Bachelor State Final Exam Topics

The role of this text is to provide a detailed description for each Bachelor State Final Examination topics in order to clarify (1) what specific knowledge belongs to each general topic, (2) how deep a knowledge will be required, and (3) what courses generally examine each topic.

Requirements Specification

The text includes common requirements from mathematics and computer science and requirements particular to individual specializations. Requirements of individual specializations always implicitly include and may explicitly extend the common requirements. All requirements also implicitly include the relevant background knowledge, for example mastering the calculus or mastering the features of the corresponding programming language.

The requirements are specified either in the form of testable knowledge and skills, or in the form of a term list:

- The requirement "perform address translation tasks given specific addresses and single level page table content" specifies a testable skill. The exam may test this exact skill or skills reasonably similar, in this particular case for example also the ability to describe the page table structure, which is needed to perform the address translation tasks.
- The requirement "complexity classes P and NP, problem reductions, NP-hardness and NP-completeness" specifies a term list. The exam may test knowledge and application of these terms at the same level as is currently done in the listed courses.

Please contact the specialization coordinator through the link below for any clarification regarding the requirements.

Optional Requirements

Although the text refers to specific courses, the requirements are valid regardless of the course enrollment, historical deviation from current course content, or any other individual circumstances. The exam board expects the students to fill potential gaps due to such circumstances.

Where the requirements explicitly define variants for particular elective course enrollment denoted with the ⓥ symbol, the exam will provide similar variants for all course enrollment alternatives supported by the study plans.

From one exam term to another, the text can be updated to reflect the needs of the study program and the feedback of the exam board and the students. The detailed requirements may both extend and reduce the terse requirements from the study plans, however, the changes are restricted so that starting one month before each term, no new requirement will be added, and starting one year before each term, no requirement that would necessitate taking an entire new course will be added.

This is a version dated January 4, 2025, prepared for the Bachelor State Final Exams from February 2025 on.

Please report any issues at https://gitlab.mff.cuni.cz/mff/statnice/informatika-public-forum.

Foundations of Mathematics

1. Fundamentals of differential and integral calculus

- Sequences of real numbers and their limits
  - Definition, arithmetic of limit
  - The squeeze theorem, limits and ordering
- Series
  - Definition of a partial sum and a sum
  - Geometric series, harmonic series
- Real functions of one real variable
  - Limit of a function at a point
    - Definition, arithmetic of limit
    - Relation with ordering
    - Limit of a composite function
  - Function continuous on an interval
    - Intermediate value theorem
    - Extreme value theorem
- Derivative of a function of real variable and its applications
  - Definition and basic rules for calculation
  - L'Hospital's rule
  - Derivatives and properties of functions: extremes, monotonicity and convexity/concavity
  - Taylor polynomial (limit form)
- Integrals and their applications
  - Primitive function: definition and methods for calculation (substitution, per-partes)
  - Riemann integral: definition, connection with the primitive function (Newton integral)
  - Applications
    - Estimates of a (finite or infinite) sum
    - Area of plane figures
    - Volumes and surfaces of rotational figures in space
    - Length of a curve

Courses

- NMAI054 Matematical Analysis 1 (5 cr)

2. Algebra and Linear Algebra

- Algebraic structures:
  - groups, subgroups, permutations
  - fields, and finite fields in particular
- Systems of linear equations:
  - matrix form, elementary row operations, row echelon form
  - Gaussian elimination, Gauss-Jordan elimination, description of the solution set
- Matrices:
  - matrix operations, basic types of matrices, the rank of a matrix
  - nonsingular matrices, the inverse of a matrix
- Vector spaces:
  - vector spaces, linear combination, linear dependence and independence, linear span
  - Steinitz exchange lemma, basis, dimension, coordinates
  - vector subspaces, particularly matrix subspaces (row space, column space, kernel) and their dimensions
- Linear maps:
  - definition, matrix representations, the composition of linear maps and its matrix form
  - the image and the kernel of linear maps
  - isomorphism of vector spaces
- Inner product spaces:
  - inner product, norm induced by an inner product

  - Pythagoras theorem, Cauchy-Schwarz inequality, triangle inequality
  - orthonormal system of vectors, Fourier coefficients, Gram-Schmidt orthogonalization
  - orthogonal complement, orthogonal projection, projection as a linear map
  - orthogonal matrices and their properties
- Determinants:
  - definition and basic properties (multiplicativity, determinant of the transpose, relationships with nonsingularity and eigenvalues)
  - Laplace expansion of a determinant
  - geometric interpretation of determinants
- Eigenvalues and eigenvectors:
  - definition, geometric interpretation, basic properties, characteristic polynomial, multiplicity of eigenvalues
  - similarity and diagonalization of matrices, spectral decomposition
  - symmetric matrices, their eigenvalues and spectral decomposition
- Positive semidefinite and positive definite matrices:
  - characterization and properties, relationship with inner products and eigenvalues
  - Cholesky decomposition (statement of the theorem and its practical application)

Courses

- NMAI057 Linear Algebra 1 (5 cr)
- NMAI058 Linear Algebra 2 (5 cr)

3. Discrete Mathematics

- Relations
  - properties of binary relations (reflexivity, symmetry, antisymmetry, transitivity)
- Equivalence relation and equivalence classes
- Partial orders
  - basic concepts (minimal and maximal elements, chain, antichain)
  - height and width of a partially ordered set and the theorem about their relation
- Functions
  - types of functions (injective, surjective, bijective)
  - number of different functions between two finite sets
- Permutations and their basic properties (number of pertmutations, fixed point)
- Binomial coefficients and relations among them, binomial theorem and its use
- Principle of inclusion and exclusion
  - general formulation (and a proof)
  - usage (Euler's function for the number of divisors, number of surjections)
- Hall's theorem on systems of distinct representatives and its relation to matchings in a bipartite graph
  - proof principle and algorithmic aspects (polynomial algorithm for finding a SDR)

Courses

- NDMI002 Discrete Mathematics (5 cr)
- NDMI011 Combinatorics and Graph Theory 1 (5 cr)

4. Graph Theory

- Basic concepts of graph theory
  - graph, vertices and edges, graph isomorphism, subgraph, vertex neighbourhood and vertex degree, graph complement, bipartite graph
- Basic examples of graphs
  - complete graph and complete bipartite graph, paths and circuits
- Graph connectivity, connected components, distance in a graph
- Trees
  - definition and basic properties (existence of leafs, number of edges in a tree)
  - equivalnet characterizations of trees
- Planer graphs
  - definition and basic concepts (planar graph and planar drawing of a graph, faces)
  - Euler's formula and the maximum number of edges in a planar graph (proof and usage)

- Graph colourings
  - definition of a good colouring
  - chromatic number, clique number, and their relation
- Edge- and vertex-connectivity of graphs
  - edge and vertex version of Menger's theorem
- Directed graphs, weak and strong connectivity
- Network flows
  - definition of a network and a flow in it
  - existence of a maximum flow (w/o a proof)
  - principle of searching for a maximum flow in a network with integral csapacities (e.g. using the the Ford-Fulkerson algorithm)

Courses

- NDMI002 Discrete Mathematics (5 cr)
- NDMI011 Combinatorics and Graph Theory 1 (5 cr)

5. Probability and Statistics

- Probability space, random events, probability
  - Definitions of these terms, examples
  - Basic rules for calculating with probability
  - Independence of random events, conditional probability
  - Bayes' formula
- Random variables and their distribution
  - Discrete and continuous case
  - Description using cumulative distribution function and probability mass/density function
  - Expectation
    - Linearity of the expectation
    - Expectation of the product of independent variables
    - Markov inequality
  - Variance
    - Definition
    - Formula for the variance of the sum (of dependent or independent variables)
  - Working with specific distributions: geometric, binomial, Poisson, normal, exponential
- Limit theorems
  - Law of large numbers
  - Central Limit Theorem
- Point estimates
  - At least one method for constructing them
  - Properties
- Interval estimates: method based on approximation by normal distribution
- Hypothesis testing
  - Basic approach
  - Type I and Type II errors
  - Significance level

Courses

- NDMI002 Discrete Mathematics (5 cr)
- NMAI059 Probability and Statistics 1 (5 cr)

6. Logic

- Syntax
  - working knowledge of elementary concepts in syntax of propositional and predicate logic (language, open and closed formulas, instances of formulas, etc.)
  - normal forms of propositional formulas
    - prenex forms of formulas in predicate logic
    - knowledge of basic normal forms (CNF, DNF, PNF)

        - transformations to normal forms
        - usage in algorithms (SAT, resolution)
- Semantics
    - the concept of a model of a theory
    - valid, contradictory, independent formulas with respect to a theory
    - satisfiability, tautologies, consequence of a theory
    - analysis of propositional theories over finitely many atoms
- Extensions of theories
    - ability to compare the strength of theories
    - conservative extensions, skolemization
- Provability
    - the concept of a formal proof, refutation
    - ability to work in some formal proof system (e.g. tableau method, resolution, Hilbert's calculus)
- Compactness and completeness theorems in propositional and predicate logic
    - statement and understanding of its meaning
    - use on examples, consequences
- Decidability
    - the concept of complete theories and its criteria, importance for decidability
    - examples of decidable and undecidable theories

Courses

- NAIL062 Propositional and Predicate Logic (5 cr)

Foundations of Computer Science

1. Automata and Languages

- Regular languages
  - regular grammars
  - deterministic and nondeterministic finite automaton
  - regular expressions
- Context-free languages
  - context-free grammars, language generated by a grammar
  - push-down automaton, class of languages accepted by push-down automata
- Recursively enumerable languages
  - type 0 grammar
  - Turing machine
  - algorithmically undecidable problems
- Chomsky hierarchy
  - ability to classify a concrete language into the correct level of Chomsky hierarchy (typically by constructing the appropriate automaton or grammar)

Courses

- NTIN071 Automata and Grammars (2/5 of 5 kr)

2. Algorithms and Data Structures

- Time complexity of algorithms
  - time and space complexity of an algorithm
  - data size measurment
  - best case, worst case, and average case complexity
  - asymptotic notation
- Complexity classes
  - classes P and NP
  - problem reductions, NP-hardness and NP-completeness
  - examples of NP-complete problems and reductions among them
- Divide and conquer method
  - recursive division principle (of a problem into subproblems)
  - complexity computation using recursive equations
  - Master theorem (without a proof)
  - aplications
    - Mergesort
    - binary search
- Binary search trees
  - definition of a search tree
  - operations with unbalanced trees
  - AVL trees (definition)
- Sorting
  - primitive sorting algorithms (Bubblesort, Insertsort)
  - Quicksort
  - complexity lower bound for sorting algorithms that use comparisons
- Graph algorithms
  - BFS and DFS
  - topological sort for directed graphs
  - shortest paths in weighted graphs (Dijkstra's and Bellman-Ford's algorithms)
  - minimum spanning trees (Jarník's and Borůvka's algorithms)
  - network flows (Ford-Fulkerson's algorithm)

Relevant courses

- NTIN060 Algorithms and Data Structures 1 (3/5 of 5 kr)
- NTIN061 Algorithms and Data Structures 2 (1/5 of 5 kr)

3. Programming Languages

Some of the following items define variants for particular elective course enrollment. Knowledge of all items without the ⓥ symbol and all items with the ⓥ symbol for one of the C#, C++ or Java languages is required.

- Concepts for abstraction, encapsulation, and polymorphism.
  - related programming language constructs
    - classes, interfaces, methods, data members, inheritance, visibility
    - ⓥ encapsulation provided by modules in Java
  - (dynamic) polymorphism, static and dynamic typing
  - single inheritance
    - ⓥ virtual and non-virtual methods in C++ and C#
  - multiple inheritance and related issues
    - ⓥ multiple and virtual inheritance in C++
    - ⓥ interfaces and default methods in Java
    - ⓥ interfaces in C#
  - interface implementation
  - reasonable application of the above concepts
- Primitive and object types and their representation.
  - numeric and enumerable types
  - ⓥ pointers and references in C++
  - ⓥ value and reference types in C#
  - ⓥ references, immutable types and boxing in C# or Java
- Generic types and functional elements (in procedural programming languages).
  - ⓥ templates and static polymorphism in C++
  - ⓥ generic types in Java and C#
  - ⓥ function types in C++, C#, or Java
  - lambda functions and functional interfaces
- Working with resources and mechanisms for error handling.
  - resource lifecycle management in presence of errors
    - ⓥ RAII in C++
    - ⓥ `using` in C#
    - ⓥ try-with-resources in Java
  - language constructs for exception handling and propagation
- Object lifecycle and memory management.
  - allocation (static, stack, heap)
  - initialization (constructors and inherited constructor chaining), object
  - destruction (destructors, finalizers)
  - explicit deallocation, reference counting, garbage collection
- Threads and support for synchronization.
  - representation of threads in programming languages
  - thread function and basic thread API operations
  - race conditions and mechanisms for thread synchronization
- Implementation of basic elements of object-oriented languages.
  - basic object-oriented concepts
  - internal representation of primitive types
  - internal representation of compound types and objects
  - implementation of dynamic polymorphism (virtual method table)
- Native and interpreted execution, compilation and linking.
  - program representation, bytecode
  - just-in-time and ahead-of-time compilation
  - build process, separate compilation, linking
  - statically and dynamically linked libraries
  - process runtime environment and operating system bindings

Relevant courses

Based on the choice of the programming language

- NPRG013 Programming in Java (5 kr)
- NPRG035 Programming in C# (5 kr)
- NPRG041 Programming in C++ (5 kr)

4. Computer Architecture and Operating Systems

- Computer organization, data and program representation.
  - accessing data in memory, address, address space
  - layout of primitive and compound data types
  - basic arithmetic and logic operations
- Instruction set, correspondence to elements of high-level programming languages.
  - Implement common constructs from high-level programming languages (assignment, branch, loop, function invocation) using processor instructions
  - Write common constructs from high-level programming languages (assignment, branch, loop, function invocation) that correspond to given sequence of (explained) processor instructions
- Support for operating system execution.
  - privileged and unprivileged execution
  - operating system kernel
- Peripheral device interface and handling.
  - Describe the role of a device controller in program-driven I/O (PIO), implement program-driven device service for given I/O port addresses and functions (mouse, disk)
  - Describe the role of an interrupt in program-driven I/O (PIO), at the level of instruction execution describe the reaction of the processor (hardware) and the operating system (software) to an interrupt request
- Fundamental OS abstractions, interfaces, and mechanisms for program execution, resource sharing, and input/output.
  - non-privileged (user-mode) programs
  - processor sharing
    - processes, threads, process and thread context
    - context switching, cooperative and preemptive multitasking
    - process and thread scheduling, thread state
  - memory sharing
    - Explain the difference between a virtual and a physical address and distinguish virtual and physical addresses in given contexts or code fragments
    - Identify and explain the use of virtual and physical address components (page number, frame number, offset) on a given address example
    - Perform address translation tasks given specific addresses and single level page table content
    - Explain the role of virtual address spaces in process and thread memory protection
  - storage sharing
    - files, address space analogy
    - abstractions and interfaces for working with files
- Parallelism, threads and interfaces for thread management, thread synchronization.
  - race conditions
  - critical section, mutual exclusion
  - basic synchronization primitives, interface and application
    - locks
    - active and passive waiting

Courses

- NSWI120 Principles of Computers (3 cr)
- NSWI170 Computer Systems (5 cr)

Based on the choice of the programming language

- NPRG013 Programming in Java (1/5 of 5 cr)
- NPRG035 Programming in C# (1/5 of 5 cr)
- NPRG041 Programming in C++ (1/5 of 5 cr)

Web and Data-Oriented Programming

In addition to the common topics, all of the topics below apply to the students of this specialization.

1. Databases

- Database systems architectures
  - Explain the concepts of conceptual, logical and physical levels in data design.
  - Explain the concept of normal form in the design of relations, and the reasons for normalizing relations. Show an example of a relation that violates the given normal form and suggest a correction to the given proposition.
  - Create a conceptual model (UML, ER) for the described problem.
  - Convert a conceptual model to a (relational) logical data model.
- Transaction processing
  - Describe the ACID properties of transactions.
  - Explain and recognize the properties of schedules (serializability, recoverability).
  - Describe and use locking protocols to achieve schedule properties.
  - Explain the concept of deadlock.
- An overview of SQL.
  - Write basic SQL queries including joining tables.
  - Use data grouping and aggregation in SQL queries.
  - Use nested queries and tests for the NULL value in SQL queries.
  - Understand the meaning of a simple SQL query.
  - Explain the purpose of individual SQL query clauses.
- Modern database systems.
  - List the basic classes of modern database systems and briefly describe their specific properties. Using a simple example, compare their features with traditional (SQL) databases.
  - Explain the concept of Big Data, at least four basic properties and their meaning. Specify at least two examples of Big Data sources. Explain new challenges and problems for traditional database systems.
  - Explain the principle of MapReduce and express in pseudocode a simple example of its application to a specific task. Describe the advantages and disadvantages of this principle, and mention alternative approaches.
  - Specify the types, features, advantages and disadvantages of NoSQL databases. Show the differences of the respective models on a simple example.
  - Specify the data model of a graph database, create a simple example. Specify classes of graph queries, give simple examples of use.
  - Specify the differences, advantages and disadvantages of a multi-model database and a polystore. Create a simple example of a multi-model query and describe the possible problems of its evaluation.

Courses

- NDBI025 Database Systems (5 cr)
- NDBI026 Development of Database Applications (4 cr)
- NDBI040 Modern Database Systems (5 cr)

2. Data management

- Data formats
  - Describe basic types of structured data, give their examples and their uses. Describe which format is suitable for which situation, give examples.
  - Explain the differences between a data model, a data format and a data schema. Describe basic properties of text-based data formats. State examples of standardization organizations, describe how they operate and state examples of standards defining data formats.
  - State, describe, compare and use data models and data formats for graph data - RDF and its serializations, Labeled Property Graph. State, describe and use the RDF Schema vocabulary and show usage examples that create RDF vocabularies. State, describe and use the SPARQL and Cypher languages for graph data transformation.

    - State, describe, compare and use data formats for nested (tree-shaped, hierarchical) data - XML and JSON. State, describe and use schema languages for nested data - XML Schema, JSON Schema. Explain on an example how JSON data can be semantically described so that it can be viewed as RDF data suitable for exchange over the Web (JSON-LD). Describe and use an XML transformation language (XSLT).
    - State, use and describe the CSV format for tabular data and its specifications. Describe the usage of the CSV on the Web standard for expressing schemas and facilitating automatic transformation of CSV files to RDF suitable for data exchange over the Web.
    - Show examples and explain what a coordinate reference system is in the context of geodata. Show how geodata can be represented in various data formats such as WKT, GML, GeoJSON and GeoSPARQL.
    - Explain how data can be semantically described using RDF Schema and RDF vocabularies such as Dublin Core and SKOS. Explain the data model of Wikidata and how Wikidata can be queried.
- Data transformation
    - Provide examples and explanations for following data operations: data selection, data projection, data summarization, data reduction, data lifting, and data lowering.
    - Explain the meaning of data quality and its importance for users.
    - Provide examples of data dimensions and how they can be measured.
    - Provide an example and demonstrate a use of data provenance using the PROV-O ontology.
- Data catalogization and metadata
    - List metadata types, provide explanation and an example for each type.
    - Describe benefits of introducing a data catalog into an organization.
    - Explain role of DCAT (Data Catalog Vocabulary) in the context of data catalogs.
    - Provide an example of data catalog using DCAT, with selected classes: dcat:Dataset, dcat:DataService, dcat:Distribution a dcat:Catalog.
- Data semantics, data vocabularies
    - Describe the problem of data silos and how it can be tackled.
    - Explain advantages of using controlled vocabularies to describe data
    - Explain differences among: controlled list, taxonomy, thesaurus, classification scheme, and ontology.
    - Describe the basic structure of SKOS (Simple Knowledge Organization System).
    - Define custom ontology using SKOS and use it to annotate data.
- Basics of data encryption and compression
    - Explain Shannon's source coding theorem in context of data compression.
    - Demonstrate, with examples, the main ideas of the following compression algorithms: Run-Length Encoding, Huffman Coding, Lempel-Ziv-Welch a Arithmetic coding.
    - Explain role and use of digital certificates.
    - Describe PKI (public key infrastructure) and its relation to digital certificates.
    - Explain, using example, use of symmetric and asymmetric encryption in the context of digital certificates.
- Basics of indexing
    - Use examples to explain the basic types of file organization (heap file, sequential file, indexed sequential file) and their (dis)advantages.
    - Explain and demonstrate with an example the concepts of direct/indirect indexing, primary/secondary index.
    - Explain the principles of hashing in external memory, selected specific approach (e.g. Cormack, Larson & Kalja, Fagin, ...) demonstrate with an example.
    - Explain the purpose and advantages of hierarchical indexes. Use an example to demonstrate the B-tree data structure and related operations. Describe other modifications (B+ tree, B* tree).
- Indexing for spatial databases
    - Draw and explain the advantages of space-filling curves (Z-curve, Hilbert curve).
    - Draw a Quad-tree for the given example, explain the principle, advantages and disadvantages.
    - Draw a k-d-tree for the given example, explain the principle, advantages and disadvantages.
    - Draw an R-tree for the given example, explain the principle, advantages and disadvantages. Explain the differences from the R+ tree, or R* tree.
    - Explain the concept of a spatial join - principle, problems. Using an example, demonstrate a specific approach in more detail.

Courses

- NDBI046 Introduction to Data Engineering (5 cr)
- NDBI007 Principles of Data Organization (4 cr)
- NPRG036 Data Formats (5 cr)

3. Web

- Basic principles of www, HTML, XHTML, HTML5, and CSS
  - Create a simple static web page, personal blog or e-shop, using HTML or HTML5.
  - Explain advantages of HTML5 semantic tags.
  - Create HTML form with input validation.
  - Explain basic principles of CSS: syntax, selector specificity, how to insert CSS into HTML page.
  - Create a responsible layout for a page with a menu, a main content with columns, and a footer.
- Architecture, principles and design pattern of web applications
  - Explain and demonstrate use of: Front Controller, MVC/MVP, MVVC.
- Client side development, JavaScript, standard Web APIS
  - Write JavaScript that, in response to an event, queries the server, processes the response, and modifies the DOM.
  - Give examples of a standard API in a browser.
  - Explain approaches to asynchronous programming in JavaScript: callbacks, promises, async/await, event loop.
- Web application's API and web services
  - Explain the basic principles of the REST API.
  - Describe and explain REST API levels.
  - Create OpenAPI description for a REST API.
  - Write simple GraphQL queries and describe advantages and disadvantages of GraphQL.
- Single-page applications, state management and user sessions
  - Explain the principle of single-page applications.
  - Explain state management in the context of web application.
- Server side development, CGI and CGI-like applications
  - Explain how CGI and CGI-like applications work.
  - Explain state management and use of user sessions from the perspective of server side development.
  - Demonstrate use of PHP interleaving.
  - Using PHP and HTTP wrapper create a simple page with a connection to an SQL database.
- Basics of web applications security
  - Describe advantages of HTTPS with respect to HTTP.
  - Demonstrate the use of authentication tokens (JWT).
  - Identify basic security issues of web applications.
- Recommender systems
  - Explain the basic workflow of recommender systems, describe typical inputs and outputs, explain the challenges of the dynamic nature of recommenders (cold start problem, new item problem, online model updates, ...)
  - Collaborative filtering: explain principles, pros and cons, explain basic algorithms (user/item-based KNN, matrix factorization variants)
  - Content-based and knowledge-based recommending: explain principles, pros and cons
  - Evaluation of recommender systems: explain the differences and limitations of offline/online/ user-study evaluation protocols, give examples of frequently used evaluation metrics
- Multimedia retrieval and web search.
  - Describe boolean and vector models, word2vec.
  - Describe hypertex search, raking, PageRank.
  - Describe search engine optimization (SEO).
  - Describe metric similarity indexing (pivot-based filtering, matrix-based, tree-based, hashed and hybrid indexes)
  - Describe the basic formats for visual data (specifically BMP, JPEG, MP4) and basic principles of video compression (explain why P and B frames improve compression, explain the coding tree used by HEVC), describe the algorithms for shot boundary detection using frame similarity and convolutional neural networks (inference is sufficient)
  - Formalize and explain basic similarity search model (descriptor, similarity function, Cosine and Euclidean distances), explain basic principles of early/late fusion
  - Explain how to perform text search and classification in image databases using the CLIP network, explain basic building blocks of CLIP architecture (inference is sufficient)
  - Explain options to visualize ranked result set in 2D grid, describe how to visualize images in 2D grid with SOM (self-organizing map) and explain an algorithm to sort images (with respect to their similarity) in a 2D grid (self-sorting map)
  - Describe methods to measure effectiveness of search models (especially precision, recall, mAP, F1-score), describe options to evaluate search effectiveness of interactive search systems

Courses

- NDBI038 Searching the Web (4 cr)
- NDBI045 Video retrieval (5 cr)
- NSWI142 Web Applications Programming (5 cr)
- NSWI153 Advanced Programming of Web Applications (5 cr)
- NSWI166 Introduction to recommender systems and user preferences (4 cr)

General Computer Science

Upon exam registration, students of this specialization will pick two topics from topics 3 to 6 below, in addition to the common topics and topics 1 and 2 below.

1. Combinatorics

- Generating functions
  - using generating functions to solve linear recurrences
  - generalized binomial theorem (formulation)
  - Catalan numbers (example of combinatorial interpretation, derivation of formula in closed form)
- Estimates of factorials and binomial coefficients
  - basic estimates
    - $(n/e)^n \leq n! \leq en(n/e)^n$
    - $(n/k)^k \leq \binom{n}{k} \leq (en/k)^k$
    - $2^{2m}/(2\sqrt{m}) \leq \binom{2m}{m} \leq 2^{2m}/\sqrt{2m}$
- Ramsey's theorems
  - Ramsey's theorem (formulation of finite and infinite version for p-tuples, proof of version p=2 for 2 colors)
  - Ramsey numbers (definition, for two colors upper bound from the proof of Ramsey's theorem and lower bound by probabilistic construction)
- Extremal combinatorics
  - general understanding of what extremal combinatorics studies
  - Turan's theorem (formulations, Turan graphs)
  - Erdös-Ko-Rado theorem (formulations)
- Error-correcting codes
  - an overview of the terminology used
  - code distance and its relation to the number of correctable and detectable errors
  - Hamming estimate (formulation and proof)
  - perfect codes (definitions and examples, Hamming code without exact construction)

Courses

- NDMI011 Combinatorics and Graph Theory 1 (5 cr)
- NDMI012 Combinatorics and Graph Theory 2 (5 cr)

2. Differential and Integral Calculus in Multiple Dimensions

- Riemann integral in one and several dimensions
- Multivariable functions
  - partial derivatives: definition and calculation
  - computing extrema using partial derivatives
  - existence of extrema for multivariable functions
  - constrained extrema: calculation using Lagrange multipliers
- Metric space
  - definition and basic examples
  - open and closed sets: definitions, examples
  - continuity of a function on a metric space
  - compactness: definitions and implications for extrema of multivariable functions
  - uniform continuity

Courses

- NMAI055 Mathematical Analysis 2 (5 cr)

3. Optimization Methods

- Fundamentals of linear and integer programming
  - duality of linear programming, Farkas' lemma
  - simplex method, pivot rules
- Combinatorial geometry
  - convex hull
  - polyhedra
  - Minkowski-Weyl theorem
- Fundamentals of mathematical programming
  - unimodularity, König's lemma, network flows, connections with LP duality
  - weighted maximum matching in bipartite graphs and a primal-dual algorithm to find it
- Integer programming
  - cutting-plane method
- Matroids
  - rank function, existence and submodularity
  - greedy algorithm
- Approximation algorithms
  - definitions: approximation ratio, approximation scheme
  - greedy algorithms
    - travelling-salesman - Christofides' 3/2-approximation algorithm
    - scheduling
    - set cover
  - use of linear programming
    - algorithms for satisfiability (MAXSAT, probabilistic rounding)
    - vertex and set cover (deterministic rounding, primal-dual algorithm)
- Use of probability in algorithm design
  - minimum global cut in a graph
  - hashing and its use for constant-time dictionary search
  - probabilistic testing of matrix and polynomial identities
  - a parallel algorithm for finding the maximal independent set
  - parallel algorithms for finding matchings (bipartite graphs)

Courses

- NOPT048 Linear Programming and Combinatorial Optimization (5 cr)
- NOPT046 Discrete and Continuous Optimization (5 cr)
- NDMI084 Introduction to approximation and randomized algorithms (5 cr)

4. Advanced Algorithms and Data Structures

- Dynamic programming
  - principle of dynamic programming (solving subproblems from smallest to largest).
  - applications: longest increasing subsequence, edit distance
- Graph algorithms
  - strongly connected components of directed graphs
  - network flows (Dinic and Goldberg algorithms)
  - flows in integer-valued networks, application to bipartite matching
- Text search algorithms
  - Knuth-Morris-Pratt and Aho-Corasick algorithms
- Algebraic algorithms
  - Discrete Fourier transform and its applications
  - computation of the Fourier transform by the FFT algorithm
- RSA
  - encryption, decryption and key generation
- Approximation algorithms
  - approximation ratio and relative error
  - approximation schemes
  - examples: travelling salesman, knapsack problem
- Parallel sorting using comparator networks

Courses

- NDMI010 Graph Algorithms (3 cr)
- NMAI062 Algebra 1 (5 cr)
- NTIN060 Algorithms and Data Structures 1 (5 cr)
- NTIN061 Algorithms and Data Structures 2 (5 cr)

5. Geometry

- Basic theorems on convex sets
  - Helly's theorem
  - Rado's theorem
  - separation theorem
- Minkowski's theorem on lattices
- Convex polyhedra
  - fundamental properties
  - V-polyhedra, H-polyhedra
  - combinatorial complexity (maximum number of faces of a polyhedron, cyclic polyhedron and its properties)
- Geometric duality
  - dual set
  - dual polyhedron
  - duality between points and hyperplanes
- Voronoi diagrams
  - definitions
  - connection with polyhedra
  - upper bound for complexity
- Arrangements of hyperplanes
  - definitions
  - number of cells of a simple arrangement of n hyperplanes in d-dimensional space
  - zone theorem
- Incidence of points and lines
  - lower estimate of the number of incidences
  - Szemerédi-Trotter theorem
- Basic algorithms of computational geometry
  - construction of the arrangement of lines in plane
  - construction of a convex hull in the plane

Courses

- NDMI009 Introduction to Combinatorial and Computational Geometry (5 cr)

6. Advanced Discrete Mathematics

- Graph Colouring
  - definitions and basic properties
  - edge coloring (definition, formulation of Vizing's theorem, connection with matchings in graphs)
  - Brooks' theorem (formulation)
  - basic methods from the proofs of Vizing's and Brooks' theorems (Kempe chains, greedy algorithm)
  - strong and weak perfect graph theorem (formulation), chordal graphs and other examples of classes of perfect graphs
- Matchings in graphs
  - definitions of matching and perfect matching
  - matching in general graphs (formulation of Tutte's theorem including proof of the simpler implication, Petersen's theorem and its proof using Tutte's theorem)
  - Edmonds' algorithm (just knowing about its existence)
- Graphs on surfaces
  - basic topological concepts (homeomorphism, curve, surface)
  - construction of surfaces by adding ears and crosses (formulations), orientable and non-orientable surfaces, Euler characteristic
  - notion of 2-cell embedding

    - generalised Euler's formula, its use for upper estimation of the number of edges and minimum degree in a graph drawn on a given surface
- Graph minors
    - definition and basic properties
    - preservation of drawing in minor operations
- Sets and mappings
    - overview of the terminology used (classes and proper classes, Cartesian product, relation, mapping, sum, power set, ...)
- Subvalence and equivalence of sets
    - definitions
    - Cantor-Bernstein theorem (without proof)
    - countable sets
        - definitions
        - preservation of countability by set operations
    - cardinality of sets of rational and real numbers, proof of non-equivalence by the diagonal method
- Well-order
    - definitions
    - ordinal and cardinal numbers (definitions)

Courses

- NAIL063 Set Theory (3 cr)
- NDMI012 Combinatorics and Graph Theory 2 (5 cr)

Artificial Intelligence

Upon exam registration, students of this specialization will pick a branch from the choice of Robotics, Machine Learning and Natural Language Processing. In addition to the common topics, topic 1 below is common to all three branches, topic 2 applies to Robotics, topic 3 applies to Machine Learning, topic 4 applies to Natural Language Processing.

1. Foundations of Artificial Intelligence

- Problem solving via search
  - problem formulation
  - tree vs. graph search
  - uniformed search (DFS, BFS, uniform-cost search)
  - informed search (the A* algorithm, admissible and consistent heuristics)
- Constraint satisfaction
  - constraint satisfaction problem
  - arc consistency (the AC-3 algorithm)
  - constraint satisfaction techniques (the MAC algorithm)
- Logical reasoning
  - foundations of propositional logic (conjunctive and disjunctive normal forms)
  - algorithm DPLL
  - forward and backward chaining (Horn clauses)
  - resolution principle
- Probabilistic reasoning
  - foundations of probability theory (full joint distribution, independence, Bayes' rule)
  - probabilistic reasoning (summing out, normalization)
  - Bayesian Networks
    - construction and relation to full joint distribution
    - exact inference (enumeration, variable elimination)
    - approximate inference (Monte Carlo, rejection sampling, likelihood weighting)
- Knowledge representation
  - situation calculus, frame problem
  - Markovian models
    - filtering, prediction, smoothing, most likely explanation
    - Hidden Markov Models (HMM) vs. dynamic Bayesian networks
- Automated planning
  - formulating planning problems (planning operators)
  - forward and backward planning
- Markov decision processes (MDP)
  - problem formulation (utility functions, policy)
  - Bellman equation
  - value iteration, policy iteration
  - POMDP (basic definitions)
- Games and theory of games
  - algorithms Minimax and alfa-beta pruning
  - foundations of game theory (prisoner's dilemma, Nash equilibrium)
  - mechanism design (auctions)
- Machine learning
  - forms of learning (supervised, unsupervised, reinforcement learning)
  - decision trees (definition, construction)
  - regression, SVM (basic principles)
  - Bayesian learning, the EM algorithm
  - reinforcement learning
    - pasive learning (definition, ADP and TD techniques)
    - active learning (definition, exploration vs. exploitation, Q-learning, SARSA)

Courses

- NAIL120 Introduction to Artificial Intelligence (5 cr)

2. Robotics

- Kinematics
  - motion and transformation
    - 2D, 3D translation, rotation, general transformation
    - Cartesian and polar coordinate systems
    - Homogeneous transformation
  - basic problem-solving
    - simple transformations
    - compound transformations
    - Denavit-Hartenberg convention
- Control systems
  - architectures and their hierarchies
    - reactive, deliberative, subsumption, behavioral, three-layer architectures
  - implementation
    - monolithic systems
    - state automata
    - middleware
  - specific run-time environments
    - low- and high-level systems
    - programming for resource-limited environments
- Motion, sensorics
  - motion types
    - land moving wheeled and track-based vehicles and their control (differential, Ackermann, omnidirectional)
    - aerial vehicles
  - basic actuators and sensor types
    - electric motors
    - contact and contactless sensors
  - closed loop control, PID
  - input data processing, filtration.
- Localization and mapping
  - possibilities for determining position
    - absolute and relative
    - passive and active
    - odometry, triangulation, trilateration, satellite systems
    - probabilistic approach
  - map types
    - metric, geometric, topological maps, occupancy grid
    - abstraction levels
  - model situation solving
  - simultaneous localization and mapping
- Image processing, computer vision
  - segmentation
  - object searching and tracking

Courses

- NAIL028 Introduction to Robotics (5 cr)
- NPGR036 Computer Vision (5 cr)
- NPRG037 Microcontroller Programming (5 cr)

3. Machine Learning

- Supervised learning
  - classification, regression (basic types of tasks)

    - standard performance measures
    - model assessment (test data, cross-validation, maximum likelihood)
    - overfitting and regularization (generalization error, early stopping, L2 a L1 regularization)
    - curse of dimensionality
- Instance-based learning
    - k-nearest neighbors algorithm
- Linear regression
    - least squares method
    - stochastic gradient descent
- Logistic regression
    - binary classification (sigmoid function, training methods)
    - multi-class classification (softmax function, training methods)
- Decision trees
    - learning algorithm and splitting criteria
    - pruning
- Support vector machines
    - large margin classifier
    - soft margin classifier
    - kernel functions
    - multi-class classification
- Ensemble learning
    - bagging and boosting
    - random forest algorithm
- Statistical tests
    - student's t-test (one sample and two sample)
    - chi-squared test
- Unsupervised learning
    - clustering (k-means algorithm)
    - hierarchical clustering
    - dimensionality reduction (Principal component analysis)

Courses

- NAIL121 Seminar on Data Mining (4 cr)
- NPFL054 Introduction to Machine Learning with R (5 cr)
- NPFL129 Introduction to Machine Learning with Python (5 cr)
- NPGR035 Machine Learning in Computer Vision (5 cr)

4. Natural Language Processing

- Layers of language description, morphological and syntactic analysis
    - system of levels of language description, relations between layers
    - morphological analysis, lemmatization, parts-of-speech and other morphological categories, morphological tag sets, examples of algorithms for statistical tagging
    - formal description of syntax of natural languages, constituency and dependency approaches
    - syntactic analysis, examples of algorithms for statistical parsing
- Basic notions in probability theory and information theory
    - definition of probability, basic properties
    - statistical independence
    - Bayes' theorem
    - entropy
    - mutual information
    - Kullback–Leibler divergence
- Statistical methods in Natural Language Processing, language models
    - n-gram language model
    - language model estimation using maximum likelihood
    - smoothing of language models, usage, examples of smoothing methods
- Machine learning, classification, regression
    - supervised and unsupervised machine learning

    - construction of loss functions using maximum likelihood (NLL, MSE)
    - the task of classification, classification algorithms (decision trees, logistic regression, support vector machine, perceptron, multilayer perceptron)
    - examples of applications of classification in Natural Language Processing
    - the taks of regression, regression algorithms (decision trees, linear regression)
    - model combination (voting, ensembling, boosting)
    - clustering (k-means)
- Generalization error estimation, overfitting, regularization
    - performance evaluation in classification and regression tasks (RMSE, accuracy, confusion matrix, precision, recall, F1 score, Fbeta score, precision-recall curve, specificity, sensitivity, ROC curve)
    - the problem of overfitting, possible causes, possible solutions
    - examples of regularization in linear and logistic regression, regularization factors L1 and L2
- Vector representations of words, fundamentals of deep learning
    - basic word embedding methods (skipgram, CBOW)
    - perceptron, multilayer perceptron (activation function of hidden and output layers), stochastic gradient descent
    - Universal approximation theorem
    - convolutional networks and basic architectures for applying them in Natural Language Processing
    - recurrent networks and basic architectures for applying them in Natural Language Processing
- Natural Language Processing applications, examples of evaluation measures
    - Information Retrieval, inverted index, the bag-of-words model, TF-IDF,
    - statistical machine translation (phrase-based)
    - neural machine translation

    - using precision, recall, and f-score in Natural Language Processing
    - task-specific evaluation measures (BLEU, UAS etc.)
    - cross-validation

Courses

- NPFL012 Introduction to Computer Linguistics (3 cr)
- NPFL054 Introduction to Machine Learning with R (5 cr)
- NPFL124 Natural Language Processing (4 cr)
- NPFL129 Introduction to Machine Learning with Python (5 cr)

Requirements change history

All notable changes to the requirements are documented in this section.

2024-12-18

- typo fixes in General Computer Science
- more precise wording for vector spaces in Foundations of Mathematics
- removed requirements concerning Hadwiger hypothesis and upper bound on graph minor edges in General Computer Science
- added integer flows and their applications and removed red-black trees and transitive closures in General Computer Science

2024-01-05 (applies starting February 2024)

Miscellanea

- more precise wording for variants related to programming language selection in Foundations of Computer Science

2023-08-12 (applies starting September 2023)

Miscellanea

- more precise wording for concepts around inheritance in Foundations of Computer Science
- more precise wording for classification performance in Artificial Intelligence

2022-12-08 (applies starting February 2023)

Removals

- removed requirements concerning computer networks in General Computer Science

Miscellanea

- more precise wording for variants related to programming language selection