Division of Mathematical and Physical Sciences
MSc in Computer Science: Long Vacation 2009

Report of the examiners

Part 1

A: Statistics

1: Results

2009	2008	2007	2006	2005	2004	2003	2002	2001	2000	1999	1998	1997	1996
Distinction 10	16	11	10	5	5	6	5	3	6	7	5	7	5
Pass 36	48	44	40	35	28	25	29	24	31	25	22	28	22
Fail 5	2	8	5	3	6	3	4	5	2	2	0	2	2
Candidates 52	53	53	46	38	34	28	33	29	33	27	22	30	24

One candidate has an extension on his dissertation.

2: Number of vivas

There was one viva.

3: Number of scripts multiply marked

Each assignment was marked by an examiner or assessor (usually the lecturer on the course). The marks were moderated by the internal examiners, with the approval of the external, to ensure fair comparison between different courses.

Each dissertation was read twice, by an examiner and an assessor. Cases bordering on failure or distinction, or for which there was a large discrepancy between those two marks, were read by another examiner. The external examiner examined three dissertations.

4: Numbers taking each optional subject

Automata, Logic and Games	3
Bioinformatics and Computational Biology	2
Categories, Proofs and Processes	1
Compilers	9
Computational Complexity	1
Computational Linguistics	14
Computer Aided Formal Verification	7
Computer Animation	16
Computer Security	24
Computers in Society	7
Concurrency	5
Concurrent Programming	4
Databases	5
Database Systems Implementation	19
Foundations of Computer Science	4
Formal Program Design	1
Functional Programming	13
Game Semantics	0
Information Retrieval	37
Intelligent Systems I	28
Intelligent Systems II	34
Introduction to Specification	3
Lambda Calculus and Types	8
Logic of Multi-Agent Information Flow	5
Machine Learning	40
Object Oriented Programming	17
Program Analysis	3
Principles of Programming Languages	2
Probabilistic Model Checking	4
Quantum Computer Science	2
Randomised Algorithms	3
Software Verification	3
System Design & Refinement	0
Theory of Data and Knowledge Bases	3

B: Changes in examining methods and procedures this year

Following last year’s recommendation, the pass mark for practicals was raised from 30 to 50.

C: Changes in examining methods and procedures envisaged

All parts of the examination process worked smoothly, and no changes were recommended.

Part 2

The examiners awarded the Hoare Prize for best overall performance to Rasmus Wissman and the best project to Sean Heelan.

Appendix

Automata, Logic and Games
Scaling:{0 0}
Number of candidates: 3 *Mean: 37.0* *Standard Deviation: 24.1*

Below 50	50-60	60-70	70-80	80-90	90-100
1	2	0	0	0	0

Question 1. Quite a discriminating question. Nobody had any problem in (a) and (b), although a few answered (b) inelegantly by coding their answer to (a). For(e) both the direct proof and proof-by-contradiction were exhibited. Some (perhaps a third) of the candidates spotted the intended route to (f) and (g) - by exploiting (d) and (e) and establishing a simple lemma.

Question 2. Parts (a) and (b) are straightforward. Two different encodings of the release operator were given by the candidates.

Question 3. The hardest question: quite a few faltered on (a) and then muddled their way through. On a positive note, the few correct simulations were all clearly different from each other.

Question 4. Most had the right idea for the transition function (a) but few formulated it rigorously. Some candidates evidently found help in the literature for (b) and (c), but the overall quality of argument was lower than expected. A common omission in (d) was the direction from Parity to Muller.

Question 5. Perhaps too easy: all candidates did very well on this straightforward question.

Bioinformatics and Computational Biology
Scaling:{0 0}
Number of candidates: 2 *Mean: 59.0* *Standard Deviation: 8.0*

Below 50	50-60	60-70	70-80	80-90	90-100
0	1	1	0	0	0

Categories, Proofs and Processes
Scaling:{14 30} {60 60}
Number of candidates:1 *Mean: 30.0* *Standard Deviation: 0.0*

Below 50	50-60	60-70	70-80	80-90	90-100
1	0	0	0	0	0

The exam consisted of a few routine questions on introductory category theory, supplemented by more challenging questions requiring mathematical creativity. There were a few candidates (including finalists who also took this exam) who had difficulty with even the routine parts.

The external assessor for this course expressed some concern that candidates had consulted the library and web to answer easier questions, instead of attempting them on their own. However, there was no evidence that any candidate did so without appropriate citation of the relevant sources. Furthermore, the assessor had been told in advance that such consultation of the literature is permissible.

Compilers
Scaling:{56 50} {83 70} {96 100}
Number of candidates: 9 *Mean: 70.7* *Standard Deviation: 9.8*

Below 50	50-60	60-70	70-80	80-90	90-100
0	1	6	0	2	0

As intended, the exam was more stretching than last year. The students' answers were extensive and, on the whole, very detailed and well justified, demonstrating sound command of the subject. The marks on all four questions were quite evenly balanced. Some small parts of questions (e.g. 2b) were on topics not covered in details in the lectures, but the students used the textbooks and other material available to answer those satisfactorily. Two parts of questions caused some confusion due to the way they were stated (1d and 3b) but this was compensated for in the marking. Question 4c, which was open-ended, had a good range of interesting answers.

Computational Complexity
Scaling:{0 0}
Number of candidates: 1 *Mean: 50.0* *Standard Deviation: 0.0*

Below 50	50-60	60-70	70-80	80-90	90-100
0	**1**	**0**	**0**	**0**	**0**

Computational Linguistics
Scaling:{0 0}
Number of candidates: 13 *Mean: 59.6* *Standard Deviation: 9.5*

Below 50	50-60	60-70	70-80	80-90	90-100
2	**4**	**6**	**1**	**0**	**0**

Computer Aided Formal Verification
Scaling:{60 100}
Number of candidates: 7 *Mean: 55.5* *Standard Deviation: 13.9*

Below 50	50-60	60-70	70-80	80-90	90-100
3	**2**	**1**	**0**	**1**	**0**

Q1. All but one candidate got the right BDD.

Q2. Part 1 was done very well by virtually all candidates. Most candidates correctly identified the complexity of the general case for part 2, with the great majority opting for a theoretical analysis rather than experiments. The proofs varied widely in detail and rigour, with one or two extremely elaborate efforts. A small minority of candidates leaped to the conclusion that the rate of growth is linear, based on (sometimes erroneous) examples.

Q3. Solutions to this question were of mixed quality, with a surprising number of imprecise or vague proofs, often with missing steps. Some solutions, however, were solid.

Q4. Virtually all candidates had a correct counterexample, with explanations varying in quality.

Q5. Virtually all answers correct for this routine question, with varying degrees of detail in the proofs.

Q6. Several candidates declined to make a serious - or any - attempt at question 6. Of those who did, all had a reasonable formula for part 1, although several of these neglected to mention that the boolean variables used must not already appear in A1, A2, and C. The proofs for part 2 were mostly disappointing, with few candidates noticing that the key to the proof was an argument about valuations and variables not occurring in the formula.

Computer Animation
Scaling: {48 50}
Number of candidates: 16 *Mean: 64.2* *Standard Deviation: 8.7*

Below 50	50-60	60-70	70-80	80-90	90-100
0	**6**	**7**	**2**	**1**	**0**

This is the first time this course has been offered, but all the candidates showed at least some understanding of the material. In particular, the format of the exam seemed to work well; the first two were traditional (but taxing) exam questions, and were worth 25% of the total marks each, and the last question followed a mini-project format for the other 50%. In particular, little scaling was required of the raw marks to produce

USMs.

Question 1 involves material concerning Bezier interpolation and quarternions, with the use of the de Casteljau process and quarternions being a new twist. There was an `obvious' typographical error in the paper here, that a candidate reported and a correction posted to all; it clearly caused no problems to the candidates. At one point candidates had to generate a quarternion representing a rotation about a particular direction; the majority of the candidates did not normalise the direction vector first, and this error propagated throughout the rest of the question.

Question 2 involves the kinematics for an unusual (vehicular) mechanism, and was unpopular. Few simplified the expression as instructed, or properly applied a forward Euler scheme when asked; both these steps where designed to simplify the analysis. At one stage some angles were given in degrees, and I was shocked when the majority of candidates proceeded to treat them as radians (by using $r\times\theta$ for the length of an arc of a circle). Very few candidates could relate the Jacobian expression in part (c) to the use of the transpose Jacobian for inverse kinematics; most chose to come up with ad hoc solutions instead.

Question 3 involved the design of part of a robot football system. Most of the candidates chose to tackle this by giving a Blender implementation and report, with just two candidates taking the alternative option of a paper design study. All the Blender solutions were tested (with one failing to load properly, thanks to some hard-wired file names). Most of the implementations were good or better, although in several cases poor attention to the written report lost significant marks when the implementation itself was impressive.

Computer Security

Scaling:{46 50}

Number of candidates: 24 *Mean: 59.8* *Standard Deviation: 13.8*

Below 50	**50-60**	**60-70**	**70-80**	**80-90**	**90-100**
4	**5**	**11**	**1**	**3**	**0**

The results of this examination seem satisfactory. In my opinion, they faithfully reflect the good work that many students invested in it. Only a couple of them did not absorb the material of the course, and they are probably those who for personal reasons did not follow the course. It was difficult for them to compensate the missed lectures, because there was no suitable textbook, and the lecture notes are a work in progress. This is, or course, a serious shortcoming, that should be solved in the future. Nevertheless, the overal results of the course positive, perhaps more so than I dared to expect. Three students had 80 points or more.

I should have formulated some parts of question 2 more precisely. I did not want to make the questions too formal, or too mathematical, and I exaggerated in the opposite direction in this question. It is, of course, always better if the material can be covered and examined more precisely; but a course covering all of security in 16 lectures cannot set too narrow targets. Question 3, which required a very basic cryptographic proof, following the examples worked out in the lectures and in the exercises, remained the hardest.

The scope and importance of the material, as well as students' interest, would seem to justify assigning more time to this course.

Computers in Society

Scaling:{0 0}

Number of candidates: 7 *Mean: 73.1* *Standard Deviation: 10.7*

Below 50	**50-60**	**60-70**	**70-80**	**80-90**	**90-100**
0	**1**	**3**	**1**	**2**	**0**

Concurrency

Scaling:{59 50} {88 74}

Number of candidates: 5 *Mean: 59.6* *Standard Deviation: 7.7*

Below 50	**50-60**	**60-70**	**70-80**	**80-90**	**90-100**

0	**2**	**2**	**1**	**0**	**0**

Concurrent Programming
Scaling:{80 80} {83 80}
Number of candidates: 4 *Mean: 63.5* *Standard Deviation: 12.8*

Below 50	50-60	60-70	70-80	80-90	90-100
1	**1**	**1**	**1**	**0**	**0**

Databases
Scaling:{0 20} {25 70} {40 100}
Number of candidates: 5 *Mean: 61.6* *Standard Deviation: 8.2*

Below 50	50-60	60-70	70-80	80-90	90-100
0	**3**	**1**	**1**	**0**	**0**

Question 1: Most people did well on this, but many failed to understand what SPJ meant, and some did not manage the question that required universal quantification

Question 2: Performance on this problem was tolerable, but students did not know how to derive dependencies using the axioms. No student could do the last part correctly.

Question 3: Every student could derive the graph correctly, but few understood that there were many serializations corresponding to it. No student could give a fully correct proof for the second part, although a few had an idea.

Question 4: Most students understood the idea of an algorithm for the first part, although some of the algorithms were inefficitn. The second half of the problem students did very well on, although many of them exhaustively looked at enormous numbers of plans.

Database Systems Implementation
Scaling:{55 50} {80 70}
Number of candidates: 19 *Mean: 65.8* *Standard Deviation: 9.9*

Below 50	50-60	60-70	70-80	80-90	90-100
1	**2**	**11**	**3**	**2**	**0**

Overall, this paper was very well done by most of the candidates. Except for two weak submissions, the candidates demonstrated a sound grasp of ideas and techniques studied in the course.

Question 1 was answered well by all but two students, with marks varying from 5 to 20 (out of maximum 20) and only two of them below 10. The major difficulty is centered around the construction of extendible hash indexes (part 3).

Question 2 was in general not approached satisfactorily, with most of the students wrongly proposing main-memory based extensions of standard disk-resident join algorithms for the implementation of full-outer joins. The given marks (maximum of 15) were distributed rather uniformly with four of them below four and four of them above ten (only one solution reached the maximum).

Question 3 is a hard question involving sound cost estimation and construction of (near-optimal) query plans, and constituted a major part of the exam with a maximum of 30 marks. Overall, the candidates performed well and the solutions are good, with only one below 10 marks, three below 19, most of them between 20 and 24, and six of them above 25 (two with 29 marks). The solutions with more than 25 marks are very interesting and showed very deep understanding of the topic of query optimization.

Question 4 is an implementation question and builds upon the work done by the students during the practicals. It weights 35 marks, which is significant in comparison with the first two questions. Very importantly, all students approached this question and were able to submit compilable and runnable code and, except of one case, also understandable experimental report. Six solutions were given the maximum number of marks for the implementation part, with only one of them obtaining the maximum overall. Most implementations, which did not obtain the maximum number of points, did not treat errors (eg, I/O errors) at all or not properly. As for the reports, the points were deducted for not self-explanatory diagrams (eg, units on the x- and y-axes not specified) or for missing or poor explanations of the diagrams.

Formal Program Design
Scaling: {0 0}
Number of candidates: 1 *Mean: 45.0* *Standard Deviation: 0.0*

Below 50	50-60	60-70	70-80	80-90	90-100
1	0	0	0	0	0

Only one candidate took this option

Foundations of Computer Science
Scaling:{8 50} {21 70} {30 100}
Number of candidates: 4 *Mean: 61.5* *Standard Deviation: 7.7*

Below 50	50-60	60-70	70-80	80-90	90-100
0	2	2	0	0	0

Question 1: Most people did well on the first two parts. a few had difficulties with the "solitaire" question, missing the connection to PCP.

Question 2: Students generally did well. One part of the problem was misleadingly phrased: it asks the students to state whether the problem is in P or is NP complete, while the problem is likely not in NP. Students still handled this deftly, explaining that the problem was NP-hard but could be harder.

Question 3: In the first part of the question, most students had the right idea but many had trouble writing a correct predicate logic formula. Students understood the implications for the satisfiability problem, but had difficulty articulating it. Only D.Phil. students got the last part of the problem correct.

Functional Programming
Scaling:{0 0}
Number of candidates: 13 *Mean: 55.0* *Standard Deviation: 29.4*

Below 50	50-60	60-70	70-80	80-90	90-100
4	2	2	3	1	1

On the whole the assignment was completed well and solutions were well-presented. In Question 1 most candidates were happy to represent a bag as a function, but solutions to the final part (on prime factors) were often rather overcomplicated due to re-using functions from the practical assignment rather than starting afresh. In Question 2 some candidates lost marks for giving only informal arguments rather than proofs. Question 3 was the most difficult question, with many candidates failing to appreciate that the infinite lists constructed in parts (c) and (d) had to be built by interleaving other infinite lists, to ensure that all elements occurred in some finite initial segment of the answer. Few realised that a fold could be used in part (b). In Question 4 several candidates failed to construct a general inductive proof over trees - they dealt only with the special cases listed as examples in the question. Question 5, on evaluating a simple Command language was fully completed by almost all candidates, so probably too easy (or easily available). Question 6 was much more challenging: very few candidates fully used the powerful combinatorial parsing tools provided in the lecture notes; most resorted to ad hoc string processing techniques such as dividing the input into a list of words, looking for the presence of keywords, and so on.

Game Semantics – No candidates took this option

Below 50	50-60	60-70	70-80	80-90	90-100
0	0	0	0	0	0

Information Retrieval
Scaling: {42 50} {80 70}

Number of candidates: 37 Mean: 61.1 Standard Deviation: 10.1

Below 50	50-60	60-70	70-80	80-90	90-100
3	16	13	5	0	0

The performance on the exam overall was as I expected: a small number producing an excellent script, showing depth of understanding and interesting insights; a small number showing very little understanding and essentially copying out the lecture notes; and the majority somewhere in between.

Qu. 1 was answered well on the whole; it was intended to be an easy starter. Quite a few didn't understand that 2)i was about smoothing. Quite a few didn't understand "effectiveness of the dynamic programming", and that forming less equivalence classes makes the DP less effective. A surprising number didn't understand that "Describe the problem of cross-lingual IR" simply means give a description of cross-lingual IR, and not list all the problems associated with it. The lack of non-native knowledge of English clearly hampered some of the candidates. Question 4, the essay question, was answered surprisingly well by a small number, although most failed to show any dazzling insights, simply repeating what was said in lectures.

One candidate, 153564, made the remarkable mistake of including the answer to a database exam for question 1, and hence received no marks for that question.

Intelligent Systems I

Scaling:{44 44} {50 44} {100 94}

Number of candidates: 28 Mean: 66.3 Standard Deviation: 8.0

Below 50	50-60	60-70	70-80	80-90	90-100
1	3	18	6	0	0

The overall standard of the solutions was high, although high marks were awarded for clarity, and some candidates were penalised for not focussing on the questions and providing long `essays' that only proved that they could copy bookwork.

The first question dealt with some aspects of search and path planning. Most candidates sketched the configuration space well, although several thought that it was reasonable for a 2m wide robot to travel down a 2m wide corridor. (It isn't.) Given the nature of the take-home exam, small numbers of marks were ruthlessly taken off for missing details and errors in the search processes. The last two parts allowed candidates to think around situations that they hadn't considered before; only one candidate spotted that the last part could be solved by moving a unit-length rod around the configuration space, using a potential field.

Question two dealt with Bayes rule and Markov processes. Several candidates provided circular arguments for the `proof' in part (a), rather than following the example given in classes. Marks were lost for using alternative definitions of a Markov Process than the one given in the lectures. Despite the unusual setting, practically all the candidates could deal with the concrete example given, although some offered much more elegant (and shorter) solutions than others (and were rewarded).

Intelligent Systems II

Scaling:{36 30} {100 91}

Number of candidates: 34 Mean: 66.9 Standard Deviation: 10.1

Below 50	50-60	60-70	70-80	80-90	90-100
1	5	15	9	4	0

Introduction to Specification

Scaling:{subtract 4 marks}

Number of candidates: 3 Mean: 66.8 Standard Deviation: 16.4

Below 50	50-60	60-70	70-80	80-90	90-100
0	0	1	1	1	0

Lambda Calculus & Types

Scaling:{28 30} [47 50}

Number of candidates: 8 *Mean: 48.3* *Standard Deviation: 15.7*

Below 50	50-60	60-70	70-80	80-90	90-100
3	3	1	1	0	0

Logic of Multi-Agent Information Flow

Scaling:{0 0}

Number of candidates: 5 *Mean:67.8* *Standard Deviation: 21.5*

Below 50	50-60	60-70	70-80	80-90	90-100
1	0	2	0	1	1

Machine Learning

Scaling:{50 50} {85 70}

Number of candidates: 40 *Mean: 69.0* *Standard Deviation: 12.2*

Below 50	50-60	60-70	70-80	80-90	90-100
1	5	19	7	8	0

In general, the students have answered well or very well despite the fact that the questions required a deep understanding of the problems. There were 4 questions that covered the vast majority of the taught topics. The most difficult question was Question 4, but, remarkably, more than half of the students have provided good answers. Many students had problems with solving the second part of Question 3. Most of the students obtained marks between 65 and 85, one student has failed. Overall, the examination went very well and I am very satisfied by the level of understanding reached by students.

Object Oriented Programming

Scaling:{0 0}

Number of candidates: 17 *Mean: 60.5* *Standard Deviation: 12.1*

Below 50	50-60	60-70	70-80	80-90	90-100
1	10	2	3	1	0

The MSc exam for the Object Oriented Programming course consisted of one question which required students to implement an application based on the model-view-controller design pattern. The applications submitted by the students demonstrate several common errors.

- Many students did not apply the separation between the model and the view to the extent to which they should have. In particular, a common error was to implement hit-testing as part of the model; however, this functionality logically belongs to the view. This might be caused by the fact that the sample application used in the lectures implemented hit-testing in the model, but the students failed to see the difference.

- Many students allowed domain elements to be changed without a notification event being fired. It is unclear why this common error occurred: the sample application clearly demonstrates how the Observer pattern should be applied.

- Several students were caught by the desire to apply as many design patterns as possible, resulting in a complex and buggy application.

- A number of students produced domain models that stored data in the domain model using parallel structures. All such applications contain Airport and AirportElement, as well as Route and RouteElement classes. Furthermore, all such applications were obviously obtained by copying the sample application and adapting it to the given problem. While it is unclear what the source of this error was, the most likely cause seems to be that the students were too wrapped into the particular of the sample application and were not able to abstract its important aspects. **It is also possible (although I do not know how to prove it) that some of these students collaborated on the design of the application.**

- Two students submitted applications that do not meet the required standard. The application of the first student reflects at least remotely the structure of an MVC application. In contrast, the application of the second student does not even remotely resemble an MVC application.

Program Analysis
Scaling:{0 0}
Number of candidates: 3 *Mean: 58.7* *Standard Deviation: 18.1*

Below 50	50-60	60-70	70-80	80-90	90-100
1	1	0	0	1	0

The assignment consisted of three questions: one on attribute grammars, one on flow-sensitive type analysis, and a question on analysis of logic programs. Four candidates took the exam. It was rather disappointing to see that only one candidate attempted the harder question on attribute grammars. Two candidates achieved an outstanding mark.

Principles of Programming Languages
Scaling:{0 0}
Number of candidates: 2 *Mean: 70.0* *Standard Deviation: 8.0*

Below 50	50-60	60-70	70-80	80-90	90-100
0	0	1	1	0	0

Probabilistic Model Checking
Scaling:{52 52} {80 70} {94 88}
Number of candidates: 4 *Mean: 65.7* *Standard Deviation: 8.7*

Below 50	50-60	60-70	70-80	80-90	90-100
0	1	1	2	0	0

The students handled the exam very well. Despite latter parts intended to be more open-ended and stretching, the average raw mark was high (76.6 out of 100) and 2 students obtained raw marks in the low 90s. Questions 1-3 mostly took the same form as questions from course problem sheets, requiring execution of various algorithms/calculations and a simple modelling exercise. Except for a selection of small errors, these posed few problems. Question 4 was designed to test understanding of some key model checking algorithms, requiring the students to devise extensions of the algorithms for new scenarios. A few students struggled but the majority had no problems, producing a variety of correct solutions. Question 5 was the largest, and most open-ended, question on the paper, worth 30 marks out of 100. It involved the use of the PRISM tool to model and analyse a communication protocol described in the paper. This proved to be the best question to distinguish between the students, with answers varying from brief functional solutions to detailed and well-thought out responses illustrating excellent understanding of the techniques taught in the course.

Quantum Computer Science
Scaling:{0 0}
Number of candidates: 2 *Mean: 77.0* *Standard Deviation: 5.0*

Below 50	50-60	60-70	70-80	80-90	90-100
0	0	0	1	1	0

The marks were exceptionally high this year for the MSc CS students, in contrast to the previous years when MFoCS and Part C maths students obtained the top marks. The first question was essentially perfect for all students, and also the second was solved in a satisfactory manner. Only for the third question, which was the most difficult one, some encountered problems.

Randomised Algorithms
Scaling:{62 50} {82 68}
Number of candidates: 3 *Mean: 63.5* *Standard Deviation: 5.7*

Below 50	50-60	60-70	70-80	80-90	90-100

0	**1**	**2**	**0**	**0**	**0**

Randomised Algorithms was only taken by 3 MSc students, so it is difficult to draw any significant (statistically or otherwise) conclusions from the students' performance in it. (The course was also taken by 7 MFoCS students, 1 DPhil student, and an as-yet-unknown number of 4th-year students, but as there are considerable differences in background among these, it would not be appropriate to include them in the context of MSc students' performance.)

All three students passed, and all clearly took valuable insights away from this course. No student did spectacularly well, although one student clearly exhibited a strong aptitude for the subject, which was unfortunately marred by his apparently not spending nearly long enough working on this paper. Among the five questions, Q2 and Q3 proved slightly more difficult.

Software Verification
Scaling:{38 45}
Number of candidates: 3 *Mean: 59.2* *Standard Deviation: 9.2*

Below 50	50-60	60-70	70-80	80-90	90-100
1	**0**	**2**	**0**	**0**	**0**

System Design & Refinement - – No candidates took this option

Below 50	50-60	60-70	70-80	80-90	90-100
0	**0**	**0**	**0**	**0**	**0**

Theory of Data and Knowledge Bases
Scaling:{0 0}
Number of candidates: 3 *Mean: 41.0* *Standard Deviation: 27.5*

Below 50	50-60	60-70	70-80	80-90	90-100
2	**0**	**0**	**1**	**0**	**0**

List of successful dissertations

- Forcasting Financial Markets Using Data Mining & Machine Learning techniques
- Generating Computer Animations Based on Real World Objects Using Stereopsis
- Functional Flow Analysis and Pattern Matching Safety
- Modelling the BuddyCast Protocol used in the Tribler File Sharing Network
- A Machine Learning Approach to Word Alignment in Statistical Machine Translation
- A JIT for Oberon Compiler
- Data Cleaning Using Outliers
- Measures of Document Simularity
- Finding Bugs with JastAdd
- Analysis of Machine Learning Methods in Financial Forcasting
- Value-based Join Algorithms for XPath on XML Streams
- Object Recognition in Robotic Search and Rescue
- Investigations in Quantum Computing
- Measures of Document Similarity
- Machine Learning Technique for Web Usage Mining based on User Mouse Tracking
- Automatic Generation of Control Hijacking Exploits for Software Vulnerabilities
- Information Extraction-Smart Calender Application
- Logic Circuit Workbench
- Learning in Robot Soccer
- Map Stitching
- Neural Networks for Financial Market Forcasting
- Image Processing

- Using Artificial Intelligence Methods to Analyze Banking Risks
- Financial Forcasting Using Machine Learning Techniques
- Combining Information by Communication: The dynamics of Belief Merge
- Consequence-Based Reasoning for Datatypes in Bio-medical Ontologies
- Learning C-tables
- Interactive Belief Revision
- Well Typed Recursion Schemes and Implementation of an Intersection-Type Inference Algorithm
- Physics and Animation Textile
- Modelling and Reasoning About Complex Systems
- Reconstruction of Ancestral Metabolisms
- Football Simulation & Animation
- Computing Degrees of Safety for Belief "Probabilistic Methods in Computer Science"
- Automated Rough Terrain Navigation
- Approximate Confidence Computation in Probabilistic Databases
- XML Update Language
- Dynamic Selection for Geographically Distributed Servers Using Machine Learning Techniques
- An Investigation of Generic Properties of Biological Systems
- Secure Channels in CSO
- The classification of emotion in short text extracts
- An Investigation of Authentication Models in Ubiquitous Computing
- Query Evaluation on Uncertain Databases
- Modelling of the Financial Markets Using Agent-Based Machine Learning Methods
- Implementing Adaptive Websites Using Machine Learning and Ajax
- Geometric Modelling or a project in Imaging (exact title TBC)
- Vector Addition Systems for Verifying Boolean Programs
- Vector Space Models of Meaning