Open Informatics – International Computer Science Program

Bachelor program on Informatics and Computer Science

Michal Pěchouček, Jiří Matas, . . . *

document generated at: 15:31, November 19, 2010

Contents

*document preparation Tomáš Svoboda

1 Study plan

Open Informatics bachelor, major: Computer Science and Informatics

Semestr	Mandatory courses	Mandatory courses for the major specialization	Facultative courses		
1 (winter)	AE0B01LAG 4p+2s Linear Algebra	AE4B01DMA 2p+2s Discrete mathematics	AE0B36PR1 2p+2c Programming 1	AE4B99RPH 1p+3c Solving problems and other games	Humanistic, economy, management courses
2 (summer)	AE4B01MA2 4p+2s Calculus	AE0B01LGR 3p+2s Logic and Graph Theory	AE0B36PR2 2p+2c Programming 2	AE4B33ALG 2p+2c Algorithms	Humanistic, economy, management courses
3 (winter)	AE0B01PSI 4p+2s Probability, Statistics, and Theory of Information	AE4B01JAG 2p+2s Languages, automata and grammars	AE0B35SPS 3p+2l Computer Systems Structures	AE4B33OSS 2p+2c Operating systems and networks	AE4B01NUM 2p+2c Numerical Analysis
4 (summer)	AE4B02FYZ 2p+2l Physics for Informatics	AE0B36APO 2p+2l Computer Architectures	AE4B33DS 2p+2c Database Systems	AE4B33FLP 2p+2c Functional and Logic Programming	AE4B33ZUI 2p+2c Introduction to Artificial Intelligence
5 (winter)	AE4B33OPT 4p+2c Optimization	AE4B33RPZ 2p+2c Pattern Recognition and Machine Learning	AE4B99SVP TBD Software or Research Project		
6 (summer)			Humanistic, economy, management courses		AE4B99BAP TBD Bachelor Project

2 Mandatory courses

2.1 AE0B01LAG: Linear Algebra

Title: Linear Algebra
Lecturer: TBD
Term: winter
Lectures + seminars or labs: 4p+2s
ECTS credits: 7
FEL www: AE0B01LAG
Anotation: This course covers introductory topics of linear algebra. The main focus is on the related notions of linear spaces and linear transformations (linear independence, bases and coordinates) and matrices (determinants, inverse matrix, matrix of a linear mapping, eigenvalues). Applications include solving systems of linear equations, geometry in 3-space (including dot product and cross product), and solving linear differential equations.

Syllabus:

1. Introduction, polynomials.
2. Linear spaces, linear dependence and independence.
3. Basis, dimension, coordinates of vectors.
4. Matrices, operations, determinants. Inverse matrix.
5. Systems of linear equations.
6. Linear mappings. Matrix of a linear mapping.
7. Free vectors. Dot product and cross product.
8. Lines and planes in 3-dimensional Euclidean space.
9. Eigenvalues and eigenvectors of matrices and linear mappings.
10. Similarity of matrices, matrices similar to diagonal matrices.
11. Generalized eigenvectors.
12. Systems of linear differential equations of 1st order with constant coefficients.
13. Linear differential equations of order n with constant coefficients.
14. Back-up class.

Labs/Seminars:

1. Polynomials.
2. Examples of linear spaces, linear independence.
3. Basis, coordinates of vectors.
4. Operations with matrices, determinants. Finding inverse matrix.
5. Systems of linear equations.
6. Examples of linear mappings.
7. Matrix of a linear mapping, change of basis.
8. Dot product and cross product in geometry. Lines and planes.
9. Eigenvalues and eigenvectors of matrices.
10. Diagonalization of matrices.
11. Generalized eigenvectors and applications.
12. Systems of linear differential equations.
13. Linear differential equations of order n.
14. Back-up class.

Literature:

1. P. Pták: Introduction to Linear Algebra. ČVUT, Praha, 2005.

2.2 AE4B01DMA: Discrete mathematics

Title: Discrete mathematics
Lecturer: TBD
Term: winter
Lectures + seminars or labs: 2p+2s
ECTS credits: 7
FEL www: AE4B01DMA
Anotation: The aim of the course is to introduce students to some areas of mathematics outside of the customary continuous mathematics. The common denominator here is a discrete approach, combinatorial thinking and insight into mathematical reasoning and notation. The course will explore notions of cardinality and properties of natural numbers, relations on sets, binomial theorem and combinatorics, mathematical induction and recurrence.

Syllabus:

1. Sets and their cardinality, countable and uncountable sets.
2. Integers, primes, Euclid's algorithm.
3. Binary operations and their properties.
4. Binary relations on a set, equivalence.
5. Modulo n relation on the set of integers.
6. Partial ordering.
7. Matrix of relation, relation database.
8. Binomial theorem and its applications, properties of combinatorial numbers.
9. Estimating binomial coefficients, principle of inclusion and exclusion.
10. Matematical induction and its applications.
11. Matematical induction as a tool for solving recurrence relations.
12. Evaluating time complexity of recursive algorithms, solving homogeneous recurrence equations with constant coefficients.
13. Solving non-homogeneous recurrence equations with constant coefficients.
14. Back-up class.

Labs/Seminars:

1. Bijections, countable sets.
2. Properties of numbers.
3. Properties of binary operations.
4. Counting modulo n.
5. Properties of binary relations.
6. Equivalence and partial ordering.
7. Relation matrix, relation database.
8. Combinatorics, binomial theorem.
9. Combinatorics.
10. Proofs by mathematical induction.
11. Matematical induction and recurrence.
12. Complexity of algorithms. solving homogeneous recurrence equations.
13. Solving non-homogeneous recurrence equations with constant coefficients.
14. Back-up class.

Literature:

1. M. Demlová: Mathematical Logic. ČVUT Praha, 1999.
2. R. Johnsonbauch: Discrete Mathematics, 4th edition, 1997,
3. K.H.Rosen: Discrete matematics and its aplications, McGraw-Hill, 1998.
4. Lecturer's official homepage.

2.3 AE0B36PR1: **Programming 1**

Title: Programming 1
Lecturer: TBD
Term: winter
Lectures + seminars or labs: 2p+2c
ECTS credits: 6
FEL www: AE0B36PR1
Anotation: The aim of the course is to teach the students: basic interactions with user interface and to program development system, introduction to JAVA, basic control flow structures and data structures, functions, arrays, object-oriented programming concepts, streams and files. The students are able to construct and debug a simple program in Java.

Syllabus:

1. Basic terms of information technology and computer technology, operating systems, software, compiler, interpreter, programming languages, syntax, semantics
2. Java conception, basic properties, course and trends, introduction in the language, program processing, byte code
3. Program structure in Java, program construction, program debugging, applications development systems, a simple program
4. Basic properties of imperative programming, algorithm development, variables, operators, simple data objects, assignment
5. Expressions, I/O- reading and writing, control flow structures, conditions, loops, iteration
6. Functions, procedures, parameters, parameter passing, static variables, local variables, block, memory management for functions and variables, heap, stack
7. Array, reference variable, array as a parameter, function of array type
8. Decomposition of problem into sub-problems, principle of recursion and iteration
9. Implementation of tables and sets by means of array, multidimensional array, collections, containers in Java
10. Principles of object-oriented programming, classes, class as a programming unit, class as an origin of data type, static and instance variable
11. Structure of the object, dynamic character of objects, constructors, constructor properties, instance of a class, overloading, authorized access
12. Classes, inheritance, hierarchy of classes, composition, abstract classes, polymorphism,
13. Files and streams, file as a sequence of bytes, introduction to exceptions, writing/reading of primitive types, primitive types and objects (strings), objects, serialization
14. Reserve

Labs/Seminars:

1. Introduction to computing system of the university
2. Introduction to program development system
3. Structure of the program in Java
4. Debugging of trivial tasks in program development system, running out of it
5. Program sequences, input, output, control flow structures
6. Cycles
7. Arrays, assignment of a semester task
8. Non-simple problem solving, decomposition problem to sub-problems
9. Functions and procedures
10. Classes and objects I

11. Classes and objects II
12. Files and streams
13. Test
14. Credit

Literature:

1. Zakhour, S: The Java Tutorial: A Short Course on the Basics, 4th Edition, Amazon, 2006
2. Hawlitzek, F: Java 2, Addison-Wesley, 2000
3. Eckel, B: Thinking in Java 2, Prentice Hall, 2000

Notes from Faculty Information System:

2.4 AE4B99RPH: Solving problems and other games

Title: Solving problems and other games
Lecturer: Ing. Svoboda Tomáš Ph.D.
Term: winter
Lectures + seminars or labs: 1p+3c
ECTS credits: 6
FEL www: AE4B99RPH
Anotation: The main motivation is to let students to deal with real-world problems properly. When working in teams on real problems the student shall learn how to decompose the big problem, how to define interfaces, how to test and validate individual steps and so on. Many problems will actually be beyond the first-year-student skills. And many problem will not be solved in the optimal way. The unsolved parts should motivate the students to study difficult theoretical subjects. They should generate the important questions. Ideally, at the end of the subject, the student should be eager to study deeper about informatics.

Syllabus:

1. Motivation lecture about selected problems in computer science and artificial intelligence.
2. motivation lecture about selected problems in computer science and artificial intelligence.
3. Essentials of engineering work. Problem decomposition, testing, verification.
4. Bibliographic resources, finding relevant previous work, how to avoid plagiarism.
5. Project organization, planning, lifecycle.
6. Project documentation and presentation
7. Team work, software for versioning, collaborative work.

Labs/Seminars:

1. An incomplete list of possible tasks.
2. Spam filter. How to represent frequency of bad words.
3. Plagiarism detection in text documents. Similar task as the spam filter.
4. Zip code optical character recognition.
5. Finding the optimal path for a mobile robot through the labyrinth.
6. Traffic light switching for the maximally fluent traffic in urban areas.
7. Organization of multimedia data. How to organize them automatically, ideally according to the content.
8. Software for checking conference deadlines. Call for papers are coming by email. How to transfer the content into a structured www page.

Literature:

1. Will be provided individually depending on the selected task.

Notes from Faculty Information System:

2.5 AE4B01MA2: Calculus

Title: Calculus
Lecturer: TBD
Term: summer
Lectures + seminars or labs: 4p+2s
ECTS credits: 8
FEL www: AE4B01MA2
Anotation: This course covers the standard basics of continuous mathematics. First, for functions of one variable we cover limits, derivatives and integration, which is followed by sequences and series of real numbers. The acquired skills are then applied to functions of more variables, where we use partial derivatives to find extrema. The focus is on practical computational skills and on understanding the meaning of notions and calculations. The course is concluded by a survey of power series and a brief introduction to ordinary differential equations, whose main purpose is to show students that continuous mathematics is a powerful tool.

Syllabus:

1. Introduction. Limit of a function.
2. Continuity. Introduction to derivatives.
3. Differentiation and basic theorems, l'Hospital's rule.
4. Monotonicity and extrema. Applications of derivative (Taylor polynomial).
5. Graph sketching. Introduction to indefinite integral.
6. Properties of integral, methods of evaluation.
7. Definite integral.
8. Improper integral. Applications of integral.
9. Sequences. Introduction to series.
10. Series. Introduction to functions of more variables.
11. Functions of more variables (including extrema without and with constraints).
12. Series of functions (region of convergence, expanding a function in a power series).
13. Brief introduction to differential equations.
14. Back-up class.

Labs/Seminars:

1. Review, domains of functions.
2. Limit of a function.
3. Differentiation, tangent and normal lines.
4. Limit using l'Hospital's rule.
5. Monotonicity and extrema.
6. Taylor polynomial. Graph sketching.
7. Basic methods of integration.
8. Definite integral.
9. Improper integral. Applications of integral.
10. Limit of a sequence, intuitive evaluation. Scale of powers.
11. Testing series convergence.
12. Partial derivative, local extrema.
13. Constrained extrema. Power series.
14. Solving differential equations by separation.

Literature:

1. M. Demlová, J. Hamhalter: Calculus I. ČVUT Praha, 1994.
2. P. Pták: Calculus II. ČVUT Praha, 1997.
3. Habala, P.: Math Tutor, http://math.feld.cvut.cz/mt/

2.6 AE0B01LGR: Logic and Graph Theory

Title: Logic and Graph Theory
Lecturer: TBD
Term: summer
Lectures + seminars or labs: 3p+2s
ECTS credits: 6
FEL www: AE0B01LGR
Anotation: The course covers basics of logic and theory of graphs. Propositional logic contains: truth validation, semantical consequence and tautological equivalence of formulas, CNF and DNF, complete systems of logical connectives, and resolution method in propositional logic. In predicate logic the stress is put on formalization of sentences as formulas of predicate logic, and resolution method in predicate logic. Next topic is an introduction to the theory of graphs and its applications. It covers connectivity, strong connectivity, trees and spanning trees, Euler?s graphs, Hamilton?s graphs, independent sets, and colourings.

Syllabus:

1. Formulas of propositional logic, truth validation, tautology, contradiction, satisfiable formulas.
2. Semantical consequence and tautological equivalence in propositional logic.
3. CNF and DNF, Boolean calculus.
4. Resolution method in propositional logic.
5. Predicate logic, syntactically correct formulas
6. Interpretation, sematical consequence and tautological equivalence.
7. Resilution method in predicate logic.
8. Directed and undirected graphs.
9. Connectivity, trees, spanning trees.
10. Strong connectivity, acyclic graphs.
11. Euler?s graphs and their application.
12. Hamilton?s graphs and their application.
13. Independent sets, cliques in graphs.
14. Colourings.

Labs/Seminars:

1. Formulas of propositional logic, truth validation, tautology, contradiction, satisfiable formulas.
2. Semantical consequence and tautological equivalence in propositional logic.
3. CNF and DNF, Boolean calculus.
4. Resolution method in propositional logic.
5. Predicate logic, syntactically correct formulas
6. Interpretation, sematical consequence and tautological equivalence.
7. Resilution method in predicate logic.
8. Directed and undirected graphs.
9. Connectivity, trees, spanning trees.
10. Strong connectivity, acyclic graphs.
11. Euler?s graphs and their application.
12. Hamilton?s graphs and their application.
13. Independent sets, cliques in graphs.
14. Colourings.

Literature:

1. M. Demlová: Mathematical Logic. ČVUT Praha, 1999.
2. R. Diestel: Graph Theory, Springer-Verlag, 1997

2.7 AE0B36PR2: **Programming 2**

Title: Programming 2
Lecturer: TBD
Term: summer
Lectures + seminars or labs: 2p+2c
ECTS credits: 6
FEL www: AE0B36PR2
Anotation: The course moves along the understanding of programming skills from Programming 1, the aim is to design an interactive application with a graphic user interface (GUI), with knowledge of polymorphism abstract classes, interfaces, events handling, applets, user libraries, library practical application. Further students continue by the comparative way in getting acquainted in C language on the base of Java language, dynamic memory management, students are able to analyze the simple programs in C language.

Syllabus:

1. Revision and resume of programming basic in Java, arrays, functions, parameters, object-oriented programming, classes structures, structure of Java program, abstract classes
2. Graphic user interface (GUI) in Java, communication types, AWT and SWING libraries, GUI principles, components, containers, layout managers, events handling
3. Polymorphisms abstract classes, interfaces, interface as a variable type, interface and inheritance, interface type
4. Event as an object, events handling, event source, listeners, the model of events spreading, events model implementation, events processing
5. Event definition handling, more then one event sources and listeners, events source distinguishing
6. Exceptions, the principle of exceptions processing, complete exception handling
7. Exception throwing, exception propagation, exception generating, exception hierarchy, checked and unchecked exceptions
8. Applets, properties, applications, activation of the applet, the life cycle of the applet, parameters passing to applet, applet restrictions
9. Libraries, class library, documentation use, utilization of collections, containers, lists, sets, examples
10. Basic programming in C language, compilation, language description, the model of compilation, program structure, structure of function, example of program
11. Comparative presentation of C language to Java language, macros, conditional translation, language syntax, struct, union, enum types
12. Systematic programming v C, statement semantic, preprocessor, basic libraries, I/O, input, output
13. Pointers, memory management, function and pointers, pointers and arrays
14. Reserve

Labs/Seminars:

1. Introductory test, revision and resume of programming basic in Java and object-oriented programming
2. Graphic user interface (GUI) in Java layout design
3. Graphic user interface (GUI) in Java interaction, assignment of semester task
4. Polymorphisms, abstract classes, interfaces
5. Events handling
6. Events generation processing
7. Exceptions handling
8. Applets I
9. Applets II
10. Libraries, class library, documentation use,

11. Introduction to development system for C language, analysis of programs in C languages, a simple C program debugging
12. A systematic programming in C language
13. Pointers, memory management, pointers and arrays
14. Reserve

Literature:

1. Zakhour, S: The Java Tutorial: A Short Course on the Basics, 4th Edition,Amazon, 2006
2. Hawlitzek, F: Java 2, Addison-Wesley, 2000
3. Eckel, B: Thinking in Java 2, Prentice Hall, 2000

Notes from Faculty Information System:

2.8 AE4B33ALG: Algorithms

Title: Algorithms
Lecturer: TBD
Term: summer
Lectures + seminars or labs: 2p+2c
ECTS credits: 6
FEL www: AE4B33ALG
Anotation: In the course, the algorithms development is constructed with minimum dependency to programming language; nevertheless the lectures and seminars are based on Java. Basic data types a data structures, basic algorithms, recursive functions, abstract data types, stack, queues, trees, searching, sorting, special application algorithms. Students are able to design and construct non-trivial algorithms and to evaluate their affectivity.

Syllabus:

1. Algorithms, programs, programming languages, introduction to problems solving
2. One-dimensional array, simple problems v 1-D array
3. Sorting in 1-D array pole (mergesort, quicksort, heapsort),
4. Searching in 1-D array
5. 2-D array, simple tasks in 2-D array
6. Strings, simple problems in string processing, text files
7. Asymptotic complexity, evaluation of space and complexity of algorithms from lectures No. 3.-6.
8. Simple recursion, recursive functions, advanced techniques
9. File conception, sequential files, conception of the record, file of records
10. Data types, list, stack, queue, examples of application
11. Linked lists, linearly-linked list, other types of linked list, trees
12. Trees, their properties, binary trees, basic algorithms of tree search.
13. Basic algorithms of linear algebra and mathematical analysis
14. Reserve

Labs/Seminars:

1. Introductory test, repeating of the ways of program construction in development environment, examples of functions and procedures, parameters, simple classes, assignment of semester task
2. One-dimensional array processing
3. Sorting and searching in 1D array algorithms
4. Multidimensional array processing algorithms
5. Text and string algorithms
6. Experimentation with space and complexity of algorithms
7. Sequential files
8. Implementation of abstract data types
9. Recursion and iteration
10. Linked lists, linearly-linked list
11. Tree construction, tree search
12. Test, consultation to semester task
13. Algorithms of linear algebra and geometry, mathematical analysis
14. Credit

Literature:

1. Sedgewick, R: Algorithms (Fundamentals, Data structures, Sorting, Searching), Addison Wesley, 2003
2. Weiss, M: Data structures and Algorithm Analysis in Java, Addison Wesley, 1999
3. Keogh, J: Data Structures Demystified, McGraw-Hill, 2004
4. Wróblevski, P: Algorytmy, Helion, 2003

Notes from Faculty Information System:

2.9 AE0B01PSI: Probability, Statistics, and Theory of Information

Title: Probability, Statistics, and Theory of Information
Lecturer: TBD
Term: winter
Lectures + seminars or labs: 4p+2s
ECTS credits: 6
FEL www: AE0B01PSI
Anotation: Basics of probability theory, mathematical statistics, information theory, a coding. Includes descriptions of probability, random variables and their distributions, characteristics and operations with random variables. Basics of mathematical statistics: Point and interval estimates, methods of parameters estimation and hypotheses testing, least squares method. Basic notions and results of the theory of Markov chains. Shannon entropy, mutual and conditional information, types of codes. Correspondence between entropy and the optimal code length. Information channels and their capacity, compression.

Syllabus:

1. Basic notions of probability theory. Random variables and their description.
2. Characteristics of random variables. Random vector, independence, conditional probability, Bayes formula.
3. Operations with random variables, mixture of random variables. Chebyshev inequality. Law of large numbers. Central limit theorem.
4. Basic notions of statistics. Sample mean, sample variance.
5. Method of moments, method of maximum likelihood. EM algorithm.
6. Interval estimates of mean and variance. Hypotheses testing.
7. Goodness-of-fit tests, tests of correlation, non-parametic tests.
8. Applications in decision-making under uncertainty and pattern recognition. Least squares method.
9. Discrete random processes. Stationary processes. Markov chains.
10. Classification of states of Markov chains. Overview of applications.
11. Shannon's entropy of a discrete distribution and its axiomatical formulation. Theorem on minimal and maximal entropy. Conditional entropy. Chain rule. Subadditivity. Entropy of a continuous variable.
12. Fano's inequality. Information of message Y in message X. Codes, prefix codes, nonsingular codes. Kraft-MacMillan's inequality.
13. Estimation of the average codelength by means of entropy. Huffman codes. Data compression using the law of large numbers. Typical messages. Entropy speed of stationary sources.
14. Information channel and its capacity. Basic types of information channels. Shannon's coding theorem. Universal compression. Ziv-Lempel codes.

Labs/Seminars:

1. Elementary probability. Random variables and their description.
2. Mean and variance of random variables. Unary operations with random variables.
3. Random vector, joint distribution.
4. Binary operations with random variables. Mixture of random variables. Central limit theorem.
5. Sample mean, sample variance. Method of moments, method of maximum likelihood.
6. Interval estimates of mean and variance.
7. Hypotheses testing.
8. Least squares method.
9. Goodness-of-fit tests.
10. Discrete random processes. Stationary processes. Markov chains.

11. Shannons's entropy of a discrete distribution and its axiomatical formulation. Theorem on minimal and maximal entropy. Conditional entropy. Chain rule. Subadditivity. Entropy of a continuous variable.
12. Fano's inequality. Information of message Y in message X. Codes, prefix codes, nonsingular codes. Kraft-MacMillan's inequality.
13. Estimation of the average codelength by means of entropy. Huffman codes. Data compression using the law of large numbers. Typical messages. Entropy speed of stationary sources.
14. Information channel and its capacity. Basic types of information channels. Shannon's coding theorem. Universal compression. Ziv-Lempel codes.

Literature:

1. David J.C. MacKay: Information Theory, Inference, and Learning Algorithms, Cambridge University Press, 2003.
2. T.M.Cover and J.Thomson: Elements of information theory, Wiley, 1991.
3. Papoulis, A.: Probability and Statistics, Prentice-Hall, 1990. Mood, A.M., Graybill, F.A., Boes, D.C.: Introduction to the Theory of Statistics. 3rd ed., McGraw-Hill, 1974.

2.10 AE4B01JAG: Languages, automata and grammars

Title: Languages, automata and grammars
Lecturer: TBD
Term: winter
Lectures + seminars or labs: 2p+2s
ECTS credits: 6
FEL www: AE4B01JAG
Anotation: The course covers basics of the theory of finite automata and grammars: deterministic and non-deterministic finite automata, characterization of the class of languages accepting by a finite automaton and description of such a language by a regular expression. Grammars and languages generated by a grammar, context-free grammars will be emphasized. The relation will be shown between context-free grammars and push down automata. Next topic is a Turing machine and the existence of non-decidable problems.

Syllabus:

1. Alphabet, strings over an alphabet, concatenation of words, language.
2. Deterministic finite automaton, state diagram.
3. Language accepted by a finite automaton, Nerode?s Theorem.
4. Nondeterministic finite automata.
5. Equivalence of deterministic and nondeterministic finite automata.
6. Regular expressions and regular languages, Kleen?s Theorem.
7. Properties of regular languages.
8. Grammars, regular grammars, context-free grammars.
9. Push down automata and their relation to context-free languages.
10. Properties of context-free languages, Pumping Lemma for context-free languages.
11. Algorithms for some problems concerning context-free languages.
12. Turing machines.
13. Non-decidable problems.

Labs/Seminars:

1. Alphabet, strings over an alphabet, concatenation of words, language.
2. Deterministic finite automaton, state diagram.
3. Language accepted by a finite automaton, Nerode?s Theorem.
4. Nondeterministic finite automata.
5. Equivalence of deterministic and nondeterministic finite automata.
6. Regular expressions and regular languages, Kleen?s Theorem.
7. Properties of regular languages.
8. Grammars, regular grammars, context-free grammars.
9. Push down automata and their relation to context-free languages.
10. Properties of context-free languages, Pumping Lemma for context-free languages.
11. Algorithms for some problems concerning context-free languages.
12. Turing machines.
13. Non-decidable problems.

Literature:

1. J.E. Hopcroft, R. Motwani, J. D. Ullman: Introduction to Automata Theory, Languages, and Computation, Second Edition, Addison-Wesley, 2001

2.11 AE0B35SPS: Computer Systems Structures

Title: Computer Systems Structures
Lecturer: Ing. Šusta Richard Ph.D.
Term: winter
Lectures + seminars or labs: 3p+2l
ECTS credits: 6
FEL www: AE0B35SPS
Anotation: The subject introduces into basic hardware structures of computer systems, into their design and architecture. It explains technical background of classic computer systems but also special computer for digital and logic control.

Syllabus:

1. Synthesis of combinational logic circuits. Hazards in logic circuits.
2. Introduction into HDL languages for design of circuits for computers
3. Minimization of logic functions. Combinational circuits used in computers - multiplexors, demultiplexors, decoders, comparators, adders. Their descriptions in HDL language.
4. Programmable logic circuits PLD, GAL, iPLSI, XILINX. Their descriptions in HDL language.
5. Event driven systems and finite automaton as its mathematical model. Design and minimization of synchronous and asynchronous automata.
6. Sequential logic systems. Synthesis of asynchronous sequential systems as combinational circuits with feedback. RS, JK a D circuits.
7. Synthesis of sequential logic circuits with clock and circuits used in computers: binary and decade counters, Gray counters, shift registers, interrupt controllers. Examples of HDL descriptions.
8. From automata to processors. Fix and programmable controller. Automaton with micro program. Microprocessor. Instruction cycles. Classic architecture of CPU, bus, memory. von Neumannova, Harvard and modified Harvard architecture.
9. Structure of CPU, data and address registers, counter of instructions, stack pointer, types of instructions, address modes in linear addres space.
10. Machine code of general processor. Basic instructions.
11. Structure and hierarchy of memory: Cache as an associative memory, operational memory, secondary memories (hard drives), fragmentation of memory. Reliability of memories.
12. Interrupts and exceptions. Sources of interrupts, external interrupts, interrupt vectors, interrupts from timers, interrupts generated by CPU and controllers of memory bus.
13. Different width of addresses generated by CPU and physical memory. Mapping of memory, paging, segmentation. Protection of memory, DMA transfers.
14. Differences of industrial programmable controlles (PLC) from classic computers: Structer of PLCs, their properties and methods of programming.

Labs/Seminars:

1. Introduction, safety rules in laboratory, organization.
2. Minimization of maps, demonstration of design in HDL language.
3. Design in HDL, part II.
4. Examples of HDL uses and programming of PLD circuits.
5. Independent work - design of counter.
6. Independent work - Code lock.
7. Written test.
8. Design of controllers and its description in HDL language.
9. Independent work - Simple automaton I.

10. Independent work - Simple automaton II.
11. Independent work - Small controller I.
12. Independent work - Small controller II.
13. Independent work - Small controller III.
14. Credits. Tests repetitions.

Literature:

1. Mano, M. Morris: Digital Design, 4/E, Prentice Hall 2007, ISBN-10: 0131989243
2. Sasao, Tsutomu: Switching Theory for Logic Synthesis, Springer 1999, 376 p., Hardcover, ISBN: 978-0-7923-8456-4
3. Hachtel, G. D., Somenzi, F., Logic Synthesis and Verification Algorithms, Kluwer Academic. 1996.
4. DeMicheli G., Synthesis and Optimization of Digital Circuits, McGraw-Hill, 1994.
5. P. Ashar, S.Devadas, and A.R. Newton, Sequential Logic Synthesis, Kluwer Academic Publishers, Boston, 1992, Chapters 3 - 5.

Notes from Faculty Information System:

2.12 AE4B02FYZ: Physics for Informatics

Title: Physics for Informatics
Lecturer: Doc.Dr.Ing. Bednařík Michal / Prof.Ing. Pekárek Stanislav CSc.
Term: summer
Lectures + seminars or labs: 2p+2l
ECTS credits: 6
FEL www: AE4B02FYZ
Anotation: Within the framework of this course students gain the knowledge of selected parts of classical physics and dynamics of the physical systems. The introductory part of the course deals with the mass particle kinematics; dynamics, with the system of mass particles and rigid bodies. The students should be able to solve basic problems dealing with the description of mechanical systems. The introduction to the dynamics of the systems will allow to the students deeper understanding as well as analysis of these systems. The attention will be devoted namely to the application of the mathematical apparatus to the solution of real physical problems. Apart of this, the knowledge gained in this course will help to the students in the study of other disciplines, which they will meet during their further studies.

Syllabus:

1. Units, system of units. Physical fields. Reference frames.
2. Particle kinematics (rectilinear motion, circular motion, motion in three dimensions.
3. Newton?s laws, inertial and non-inertial reference frames. Equations of motion in inertial and non-inertial reference frames.
4. Work, power, conservative fields, kinetic and potential energy. Conservation of mechanical energy law.
5. Newton´s law of universal gravitation, gravitational field of the system of n particles and extended bodies. Gravitational field intensity, potential.
6. Gravitational field outside and inside a spherical mass shell and homogeneous mass sphere.
7. Mechanical oscillating systems. Simple harmonic motion, damped and forced oscillations. Resonance of displacement and velocity.
8. System of n-particles, isolated and non-isolated systems, conservation of linear and angular momentum laws. Conservation of mechanical energy law for the system of n-particles. Center of mass and center of gravity.
9. Rigid bodies, equations of motion, rotation of the rigid body with respect to the fixed axis. Moment of inertia, Steiner?s theorem
10. Classification of dynamical systems (linear, nonlinear, autonomous, nonautonomous, conservative, continuous, discrete, one-dimensional, multidimensional, time-reversal, time-irreversal). Phase portraits, phase trajectory, fixed points, dynamical flow. Stability of linear systems.
11. Topological classification of linear systems (saddle points, stable and unstable spiral, stable and unstable node, center point).
12. Stability of nonlinear systems, Liapunov stability, limit cycles, bifurcation (Hopf, subcritical, supercritical, transcritical etc.), bifurcation diagram, Poincaré sections, attractors.
13. Deterministic chaos, Lorenz equations, strange attractor.
14. One-dimensional maps, Feigebaum numbers, the logistic equation, fractals.

Labs/Seminars:

1. Introduction, safety instructions, laboratory rules, list of experiments, theory of errors - measurement of the volume of solids.
2. Uncertainties of measurements.
3. 2nd Newton´s law and collisions.
4. Torsion pendulum, shear modulus and moment of inertia.

5. Measurement of the acceleration due to the gravity with a reversible pendulum and study of the gravitational field.
6. Young?s modulus of elasticity.
7. Forced oscillations - Pohl´s torsion pendulum.
8. Coupled pendulum.
9. Franck-Hertz experiment and measurement of excitation energy of the mercury atom.
10. Test.
11. Statistical distributions in physics. Poisson´s and Gauss´ distribution - demonstration using the radioactive decay.
12. Measurement of the speed of sound using sonar and acoustic Doppler effect. Diffraction of acoustic waves.
13. Test.
14. Grading of laboratory reports. Assessment.

Literature:

1. Halliday, D., Resnick, R., Walker, J.: Fyzika, VUTIUM-PROMETHEUS, 2000.
2. Kvasnica, J., Havránek, A., Lukáč, P., Sprášil, B.: Mechanika, ACADEMIA, 2004.
3. Sedlák, B., Štoll, I.: Elektřina a magnetismus, ACADEMIA, 2002.
4. Fyzika I a II - fyzikální praktikum, M. Bednařík, P. Koníček, O. Jiříček.
5. Physics I, S. Pekárek, M. Murla, Dept. of Physics FEE CTU, 1992.
6. Physics I - Seminars, M. Murla, S. Pekárek, Vydavatelství ČVUT, 1995.
7. Physics II, S. Pekárek, M. Murla, Vydavatelství ČVUT, 2003.
8. Physics II - Seminars, S. Pekárek, M. Murla, Vydavatelství ČVUT, 1996.
9. Physics I - II, Laboratory manual, S. Pekárek, M. Murla, Vydavatelství ČVUT, 2002.

2.13 AE0B36APO: Computer Architectures

Title: Computer Architectures
Lecturer: TBD
Term: summer
Lectures + seminars or labs: 2p+2l
ECTS credits: 6
FEL www: AE0B36APO
Anotation: Subject provides overview of basic building blocks of computer systems. Explanation starts from hardware side where it extends knowledge presented in the subject Structures of computer systems. Topics covers building blocks description, CPU structure, multiple processors interconnections, input/output subsystem and basic overview of network and busses topologies. Emphasis is placed on clarification of interconnection of hardware components with software support, mainly lower levels of operating systems, device drivers and virtualization techniques. General principles are more elaborated during presentation of examples of multiple standard PU architectures. Exercises are more focused on the software view to the contrary. Students are lead from basic multi threading programming to the interaction with raw hardware.

Syllabus:

1. Architecture, structure and organization of computers and its subsystems. Data and numbers representation and storage in computer systems (signed integer numbers, IEEE-754).
2. CPU concept and techniques development. RISC (load-store) and CISC processors comparison. Pipelining, jump prediction and super-scalar CPU.
3. Processors networks, parallel systems and interconnection networks. Topologies, communication. Symmetric multiprocessing, NUMA architectures.
4. Hierarchical concept of memory subsystem, memory management, data consistency, from MSI till MOESI protocols, control instructions and atomic operations.
5. I/O computer subsystem, I/O control. Peripherals, DMA, data consistency considerations for DMA operations.
6. Computer system, Technical and organization means. External events processing (interrupts), exceptions, real time clocks.
7. Computer networks, network topologies, communication. LAN, MAN, WAN and control area networks.
8. Parameters passing for subroutines and operating system implemented virtual instructions. Stack frames, register windows, privilege modes switching and system calls implementation.
9. Multi-level computer organization, virtual machines. Conventional (ISA) architecture and implementation dependant microarchitecture. Portable bytecode and virtual programming environments (Java, C#/.Net). Virtualization techniques (i.e. XEN, VMWARE) and paravirtualization.
10. Classic register memory-oriented CISC architecture. Principles demonstrated on FreeScale M68xxx/ColdFire architecture. MMU implementation, cache, busses etc.
11. INTEL x86 processor family from 8086 to EMT64, main focus on 32-bit and 64-bit operating modes supplemented with compatibility dictated 16-bit 8086 mode and 80286 segmented approach and why it is used minimally by todays OSes. SIMD instruction examples (MMX, SSE).
12. Short overview of RISC architectures and CPUs optimized for embedded applications - ARM, ColdFire and PowerPC.
13. Common system and I/O buses used in computer systems (ISA, PCI, PCIexpress, USB, SCSI, SATA, VME,.). Main focus paid to replacement of parallel busses by multilane serial busses. Advantages and disadvantages of this approach for RT control.
14. Analog and digital I/O interfacing, data acquisition and processing system.

Labs/Seminars:

1. Introduction, labs program, safety, real time (RT) control

2. Basic introduction to Linux operating system environment Task 1 - Practice of algorithm implementation in C language
3. Task 2 - Processes and signals
4. Task 3 - Threads
5. Task 4 - Networking and sockets
6. Task 5 - Serial communication
7. Reserve for task 1 - 5 completion
8. Description of the main task - position and revolution control of DC motor. Input IRC, output PWM, visualization
9. Test. 3 theoretical tasks from seminaries topics and practical part
10. Independent solving of main task
11. Independent solving of main task
12. Independent solving of main task
13. Main task hand in and presentation
14. Assessment

Literature:

1. Hennessy, J. L., and D. A. Patterson. Computer Architecture: A Quantitative Approach, 3rd ed. San Mateo, CA: Morgan Kaufman, 2002. ISBN: 1558605967.
2. Patterson, D. A., and J. L. Hennessy. Computer Organization and Design: The Hardware/Software Interface, 3rd ed. San Mateo, CA: Morgan Kaufman, 2004. ISBN: 1558606041.
3. Andrew S. Tanenbaum: Structured Computer Organization. Printice Hall, 2006. ISBN-10:0131485210.
4. Andrew S. Tanenbaum: Computer Networks. Prentice Hall 2003. ISBN-10:0-13-066102-3.
5. Andrew S. Tanenbaum: Modern Operating Systems. Prentice Hall 2001
6. Hyde, R.: The Art of Assembly Language, 2003, 928 pp. ISBN-10 1-886411-97-2 ISBN-13 978-1-886411-97 http://webster.cs.ucr.edu/AoA/
7. Bach., M., J.: The Design of the UNIX Operating System, Prentice Hall, 1986
8. Bayko., J.: Great Microprocessors of the Past and Present http://www.cpushack.com/CPU/cpu.html

Notes from Faculty Information System: Computer architectures

2.14 AE4B33OPT: Optimization

Title: Optimization
Lecturer: Ing. Werner Tomáš Ph.D.
Term: winter
Lectures + seminars or labs: 4p+2c
ECTS credits: 6
FEL www: AE4B33OPT
Anotation: The topic of the subject is the optimization of functions of continuous variables without and with constraints. Considerable space is devoted to convex optimization and linear programming (LP). The student will understand the scope, generality and usefulness of the discipline of optimization and receive theoretical background and practical skills to formulate optimization problems, estimate their level of difficulty, and propose ways of solution.

Syllabus:

1. Taxonomy of optimization problems. What belongs and what does not belong to the subject. Motivating examples. Local and global extremes.
2. Algorithms of unconstrained local optimization: bracketing an extreme, gradient optimization, coordinate optimization, Newton's method.
3. Algorithms of constrained local optimization: Lagrange multipliers, projected gradient methods.
4. Convex sets and functions.
5. LP1: convex polyhedra and their geometry
6. LP2: simplex method
7. LP3: Farkas lemma, LP duality
8. Application of LP (linear regression in L1 metric etc.)
9. Quadratic programming. Perceptron and Kozinec algorithms.
10. More general tasks of convex optimization: semidefinite programming (SDP), geometric programming. Examples, applications.
11. Lagrange duality.
12. Conjugated gradients.
13. Optimization of nonsmooth functions: subgradients, Shor's theorem.
14. Reserve.

Labs/Seminars:

1. The labs consist of solving practically motivated problems, which can hardly or not at all be tackled without using the knowledge acquired at the lectures. This does not require lengthy coding but will be mathematically nontrivial – therefore preparation at home is necessary for each lab lesson. The MATLAB programming language is used.

Literature:

1. Selected parts of the book "Boyd and Vanderberghe: Convex Optimization" (freely available on www).

Notes from Faculty Information System:

2.15 AE4B99BAP: Bachelor Project

Title: Bachelor Project
Lecturer: TBD
Term: summer
Lectures + seminars or labs: TBD
ECTS credits: 20
FEL www: AE4B99BAP
Anotation: TBD

2.16 AE4B33OSS: Operating systems and networks

Title: Operating systems and networks
Lecturer: Doc.Ing. Lažanský Jiří CSc.
Term: winter
Lectures + seminars or labs: 2p+2c
ECTS credits: 6
FEL www: AE4B33OSS
Anotation: The goal of this course is to introduce basic concepts and principles of operating systems (OS), like processes and threads, their scheduling, mutual communication and synchronization, time-dependent errors and deadlocks. Attention is also paid to memory management, virtual memory, management of secondary storages, file-systems and data security. The second part of the course is focused at distributed systems (DS) principles and technologies. DS communication media and topologies are explained and the basics of Internet including specific protocols are treated as typical DS applications.

Syllabus:

1. Basic concepts: operating system (OS) and its kernel, hardware dependency, processes, threads, scheduling. OS types.
2. Compiling and linking, libraries, OS services and calls, dynamic libraries and run-time linking
3. Implementation and life-cycle of processes and threads. Scheduling algorithms. Real-time OSes and scheduling in them.
4. Communication among processes and threads, critical sections, time-dependent errors, synchonization tools. Classical synchronization problems and their solution.
5. Deadlocks: definition, conditions, solution possibilities
6. Memory management: Basic techniques. Virtual memory, swapping, paging, segmentation, page-replacement algorithms.
7. File-systems, organizing data on the secondary storage, principles, standard solutions, data security.
8. Distributed computing, client-server architectures. OS components supporting computer networks (sockets) and their programming.
9. Introduction to computer networks, basic definitions, ISO/OSI model and its layers functionality
10. Local-area networks, active componets, physical addressing
11. Interconnecting networks, internetworking, addressing, routing principles
12. Protocols and technologies in the Internet. Protocols for network management
13. DS applications (deeper insight into some common TCP/IP protocols)
14. Wrap-up, spare time.

Labs/Seminars:

1. Introdution, exercise organization, 1st task assignment
2. Brief OS overview, introduction to scripting languages
3. Advanced script programming, regular expressions
4. Exercising the scripting languages, 1st task delivery
5. 2nd task assignment. Processes and threads, resource sharing - practical examples
6. Interprocess communication, deadlocks
7. Client-server programming
8. 2nd task delivery and results presentation
9. 3rd task assignment, network configurations
10. Tools for network management
11. Firewalls, configuration principles

12. Individual work in a computer room
13. Presentation of 3rd task results
14. Conclusion, QandA, spare time

Literature:

1. Silberschatz A., Galvin P. B., Gagne G.: Operating System Concepts. J. Willey, 2005
2. Tanenbaum A. S.: Modern Operating Systems. Prenice Hall, 2001
3. Halsall, F.: Data Communications, Computer Networks and Open Systems, Adison Wesley 1996
4. Comer D. E.: Internetworking with TCP/IP: Principles, Protocols and Architectures. Prentice Hall, 2005

Notes from Faculty Information System:

2.17 AE4B01NUM: Numerical Analysis

Title: Numerical Analysis
Lecturer: TBD
Term: winter
Lectures + seminars or labs: 2p+2c
ECTS credits: 6
FEL www: AE4B01NUM
Anotation: The course introduces to basic numerical methods of interpolation and approximation of functions, numerical differentiation and integration, solution of transcendent and ordinary differential equations and systems of linear equations. Emphasis is put on estimation of errors, practical skills with the methods and demonstration of their properties using Maple and computer graphics.

Syllabus:

1. Overview of the subject of Numerical Analysis
2. Approximation of functions, polynomial interpolation
3. Errors of polynomial interpolation and their estimation
4. Hermite interpolating polynomial. Splines
5. Least squares approximation
6. Basic root-finding methods
7. Iteration method, fixed point theorem
8. Basic theorem of algebra, root separation and finding roots of polynomials
9. Solution of systems of linear equations
10. Numerical differentiation
11. Numerical integration (quadrature); error estimates and stepsize control
12. Gaussian and Romberg integration
13. One-step methods of solution of ODE's
14. Multistep methods of solution of ODE's

Labs/Seminars:

1. Instruction on work in laboratory and Maple
2. Individual work - training in Maple
3. Polynomial interpolation, estimation of errors
4. Individual work on assessment tasks
5. Least squares approximation
6. Individual work on assessment tasks
7. Root-finding methods, root separation
8. Individual work on assessment tasks
9. Solution of systems of linear equations
10. Numerical differentiation
11. Numerical differentiation and integration, modification of tasks
12. Individual work on assessment tasks
13. Solution of ODE's
14. Individual work on assessment tasks; assessment

Literature:

1. Press, W. H., Flannery, B. P., Teukolsky, S. A., Vetterling, W. T.: Numerical Recipes (The Art of Scientific Computing), Cambridge University Press, Cambridge, 1990.
2. Knuth, D. E., The Art of Computer Programming, Addison Wesley, Boston, 1997.

2.18 AE4B33DS: Database Systems

Title: Database Systems
Lecturer: Doc.Ing. Kouba Zdeněk CSc.
Term: summer
Lectures + seminars or labs: 2p+2c
ECTS credits: 6
FEL www: AE4B33DS
Anotation: Database Systems, Web Applications

Syllabus:

1. Basic information system architectures (client-server, multi-tier, thin client), analysis of information flows, UML use-case diagrams
2. Basic data modeling, E-R diagrams, relational model, database access API, JDBC
3. Integrity constraints, referential integrity, normal forms
4. Querying in relational databases, SQL basics, referential integrity in SQL
5. Advanced queries, aggregation functions, nested queries
6. Cursor, view, stored procedures, triggers
7. Transactions, their serializability, locking, isolation levels, transaction deadlock, its prevention and resolution
8. UML class diagrams, sequence diagrams
9. Object-relational mapping, object persistence and access API (JPA)
10. Design of multithreaded applications and guidelines for their implementation, UML activity diagrams
11. Basic design patterns
12. Overview of component architectures (CORBA, COM, EJB) and communication protocols (RPC, RMI, IIORB, JMS, http, web services)
13. Design of a distributed system with component architecture, web-based interface
14. Enterprise applications and major design problems (load balancing, data replication)

Labs/Seminars:

1. Organization of labs, safety rules, making up working groups
2. Basics of relational modeling
3. Conceptual model creation
4. Application interface of a database, connection methods
5. Logical data model creation
6. Interactive query composition
7. Practical examples of transactional processing significance
8. Realization of selected data model in the relational database environment
9. Object-relational mapping
10. Autonomous work
11. Autonomous work
12. Autonomous work
13. Submission and presentation of the working group results
14. Submission and presentation of the working group results, credits

Literature:

1. Kroha, P.: Objects and Databases. McGraw-Hill Book Company, London, 1993

Notes from Faculty Information System:

2.19 AE4B33FLP: Functional and Logic Programming

Title: Functional and Logic Programming
Lecturer: Ing. Železný Filip Ph.D.
Term: summer
Lectures + seminars or labs: 2p+2c
ECTS credits: 6
FEL www: AE4B33FLP
Anotation: This course introduces students into the techniques of functional programming in the LISP language and logic programming in the PROLOG language. Both languages are declarative in that the programmer symbolically describes the problem to be solved, rather than enumerating the exact sequence of actions to be taken. In PROLOG, one describes the problem by specifying properties of objects and relations thereamong through logic formulas. In LISP, the problem description takes the form of function definitions. Both languages have found significant applications in artificial intelligence fields, such as agent systems or symbolic machine learning. Motivating tasks from these domains will be used throughout the course.

Syllabus:

1. Prolog: facts, rules and queries. Recursion. Query answering.
2. Functions, unification, list operations.
3. Prolog and logic: clauses, Herbrand base, interpretation, model, closed-world assumption, decidability.
4. Cut and negation. Extralogical operators, arithmetics.
5. Combinatorial search in Prolog.
6. Constraint logic programming.
7. Programming practices, debugging, built-in predicates.

Labs/Seminars:

1. Prolog as a database. Facts, rules, queries.
2. Recursion. Program debugging.
3. Unificaton. List operations.
4. List, cut and negation operations.
5. Search algorithms, individual task assignment
6. Search algorithms
7. Constraint logic programming
8. Constraint logic programming
9. Credits

Literature:

1. I. Bratko: Prolog programing for AI, Addison Wesley 2001 (3rd edition)
2. P. Flach: Simply Logical, John Wiley 1994
3. V. Mařík et al: Umělá inteligence I, II, Academia 1993 (In Czech)
4. P. Jirků, P. Štěpánek, O. Štěpánková: Programování v Jazyku Prolog, SNTL 1991 (In Czech)

Notes from Faculty Information System:

2.20 AE4B33ZUI: Introduction to Artificial Intelligence

Title: Introduction to Artificial Intelligence
Lecturer: Prof.Dr. Pěchouček Michal MSc.
Term: summer
Lectures + seminars or labs: 2p+2c
ECTS credits: 6
FEL www: AE4B33ZUI
Anotation: This course provides introduction to symbolic artificial intelligence. It presents the algorithms for informed and non-informed state space search, nontraditional methods of problem solving, knowledge representation by means of formal logic, methods of automated reasoning and introduction to markovian decision making.

Syllabus:

1. Introduction to artificial inteligence
2. Problem solving using state space search
3. Non-informed state space search
4. Informed state space search - A* algorithm
5. Nontraditional state space search methods
6. Knowledge representation and rule-based systems reasoning
7. Introduction to two-player games
8. Logics and knowledge representation
9. Reasoning in first-order predicate logic
10. Resolution. Theorem proving.
11. Resolution, resolution strategies, theorem proving tools.
12. Introduction to uncertainty in knowledge representation. Markov models.
13. Markov chains.
14. Back-up class.

Labs/Seminars:

1. Review of mathematical logic.
2. Knowledge representation in first-order logic I.
3. Knowledge representation in first-order logic II.
4. Resolution principle I.
5. Resolution principle II.
6. Non-informed state space search.
7. Informed state space search.
8. A* algorithm.
9. -12. Implementation of state space search problem in Lisp programming language.

Literature:

1. Stuart Russell and Peter Norvig: Artificial Intelligence: A Modern Approach, Prentice Hall, Second Edition, 2003

Notes from Faculty Information System:

2.21 AE4B33RPZ: Pattern Recognition and Machine Learning

Title: Pattern Recognition and Machine Learning
Lecturer: Doc.Dr.Ing. Matas Jiří
Term: winter
Lectures + seminars or labs: 2p+2c
ECTS credits: 6
FEL www: AE4B33RPZ
Anotation: The basic formulations of the statistical decision problem are presented. The necessary knowledge about the (statistical) relationship between observations and classes of objects is acquired by learning on the raining set. The course covers both well-established and advanced classifier learning methods, as Perceptron, AdaBoost, Support Vector Machines, and Neural Nets.

Syllabus:

1. The pattern recognition problem. Overview of the Course. Basic notions.
2. The Bayesian decision-making problem, i.e. minimization of expected loss.
3. Non-bayesian decision problems.
4. Parameter estimation. The maximum likelihood method.
5. The nearest neighbour classifier.
6. Linear classifiers. Perceptron learning.
7. The Adaboost method.
8. Learning as a quadratic optimization problem. SVM classifiers.
9. Feed-forward neural nets. The backpropagation algorithm.
10. Decision trees.
11. Logistic regression.
12. The EM (Expectation Maximization) algorithm.
13. Sequential decision-making (Wald´s sequential test).
14. Recap.

Labs/Seminars:

1. Students solve four or five pattern recognition problems, for instance a simplified version of OCR (optical character recognition), face detection or spam detection using either classical methods or trained classifiers.
2. Introduction to MATLAB and the STPR toolbox, a simple recognition experiment
3. The Bayes recognition problem
4. Non-bayesian problems I: the Neyman-Pearson problem.
5. Non-bayesian problems II: The minimax problem.
6. Maximum likelihood estimates.
7. Non-parametric estimates, Parzen windows.
8. Linear classifiers, the perceptron algorithm
9. Adaboost
10. Support Vector Machines I
11. Support Vector Machines II
12. EM algoritmus I
13. EM algoritmus II
14. Submission of reports. Discussion of results.
15. Submission of reports. Discussion of results.

Literature:

1. Duda, Hart, Stork: Pattern Classification, 2001.
2. Bishop: Pattern Recognition and Machine Learning, 2006.
3. Schlesinger, Hlavac: Ten Lectures on Statistical and Structural Pattern Recognition, 2002.

Notes from Faculty Information System:

3 Facultative courses

4 Index

Index