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 cover each topic.

For each topic, a frame with the study plan requirements text is provided.

Below the frame, the requirements are elaborated in term of specific knowledge and supplemented with the list of courses with credit budgets that normally cover the topic.

The text includes common requirements from mathematics and computer science and requirements particular to individual specializations. Requirements of individual specializations may overlap with common requirements where the specializations require more depth - in such cases the common requirements are included implicitly. Requirements that concern specific programming languages apply to the same degree as elective course requirements of each specialization.

From one exam term to another, the text can be updated to reflect the requirements of the study program and the feedback of the exam board and the students. Starting one month before each term, potential modifications are limited so that no new requirement is added.

This is a version dated June 1, 2022, prepared for the Bachelor State Final Exams from June 2022 on.

Please address any comments to the head of the exam board.
Foundations of Mathematics

1. Fundamentals of differential and integral calculus


- Real numbers
  - The square root of two is irrational
  - \( \mathbb{R} \) is an uncountable set
- 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


- Groups and subgroups (definition, examples, commutativity)
- Fields (definition, characterization of a field, finite fields)
- Vector spaces and subspaces
  - properties and basic concepts (linear combination, linear cover, generators, linear dependence and independence, basis, dimension, coordinates) and their use
- practical skill to test linear dependence and independence, finding a basis, determining the dimension etc.
- scalar product and its properties
- norm and its relation to scalar product, examples
- orthogonality, orthonormal basis, its properties and use (finding coordinates and projection)
- Systems of linear equations and the solution set
  - solution methods, Gauss and Gauss-Jordan elimination, block shape of a matrix and its uniqueness (w/o a proof)
- Matrices and operations with matrices (sum, product, transposition)
  - interpretation of matrix product by composition of linear mappings
  - rank of a matrix and of its transposition
  - eigenvalues and eigenvectors of a matrix and their geometric meaning and properties, multiple eigenvalues, spectral radius
  - characteristic polynomial, relation of eigenvalues to roots of polynomials

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 permutations, 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, basic examples of graphs. Connected graphs, connected components. Trees, their properties, equivalent characterizations of trees. Planar graphs, Euler’s formula and the maximum number of edges in a planar graph. Graph colourings, chromatic number and clique number. Edge- and vertex-connectivity, Menger’s theorem. Directed graphs, weak and strong connectivity. Network flows.

- 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 leaves, number of edges in a tree)
  - equivalent 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 capacities (e.g. using the Ford-Fulkerson algorithm)

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

5. Probability and Statistics

Random events, conditional probability, independence of random events, Bayes’ formula, applications. Random variables, mean (expectation), distribution of random variables, geometric, binomial, and normal distribution. Linear combination of random variables, linearity of expectation. Point estimates, confidence intervals, hypothesis testing.

- 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
6. Logic

Syntax - language, open and closed formulas. Normal forms of propositional formulas, prenex forms of predicate logic formulas, converting to normal form, applications in algorithms (SAT, resolution). Semantics, truth, falsity, independence of a formula with respect to a theory, satisfiability, tautologies, logical consequence, the notion of a model of a theory, extensions of theories.

- 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
  - finite automaton, language accepted by a finite automaton
  - deterministic and nondeterministic automaton, lambda transitions
  - regular expressions
  - Kleene’s theorem
  - iteration (pumping) lemma for finite automata
- Context-free languages
  - context-free grammars, language generated by a grammar
  - push-down automaton, class of languages accepted by push-down automata
- Turing machine
  - type 0 grammar
  - diagonal language
  - universal language
- Chomsky hierarchy
  - determining an equivalence or inclusion among classes of languages generated by the above described automata and grammars
  - ability to classify a concrete language into the correct level of Chomsky hierarchy (typically by constructing the appropriate automaton or grammar and proving that the language is not on a lower level using the iteration lemma)

Courses

- NTIN071 Automata and Grammars (5 kr)

2. Algorithms and Data Structures

- Time complexity of algorithms
  - time and space of a computation for a concrete input
  - time and space complexity of an algorithm
  - data size measurement
  - 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
  - applications
– Mergesort
– long number multiplication
– Strassen’s algorithm
– Binary search trees
  – definition of a search tree
  – operations with unbalanced trees
  – AVL trees (definition)
– Heaps, binary heap
– Hashing, hashing with buckets, open addressing
– Sorting
  – primitive sorting algorithms (Bubblesort, Insertsort)
  – Heapsort
  – Quicksort
  – complexity lower bound for sorting algorithms that use comparisons
  – Bucket sort for numbers and strings
– Graph algorithms
  – BFS and DFS
  – detection of connected components
  – topological sort for directed graphs
  – shortest paths in valued graphs (Dijkstra’s and Bellman-Ford’s algorithms)
  – minimum spanning trees (Jarník’s and Borůvka’s algorithms)
  – network flows (Ford-Fulkerson’s algorithm)
– Algebraic algorithms, Euclid’s algorithm

Relevant courses
– NTIN060 Algorithms and Data Structures 1 (5 kr)
– NTIN061 Algorithms and Data Structures 2 (5 kr)

3. Programming Languages


– Concepts for abstraction, encapsulation, and polymorphism.
  – related programming language constructs
    – classes, interfaces, methods, data members, inheritance, visibility
  – (dynamic) polymorphism, static and dynamic typing
  – single inheritance
    – virtual and non-virtual methods in C++ and C#
    – default methods in Java
  – multiple inheritance and related issues
    – multiple and virtual inheritance in C++
    – interfaces in Java and C++
  – interface implementation
– Primitive and object types and their representation.
  – numeric and enumerable types
  – pointers and references in C++
  – value and reference types in C#
  – immutable types and boxing/unboxing 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 in C++, C#, or Java
  - 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)


Computer organization, data and program representation. Instruction set architecture as a hardware/software interface, connection to elements of high-level programming languages. Support for operating system execution. Peripheral device interface and handling. Fundamental OS abstractions, interfaces, and mechanisms for program execution, resource sharing, and input/output. Parallelism, threads and interfaces for thread management, thread synchronization.

- 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)
Databases and Web

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

1. Databases


- Transaction processing
  - Properties of transactions and scheduling, locking protocols, blocking.

- An overview of SQL
  - Core SQL queries.
  - Data grouping and aggregation.
  - Joining of tables.
  - Nested queries, tests for a NULL value.

- Modern database systems
  - Basic classification, specific properties, comparison with traditional (SQL) databases.
  - Big Data - features, challenges for traditional database systems.
  - MapReduce - principle, an example, properties, critique, alternative approaches.
  - NoSQL database - types and features, advantages and disadvantages.
  - Graph databases - principle and specific properties, classifications and examples of queries.
  - Multi-model databases vs. polystores - differences, advantages and disadvantages, an example of a multi-model query and problem of its evaluation.

Courses

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

2. Data management


- Data formats
  - Basic structured data types, their representants and use cases.
  - Data model, data format, data schema. Basic properties of text-based data formats, standardization of data formats.
  - Formats for graph data, schemas, languages for data transformation - RDF and its serializations, RDF Schema, SPARQL, Labeled Property Graph, Cypher.
- Formats for nested (tree-shaped, hierarchical) data, schemasm languages for data transformation - Basics of XML, basics of XML Schema, basics of XSLT, RDF/XML, JSON, JSON Schema, JSON-LD.
- Formats for tabular data, schemas, languages for data transformation - SQL dump, CSV and its specification, CSV on the Web, transformation of CSV to RDF.
- Formats for spatial data - Coordinate reference systems, relations of spatial objects, WKT, GML, GeoJSON, GeoSPARQL.

- Data management
  - Data processing (data selection, data projection, data summarization, data reduction, data lifting and data lowering).
  - Data cleaning, data quality, data quality dimensions.
  - Data provenance.
- Data catalogues, metadata.
  - Purpose and use of different types of metadata.
  - Data catalog and its use. Awareness of DCAT (Data Catalog Vocabulary) and its basic classes.
- Semantic data
  - Data silos and their limitations.
  - Benefits of using controlled vocabularies to describe data.
  - Types of controlled vocabularies (controlled list, taxonomy, thesaurus, classification scheme, ontology).
  - SKOS (Simple Knowledge Organization System).
  - Basic data ontology and its usage to describe data.
- Data encryption and compression
  - Shannon’s source coding theorem in the context of data compression.
  - Principles of data compression (Run-Length Encoding, Huffman Coding, Lempel-Ziv-Welch an Arithmetic coding).
  - Use of symmetric and asymmetric data encryption in the context of digital certificates.
- Basics of indexing
  - File organization techniques, direct/indirect indexing, primary/secondary index.
  - Principles of hashing in external memory, a selected approach in more detail (e.g., Cormack, Larson & Kalja, Fugin, ...).
  - Hierarchical indexing (B-Tree, B+-Tree, B*-Tree).
- Indexing for spatial databases
  - Space-filling curves (Z-Curve, Hilbert Curve).
  - Quad-Tree, k-d-Tree - basic principles, advantages and disadvantages
  - R-Tree, R+-Tree, R*-Tree - basic principles, differences
  - Spatial join, spatial query - basic principles and problems, a selected approach in a more detail.

Courses
- NDBI046 Data Management (5 cr)
- NDBI007 Database Access Methods (4 cr)
- NPRG036 Data Formats (5 cr)

3. Web


- World Wide Web
  - Static websites. Principles of markup languages (HTML, HTML5) and their interpretation by web browsers (visualisation, DOM).
  - Web site styling using CSS (syntax, basic principles and properties).
  - User interaction using links and form.
  - Architecture and principles of web applications
- Design pattern (Front Controller, MVC/MVP, MVCC).
- Client site programming (JavaScript). ECMAScript syntax, principles of OOP, functional programming (scope chaining, closure).
- Modification of web page via DOM (principles, API overview). DOM event handling, event-driven model, asynchronous programming in JavaScript (callbacks, promises).

- Web API
  - REST API, GraphQL API, OpenAPI.
- Web application security
  - Authentication, tokens (JWT).
  - Authorization, security models (ACL).
- Recommender systems
  - Collaborative filtering: principles, pros and cons, algorithm examples (user/item-based KNN, matrix factorization variants)
  - Content-based and knowledge-based recommending: principles, pros and cons
  - Evaluation of recommender systems (offline / online / user-studies, metrics)
- Web search
  - Boolean and vector models, word2vec.
  - Hypertext search, raking, PageRank.
  - Search engine optimization (SEO).
  - Similarity search in multimedia databases.

- Metric similarity indexing (pivot-based filtering, matrix-based, tree-based, hashed and hybrid indexes)

Courses
- NDBI038 Searching the Web (4 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 4 to 7 below, in addition to the common topics and topics 1 to 3 below.

1. Networking Fundamentals

<table>
<thead>
<tr>
<th>Topic</th>
<th>Description</th>
</tr>
</thead>
</table>
| - Basic concepts and taxonomy of computer networks | - connection-oriented/connectionless, block/stream, reliable/unreliable data transfer  
- circuit switching vs. packet switching, store and forward principle, best effort vs. QoS  
- network sizing, Internet, server-type networks, P2P networks |
| - ISO/OSI architecture and its relation to TCP/IP | - principle of layered architecture and its importance  
- functions of each layer, communication between layers  
- the importance of ISO/OSI in modern TCP/IP-based networks  
- examples of technologies/protocols commonly used in TCP/IP at each layer |
| - Routing and IP protocol | - importance of IPv4 protocol and IPv4 addresses  
- IP datagram routing algorithm  
- the meaning of routing tables and how to configure them  
- typical problems in routing and delivery of IP datagrams, ICMP protocol |
| - Transport layer | - TCP and UDP protocols (basic overview and functions)  
- port concept and its meaning, communication link concept, sockets |
| - Client-server architecture | - relationship to transport protocols  
- the importance of this architecture in terms of network application design and application protocols |
| - The most important TCP/IP application protocols | - HTTP  
- basic principle, use in applications  
- message format, overview of important headers, content encoding methods  
- basic mechanisms (content negotiation, range serving, caching, sessions) |
| - FTP | - basic functioning, active vs. passive mode, login vs. anonymous mode  
- overview of basic command sequences and how they work |
| - Electronic mail | - basic principles, terms and protocols (mailbox, MTA, SMTP, POP3, IMAP)  
- email message format, important headers  
- SMTP protocol, overview of basic commands, message forwarding  
- spam and how to defend against it (configuring SMTP servers, sender identification, message filtering) |

Courses

- NSWI141 Introduction to Networking (3 cr)

2. Combinatorics

<table>
<thead>
<tr>
<th>Topic</th>
<th>Description</th>
</tr>
</thead>
</table>

Courses
- 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
    - \((n/e)^n \leq n! \leq n(n/e)^n\)
    - \((n/k)^k \leq \left(\frac{n}{k}\right) \leq (en/k)^k\)
    - \(2^{2m}/(2\sqrt{m}) \leq \left(\frac{2^m}{m}\right) \leq 2^{2m}/\sqrt{2m}\)
- 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)

3. 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)

4. 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)

5. 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
  - transitive closure
  - strong connected components of oriented graphs
  - network flows (Dinic and Goldberg algorithms)

- 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
- Red-black trees and their balancing

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)

6. 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)

7. 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
  - formulation of the Hadwiger hypothesis
  - upper bound on the number of edges (without proof)
  - 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 (algorithm A*); constraint satisfaction. Logical reasoning (forward and backward chaining, resolution, SAT); probabilistic reasoning (Bayesian networks); knowledge representation (situation calculus, Markovian models). Automated planning; Markov decision processes. Games and theory of games. Machine learning (decision trees, regression, reinforcement learning).

- 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
  - passive 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, basic problem-solving. Control systems: architectures, implementation, specific run-time environments. Motion, sensorics: motion types, basic actuators and sensor types, closed loop control, input data processing. Localization and mapping: possibilities for determining position, map types, model situation solving, simultaneous localization and mapping. Image processing, computer vision: object searching and tracking.

- 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, omni-directional)
    - 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 and regression, error measure, model assessment (test data, cross-validation, maximum likelihood), overfitting and regularization, the curse of dimensionality. Instance-based learning, linear and logistic regression, decision trees, pruning, ensemble learning (bagging, boosting, random forest), support vector machines, t-test, chi-squared test. Unsupervised learning, clustering.

- 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

- 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 task 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
  - 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)