1991/92

Cognitive Psychology

Recommended :

Graduate students	Lectures	30 h
	Seminars	16h

The course last ran:
This is a new course.

Goals:
To give an overview of the field of cgnitive psychology. To relate research in cognitive psychology to research in computer science.

Prerequisites:
None (though see below)

Organization:
The undergraduate course is used as a backbone. These lectures present the basic material in the field for students with no previous knowledge in the area. This part gives three credit points. Students that have taken this course previously can get credits for this part too, but the details of this will be decided upon on an individual basis.
The seminars will be for graduate students only.The aim of these is twofold; to further develop some of the topics in the course, and to relate the contents to other areas in cognitive science. Active participation in the seminar series will give two credit points, and another additional one will be given to those students that write a short paper on some aspect of the field.

Contents:
Perception, attention, memory, thinking, problem-solving, etc.

Literature:
Solso, R. *Cognitive Psychology* (or probably some other similar textbook)
Articles.

Teachers:
Nils Dahlbäck

E*xaminer:*
Nils Dahlbäck

Schedule:
April-june 1992

Examination:
Written exam, seminar presentation, and paper

Credit:
3+2+1

Compiling functional languages

Recommended for:
For computer science and systems students

Lectures:
36 h

The course last ran:
The course has not been given before, but a course given by Gary Lindström march 1990 contained approximately 10% of this material.

Goals:
The course gives a basic understanding of methods for implementing lazy functional languages.

Prerequisites:
Some knowledge of ML and denotational semantics, e.g. from Pteori I or the Compiler Construction - advanced course.

Organization:
Mainly lectures. Possibly a few seminars for student presentations.

Contents:
Part I: The lambda calculus, translation of a functional language into lambda calculus, types and pattern-matching, compilation of pattern-matching, transforming enriched lambda calculus, list comprehensions, polymorphic type-checking.
Part II: Graph reduction. Program representation, selecting redex, supercombinators, lambda-lifting, fully-lazy lambda-lifting, sk combinators, storage management and garbage collection.
Part III: Advanced graph reduction. The G-machine, G-code, optimizations to the G-machine, optimizing tail calls, strictness analysis, parallel graph reduction.

Literature:
Simon L. Peyton Jones: *The Implementation of Functional Programming Languages*, Prentice-Hall 1987.
Articles.

Teachers:
Peter Fritzson, + invited lecturer (Thomas Johnsson or Lennart Augustsson)

Examiner:
Peter Fritzson

Schedule:
Mid-September 1991. Four hours/week

Examination:
Written examination and small implementation projects
Seminar presentation gives extra point

Credit:
5 p(+2 p)

1991/92

Computer-Supported Cooperative Work

Recommended for:
Graduate students interested in computer applications

Lectures: 20 h

The course last ran:
This is a new course

Goals:
Communicate Basic concepts within CSCW.

Prerequisites:
None

Organization:
Problem-based seminars and a project.

Contents:
Group processes, computers in work life, shared computer environments, group interfaces, distributed system.

Literature:
Since the course is basic and problem-oriented, an extensive literature list will be distributed at the course start.

Teachers:
Toomas Timpka and James Nyce

Examiner:
Toomas Timpka

Schedule:
March - June, 1992

Examination:
Term paper and attendance at seminars

Credit:
2 (+some points)

1991/92

Concurrent Programming and Concurrent Logic Programming

Recommended for:
Graduate students

Lectures: 24 hours

The course last ran:
This is a new course.

Goals:
To provide an introduction to various Concurrent Programming paradigms and show how Concurrent Programming might be done in a Prolog.

Prerequisites:
Knowledge of C and Prolog

Organization:
The course will be given as a series of lectures

Contents:
- “Conventional” concurrent programming - co-routines, semaphores, monitors and message-passing. I intend using the uSystem from University of Waterloo for this.
- Occam
- Early multiprocessor implementations of Prolog
- Committed Choice Logic Programming Languages
- A message-passing Prolog, PMS-Prolog, which shows the crossover between the conventional (message-passing) concurrent programming paradigm and Prolog

Literature:
To be decided later.

Teacher:
Michael Wise

Examiner:
Jan Maluszynski

Schedule:
Spring 1992

Examination:
Three programming assignments and an exam

Credit:
3 points

1991/92

Constraint Logic Programming

Recommended for:
Computer Science and Systems students

Lectures: 24 h

The course last ran:
This is a new course

Goals:
1. To give a survey of systems for solving constraint satisfaction problems, in particular constraint logic programming systems. Systems that will be discussed are **CLP(X), CHIP,** (possibly Prolog III) ...
2. To clarify the declarative and operational principles underlying such systems
3. To discuss their applications to AI and to programming.

Prerequisites:
Some knowledge about modeltheoretic and operational semantics of Horn clause logic considered as a programming language.

Organization:
The course will be given as a number of lectures, possibly with student presentations.

Contents:
Systems: CLP(X), CHIP, (Possibly Prolog III) ...
Declarative principles: Model theoretic and fixed point semantics of CLP(X), satisfaction complete and solution compact domains, negated constraints ...
Operational principles: Extensions of SLD-resolution, extended unification algorithms, forward checking, (possibly adaptions of the simplex algorithm for solving constraints over the reals) ..

Literature:
P. van Hentenryck: **Constraint Satisfaction in Logic Programming**, MIT Press 1989.
A number of articles.

Teachers:
Jan Maluszynski, Staffan Bonnier (plus invited lecturers)

Examiner:
Jan Maluszynski

Schedule:
Spring 1992.

Examination:
Exercises given as homework

Credit:
3 points

1991/92

Datasäkerhet

Recommended for:
I första hand för Ekonomiska Informationssystems särskilda lic utbildning.

The course last ran:
Ny kurs

Goals:
Att ge fördjupad insikt i såväl tekniska som organisatoriska datasäkerhetsfrågor.

Literature:
Carrol, J, Computer Security. 2:a uppl. Butterworth, 1987.
Ytterligare litteratur tillkommer

Teachers:
Viiveke Fåk m fl

Examiner:
Birger Rapp

Schedule:
Veckorna 36 och 37

Examination:
Tentamen och skriftlig rapport.

Credit:
3 - 5 poäng

För ytterligare information hänvisas till Eva Elfinger tel 1524.

1991/92

Empirical Research Methods II

Recommended for:
Graduate students.

Lectures: 20 h

The course last ran:
New course.

Goals:
Prefare the student to a level which enables publication of an empirical study in an international journal.

Prerequisites:
ERM I.

Organization:
Seminars and project.

Contents:
Advanced topics within quantitative and qualitative research methods.

Literature:
Handouts will be distributed during the course.

Teachers:
Toomas Timpka and James Nyce.

Examiner:
Toomas Timpka.

Schedule:
92/93.

Examination:
Attendance at seminars and term paper.

Credit:

Hypermedia III

Recommended for:
Graduate students.

Lectures: 20 h

The course last ran:
New course.

Goals:
Scrutirize advanced Topics within the field of hypermedia.

Prerequisites:
Hypermedia II (Hypermedia I with complementations).

Organization:
Problem-based seminars and project.

Contents:
Advanced topics within hypermedia. The course is mainly oriented towards applications.

Literature:
Since the course is advanced and problem-oriented, no literature list is provided.

Teachers:
Toomas Timpka and James Nyce.

Examiner:
Toomas Timpka.

Schedule:
92/93.

Examination:
Attendance at seminars and term paper.

Credit:
2(+6)p

1991/92

Doktrinhistoria

Recommended for:
I första hand för Ekonomiska Informationssystems särskilda lic utbildning.

The course last ran:
Ny kurs

Goals:
Att ge fördjupad insikt i både den företagsekonomiska utvecklingen i Sverige och utvecklingen av betydelsefulla nationalekonomiska teorier.

Literature:
Dickson, H, Nationalekonomin genom tiderna. Esselte Studium 1983.
Engwall, L, m fl, Mercury meets Minerva. Business Administration in Academia: The Swedish Case. Bokmanus, Uppsala, 1990.
Staley, C, A History of Economic Thought. From Aristotle to Arrow. Basil Blackwell, 1989.
Kompendiematerial.

Teachers:
Björn Elsässer, Lars Engwall

Examiner:
Birger Rapp

Schedule:
-v 36

Examination:
Skriftlig rapport och muntlig tentamen

Credit:
3 poäng

För ytterligare information kontakta Eva Elfinger tel 28 15 24.

Human-Computer Interaction

Recommended: *Lectures:* 20 h.
For graduate students.

The course last ran:
In 87/88, when it was given by Sture Hägglund, Lars Ahrenberg and Arja Vainio-Larsson. This course is not the same, but it covers a lot of the same ideas albeit in a different way.

Goals:
To gain some practical and theoretical knowledge of user interface development, based on an understanding of the process of human-computer interaction.

Prerequisites:
Undergraduate level computer science and software development courses.

Organization:
The lectures and seminars are intended to give a theoretical background for the individual and group assignments which form the bulk of the course.

Contents:
Background: human-computer interaction.
The three central phases in user interface development: *design*, *implementation* and *evaluation*.

Literature:
To be decided.

Teachers:
Jonas Löwgren.

Examiner:
Jonas Löwgren.

Schedule:
Spring or Fall 92.

Examination:
Individual and group assignments during the course.

Credit:
3 points.

Information Retrieval I: Traditional and newer approaches

Recommended for:
Graduate Students

Lectures: 24 h

The course last ran:
This is a new course

Goals:
Provide an overview of traditional issues and solutions in information retrieval and some of the current approaches in selected areas.

Prerequisites:
None

Organization:
Reading assignment will be specified or distributed one week before each meeting. Students will take turns in presenting the contents of the reading assignment and in leading the discussions of issues arising from it.

Contents:
In Information Retrieval (IR) some of the main concerns are:
- the design of databases from a descriptive and a technical point of view,
- the design and use of vocabularies for description as well as query formulation,
- the translation of “documents” and information requests to matchable items,
- methods of matching, and file organizations,
- interfaces and interaction in searching,
- methods for evaluation,

Literature:
To be decided later

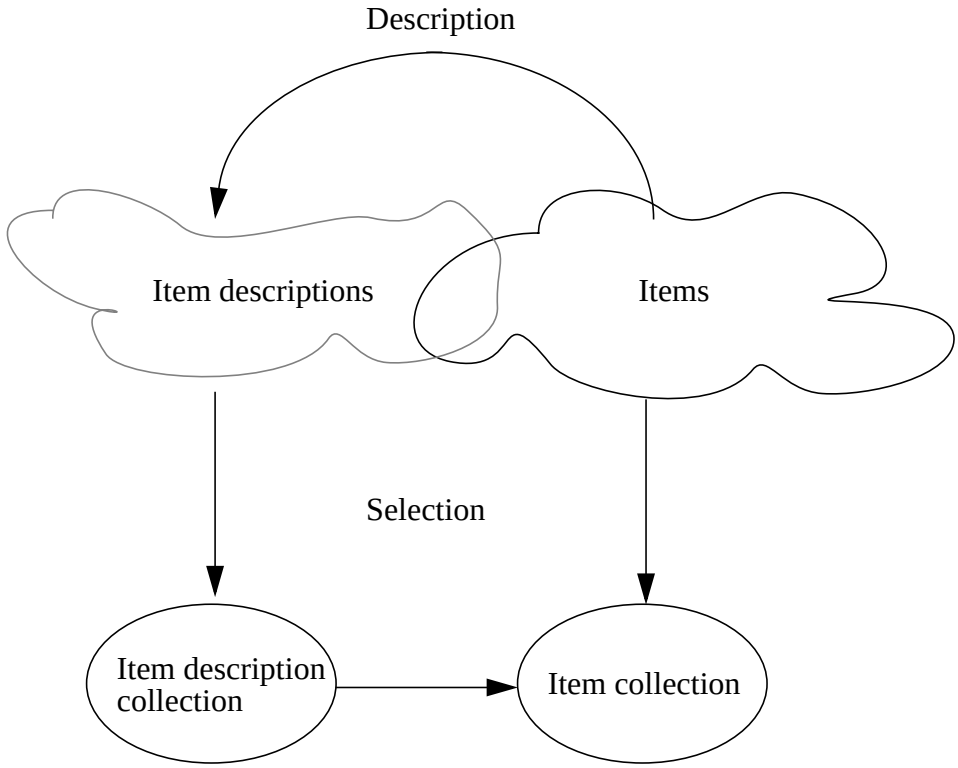
Teachers:
Roland Hjerppe

Examiner:
Roland Hjerppe

Schedule:
Fall 1991

Examination:
A written report on a chosen topic

Credit:
3 points


Description
Item descriptions
Items
Selection
Item description collection
Item collection

1991/92

Intelligent Autonomous Systems Architectures

Recommended for:
Computer Science and Systems students

Lectures: 20h

The course last run:
New course.

Goals:
To give a survey of previous and current approaches to Intelligent Autonomous System design. In particular, I am going to discuss the “classical” paradigm: SENSE-THINK-ACT, with its variations, and contrast it to the “emergent functionality” paradigm originating from Brooks’s layered subsumption architecture.

Prerequisites:
Introduction to AI (TDDA 58)
Knowledge Representation (TDDA 16)
Automata Theory (at basic level)

Organization:
A series of several lectures presenting general ideas, followed by seminars for discussing more detailed topics.

Contents:
1. SENSE-THINK-ACT paradigm;
 - description of several “flagships” of this approach;
 - analysis of complexities involved;
 - possible solutions.
2. EMERGENT FUNCTIONALITY paradigm;
 - description of several systems;
 - methods for combining primitive behaviors into complex ones.

Although the course will focus on autonomous system architectures, it will also discuss architectures used in other AI application areas, and compare them with requirements posed by ‘autonomousness’.

Literature:
A series of articles will be handed out to the course participants.

Teacher:
Jacek Malec

Examiner:
Jacek Malec

Schedule:
November - December 1991

Examination:
Term report.

Credit:
3 points

Introduction to research methodology in computer science

Recommended for:
New graduate students

Lectures:16 h

Last offering:
This course is a slightly modified version of a subset of the course given in 90/91.

Goals:
To give an introduction to the philosophy of science, the special characteristics of computer science research and to discuss practical aspects of graduate studies and scientific activities.

Prerequisites:
None

Organization:
Lectures and seminars.

Contents:
Introduction to the philosophy of sience. Scientific writing, publication and information retrieval. Science and technology, methodological issues. Survival strategies for young scientists. Project management. Research paradigms in computer science.

Literature:
Chalmers: *What is this thing called science*?
Sindermann: *Survival strategies for young scientists.*
Lecture Notes.

Teacher:
Sture Hägglund, et al.

Examiner:
Sture Hägglund

Schedule:
September-October, 1991.

Examination:
Written examination and seminar activity.

Credit:
2 points

Kunskapsutveckling om informationssystem

Recommended for: *Lectures:* 24 h
I första hand för Ekonomiska Informationssystem, ss systemutveckling

The course last ran:
Detta är en ny kurs.

Goals:
Ge översikt över kunskapsteoretiska och forskningsmetodologiska grunder för empiriska studier av informationssystem/systemutveckling, samt öka förmågan att genomföra sådana studier.

Prerequisites:
Inga absoluta krav. Önskvärt med grundläggande kurs i systemutveckling/informationssystem.

Organization:
Delvis samläsning med kursen Samhällsvetenskaplig metodik, SVL3.

Contents:
Kunskapsteoretiska och forskningsmetodologiska grunder för studier kring informationssystem. Kunskapsprojektering (planering av forskningsarbete). Genomförande av en mindre empirisk undersökning med särskild analys av forskningsmetodik och kunskapskaraktärisering.

Literature:
Ännu ej bestämd

Teachers:
Göran Goldkuhl

Examiner:
Göran Goldkuhl

Schedule:
September - november -91.

Examination:
Skriftlig rapport. Eventuellt tentamen.

Credit:
3 poäng

Logic for Ida-ites

Recommended for: *Lectures:* *20 h*
Computer Science and Systems students and Information Science Systems students

The course last ran:
This is a new course.

Goals:
To provide an introductory logic course to those graduate students, IDA researchers or teachers that have not had the time or opportunity to take a course in basic logic.The purpose of the course is to provide a sufficient basis for the participant to continue studies in logic on his or her own and to get more out of IDA seminars etc, where a working knowledge of logic is assumed.

Prerequisites:
The ability to reason,along with a knowledge of mathematics at a level equivalent to “naturvetenskapliga linjen” at the “gymnasium” level.

Organization:
10 x 2 hours lectures + additional problem solving sessions.

Contents:
- Compiling natural language sentences into well-formed formulas.
- The notions of deduction, derivability, proof , entailment etc.
- Proof theory, model theory.
- Model generation, resolution, natural deduction.
- Modal and temporal logic.
- Perhaps more.

Literature:
Logic for Information Technology, Galton.

Teacher:
Patrick Doherty

Examiner:
Patrick Doherty

Schedule:
Late Spring 1992.

Examination:
Written exam and/or problem solutions

Credit:
3 points

1991/92

Metodik för Systemutveckling

Recommended for: *Lectures:* Ett antal seminarier
I första hand för Ekonomiska informationssystem, ss systemutveckling.

The course last ran:
Detta är en ny kurs.

Goals:
Ge ökad kunskap om olika systemutvecklingsmetoder; öka förmågan att analysera och jämföra olika metoder.

Prerequisites:
Inga absoluta krav. Önskvärt med grundläggande kurs i systemutveckling/informationssystem.

Organization:
Seminarieserie under hela läsåret 91/92.

Contents:
Doktorandseminarier med analys av
- olika metoder,
- olika aspekter av metodanvändning,
- komparativ analys av metoder,
- kriterier för bedömning av metoder,
- hur empiriskt studera metoder,
- metodteoretiska grunder, m fl tänkbara aspekter

Literature:
Ännu ej bestämd.

Teachers:
Göran Goldkuhl

Examiner:
Göran Goldkuhl

Schedule:
Utspridd under hela läsåret

Examination:
Aktivt deltagande på seminarier samt skriftlig rapport.

Credit:
2-5 poäng

Non-Monotonic Reasoning

Recommended for:
Computer Science and Systems students

Lectures: 24 hours
Practical exercises: 6 hours

The course last run:
Partial overlap with “Non-Monotonic Logics” given by Michael Reinfrank in 88/89.

Goals:
The purpose of the course is to present both practical and theoretical aspects of non-monotonic reasoning and to explain its relationship to computer science, particularly AI, deductive data bases and logic programming.

Prerequisites:
Introduction to standard logic. Familiarity with modal logic and three-valued logic is recommended but not necessary.

Organization:
12 two-hour lectures, three times a week. Independently, 6 hours of the laboratory work involving practical exercises.

Contents:
1. Foundations of non-monotonic reasoning.
2. Approaches to non-monotonic reasoning.
3. Non-monotonic formalisms:
 (i) Modal non-monotonic logics
 (ii) Default logic
 (iii) Circumscriptive logics
 (iv) Closed-World Assumption formalisms
 (v) Preferential entailment approach

Literature:
Lukaszewicz, W.: “Non-Monotonic Reasoning. A Formalization of Comonsense Reasoning”, Ellis Horwood , 1990.

Teacher:
Witold Lukaszewicz.

Examiner:
Witold Lukaszewicz.

Schedule:
15th September-10th October

Examination:
There will be a set of obligatory exercises. Each student must submit written solutions to these exercises and be prepared to orally present his/hers solutions.

Credit: 4 points

Operativsystem

Recommended for: *Föreläsningar* *26 h*
??????

Kursen gavs senast:
??????

Mål:
Kursen ska ge kunskap om hur ett operativsystem är uppbyggt. Detaljer kommer att tas från UNIX. Kursen ska ses som en fördjupning jämfört med OS-kurser i grundutbildningen. (Vissa begrepp kommer att repeteras).

Förkunskaper:
TDDA 21 Processprogrammering eller motsvarande kurs.

Kursinnehåll
1. Allmän introduktion, historisk översikt, användarens synvinkel
2. Introduktion till kärnan, bufferthantering
3. Filsystem, intern beskrivning (inodes, directories, ...)
4. Filsystem, systemanrop (open, read, mount, ...)
5. Processbeskrivningar och adressrymder
6. Schemaläggning
7. Minneshantering I
8. Minneshantering II
9. I/O (drivrutiner, IORB, ...)
10. Interprocesskommunikation
11. SUN-OS
12. - “ -
13. - “ -

Litteratur:
Maurice J. Bach “The design of the UNIX operating system”, Prentice-Hall, 1986

Lärare:
Föreläsning 1 - 10: Johan Fagerström
11 -13 Någon SUN OS-expert

Schema:
Fall 1991 JOHAN KAN DU GE DEN TIDIGT I VÅR??????

Examination:
Olika alternativ möjliga, tentamen med öppen bok, ...

Poäng:
5 poäng

Operating Systems

Recommended for:

Lectures: 26 h

The course last ran:

Goals:

This course will discuss the design and implementation of operating systems in general with UNIX as an example. It can be considered as a continuation of a basic undergraduate course. In particular, it includes“difficult” topics such as device drivers and network programming.

Prerequisites:
Basic operating system course.

Contents:
1. Introduction, using an operating system
2. Kernel, buffer cache
3. File system I: implementation (inodes, directories, ...)
4. File system II: system calls (open, read, mount, ...)
5. Processes
6. Scheduling and resource control
7. Memory management I
8. Device drivers II
9. Networks (sockets, streams,..
10. Process communication
11. SUN
12. - “ -
13. - “ -

Literature:
Maurice J. Bach “The design of the UNIX operating system”, Prentice-Hall, 1986

Teachers:
Johan Fagerström

Schedule:
September-October

Examination:
Written exam (open book) or term paper

Credit:
4 p

1991/92

Introduction to Parallel Algorithms

Recommended for:
Computer Science students

Lectures:20 h

The course last ran:
Partial overlap with “A Survey of Parallel algorithms for Shared-Memory Machines”, 1989

Goals:
The course addresses basic concepts and techniques concerning the design and analysis of parallel algorithms. Parallel algorithms for certain basic problems are presented. It is shown how these solutions can be used in the solution of other problems.

Prerequisites:
Basic course on design and analysis of algorithms.

Organization:
Two lectures weekly, obligatory homework.

Contents:
1. Introduction; Models of computation and analysis of algorithms.
2. Some basic problems: maximum, prefix sums, list ranking, etc...
3. Selection, merging and sorting.
4. Some basic graph algorithms.
5. Algorithms for various problems.

Literature:
Selim G Akl: the Design and Analysis of Parallel Algorithms,
Prentice-Hall International Edition, 1989.

Teachers:
Per-Olof Fjällström

Examiner:
Per-Olof Fjällström

Schedule:
April-May 1992

Examination:
Oral examination and obligatory homework.

Credit:
3 points

Parallel Computers: Architecture and Programming

Recommended for.
Computer Science and Systems students

Lectures: 24 h

The course last ran:
New course.

Goals:
The goal of the course is to give students basic knowledge of parallel computer architectures which are currently in use and basic programming methods for this computer systems. Some major commercially available systems will be discussed in the course

Prerequisites:
Basic knowledge about computer architecture and programming languages.

Organization:
The course will consist of lectures introducing basic concepts in the area and seminars discussing particular architectures. Some seminars will be presented by course participants.

Contents:
- Introduction and basic concepts
- Pipelined computers and vector computers
- Multiprocessors
- Parallel languages
- Parallel algorithms
- Case studies (Alliant, Connection Machine, CRAY X-MP and CRAY-2, Sequent, etc)

Literature:
R.W. Hochney, C.R. Jesshope, *Parallel Computers*, Adam Hilger, Bristol and Philadelphia, 1989
(D. Tabak, *Multiprocessors*, Prentice-Hall, Inc. 1990.)
Selected papers.

Teachers:
Krzysztof Kuchcinski, Zebo Peng

Examiner:
Krzysztof Kuchcinski

Schedule:
February - April 1992

Examination:
A written report on a chosen topic.

Credit:
3 points

1991/92

Parallel Execution Models

Recommended for:
Graduate students

Lectures+seminars:
16h

The course last ran:
This course has not been given before. The course *Compiling for Parallelism* which ran 1990/91 covered parallelizing transformations and compilation techniques, which is not the same as the material covered by this course.

Goals:
To give an understanding of a number of models for implementing and expressing parallelism.

Prerequisites:
Process programming course.

Organization:
Three introductory lectures followed by five seminars, where the course participants present papers.
A small one-week implementation project gives an extra point.

Contents:
Parallel execution models, languages and implementation strategies such as: Unity, Linda, Futures, etc.

Literature:
Scientific Papers

Teachers:
Johan Fagerström and Peter Fritzson

Examiner:
Johan Fagerström

Schedule:
Spring 1992

Examination:
Seminar presentation and exercises.
Implementation project gives extra point.

Credit:
2p (+1p)

Process Algebra

Recommended for.
Computer Science and Systems students

Lectures: 24 h

The course last ran:
This is a new course.

Goals:
An abstract notion of ‘process’ is introduced in order to study the behaviour of and interaction between processes in a general setting. The main tools for this study are Structured Operational Semantics and Axiom Systems.

Prerequisites:
Logic I

Organization:
Probably 12 2-hour lectures

Contents:
- CCS and CSP
- Basic Process Algebra
- Structured Operational Semantics (SOS)
- Process Equivalences
- Axiom Systems
- Parallelism and communication
- Silent step, abstraction, observational equivalence
- SOS with negative premises, priority
- Real Time Process Algebra
- Processes with Data

Literature:
To be decided later.

Teachers:
Roland Bol and Jan Maluszynski

Examiner
Jan Maluszynski:

Schedule:
November - December 1991

Examination:
Written exercises.

Credit:
3 points

1991/1992

Real-Time Systems

Recommended for.
Computer Science and systems graduate students.

Lectures: 24 h

The course last ran:
This is a new course

Goals:
The course will give a survey of different research issues and state of the art in the design of real time systems.

Prerequisites:
Processprogrammering TDDA21.
Basic knowledge in control theory and operating system.

Organization
The course will consist of 5 lectures and 7 seminars. The participants are expected to give presentations during the seminars.

Contents:
- Specification and verification - inclusion of time metrics.
- Real-time scheduling theory
- Real-time operating system.
- Distributed real-time databases.
- Real-time symbolic systems
- Architectures and models issues.)

Literature
S-T. Levi, A.K. Agrawala: Real Time systems design
Selected papers.

Teachers:
Anders Törne

Examiner:
Anders Törne

Schedule:
September 1991 - October 1991.

Examination:
Seminar presentation, written summaries
Term paper on chosen subject gives extra points.

Credit 3p (+ 2p)

Representation of Knowledge About Dynamic Systems

Recommended for:
Computer Science and Systems students

Lectures: 24 h

The course last ran:
This is a new course

Goals:
To present an integrated account of temporal reasoning, planning, and some aspects of qualitative reasoning. The course reflects on-going research both at IDA and elsewhere, and presents recent results as well as currently open research problems.

Prerequisites:
C-line A.I. courses; “A.I. paradigms. Courses on “Knowledge based planning” and “Qualitative Reasoning” are also recommended background.

Contents:
Choice of methodology: entailment based specifications; preferential (non-monotonic) logic; lexical domains. Time-dependent properties, chronicle completion, and the frame problem. Chronicle completion with incomplete information: different approaches including the use of the occlusion predicate. Reasoning about actions and their effects. Definition of plan generation, plan recognition, and other plan analysis operations in terms of entailment. Plan analysis algorithms. Special cases, in particular for strips ontologies and action-structure ontologies. Continuous change. Discontinuity equations.

Literature:
Manuscript in preparation: Features and Fluents. An approach to the representation of Knowledge about Dynamic Systems” by Erik Sandewall.
(Available September 1991).

Teacher:
Erik Sandewall

Examiner:
Erik Sandewall

Schedule:
January - March1992

Examination:
Term paper.

Credit:
3 points

1991/92

Risker och nyföretagande

Recommended for:
I första hand för Ekonomiska Informationssystems särskilda lic utbildning

The course last ran:
Ny kurs

Goals:
Att ge fördjupad insikt i nyföretagande

Contents:
Kursen kommer att belysa risker i samband med företagandet. Särskilt skall nyföretagandet och obeståndsfrågor belysas. Detta sker både utifrån ett teoretiskt och ett empiriskt perspektiv.
Bankers, revisorers och leverantörers roller kommer att behandlas. Vikt läggs vid individens psykologiska beteende. Särskilt skall relationen mellan uppdragsgivare och uppdragstagare behandlas.

Litterature:
Rapp, B, Thorstensson, Vem tar risken. Om uppdragsgivarens och uppdragstagarens resultatsdelning och incitament i riskprojekt. Studentlitteratur 1991.
Kompendium
Ytterligare litteratur tillkommer

Teachers:
Birger Rapp m fl

Examiner:
Birger Rapp

Schedule:
Veckorna 41,45,49 och 50

Examination:
Tentamen och skriftlig rapport

Credit:
3 - 10 poäng

För ytterligare information hänvisas till Eva Elfinger, tel 28 15 24.

Semiotics: History, Basic Concepts and Computer Applications

Recommended for.
Graduate students

Lectures: 18 h

The course last ran:
This is a new course.

Goals:
The purpose of the course is to present the concepts of the general theory of signs and their historical development within different traditions of investigation. With a firm grasp of the basic theoretical concepts in hand, various concrete computer applications will be presented and discussed.

Prerequisites:
None.

Organization:
9 two-hour lectures once a week in the beginning of the term covering the history and basic concepts of semiotics followed by a seminar series at the end of the term for the presentation of term papers.

Contents:
The course consists of two parts. The first part consists of a historical survey of the development of the key concepts of semiotic theory and a comparison of different traditions of research within semiotics. The concepts of sign, icon, index, and symbol constitute the focus for the historical, theoretical, and comparative part of the course. The syntactic, semantic, and pragmatic dimensions of semiotic systems are also defined, explained, and exemplified.
The second part of the course consists of independent supervised investigations by the students into different aspects of the theoretical concepts and their relevance for applications in computer science.

Literature:
A collection of articles and readings.

Teachers:
Richard Hirsch

Examiner:
Richard Hirsch

Schedule:
Fall term 1991.

Examination:
A written paper on a chosen theoretical topic or a report on an investigation of computer applications of semiotic theory.

Credit:
3 points.

1991/92

Software Engineering Design and Methodology

Recommended for:
Graduate students

Lectures: 40 h

The course last ran:

Goals:
To develop an understanding of and practise using software engineering design principles

Prerequisites:
None

Organization:
Lectures, group projects, individual homeworks

Contents:
The main focus of the course will be on software design methods. At least two design methods Yourdan and Bruch will be presented in detail and mention will be made of others. The course will also include some material on specification, testing and maintenance. These areas will not be presented in the same level of detail as the material on design, but are intended to give an overview and a context. Lectures will be used primarily to present issues and methods and also to present and analyze design exercises done by students (homeworks).

Literature:
Yourdan: Structured Design, Fundamentals of a Discipline of Computer Program and Systems Design
Grady Booch: Object Oriented Design with Applications

Teacher:
Lin Padgham

Examiner:
Lin Padgham

Schedule:
Spring 1992

Examination:
Group Project and indiviual exercises+ take home exam.

Credit:
6 points

Teorier och strategier för informationssystem

Recommended for: *Lectures:* 24h
I första hand för Ekonomiska informationssystem, ss systemutveckling.

The course last ran:
Detta är en ny kurs.

Goals:
Ge översikt över olika teori- och strategibildningar avseende informationssystem, samt öka förmågan att kritiskt granska sådana.

Prerequisites:
Inga absoluta krav. Önskvärt med grundläggande kurs i systemutveckling/informationssystem.

Organization:
Delvis samläsning med kurs SVL3.

Contents:
Teorier för informationssystem och systemutveckling. Olika synsätt på informationssystem: Objektivistiskt, systemteoretiskt, socioteknisk, humanistiskt. Strategier och principer för strukturering av informationssystem (informationssystemarkitektur): dataadministrativt vs verksamhetsbaserat. AU/ADB-strategier: innehåll, betydelse, utveckling.

Literature:
Ännu ej bestämd.

Teachers:
Göran Goldkuhl

Examiner:
Göran Goldkuhl

Schedule:
November 91 - januari 92

Examination:
Skriftlig rapport. Eventuellt tentamen.

Credit:
3-5 poäng