, odometry, triangulation, and trilateration, inertial navigation. Navigation and exploration algorithms. Action planning, planning problem formulation, basic planning and planning with time and resources.

Recommended courses

Code	Subject	Credits	Winter	Summer
NAIL028	Introduction to Robotics	6	2/2 C+Ex	—
NAIL108	Mobile Robotics	3	—	1/1 MC
NAIL071	Planning and Scheduling	3	—	2/0 Ex

7 Degree Plans - Computer Graphics and Game Development

Coordinated by: Department of Software and Computer Science Education

Study branch coordinator: Doc. Ing. Jaroslav Křivánek, Ph.D.

Specializations:

- Computer graphics
- Computer game development

The study branch consists of two closely related specializations, Computer graphics and Computer game development. The specialization Computer graphics offers training in a wide range of visual sciences, including geometric modeling, rendering (image synthesis) as well as the basics of image analysis and computer vision. The specialization Computer game development focuses - apart from computer graphics techniques - mainly on artificial intelligence and intelligent agent systems, as well as on software engineering skills necessary for the development of large-scale gaming projects. Both specializations place emphasis on general programming skills, both at the system level closer to the hardware, as well as on the higher level of modern programming languages.

Graduates have expertise in the design and development of graphical systems and computer games, but they can work in any position which requires logical thinking, analytic and algorithmic approaches or the use of methods of computer science. Depending on the chosen specialization, graduates have a deep knowledge of computer graphics and image analysis, and their expertise covers the development of large-scale gaming projects, real-time applications, programming of portable devices, as well as the foundations of artificial intelligence and computer graphics in the context of computer games. Graduates can apply this knowledge to solve specific practical tasks. They can work in research and development both in the private sector and in academia.

8.1 Obligatory courses

Code	Subject	Credits	Winter	Summer
NTIN090	Introduction to Complexity and Computability	5	2/1 C+Ex	—
NTIN066	Data Structures I	5	2/1 C+Ex	—
NMAI060	Probabilistic Methods	3	2/0 Ex	—
NPRG027	Credit for Project	6	0/4 C	0/4 C
NPRG023	Software Project	9	0/6 C	0/6 C
NSZZ023	Diploma Thesis I	6	0/4 C	0/4 C
NSZZ024	Diploma Thesis II	9	0/6 C	0/6 C
NSZZ025	Diploma Thesis III	15	0/10 C	0/10 C

8.2 Elective courses

The student needs to obtain at least 47 credits for the courses from the following set:

Code	Subject	Credits	Winter	Summer
NPGR007	Advanced 2D Computer Graphics	5	2/1 C+Ex	—

NPGR010	Computer Graphics III	6	2/2 C+Ex	—
NPGR001	Computer vision and robotics	3	2/0 Ex	—
NPGR021	Geometric Modelling	6	—	2/2 C+Ex
NPGR026	Predictive Image Synthesis Technologies	6	—	2/2 C+Ex
NPGR027	Shading Languages	5	—	2/1 C+Ex
NPGR028	High Performance Ray Tracing	3	—	2/0 Ex
NPGR033	Computer Graphics for Game Development	6	—	2/2 C+Ex
NPGR024	Seminar on Scientific Soft Skills	3	—	0/2 C
NPGR029	Variational methods in image processing	3	—	2/0 Ex
NSWI115	Computer Games Development	6	2/2 C+Ex	—
NPRG056	Developing Applications for Mobile Devices	3	0/2 C	—
NSWI035	Principles of Distributed Systems	3	2/0 Ex	—
NPRG054	High Performance Software Development	5	—	2/2 C+Ex
NPRG042	Programming in Parallel Environment	6	—	2/2 C+Ex
NAIL069	Artificial Intelligence I	5	2/1 C+Ex	—
NAIL070	Artificial Intelligence II	3	—	2/0 Ex
NPRG059	Advanced Programming Praxis	2	0/1 C	—
NSWI041	Introduction to Software Engineering	5	—	2/2 C+Ex
NAIL068	Human-like Artificial Agents	6	—	2/2 C+Ex
NAIL071	Planning and Scheduling	3	—	2/0 Ex
NPRG043	Recommended Programming Practices	6	—	2/2 C+Ex
NMAI061	Methods of Mathematical Statistics	5	—	2/1 C+Ex
NAIL106	Multiagent Systems	6	—	2/2 C+Ex
NPGR016	Applied Computational Geometry	5	—	2/1 C+Ex
NPRG014	Concepts of Modern Programming Languages	2	0/2 C	—

8.3 Other recommended courses

Code	Subject	Credits	Winter	Summer
NPGR005	Special Seminar for Computer Graphics	2	0/2 C	0/2 C
NPGR013	Special Functions and Transformations in Image Processing	3	—	2/0 Ex
NPGR022	Advanced Seminar On Image Processing	2	0/2 C	0/2 C
NAIL028	Introduction to Robotics	6	2/2 C+Ex	—

NAIL108	Mobile Robotics	3	—	1/1 MC
NSWI072	Data Compression Algorithms	3	2/0 Ex	—
NPGR012	Virtual Reality	6	2/2 C+Ex	2/2 C+Ex
NPGR030	Optics for computer graphics	3	2/0 Ex	—
NPGR023	Visualization	5	—	2/1 C+Ex
NAIL025	Evolutionary Algorithms I	6	2/2 C+Ex	—
NAIL082	Seminar on Humanlike Artificial Agents	3	0/2 C	0/2 C
NAIL087	Computers and Cognitive Sciences I	6	3/1 C+Ex	—
NPOZ017	Video Games as a Socio-cultural Phenomenon	3	2/0 Ex	—
NSWI149	Software Engineering in Practice	3	—	2/0 C
NPRG058	Advanced Programming in Parallel Environment	3	0/2 C	—

8.4 State Final Exam

a) Specialization *Computer graphics*

In addition to the two examination areas that are obligatory for all study branches, the student will select three other examination areas. Students have to choose at least two examination areas from the following list of examination areas; the third examination area can be chosen at will also from among all the examination areas offered in the specialization Computer game development, except the area Computer graphics for games. Students will be asked one question from each selected examination area. In total, each student will get five questions.

Examination areas

1. Geometric modeling and computational geometry
2. Image analysis and processing, image compression, computer vision
3. Realistic image synthesis

Knowledge requirements

1. Geometric modeling and computational geometry

Homogeneous coordinates, affine and projective transformations in the plane and in space, quaternions in the representation of the 3D orientation, differential geometry of curves and surfaces, spline functions, cubic splines C2, interpolation by cubic splines, Bezier curves, Catmull-Rom splines, B-splines, de Casteljau and de Boor algorithms, approximation surfaces, surfaces defined by the edges, Bezier surfaces, patch stitching, B-spline surfaces, NURBS surfaces. Design of geometric algorithms and their complexity, triangulation of polygons and point sets, Voronoi diagrams and Delaunay triangulations, convex hull, intersections of geometric shapes, dualization, localization, data structures for efficient spatial search.

Recommended courses

Code	Subject	Credits	Winter	Summer
NPGR003	Computer Graphics I	6	2/2 C+Ex	—
NPGR016	Applied Computational Geometry	5	—	2/1 C+Ex
NPGR021	Geometric Modelling	6	—	2/2 C+Ex

NPGR007	Advanced 2D Computer Graphics	5	2/1 C+Ex	—
---------	-------------------------------	---	----------	---

2. *Image analysis and processing, image compression, computer vision*

Mathematical models of images, 2D Fourier transform and convolution, sampling and quantization of the image, contrast and brightness change, noise removal, edge detection, inverse and Wiener filter, determining the relative position of images - registration, point and object correspondence, geometric distortion removal, edge detection, detection of areas, features for description and recognition of 2D objects, moment invariants, wavelets and their use, statistical theory of pattern recognition, supervised and unsupervised classification, computer vision.

Reproduction of color graphics, halftoning and dithering, alpha blending and image composition, warping, morphing, compression of raster 2D graphics, scalar and vector quantization, predictive compression, transformation compression methods, hierarchical and progressive methods, video compression, temporal prediction (motion compensation), JPEG and MPEG standards, image capture in digital photography.

Recommended courses

Code	Subject	Credits	Winter	Summer
NPGR003	Computer Graphics I	6	2/2 C+Ex	—
NPGR002	Digital Image Processing	5	3/0 Ex	—
NPGR029	Variational methods in image processing	3	—	2/0 Ex
NPGR013	Special Functions and Transformations in Image Processing	3	—	2/0 Ex
NPGR007	Advanced 2D Computer Graphics	5	2/1 C+Ex	—
NPGR001	Computer vision and robotics	3	2/0 Ex	—

3. *Realistic image synthesis*

Representation of 3D scenes, visibility determination, cast shadows, reflection models and shading algorithms, recursive ray tracing, textures, anti-aliasing, acceleration methods for ray-tracing, principle of radiosity methods, physically-based models of light transport (radiometry, rendering equation), Monte Carlo integration, Monte Carlo approaches in lighting simulation, direct methods in volume data visualization, isosurface extraction, shading languages, graphics accelerator architecture, data transfer to the GPU, textures in the GPU, GPU programming: shaders, basics of OpenGL, Cg and GLSL, CUDA, advanced GPU programming techniques.

Recommended courses

Code	Subject	Credits	Winter	Summer
NPGR003	Computer Graphics I	6	2/2 C+Ex	—
NPGR004	Computer Graphics II	5	—	2/1 C+Ex
NPGR019	Hardware for Computer Graphics	5	—	2/1 C+Ex
NPGR010	Computer Graphics III	6	2/2 C+Ex	—
NPGR026	Predictive Image Synthesis Technologies	6	—	2/2 C+Ex
NPGR027	Shading Languages	5	—	2/1 C+Ex

b) Specialization **Computer game development**

In addition to the two examination areas that are obligatory for all study branches, students get one question from each of the following examination areas:

1. Multi-agent systems (this area is obligatory for the specialization Computer game development).
2. Either the area Computer graphics for games or any area from the specialization Computer graphics.
3. One of the following areas: Parallel programming, System aspects of computers, Modern concepts of programming, Software engineering, Advanced programming, Web technologies.

In total, each student will get five questions.

Examination areas

1. Multi-agent systems (obligatory for the specialization Computer game development)
2. Computer graphics for games
3. Parallel programming
4. System aspects of computers
5. Modern programming concepts
6. Software analysis and architectures
7. Advanced programming
8. Web technologies

Knowledge requirements

1. *Multi-agent systems*

Autonomous agent architectures; agent perception, agent action selection mechanism, agent memory. Psychological inspiration. Methods for agent control; symbolic and connectionist reactive planning, hybrid approaches. Path search problem, steering rules, terrain representation. Communication and knowledge in multiagent systems, ontologies, speech acts, FIPA-ACL, protocols. Distributed problem solving, cooperation, Nash equilibria, Pareto efficiency, source allocation, auctions. Agent design methodologies, agent languages and environments. Ethological inspiration, models of population dynamics. Methods for agent learning; reinforcement learning, basic forms of animal learning.

Recommended courses

Code	Subject	Credits	Winter	Summer
NAIL054	Adaptive Agents	3	—	0/2 C
NAIL068	Human-like Artificial Agents	6	—	2/2 C+Ex
NAIL096	Multi-agent systems	6	2/2 C+Ex	—

2. *Computer graphics for games*

Homogeneous coordinates, affine and projective transformations in the plane and in space, quaternions, splines, interpolation by cubic splines, Bezier curves, Catmull-Rom splines, B-splines, de Casteljau and de Boor algorithms. Character animation, skinning, rigging. Collision detection, rigid and soft body dynamics, fluid dynamics.

2D Fourier transform and convolution, sampling and quantization, image anti-aliasing, texture, change of contrast and brightness, alpha-bending and compositing, compression of raster 2D graphics, video signal compression, temporal prediction (motion compensation), JPEG and MPEG standards.

Representation of 3D scenes, visibility, cast shadows, soft shadows, subsurface scattering, lighting models and shading algorithms, recursive ray tracing, physically-based model of light transport (radiometry, imaging equation), path tracing, precomputed global illumination, real-time global illumination, spherical harmonics-based shading, precomputed radiance transfer.

Graphics accelerator architecture, data transfer to the GPU, textures on the GPU, GPU programming: shaders, basic OpenGL, Cg and GLSL, CUDA, advanced techniques for GPU programming. Game engine architecture.

Recommended courses

Code	Subject	Credits	Winter	Summer
NPGR003	Computer Graphics I	6	2/2 C+Ex	—
NPGR004	Computer Graphics II	5	—	2/1 C+Ex
NPGR019	Hardware for Computer Graphics	5	—	2/1 C+Ex
NPGR033	Computer Graphics for Game Development	6	—	2/2 C+Ex

3. *Parallel programming*

Modern memory architectures (multiprocessor, GPU, NUMA), relationship between data location and access efficiency. Virtualization support. Converting computations for data parallelism, parallel solutions of non-homogeneous tasks. Efficient parallel implementation of basic algorithms. Synchronization, load balancing.

Recommended courses

Code	Subject	Credits	Winter	Summer
NPRG042	Programming in Parallel Environment	6	—	2/2 C+Ex
NPRG054	High Performance Software Development	5	—	2/2 C+Ex

4. *System aspects of computers*

Process and thread management, communication (IPC), synchronization, memory models. Memory management (paging, heap structure, garbage collection). Runtime support, calling conventions (ABI), virtualization. Device management. File systems. Performance metrics, methods for performance evaluation, simulation and modeling. All topics considered in the context of contemporary parallel architectures.

Recommended courses

Code	Subject	Credits	Winter	Summer
NSWI004	Operating Systems	5	2/1 C+Ex	—
NSWI131	Performance Evaluation of Computer Systems	5	—	2/1 C+Ex

5. *Modern programming concepts*

Modern object-oriented concepts, metaprogramming, reflection, aspects. Features of modern programming languages (for example annotations, iterators, generics, lambda functions). Code quality metrics, documentation, refactoring, testing and debugging. Design patterns. All topics considered in the context of contemporary programming languages.

Recommended courses

Code	Subject	Credits	Winter	Summer
NPRG043	Recommended Programming Practices	6	—	2/2 C+Ex
NSWI068	Object and Component Systems	5	2/2 C+Ex	—

6. *Software analysis and architectures*

SW development processes, development phases. Business processes and their modeling using BPMN. UML and its use for analysis and design of structure and behavior of SW. Design patterns. SW testing, impact and change analysis. SW project planning, cost estimation, levels of project management. Legal aspects of SW, principal legal environment for IT projects. Types of SW architecture. Modeling and documentation of SW architecture. Classification of SW architecture quality attributes, their description using scenarios and tactics. Service oriented architectures. Algebraic methods, many sorted algebras, initial models. Formal principles of the UML language. OCL as a specification language, formal base of specification. Formal principles of RDF and OWL, the description logic.

Recommended courses

Code	Subject	Credits	Winter	Summer
NSWI130	Software System Architectures	5	2/2 C+Ex	—
NSWI041	Introduction to Software Engineering	5	—	2/2 C+Ex
NSWI094	Database Application Development	5	2/1 C+Ex	—
NTIN043	Formal Foundations of Software Engineering	5	2/2 C+Ex	—

7. *Advanced programming*

Object concepts of modern languages. Concepts of classless languages. Generic programming and metaprogramming, generics and templates, policies, traits, type inference. Modern constructs of programming languages. Object references and lifetime. Advanced aspects of imperative languages. Exceptions, exception-safe programming. Implementation of object properties, runtime support, calling conventions, garbage collection. Modern language constructs and code performance. Design patterns. Scripting languages, prototype-based languages. Domain-specific languages. Functional programming. Principles of code quality, best practices. Refactoring. Testing, debugging, monitoring.

Recommended courses

Code	Subject	Credits	Winter	Summer
NPRG059	Advanced Programming Praxis	2	0/1 C	—

NPRG014	Concepts of Modern Programming Languages	2	0/2 C	—
NPRG024	Design Patterns	3	—	0/2 MC
NPRG043	Recommended Programming Practices	6	—	2/2 C+Ex
NSWI068	Object and Component Systems	5	2/2 C+Ex	—

8. Web technologies

Overview of basic web technologies. Network services for web technologies. Web services. Architecture of client-server applications, server-side and client-side scripting, web frameworks. Database systems in web applications, NoSQL databases, multimedia databases. Indexing and document searching, principles of web search engines. Linked Data, integration of semantic data to web pages. Security of information systems in the Internet environment, authentication, authorization, security models, cryptography basics, data security.

Recommended courses

Code	Subject	Credits	Winter	Summer
NSWI130	Software System Architectures	5	2/2 C+Ex	—
NSWI153	Advanced Technologies for Web Applications	3	—	2/0 C
NSWI145	Web Services	5	—	2/2 C+Ex
NDBI034	Retrieval of Multimedia Content on the Web	4	2/1 C+Ex	—
NPRG043	Recommended Programming Practices	6	—	2/2 C+Ex

Study started in 2014 and earlier

1 General Information

The study programme Master of Computer Science consists of the following study branches and their specializations:

- 1 Theoretical Computer Science
 - Algorithms and complexity
 - Nonprocedural programming and artificial intelligence
- 2 Mathematical Linguistics
- 3 Discrete Models and Algorithms
 - Discrete mathematics and combinatorial optimization
 - Mathematical structures of computer science
 - Optimization

While your study branch has been specified already in your application, the choice of your specialization (when applicable) is up to your later decision (when enrolling for the state final exam). In this initial part of the present chapter we describe issues that are common for all study branches of the study programme Master of Computer Science.

Assumed knowledge

It is assumed that an incoming student has a sufficient knowledge of mathematics, theoretical computer science and programming. In particular, students are expected to have a good knowledge of mathematics at the level of our bachelor courses NMAI054 Mathematical Analysis I, NMAI055 Mathematical Analysis II, NMAI059 Probability and Statistics, NMAI062 Algebra I, and NOPT048 Optimization Methods.

Students are also expected to have knowledge equivalent to the courses NDMI002 Discrete Mathematics, NTIN060 Algorithms and Data Structures I, NTIN061 Algorithms and Data Structures II, NTIN071 Automata and Grammars, and NAIL062 Propositional and Predicate Logic. Knowledge from these courses is also expected for the state final exam.

We also expect students to have good knowledge of programming at least at the level of our bachelor courses NPRG030 Programming I, NPRG031 Programming II.

Students missing knowledge in some of the above mentioned areas are suggested to consider taking the relevant bachelor courses in the first year of their Master studies.

Software Project

One of the obligatory courses for students of Mathematical Linguistics is NPRG023 Software Project, a team software project; for students of the other study branches this course is optional. The goal of this subject is to practice team work in large software projects lasting typically around 9 months. The work on the project is finished by public presentation.

Students can enroll in the courses NPRG023 Software Project and NPRG027 Course Credit for Project anytime, not just in the usual enrollment period of a given academic year. Nevertheless, each student can try to complete these courses at most twice during the studies.

State Final Exam

The state final exam consists of two parts: a defense of the Master's (diploma) thesis and an oral examination. The student can enroll for each part separately. To finish the studies, both parts of the state final exam must be completed successfully.

Requirements to enroll for the state final exam

- obtaining at least 120 credits,
- passing all obligatory courses of a given branch and specialization,
- obtaining a given number of credits from the elective courses of a given branch and specialization,
- submitting the Master's thesis by the specified deadline (for defence of the Master's thesis).

Master's (diploma) thesis

Students are advised to select the topic of their Master's (diploma) thesis at the end of the first semester. The departments of the faculty offer many topics for Master's theses each year and students can also suggest their own topics. We recommend to select the topic of your thesis primarily from the offer of the department that coordinates your study branch. If you prefer a topic offered by another department or your own topic, please consult it with the coordinator of your study branch.

After the topic is assigned, the student enrolls in the following obligatory courses:

Code	Subject	Credits	Winter	Summer
NSZZ023	Diploma Thesis I	6	—	0/4 C
NSZZ024	Diploma Thesis II	9	0/6 C	—
NSZZ025	Diploma Thesis III	15	—	0/10 C

Course credits for these courses are appointed by the thesis advisor on the basis of student's work on the thesis. We suggest the students to discuss with their advisor the expected amount of work and the milestones for each of these courses. All three courses can be enrolled in both winter and summer semesters.

Oral examination

The oral part of the state final exam has the same structure for all study branches. The student is examined from two or three obligatory areas covering the theoretical foundations of computer science (complexity, computability, and data structures) and from three optional areas specific to a given study branch and selected specialization. For the study branches Theoretical Computer Science and Discrete Models and Algorithms the student will select these three optional examination areas when registering for the final exam. Two of these areas must be from the selected specialization, the remaining area can be selected from other specializations of the study branch. The study branch Mathematical Linguistics has no specializations and the optional part of the oral examination has specific rules (see below).

To simplify the orientation in the courses offered by the faculty, the description of each examination area is accompanied by a list of recommended and supplementary courses. Note that not all the courses are available in English every year. We recommend students to contact the study branch coordinator for the selected study branch and discuss individual study plans prior the beginning of a semester.

The obligatory examination areas for the study branches Theoretical Computer Science and Discrete Models and Algorithms:

1. Complexity

Hierarchical theorems, constructible functions, relationships among time and space complexities and among determinism and nondeterminism, Savitch's theorem, complete problems for distinct complexity classes (NP, PSPACE, P, P), polynomial time hierarchy, pseudopolynomial algorithms, strong NP-completeness, complexity class P and P-completeness, approximation algorithms and schemes, algorithmic paradigms: dynamic programming, exhaustive search, greedy algorithms, randomized algorithms.

Recommended courses: NTIN062 Complexity I, NTIN063 Complexity II

Supplementary courses: NTIN081 Structural Complexity I, NTIN085 Selected Topics in Computational Complexity I, NTIN017 Parallel Algorithms, NDMI025 Randomized Algorithms

2. Computability

Algorithmically computable functions, their properties, equivalence of distinct mathematical definitions, recursive and partial recursive functions, recursive and recursively enumerable sets and their properties, algorithmic undecidable problems (halting problem), recursion theorem and its applications, Rice theorem, Gödel theorems.

Recommended courses: NTIN064 Computability I, NTIN065 Computability II

Supplementary courses: NTIN073 Recursion I, NTIN074 Recursion II

3. Data structures

Tree data structures, binary search trees and balanced binary search trees, heaps, trie, B-trees and their modifications, hashing (collisions), universal hashing, perfect hashing, sorting in internal and external memory, lower bound on sorting (decision trees), dynamisation of data structures, self-organizing data structures, relaxed search trees.

Recommended courses: NTIN066 Data Structures I, NTIN067 Data Structures II

Supplementary courses: NTIN083 Seminar on Data Structures I

The obligatory examination areas for the study branch Mathematical Linguistics

1. Complexity and computability

Methods of algorithm development: divide and conquer, dynamic programming, greedy algorithms. Amortized complexity, NP-complete problems, Cook-Levin theorem, pseudopolynomial algorithms, strong NP-completeness, complexity class P and P-completeness, approximation algorithms and schemes, algorithmic paradigms: dynamic programming, exhaustive search, greedy algorithms, divide and conquer, algorithmically computable functions, their properties, equivalence of distinct mathematical definitions, partial recursive functions, recursive and recursively enumerable sets and their properties, algorithmically undecidable problems (halting problem), recursion theorem and its applications, Rice theorem.

Recommended courses: NTIN090 *Introduction to Complexity and Computability Theory*

Supplementary courses: see above for the courses for topics 1 and 2 of the study branches I1 and I4.

2. Data structures

Tree data structures, binary search trees and balanced binary search trees, heaps, tries, B-trees and their modifications, hashing (collisions), universal hashing, perfect hashing, sorting in internal and external memory, lower bound on sorting (decision trees), relaxed search trees.

Recommended courses: NTIN066 *Data Structures I*

Supplementary courses: NTIN067 *Data Structures II*, NTIN083 *Seminar on Data Structures I*

2 Degree Plans - Theoretical Computer Science I1

Coordinated by: Department of Theoretical Computer Science and Mathematical Logic

Study branch coordinator: Prof. RNDr. Roman Barták, Ph.D.

2.1 Obligatory courses

Code	Subject	Credits	Winter	Summer
NTIN062	Complexity I	5	2/1 C+Ex	—
NTIN064	Computability	3	—	2/0 Ex
NTIN066	Data Structures I	5	2/1 C+Ex	—
NMAI064	Mathematical Structures	6	—	2/2 C+Ex
NSZZ023	Diploma Thesis I	6	—	0/4 C
NSZZ024	Diploma Thesis II	9	0/6 C	—
NSZZ025	Diploma Thesis III	15	—	0/10 C

2.2 Elective courses

The student needs to obtain at least 60 credits for the courses from this set.

Code	Subject	Credits	Winter	Summer
NTIN063	Complexity	5	—	2/1 C+Ex
NTIN065	Computability II	3	—	2/0 Ex
NTIN067	Data Structures II	3	—	2/0 Ex
NAIL076	Logic Programming I	3	2/0 Ex	—
NAIL077	Logic Programming II	3	—	2/0 Ex
NAIL069	Artificial Intelligence I	5	2/1 C+Ex	—
NAIL070	Artificial Intelligence II	3	—	2/0 Ex
NMAI060	Probabilistic Methods	3	2/0 Ex	—
NMAI061	Methods of Mathematical Statistics	5	—	2/1 C+Ex

NTIN073	Recursion	3	2/0 Ex	—
NTIN074	Recursion II	5	—	2/1 C+Ex
NDMI010	Graph Algorithms	3	2/0 Ex	—
NTIN017	Parallel Algorithms	3	—	2/0 Ex
NDMI007	Combinatorial Algorithms	6	—	2/2 C+Ex
NTIN087	String Algorithms	3	2/0 Ex	—
NAIL078	Lambda Calculus and Functional Programming I	5	2/1 C+Ex	—
NAIL079	Lambda Calculus and Functional Programming II	5	—	2/1 C+Ex
NAIL021	Boolean Functions and Their Applications	3	2/0 Ex	—
NAIL031	Representations of Boolean Functions	3	—	2/0 Ex
NAIL002	Neural Networks	9	4/2 C+Ex	—
NDBI023	Data Mining	9	—	4/2 C+Ex
NAIL013	Applications of Neural Networks Theory	3	—	2/0 Ex
NAIL060	Neural Networks Implementation I	6	2/2 C+Ex	—
NAIL015	Neural Networks Implementation II	6	—	2/2 C+Ex
NTIN018	Probabilistic Analysis of Algorithms	3	2/0 Ex	—
NAIL071	Planning and Scheduling	3	—	2/0 Ex
NAIL029	Machine Learning	3	—	2/0 Ex
NAIL022	Methods of Logic Programming	3	2/0 Ex	—
NOPT042	Constraint Programming	6	2/2 C+Ex	—
NDMI025	Randomized Algorithms	6	—	2/2 C+Ex
NTIN081	Structural Complexity	3	—	2/0 Ex
NTIN082	Computational Complexity	3	2/0 Ex	—
NTIN084	Bioinformatics Algorithms	6	2/2 C+Ex	—
NTIN085	Selected Topics in Computational Complexity I	5	2/1 C+Ex	—
NTIN086	Selected Topics in Computational Complexity II	5	—	2/1 C+Ex
NAIL025	Evolutionary Algorithms I	6	2/2 C+Ex	—
NAIL086	Evolutionary Algorithms II	6	—	2/2 C+Ex
NAIL065	Evolutionary Robotics	5	—	2/1 C+Ex
NAIL068	Human-like Artificial Agents	6	—	2/2 C+Ex
NAIL087	Computers and Cognitive Sciences I	6	3/1 C+Ex	—
NAIL094	Decision Procedures and Verification	6	2/2 C+Ex	—
NAIL028	Introduction to Robotics	6	2/2 C+Ex	—
NPGR001	Computer vision and robotics	3	2/0 Ex	—
NSWE001	Embedded and Real Time Systems	6	—	2/2 C+Ex
NAIL101	Probabilistic Robotics	6	—	2/2 C+Ex

2.3 State Final Exam

a) Specialization *Algorithms and Complexity*

Examination areas

1. Recursion and structural complexity
2. Theory of algorithms
3. Concrete algorithms

Knowledge requirements

1. *Recursion and structural complexity*

Arithmetical hierarchy of classes of sets, classes of infinite branches of computable trees, low basis theorem, diagonally non-recursive functions, arithmetical forcing, 1-generic sets, algorithmic randomness, properties of 1-random sets and their degrees probabilistic complexity classes, Shannon theorem, non-uniform complexity classes, polynomial time hierarchy, relations of distinct complexity classes, separation of distinct complexity classes, properties of sparse sets, foundations of cryptography.

Recommended courses: NTIN073 Recursion I, NTIN074 Recursion II, NTIN081 Structural Complexity I, NTIN082 Structural Complexity II
Supplementary courses: NTIN085 Selected Topics in Computational Complexity I, NTIN086 Selected Topics in Computational Complexity II

2. *Theory of algorithms*

Randomized algorithms: parameters of randomized algorithms, probabilistic complexity classes (BPP, RP, ZPP) and examples of problems from these classes), random binary search trees.

Parallel algorithms: models of parallel computers, parallel computation thesis, technique of parallel algorithms, lower bounds, P-complete problems, NC and AC-complexity classes.

Deterministic algorithms: measures of complexity (the worst case complexity, expected complexity, amortized complexity), probabilistic distribution of input data, statistical method for estimates of time complexity based on experiments, interpretation of statistical results.

Recommended courses: NTIN063 Complexity II, NTIN017 Parallel Algorithms, NTIN018 Probabilistic Analysis of Algorithms, NTIN081 Structural Complexity I, NMAI060 Probabilistic Methods, NMAI061 Methods of Mathematical Statistics
Supplementary courses: NDMI025 Randomized Algorithms

3. *Concrete algorithms*

Sorting: Comparison-based sorting algorithms and their complexity (Shellsort, Mergesort, Heapsort, Quicksort), general sorting algorithms (bucketsort, hybridsort), median algorithms, sorting networks, parallel Mergesort, external sorting.

Algebraic algorithms: matrix multiplication algorithms and their applications, fast discrete Fourier transformation, LUP-decomposition, primality testing algorithms.

Graph algorithms: planarity algorithm, algorithms for maximum network flow and their application, algorithms for transitive closure, Euler tour, parallel algorithms for

connectedness and bi-connectedness, least cost path algorithms, minimum spanning tree algorithms.

Algorithms for satisfiability.

Recommended courses: NTIN067 Data Structures II, NDMI010 Graph Algorithms, NTIN017 Parallel Algorithms, NAIL021 Boolean Functions and Their Applications, NDMI025 Randomized Algorithms

Supplementary courses: NDMI007 Combinatorial Algorithms, NTIN081 Structural Complexity I, NTIN084 Bioinformatics Algorithms, NAIL025 Evolutionary Algorithms I, NAIL086 Evolutionary Algorithms II, NTIN087 Text Algorithms, NAIL094 Decision Procedures and Verification, NTIN085 Selected Topics in Computational Complexity I, NTIN086 Selected Topics in Computational Complexity II.

b) Specialization **Non-procedural programming and artificial intelligence**

Examination areas

1. Logic and Computational Complexity
2. Artificial Intelligence
3. Nonprocedural Programming
4. Neural Networks
5. Adaptive Agents and Evolutionary Algorithms
6. Robotics

Knowledge requirements

1. Logic and computational complexity

Formal systems, first-order logic, syntax, axioms, inference rules. Propositional logic, semantics of propositional logic, tautology and satisfiability, provability, deduction theorem, compactness theorem, completeness theorems. Conjunctive and disjunctive normal forms of formulae.

Predicate logic, interpretation of a first-order language, satisfiability and validity of formulae. First-order theories, provability, deduction theorem, free constants theorem, prenex normal form of formulae. Soundness theorem. Completeness theorem, Henkin theories, complete theories. Extensions of theories, conservative extensions, extensions by definition of functions and predicates.

Measures of computational complexity, complexity classes (P, NP, PSPACE, NPSPACE, LOGSPACE), NP-hard and NP-complete problems. Complexity of algorithms in artificial intelligence, searching, resolution.

Recommended courses: NAIL062 Propositional and Predicate Logic, NTIN062 Complexity I

2. Artificial Intelligence

Knowledge representation: state space, production systems, logical representation. Search algorithms: tree, graph, and local search, heuristics, algorithm A* and its variants. Games: algorithm minimax, alfa-beta pruning. Constraint satisfaction. Automated theorem proving, model checking (DPLL), forward and backward chaining, resolution and unification. Automated planning: planning domain and problem,

planning operators. Uncertainty reasoning: Bayesian networks, conditional probability, evaluation in Bayesian networks, decision graphs, Markov models, Kalman filtr. Machine learning: version space, decision trees, Bayesian learning, maximum likelihood hypothesis, algorithm expectation-maximization, reinforcement learning.

Recommended courses: NAIL069 Artificial Intelligence I, NAIL070 Artificial Intelligence II

Supplementary courses: NAIL004 Seminar on Artificial Intelligence I, NAIL052 Seminar on Artificial Intelligence II, NAIL021 Boolean Functions and Their Applications, NAIL031 Representations of Boolean Functions, NAIL029 Machine Learning, NOPT042 Constraint Programming, NAIL071 Planning and Scheduling, NAIL068 Human-like Artificial Agents, NAIL094 Decision Procedures and Verification

3. Nonprocedural programming

Differences between procedural and non-procedural styles of programming. Principles of functional and logic programming. Lambda calculus, its syntax, free and bound variables, and reduction principles. Church and Rosser property and consistency of calculus. Fixed point theorems. Normal form of terms. Typed lambda calculus. Curry and Church type systems. Basic characteristics of functional languages.

Horn logic, Horn clauses. Substitution, unification and their properties. SLD resolution and logic programs. Soundness and completeness of SLD-resolution. Negation as failure, logic programs with negation. Pure Prolog as Prolog subset. Sufficient conditions of program termination. Unification without occur-check. Implementation of Prolog. Programming with constraints: inference and search algorithms of constraint satisfaction problems.

Recommended courses: NAIL078 Lambda Calculus and Functional Programming I, NAIL076 Logic Programming I, NOPT042 Constraint Programming

Supplementary courses: NAIL079 Lambda Calculus and Functional Programming II, NAIL077 Logic Programming II, NAIL022 Methods of Logic Programming, NAIL006 Seminar on Logic Programming I, NAIL009 Seminar on Logic Programming II

4. Neural networks

Neurophysiological fundamentals: the structure of a neuron, types of synapses, main parts of the brain. Models for supervised learning: perceptron, back-propagation training algorithm, strategies to speed up the training process, internal knowledge representation, generalization, regulation techniques. Associative memories: Hebbian learning, BAM, Hopfield model, energy function and the search for sub-optimal solutions. Stochastic models: simulated annealing, Boltzmann machine. Artificial neural network based on self-organization: Oja's learning algorithm, lateral inhibition, Kohonen maps and their variants for supervised learning, ART networks. Modular, hierarchical and hybrid models of neural networks: adaptive mixtures of local experts, multi-layer Kohonen networks, RBF-networks, cascade correlation. Genetic algorithms, schema theorem. Applications of neural networks and evolutionary techniques (data analysis, bioinformatics, image processing, robotics, etc.).

Recommended courses: NAIL002 Neural Networks, NAIL013 Applications of Neural

Networks Theory

Supplementary courses: NTIN084 Bioinformatics Algorithms, NAIL060 Neural Networks Implementation I, NAIL015 Neural Networks Implementation II, NAIL065 Evolutionary Robotics, NDBI023 Data Mining

5. Adaptive agents and evolutionary algorithms

An architecture of autonomous agents; perception, action selection mechanism, memory; psychological motivations. Action selection methods, control architectures according to Wooldridge, symbolic vs. connectionist reactive planning, hybrid architectures (Belief Desire Intention, Soar), comparing to classical planning techniques. Path-finding problem, steering rules, terrain representation. Communication in multi-agent systems, ontologies, bounded rationality, possible-worlds semantics (Kripke). Ethological inspirations, models of population dynamics. Methods for agents learning, reinforcement learning, fundamental methods of animal learning.

Artificial evolution, genetic algorithms, genetic programming, evolutionary programming. Basic approaches and notions: population, fitness, recombination, genetic operators, dynamic vs. static selection, roulette-wheel selection, tournaments, elitism. Schema theory, building blocks hypothesis. Probabilistic models of simple genetic algorithm. Co-evolution, open evolution. Applications of evolutionary algorithms (action selection, expert systems evolution, finite automata evolution, adaptation of evolutionary rules, neuroevolution, combinatorial optimization).

Recommended courses: NAIL068 Human-like Artificial Agents, NAIL025 Evolutionary Algorithms I, NAIL086 Evolutionary Algorithms II, NAIL087 Computer and Cognitive Sciences I

Supplementary courses: NAIL071 Planning and Scheduling, NAIL054 Adaptive Agents, NAIL082 Seminar on Human-like Artificial Agents, NAIL065 Evolutionary Robotics, NAIL002 Neural Networks, NAIL088 Computer and Cognitive Sciences II, NAIL096 Multi-agent Systems

6. Robotics

Robot control system: Basic models, control, kinematics, autonomous systems, mobile systems. Localization: absolute and relative, local and global, localization in dynamic environment, probabilistic localization. Mapping, simultaneous localization and mapping. Activity planning: state space planning and plan space planning, planning with time and resources. Cognitive robotics: models, sensors, data processing, recognition. Multi-robot systems: basic models, synchronization and cooperation, planning. Software implementation: system design, modeling, simulation, programming for specific runtime environments.

Recommended courses: NAIL028 Introduction to Mobile Robotics, NPGR001 Computer Vision and Robotics, NAIL071 Planning and Scheduling, NSWE001 Embedded and Real Time Systems

Supplementary courses: NAIL029 Machine Learning, NAIL065 Evolutionary Robotics, NAIL068 Human-like Artificial Agents, NAIL025 Evolutionary Algorithms I, NAIL101 Probabilistic Robotics, NAIL070 Artificial Intelligence II

3 Degree Plans - Mathematical Linguistics I3

Coordinated by: Institute of Formal and Applied Linguistics

Study branch coordinator: Doc. RNDr. Markéta Lopatková, Ph.D.

3.1 Obligatory courses

Code	Subject	Credits	Winter	Summer
NPFL067	Statistical Methods in Natural Language Processing I	6	2/2 C+Ex	—
NPFL092	NLP Technology	5	1/2 MC	—
NPRG027	Credit for Project	6	0/4 C	—
NPRG023	Software Project	9	—	0/6 C
NTIN090	Introduction to Complexity and Computability ¹	5	2/1 C+Ex	—
NTIN066	Data Structures I	5	2/1 C+Ex	—
NSZZ023	Diploma Thesis I	6	—	0/4 C
NSZZ024	Diploma Thesis II	9	0/6 C	—
NSZZ025	Diploma Thesis III	15	—	0/10 C

¹Instead of the course NTIN090 Introduction to Complexity and Computability Theory students may take the pair of courses NTIN062 Complexity I, NTIN064 Computability I

The courses NPFL067 Statistical Methods in Natural Language Processing I and NPFL092 NLP Technology can be studied already during the bachelor studies.

3.2 Elective courses

The student needs to obtain at least 35 credits for the courses from this set.

Code	Subject	Credits	Winter	Summer
NPFL068	Statistical Methods in Natural Language Processing II	6	—	2/2 C+Ex
NPFL083	Linguistic Theory and Grammar Formalisms	6	—	2/2 C+Ex
NPFL070	Language Data Resources	5	—	1/2 MC
NPFL075	Prague Dependency Treebank	6	—	2/2 C+Ex
NPFL054	Introduction to Machine Learning	6	2/2 C+Ex	—
NPFL093	NLP Applications	5	—	2/1 MC
NJAZ097	Czech for Beginners I	3	0/2 C	—
NJAZ098	Czech for Beginners II	3	—	0/2 C
NPFL087	Statistical Machine Translation	6	—	2/2 C+Ex
NPFL094	Morphological and Syntactic Analysis I	3	2/0 MC	—
NPFL006	Introduction to Formal Linguistics	3	2/0 Ex	—
NPFL095	Modern Methods in Computational Linguistics	3	0/2 C	—

NPFL038	Fundamentals of Speech Recognition and Generation	6	2/2 C+Ex	—
NPFL082	Information Structure of Sentences and Discourse Structure	3	—	1/1 Ex
NPFL096	Computational Morphology	4	—	2/1 Ex
NPFL079	Algorithms in Speech Recognition	6	—	2/2 C+Ex
NPFL099	Statistical Dialogue Systems	5	—	2/1 C+Ex
NPFL106	Linguistics	3	—	1/1 MC
NPFL103	Information Retrieval	6	2/2 C+Ex	—

All other courses with the code NPFL are recommended as elective.

3.3 State Final Exam

Examination areas

The I3 branch is not divided into specializations. The examination area 1 is obligatory for all students of the branch; the student must select the second and third examination area from the areas 2-5. Upon explicit request, the student can choose one area from the areas 2-5 and one of the examination areas from the following list: Artificial Intelligence, Neural Networks, Adaptive agents and evolutionary algorithms (from the branch I1, specialization Non-procedural Programming and Artificial Intelligence), or the area Image Analysis and Processing, Computer Vision and Robotics (branch I2, specialization Computer Graphics).

1. Fundamentals of Natural Language Processing
2. Statistical Methods and Machine Learning in Computational Linguistics
3. Applications in Natural Language Processing
4. Linguistic Theories and Formalisms
5. Speech Analysis and Synthesis, Dialog Systems

Knowledge requirements

1. *Fundamentals of Natural Language Processing*

Fundamentals of general linguistics (basic linguistic terms and concepts, function and form). System of layers in language description (phonetics, phonology, morphology, surface/deep syntax, semantics, pragmatics). Dependency syntax, formal definition of dependency trees, their characteristics (dependency relation, coordination, projectivity). Chomsky hierarchy of languages, context free languages, phrase grammars for a natural language. Design and evaluation of linguistic experiments, evaluation metrics (precision, recall, f-measure, statistical significance etc.) Basic stochastic methods (generative, discriminative; source-channel model; HMM). Language modeling, basic methods for training stochastic models (maximal likelihood, EM). Basic algorithms (Trellis, Viterbi, Baum-Welch).

Recommended courses: NPFL067 Statistical Methods in Natural Language Processing I and a selection of one of the courses from NPFL063 Introduction to General Linguistics, NPFL075 Prague Dependency Treebank and NPFL106 General Linguistics

2. *Statistical Methods and Machine Learning in Computational Linguistics*

Generative and discriminative models. Language data for machine learning. Language models. Smoothing of models. Noisy channel models, decoding. Model parameters, space of hypotheses. Theoretical aspects of machine learning (PAC). Supervised machine learning (Naive Bayes, Maximal Entropy, SVN, perceptron, decision trees, logistic regression, Bayesian Networks, Example-based learning). Unsupervised machine learning methods (clustering, expectation-maximization). HMM, Viterbi. Tests of significance, intervals for reliability. Statistical parsing algorithms (PCFG, MST).

Recommended courses: NPFL067 Statistical Methods in Natural Language Processing I, NPFL068 Statistical Methods in Natural Language Processing II, NPFL054 Introduction to Machine Learning, NPFL070 Language Data Resources

3. *Applications in Natural Language Processing*

Processing morphology (morphological categories, tagset; analysis, tagging, lemmatization, segmentation, generating, algorithms). Syntactic analysis (surface, deep, dependency, phrase-based syntax, algorithms). Natural language generation. Spell-checking and grammar-checking. Machine translation (direct translation, transfer, interlingua; systems for Czech, machine aided translation, statistical methods: IBM models, phrase-based models, hierarchical models, syntactic models). Models for information retrieval (Boolean, vector-based, probabilistic, language-based), information retrieval evaluation.

Recommended courses: NPFL093 NLP Applications, NPFL094 Morphological and Syntactic Analysis, NPFL087 Statistical Machine Translation, NPFL103 Information Retrieval

4. *Linguistic Theories and Formalisms*

Functional Generative Description (basic characteristics, system of layers, theory of valency, language meaning). Government and binding (nativism, X-bar, movement, trace, binding). Other basic grammar formalisms (unification-based grammars, feature structures, HPSG, LFG, categorial grammars, TAG). Formal semantics. Prague Dependency Treebank. Computer lexicography (types of lexicons, wordnets, ontologies). Topic-focus articulation. Anaphora. Discourse.

Recommended courses: NPFL106 General Linguistics, NPFL083 Linguistic Theories and Grammar Formalisms, NPFL075 Prague Dependency Treebank, NPFL082 Information Structure of Sentence and Discourse Structure, NPFL006 Introduction to Formal Linguistics

5. *Speech Analysis and Synthesis, Dialog Systems*

Fundamentals of language production and recognition. Methods of speech signal processing. HMM modeling of the acoustics of phonemes. The implementation of Baum-Welch and Viterbi algorithms for speech recognition. Continuous speech recognition exploiting large dictionaries. Adaptation techniques. Speech recordings summarization. Topic and word search in speech corpora. Speaker recognition. Speech

synthesis methods. Text processing for speech synthesis. Prosody modeling. Basic components of a dialog system. Speech understanding. Dialog modeling - MDP and POMDP systems. User simulation. Speech generation. Dialog systems quality evaluation.

Recommended courses: NPFL038 Fundamentals of Speech Recognition and Generation, NPFL079 Algorithms in Speech Recognition, NPFL099 Statistical Dialog Systems

4 Degree Plans - Discrete Models and Algorithms I4

Coordinated by: Department of Applied Mathematics

Study branch coordinator: Doc. RNDr. Martin Klazar, Dr.

4.1 Obligatory courses

Code	Subject	Credits	Winter	Summer
NTIN062	Complexity I	5	2/1 C+Ex	—
NTIN064	Computability	3	—	2/0 Ex
NTIN066	Data Structures I	5	2/1 C+Ex	—
NMAI064	Mathematical Structures	6	—	2/2 C+Ex
NDMI073	Combinatorics and Graph Theory III ¹	6	2/2 C+Ex	—
NOPT018	Fundamentals of Nonlinear Optimization ²	6	2/2 C+Ex	—
NSZZ023	Diploma Thesis I	6	—	0/4 C
NSZZ024	Diploma Thesis II	9	0/6 C	—
NSZZ025	Diploma Thesis III	15	—	0/10 C

¹The course is compulsory only for the two specializations Discrete Mathematics and Combinatorial Optimization, Mathematical Structures of Computer Science; it is elective for the specialization Optimization.

²The course is compulsory only for the specialization Optimization; it is elective for the other specializations.

4.2 Elective courses

The student needs to obtain at least 45 credits for the courses from this set.

Code	Subject	Credits	Winter	Summer
NTIN063	Complexity	5	—	2/1 C+Ex
NTIN065	Computability II	3	—	2/0 Ex
NTIN067	Data Structures II	3	—	2/0 Ex
NDMI073	Combinatorics and Graph Theory III ¹	6	2/2 C+Ex	—
NOPT018	Fundamentals of Nonlinear Optimization ²	6	2/2 C+Ex	—
NDMI013	Combinatorial and Computational Geometry II	6	—	2/2 C+Ex
NDMI010	Graph Algorithms	3	2/0 Ex	—

NDMI025	Randomized Algorithms	6	—	2/2 C+Ex
NDMI015	Combinatorial Counting	3	—	2/0 Ex
NMAI066	Topological and Algebraic Methods	3	—	2/0 Ex
NTIN022	Probabilistic Techniques	6	2/2 C+Ex	—
NMAI065	Fundamentals of Category Theory for Computer Scientists	3	2/0 Ex	—
NMAI040	Introduction to Number Theory	3	2/0 Ex	—
NMAI067	Logic in Computer Science	3	2/0 Ex	—
NOPT008	Nonlinear Optimisation Algorithms	6	—	2/2 C+Ex
NOPT004	Optimization Processes I	6	2/2 C+Ex	—
NOPT005	Optimisation Processes II	3	—	2/0 Ex
NOPT001	Dynamic Programming	3	2/0 Ex	—
NOPT015	Parametric Optimization	6	—	2/2 C+Ex
NOPT017	Multiobjective Optimisation	3	—	2/0 Ex
NOPT016	Integer Programming	6	—	2/2 C+Ex
NAIL076	Logic Programming I	3	2/0 Ex	—
NTIN017	Parallel Algorithms	3	—	2/0 Ex
NAIL083	Mathematical Models of Cell Activity	3	2/0 Ex	—
NMAG337	Introduction to Group Theory	5	2/2 C+Ex	—
NDMI018	Approximation and Online Algorithms	6	—	2/2 C+Ex
NDMI028	Linear Algebra Applications in Combinatorics	6	2/2 C+Ex	—
NDMI036	Combinatorial Structures	3	—	2/0 Ex
NDMI037	Geometric Representations of Graphs I	3	—	2/0 Ex
NDMI045	Analytic and Combinatorial Number Theory	3	—	2/0 Ex
NDMI055	Selected Chapters on Combinatorics I	3	2/0 Ex	—
NDMI056	Selected Chapters on Combinatorics II	3	—	2/0 Ex
NDMI059	Graph Minors and Tree Decompositions	3	2/0 Ex	—
NDMI060	Coloring of Graphs and Other Combinatorial Structures	3	2/0 Ex	—
NDMI064	Applied Discrete Mathematics	3	2/0 Ex	—
NDMI065	Matroid Theory	6	2/2 C+Ex	—
NDMI066	Algebraic Number Theory	3	2/0 Ex	—
NDMI067	Flows, Paths and Cuts	3	2/0 Ex	—
NOPT013	Mathematical Economics	6	—	4/0 Ex
NOPT021	Game Theory	3	2/0 Ex	—
NOPT034	Mathematical Programming and Polyhedral Combinatorics	5	2/1 C+Ex	—
NOPT042	Constraint Programming	6	2/2 C+Ex	—

NMAA069 Measure and Integration Theory I	3	2/0 Ex	—
NMMA901 Introduction to Complex Analysis (O)	5	2/2 C+Ex	—
NMMA931 Introduction to Functional Analysis (O)	8	4/2 C+Ex	—

¹The course is elective only for the specialization Optimization; for the other specializations it is compulsory.

²The course is elective only for the two specializations Discrete Mathematics and Combinatorial Optimization, Mathematical Structures of Computer Science; it is compulsory for the specialization Optimization.

4.3 State Final Exam

a) Specialization *Discrete Mathematics and Combinatorial Optimization*

Examination areas

1. Combinatorics and graph theory
2. Probabilistic methods and randomized algorithms
3. Combinatorial optimization

Knowledge requirements

1. *Combinatorics and graph theory*

Coloring of graphs, regular graphs, connectivity of graphs, special properties of oriented graphs, algebraic properties of graphs, matching theory, Ramsey theory, infinite combinatorics, structural properties of set systems.

2. *Probabilistic methods and randomized algorithms*

Combinatorial counting, generating functions, recurrences, basic probability models, linearity of expectation, applications of variance, application in specific examples, asymptotic estimates of functions, random constructions and algorithms.

3. *Combinatorial optimization*

Graph algorithms, algebraic and arithmetic algorithms, theory of polyhedra, problem of the traveling salesman, special matrices, integrality, matchings and flows in networks, matroid theory, ellipsoid method.

Recommended courses

Code	Subject	Credits	Winter	Summer
NTIN022	Probabilistic Techniques	6	2/2 C+Ex	—
NDMI009	Combinatorial and Computational Geometry I	6	2/2 C+Ex	—
NDMI025	Randomized Algorithms	6	—	2/2 C+Ex
NDMI015	Combinatorial Counting	3	—	2/0 Ex
NDMI018	Approximation and Online Algorithms	6	—	2/2 C+Ex
NDMI028	Linear Algebra Applications in Combinatorics	6	2/2 C+Ex	—
NDMI055	Selected Chapters on Combinatorics I	3	2/0 Ex	—

NDMI060	Coloring of Graphs and Other Combinatorial Structures	3	2/0 Ex	—
NDMI065	Matroid Theory	6	2/2 C+Ex	—
NDMI067	Flows, Paths and Cuts	3	2/0 Ex	—
NOPT034	Mathematical Programming and Polyhedral Combinatorics	5	2/1 C+Ex	—

b) Specialization ***Mathematical Structures of Computer Science***

Examination areas

1. Combinatorial and computational geometry
2. Algebraic and topological methods in Computer Science
3. Number theory and category theory in Computer Science

Knowledge requirements

1. Combinatorial and computational geometry

Geometric problems in spaces with finite dimension, combinatorial properties of geometric configurations, algorithmic applications, design of geometric algorithms, geometric representations of graphs.

2. Algebraic and topological methods in Computer Science

Partially ordered sets, suprema and infima, semilattices, lattices. Fixed-point theorems. Special ordered structures in Computer Science (DCPO, domains). Fundamentals of general topology; topological constructions. Special topological problems relevant for Computer Science (Scott topology, continuous lattices). Categories of topological spaces and some other partial orders relevant for Computer Science

3. Number theory and category theory in Computer Science

Categories, functors, transformations, specific examples. Limits and colimits, specific constructions and generation of further constructions. Adjunction, relation to categorical constructions. Reflections and coreflections. Specific examples of adjoint situations. Cartesian closed categories. Categories and structures, especially structures used in Computer Science. Monadic algebras.

Recommended courses

Code	Subject	Credits	Winter	Summer
NTIN022	Probabilistic Techniques	6	2/2 C+Ex	—
NMAI066	Topological and Algebraic Methods	3	—	2/0 Ex
NMAI065	Fundamentals of Category Theory for Computer Scientists	3	2/0 Ex	—
NMAI040	Introduction to Number Theory	3	2/0 Ex	—
NMAI067	Logic in Computer Science	3	2/0 Ex	—
NDMI009	Combinatorial and Computational Geometry I	6	2/2 C+Ex	—
NDMI013	Combinatorial and Computational Geometry II	6	—	2/2 C+Ex
NDMI036	Combinatorial Structures	3	—	2/0 Ex

NDMI037	Geometric Representations of Graphs I	3	—	2/0 Ex
NDMI045	Analytic and Combinatorial Number Theory	3	—	2/0 Ex
NDMI056	Selected Chapters on Combinatorics II	3	—	2/0 Ex
NDMI059	Graph Minors and Tree Decompositions	3	2/0 Ex	—

c) Specialization ***Optimisation*****Examination areas**

1. Nonlinear programming
2. Optimisation processes
3. Parametric, multiobjective and integer programming
4. Non-smooth optimization and stochastic dynamical models

Knowledge requirements*1. Nonlinear programming*

Properties of convex sets and convex functions. Generalizations of convex functions. Necessary and sufficient conditions for optimality for free and constrained extrema in nonlinear programming. Quadratic programming. Duality in nonlinear programming. Solution methods for free and constrained extrema, including penalty and barrier methods. One-dimensional optimization.

2. Optimisation processes

Continuous: the maximum principle for nonlinear problems of various types. Conditions for optimality in the basic problems of the calculus of variations. Linear problems on the minimization of time.

Discrete: Classification of problems and their relation to the problem of nonlinear programming. Linear and quadratic problems. Fundamentals of the control of Markov systems. Discrete dynamical programming - optimization with respect to the initial state, final state, and both the initial and final state.

3. Parametric, multiobjective and integer programming

Domains of stability of the solution. Domains of solvability. Solvability functions for one-parametric and multiobjective programming. Various approaches for solving problems with several criteria.

The functional associated to a problem of vector programming. Efficient points. Problems of linear and nonlinear vector optimization. Methods for obtaining efficient points. Problems of linear programming with integrality conditions, or with bivalent variables. Nonlinear optimization problems with integrality conditions.

4. Non-smooth optimization and stochastic dynamical models

Clarke's calculus and fundamentals of non-smooth optimization. Conditions for optimality. Numerical methods of non-smooth optimization. Models with discrete states (a Poisson process, queueing theory, Markov processes and chains). Comparison

of statistical and deterministic models. Models with continuous states (the stochastic integral and differential, linear stochastic differential equations).

Recommended courses

Code	Subject	Credits	Winter	Summer
NOPT018	Fundamentals of Nonlinear Optimization	6	2/2 C+Ex	—
NOPT008	Nonlinear Optimisation Algorithms	6	—	2/2 C+Ex
NOPT004	Optimization Processes I	6	2/2 C+Ex	—
NOPT005	Optimisation Processes II	3	—	2/0 Ex
NOPT001	Dynamic Programming	3	2/0 Ex	—
NOPT015	Parametric Optimization	6	—	2/2 C+Ex
NOPT017	Multiobjective Optimisation	3	—	2/0 Ex
NOPT016	Integer Programming	6	—	2/2 C+Ex
NOPT034	Mathematical Programming and Polyhedral Combinatorics	5	2/1 C+Ex	—
NDMI067	Flows, Paths and Cuts	3	2/0 Ex	—